# **Projet de gestion d’un parc automobile**

# **D’une société**

## 

#### Projet réalisé par:

#### **Ayoub Achak**

#### **Ismail Essagar**

#### **Ammar Nafir**

#### Projet encadré par:

#### **Mme.Loubna el faquih**

#### 

#### 

# 

# 

# 

# 

# 

# 

# **Cahier des charges**

## **a)** **Etude de l’existant**

## **b)** **Etude critique**

## **c)** **Les cibles du projet**

## **d)** **Le périmètre du projet**

## **e)** **Les objectifs de l’application**

## **f)** **Etude des besoins**

### **Besoins fonctionnels**

### **Caractéristiques techniques**

## **g)** **Règles de gestion**

## **h)** **Méthodologie et solution**

# **Conception**

## **1)** **Définition de la modélisation**

## **2)** **L’avantage d’utiliser UML comme méthode d’analyse et de conception**

## **3)** **Diagrammes UML**

### **1.** **Le diagramme des cas d'utilisation**

### **2.** **Les diagrammes de séquence**

### **3.** **Le diagramme d’activités**

### **4.** **Le diagramme de classes**

### **5.** **Modèle logique de données (MLD)**

## **4)** **Conclusion**

# **Conception des interfaces**

**Cahier des charges**

a. Etude de l’existant

Actuellement, plusieurs systèmes sont mis à la disposition des utilisateurs afin de mettre en œuvre la notion de la gestion de parc automobile. Les applications existantes pour la gestion de parc automobile sont nombreuses et différentes de point de vue design et ergonomie mais elles offrent pratiquement les mêmes fonctionnalités : Elles assurent la gestion administrative, la gestion technique et la gestion opérationnelle du parc de véhicules.

b. Etude critique

Malgré la multiplicité et la diversité de ces applications, ils présentent tous une défaillance au niveau des services présentés. Parmi les défaillances que nous avons constatées dans les applications c’est l’absence de la gestion des ordres de mission, qui est un facteur important dans la gestion de parc automobile. De plus, l'absence de session du mécanicien implique que la réparation et l’entretien des véhicules sont faits en dehors du parc ce qui est plus coûteux.

c. Les cibles du projet

*Les tranches visées :*

L’application vise par ses fonctionnalités une tranche principale :

- L’application s’adresse aux directeurs responsables du parc automobile, aux gens qui s’occupent d’un parc de véhicules et ont du mal à travailler avec les méthodes traditionnelles et manuels et cherche une grande performance et une simplicité de travail.

*Où se trouvent ces personnes ?*

Les gens qui peuvent accéder à cette application peuvent être partout dans le monde. Il n’y a pas de restrictions.

d. Le périmètre du projet

Description du contenu du résultat final :

Concevoir, créer et maintenir une application qui rassemble toute l’information nécessaire à la vie du parc automobile, qui le rende accessible aux acteurs, permette sa mise à jour en continu, et son exploitation sous des formes multiples.

Livrables :

1-Définitions des besoins ;

2-Cahier de charges ;

3-Validation de la conception ;

4-Code source ;

5-Implémentation ;

6-Validation des tests ;

7-Plateforme.

e. Les objectifs de l’application

Cette application doit permettre :

· De gérer l’ensemble des véhicules qui ont recours au parc ;

· De suivre et gérer les ordres de sorties et de missions ;

· De suivre et gérer la réparation automobile ;

· De documenter et gérer la consommation automobile ;

· D’assurer une meilleure gestion au parc automobile.

f. Etude des besoins

- Besoins fonctionnels

• Meilleure organisation et exploitation de véhicules existants.

• Alerte des défaillances techniques pour la sécurité du personnel.

• Suivi des échéances des assurances avec préservation des droits des individus et de l’entreprise.

• Organisation de l’utilisation des voitures

• Suivi et contrôle du niveau de consommation des carburants.

• Suivi de limite de carburant

• Gestion des marques, sous marques, services, sociétés et catégories de véhicules

• Gestion des modèles : consulter les caractéristiques techniques complètes (général, consommation, etc.)

• Suivi des opérations de maintenance dans les délais

• Gestion des conducteurs (matricule, nom, prénom, service, département, type et validité de permis, etc.)

• Suivi et gestion de la réparation automobile

• Suivi des ordres de sorties et de missions

• Gestion des ordres de mission (autorisation de circulation, permanence, déplacement, pointage...)

• Gestion des ressources humaines, de l'équipement du travail et des consommables du parc.

• Gestion des primes de déplacement affectés au personnel suivant les missions affectées.

- Caractéristiques techniques

Le logiciel à mettre en place devra regrouper les caractéristiques suivantes :

• Souplesse, facilité d’exploitation, convivialité en offrant le maximum de paramétrage permettant l’évolution en fonction des futurs besoins du parc.

• Contrôle automatique de la validité et la cohérence des données saisies.

• Consultations paramétrées avec la possibilité de recherches multicritères.

• Permettre le paramétrage des profils et des droits associés aux utilisateurs

• La pérennité et la sécurité des informations

• La compatibilité avec les environnements Windows

g. Règles de gestion

▪ Un véhicule est caractérisé par un identifiant, un matricule, une date de mise en service, une couleur, une catégorie, un numéro châssis, un taux de puissance, un service, et son type marque.

▪ L’utilisateur peut filtrer les véhicules soit par service, catégorie ou type marque

▪ On cherche un véhicule avec son matricule.

▪ Un véhicule appartient à un service, une catégorie et un type de marque.

▪ Un état est associé à un véhicule à l’addition d’une date de début et de fin.

▪ Une vidange contient la matricule d’un véhicule, une date et une distance.

▪ Une réparation comprend le matricule du véhicule réparé, une référence, une date, une facture et une distance.

▪ Pour une réparation détaillée on ajoute à la réparation une pièce, un type et un prix.

▪ On associe à la consommation automobile le matricule du véhicule, la quantité, le montant total, le montant par lettre, le kilométrage et on y ajoute une société par ordonner son identifiant.

▪ Le personnel de l’ordre mission est décrit par son nom, son prénom et sa fonction.

▪ Pour les ordres de mission en réserve un ensemble d’informations y compris le type de permis, le matricule, le nom et prénom de la personne concernée, sa fonction, le conducteur, l’itinéraire, l’objet de mission, date de départ, date de retour et les personnes à bord.

▪ On aperçoit la consommation d’un véhicule en donnant son matricule.

▪ On cherche la réparation automobile par un matricule ou par pièce et on reçoit la réparation automatique.

h. Méthodologie et solution

- *Nous avons décidé d'adopter la méthodologie en cascade pour la réalisation de ce projet pour les raisons suivantes :*

· Les besoins sont connus et stables

· La simplicité et la facilité d'utilisation de cette méthode

· Fourniture d’une bonne visibilité de la suite du projet

- *Les principes de cette méthodologie :*

· La production des livrables définis au tout début du projet

· La livraison de ces livrables à une date précise et définie lors du cadrage du projet

· La phase ne se termine que lorsque cette dernière a été vérifiée puis validée. Si un client n’est pas satisfait d’un livrable, l’équipe devra le retravailler jusqu’à ce qu’il soit parfait.

- *Le cycle de développement en cascade :*

· Cadrage du projet

· Conception générale

· Conception détaillée

· Implémentation

· Tests et corrections

· Livraison

**Conception**

**Introduction**

Chaque système d’information a une représentation abstraite et simplifiée qui permet de le décrire et de l'expliquer. Modéliser ce système avant sa réalisation permet alors de mieux comprendre son fonctionnement.

Dans ce chapitre, on va aborder la modélisation et la conception du projet.

**3.1 Définition de la modélisation**

Pour développer une application, il faut d’abord organiser les idées, les documenter avant de commencer la réalisation tout en définissant les modules et les étapes. On appelle cette démarche "modélisation". Pour réaliser cette modélisation, on a choisi la méthode UML pour s’occuper.

Le langage UML

Langage de modélisation objet. Il faut donc lui associer une démarche (étapes. Phases et tâches de mise en œuvre) pour en faire une méthode. L'absence de démarche qui peut être perçue comme un inconvénient est plutôt un avantage car cela permet de trouver une démarche bien adaptée au système d'information à concevoir.

A l'instar des méthodes objets. UML propose une approche différente de Merise. Qui associe données et traitements et qui décrit la dynamique du système d'information comme un ensemble d'opérations attachées aux objets du système. De cette façon, l'approche UML assure un certain niveau de cohérence.

Idéal pour concevoir et déployer une architecture logicielle développée dans un langage objet (Java, C++. VB.net,...) puisque de par son origine (la programmation objet)UML s'affirme comme un ensemble de formalismes pour la conception de logiciel à base de langage objet.

**3.2 L’avantage d’utiliser UML comme méthode d’analyse et de conception**

De l'analyse comparative réalisée dans le tableau précédent, nous choisissons UML comme méthode d’analyse et conception de notre système d'information car UML :

Est un langage visuel car sa notation graphique permet d'exprimer visuellement des solutions objet facilitant ainsi la comparaison et l'évaluation de celles-ci.

Est un langage formel et normalisé doté d'un gain de précision et d'un gage de stabilité. Il sert à formaliser tous les documents techniques d'un projet et permet d'affiner les détails de l'analyse au fur et à mesure de l'avancée du projet.

Est possible d'utiliser le même atelier de génie logiciel, depuis l'expression des besoins jusqu'à la génération de tout ou partie du code.

Est un support de communication performant car il cadre l'analyse tout en facilitant la compréhension des représentations abstraites complexes.

**3.3 Diagrammes UML**

Les diagrammes sont les éléments qui permettent de décrire les différents aspects d'un système. Ces diagrammes sont au nombre de neuf et peuvent être classés en trois groupes selon qu'ils décrivent les aspects statiques ou qu'ils décrivent les aspects dynamiques ou qu’ils décrivent les aspects fonctionnels .

**3.3.1 Le diagramme des cas d'utilisation**

**Définition de diagramme de cas d’utilisation**

Le diagramme des cas d’utilisation représente les relations entre les acteurs et les fonctionnalités du système. Le diagramme des cas d'utilisation montre l'ensemble des processus du domaine d'étude. Chaque processus, ou plus précisément, chaque variante de processus, sera modélisé au moyen d'un diagramme de séquence et/ou d'un diagramme d'états-transitions et/ou d'un diagramme d'activités.

C’est un ensemble d’actions réalisées par le système en réponse à une action d’un acteur. Cet ensemble des cas d’utilisation décrivent le but du système(les objectifs).

Il permet de formaliser les besoins et de modéliser les services offerts par le système, c’est donc une vue du système dans son environnement extérieur, il modélise à la fois des activités (fonctionnalités) et des communications (interactions) pour les entités concernées (acteurs).

Le cas d’utilisation est tout le long du projet. Dans un premier temps, on les crée pour identifier et modéliser les besoins des utilisateurs. Ces besoins sont déterminés à partir des informations recueillies. Généralement lors des rencontres entre informaticiens et utilisateurs.

Le diagramme de cas d’utilisation de notre application

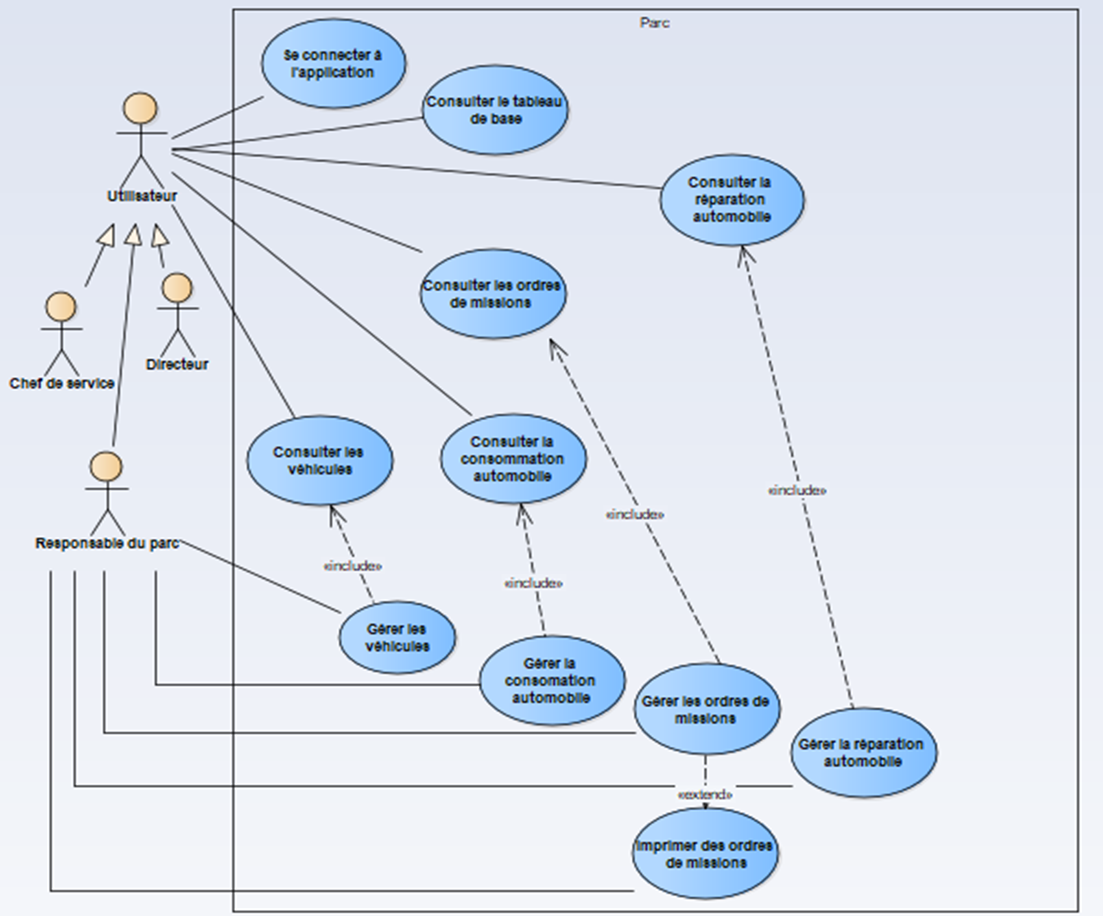


Figure 1: Diagramme de cas d’utilisation

**3.3.2 Les diagrammes de séquence**

**Définition de diagramme de séquence**

C’est une variante du diagramme de collaboration. Il permet de mieux visualiser la séquence des messages en mettant l'accent sur les aspects temporels. Le diagramme de séquence représente la succession chronologique des opérations effectuées par un acteur pour la réalisation d’un cas d’utilisation, il indique les objets que l’acteur va manipuler et les opérations qui font passer d’un objet à l’autre. Il est à invoquer que toutes les opérations mentionnées dans les différents diagrammes de cas d’utilisation se ressemblent, dans ce sens, on présentera les diagrammes de séquences de chaque opération pour une meilleure compréhension.

**Diagramme de séquence pour un scénario d’authentification**

Le diagramme décrit les scénarios possibles lors d’une opération d’authentification. En effet, après l'ajout d’un utilisateur à la base de données de l’application, l’administrateur donne un login et un mot de passe à l’utilisateur. Le système à son tour affichera une interface contenant des champs à remplir, l’utilisateur saisit leur login et leur mot de passe et valide.

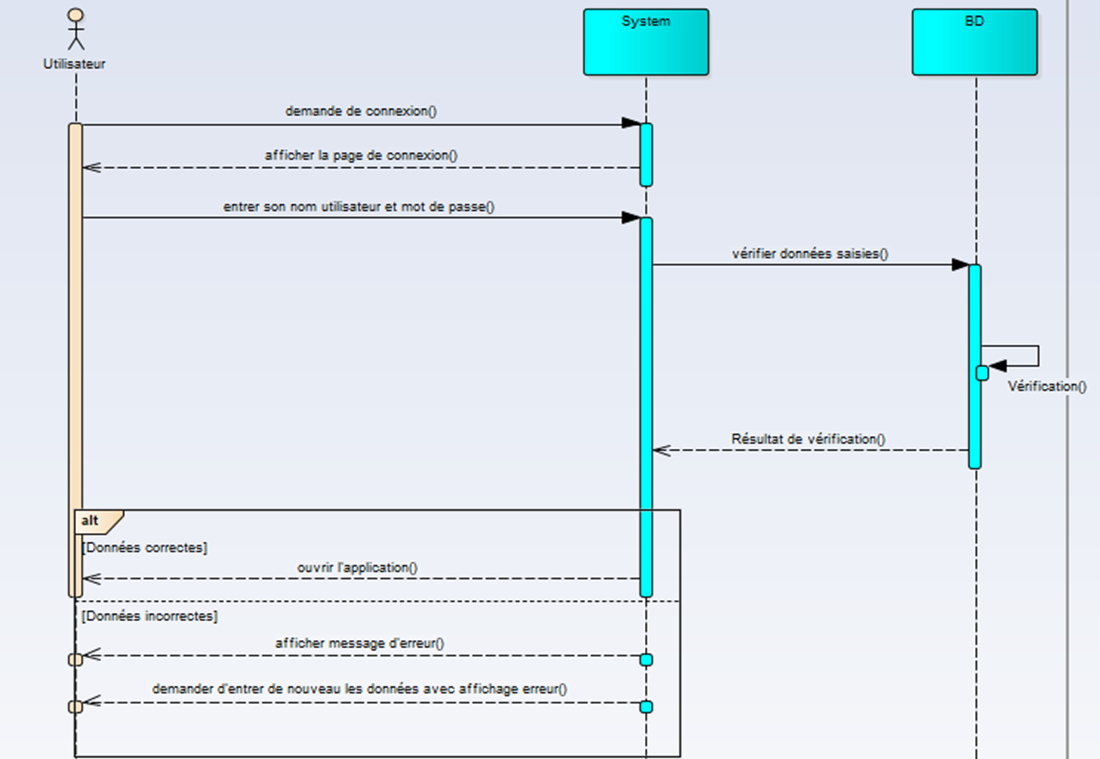


Figure 2 : diagramme de séquence d’authentification

**Diagramme de séquence pour un scénario de consultation et recherche**

Le diagramme, exposé dans la figure 6 ci-dessous, décrit les scénarios possibles lors d’une consultation des interfaces et une opération de recherche.

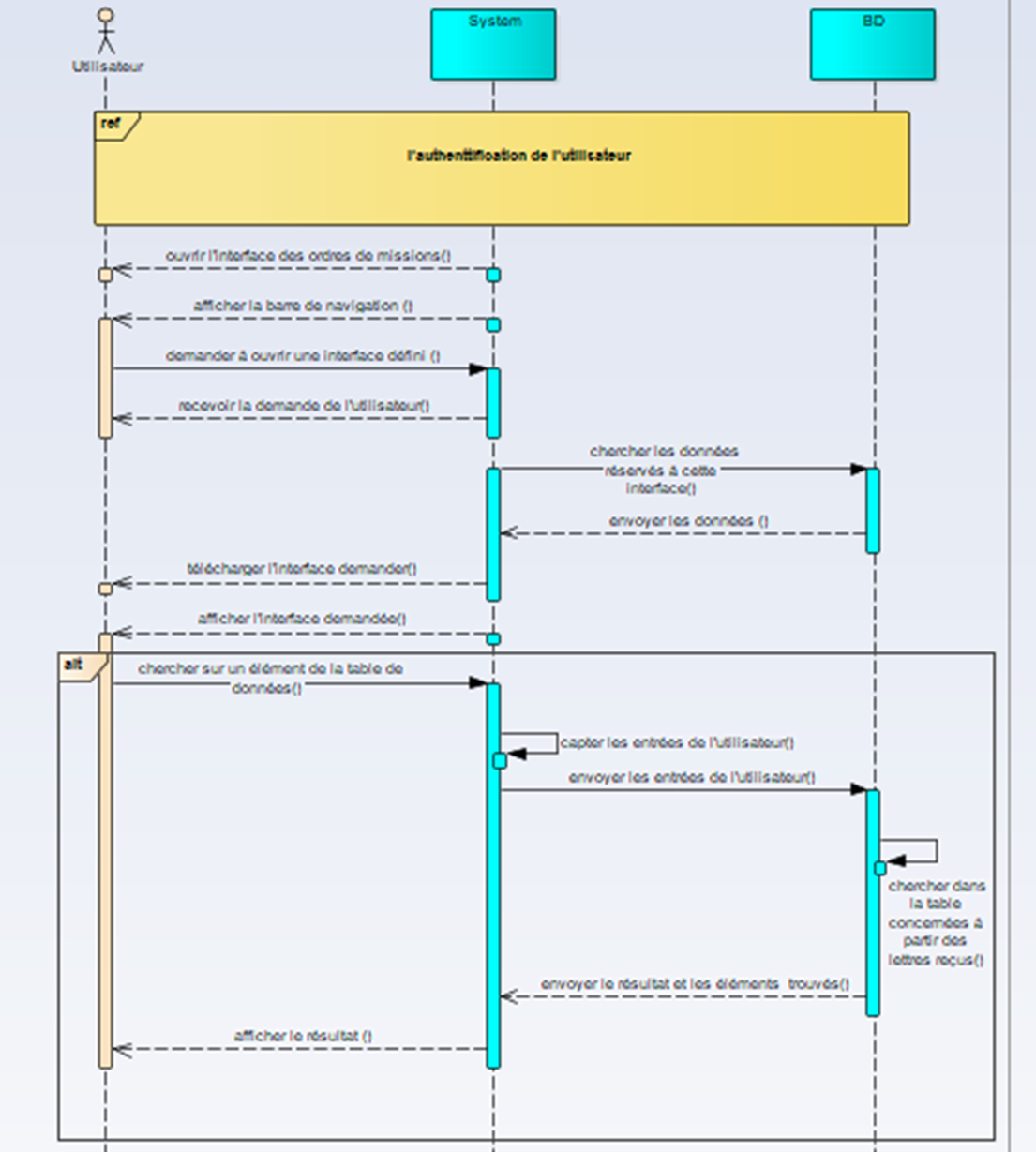


Figure 6 : diagramme de consultation des interfaces et l’option de recherche

**Diagramme de séquence pour un scénario d’ajout**

Le diagramme décrit les scénarios possibles lors d’une opération d’ajout d’un élément. En effet, si le responsable du parc est dans leur session et il choisit l’ajout d’un élément. Le système à son tour affichera une interface contenant des champs à remplir, le responsable du parc saisit ses informations et valide.

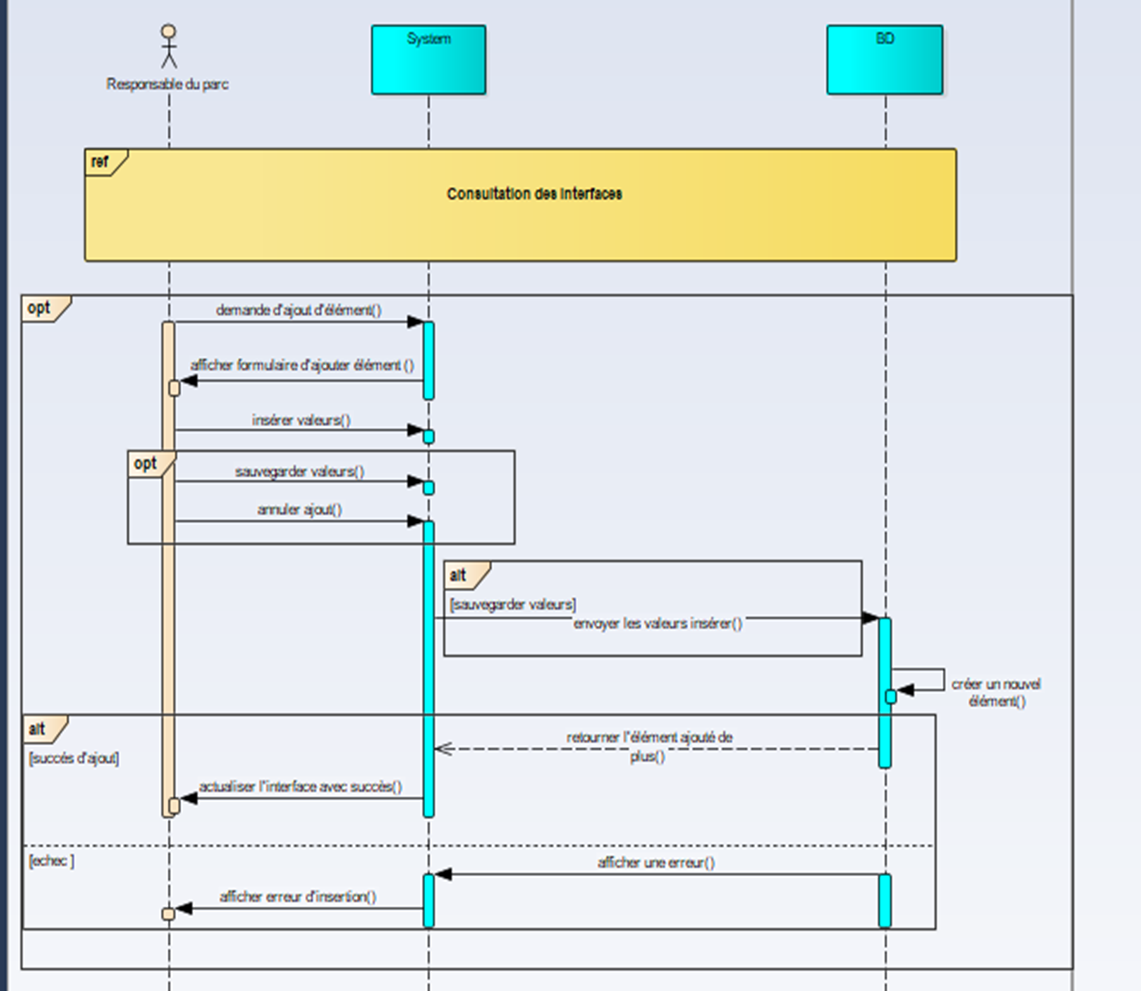


Figure 3 : diagramme de séquence d’ajout d’un élément

**Diagramme de séquence pour un scénario de suppression**

Le diagramme décrit les scénarios possibles lors d’une opération de suppression d’un élément.

Ce scénario peut être utilisé par le responsable du parc. En effet, si le responsable du parc est dans sa session et il veut supprimer un élément de leur base. Le système à son tour retourne une alerte de confirmation de suppression et le responsable du parc prend une décision (termine la tâche de suppression ou non).

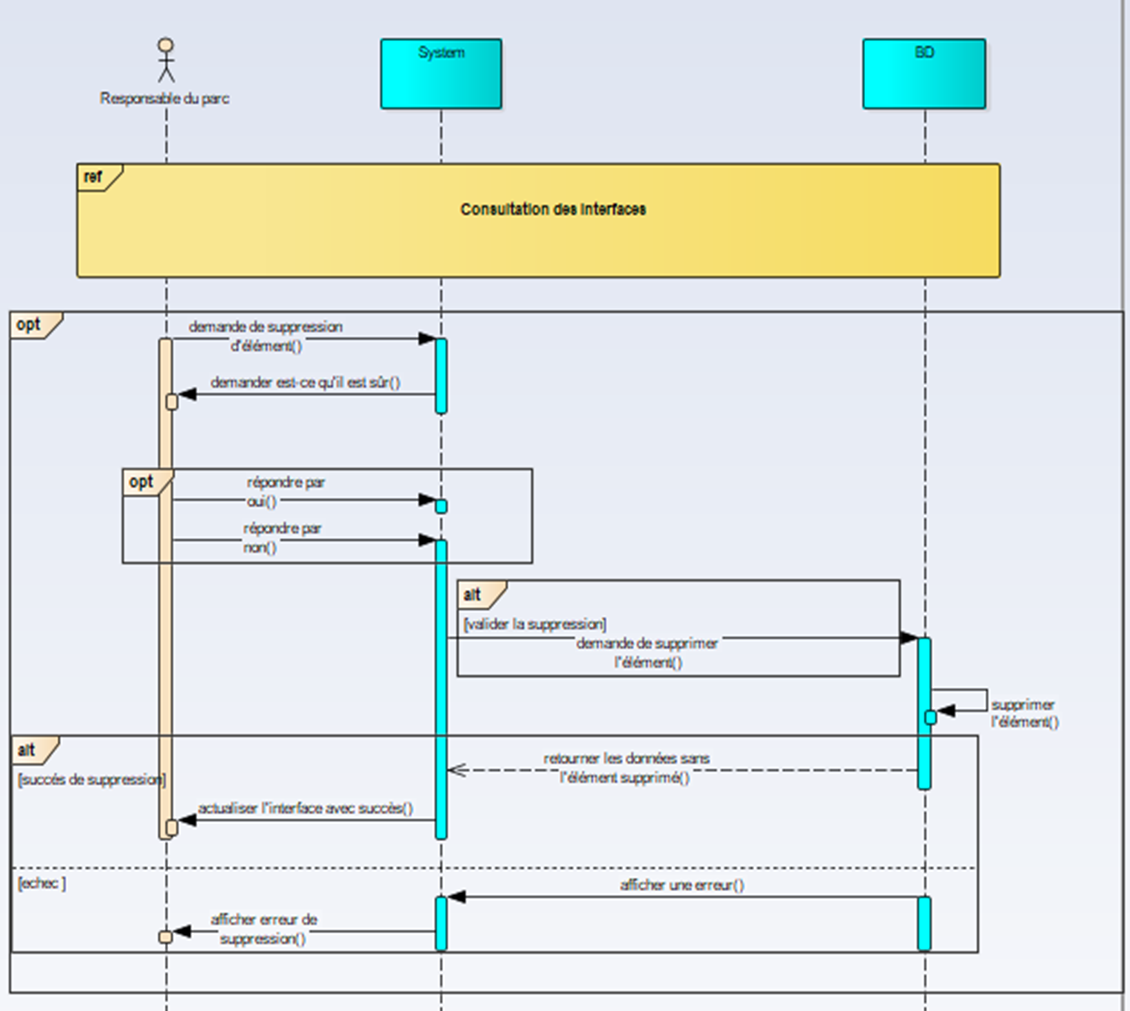


Figure 4 : diagramme de séquence de suppression d’un élément

**Diagramme de séquence pour un scénario de modification des informations d’un élément**

Le diagramme décrit les scénarios possibles lors d’une opération de modification. Le système, à son tour, cherche les anciennes informations concernant cet élément et affiche une interface contenant des champs déjà remplis par les anciennes informations de l’élément. En cas où le responsable veut changer ces informations, il saisit les nouvelles informations et valide.

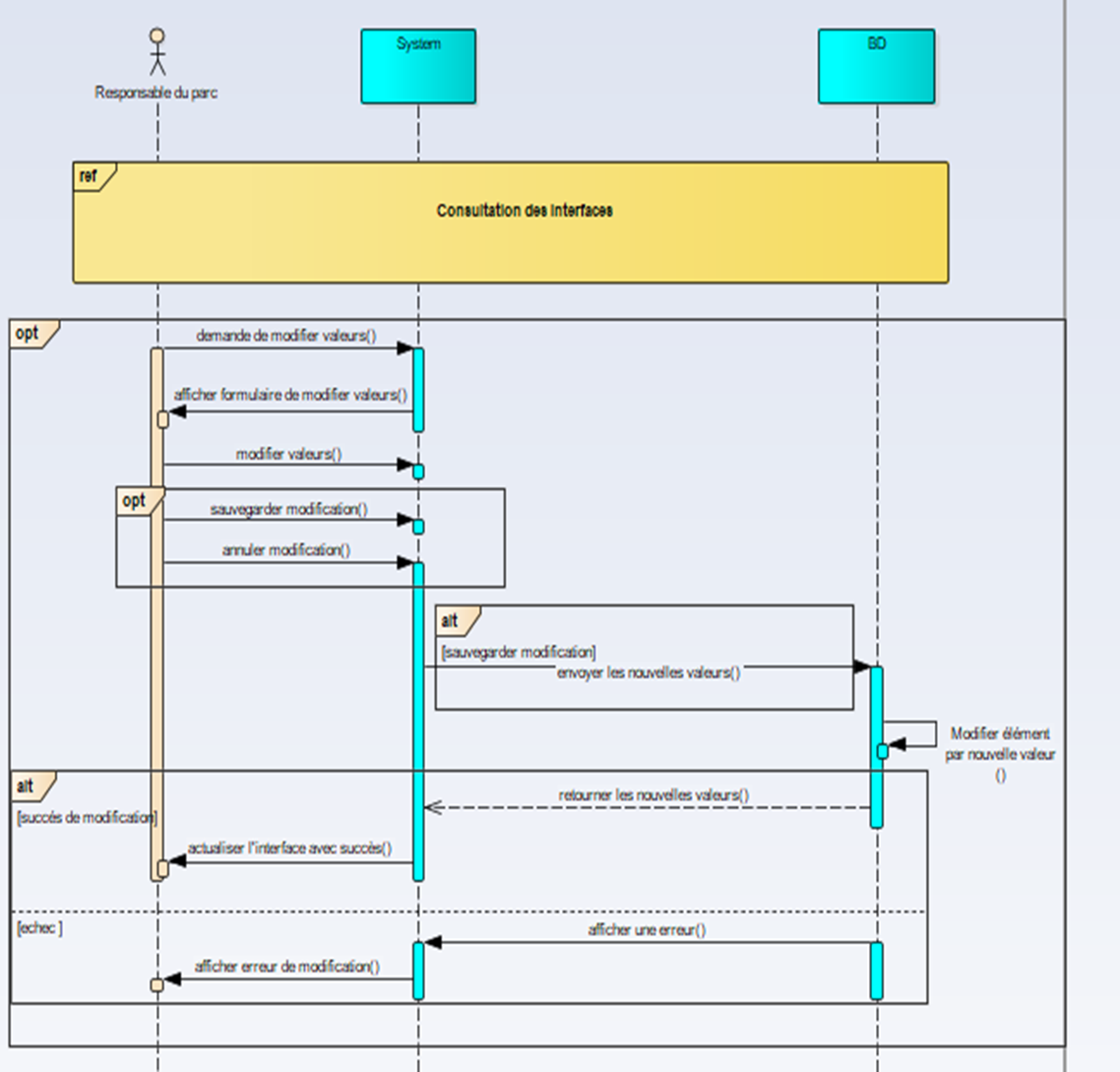


Figure 5: diagramme de séquence de modification d’un élément

**3.3.3 Le diagramme d’activités**

**Définition de diagramme d’activité**

Les diagrammes d’activité se focalisent sur le comportement internes des opérations et des cas d’utilisation du système, Ils sont très utiles dans la phase de réalisation puisqu’ils permettent une description plus précise des opérations ainsi qu’une meilleure illustration des cas d’utilisation.

**Le diagramme d’activités de l’application**

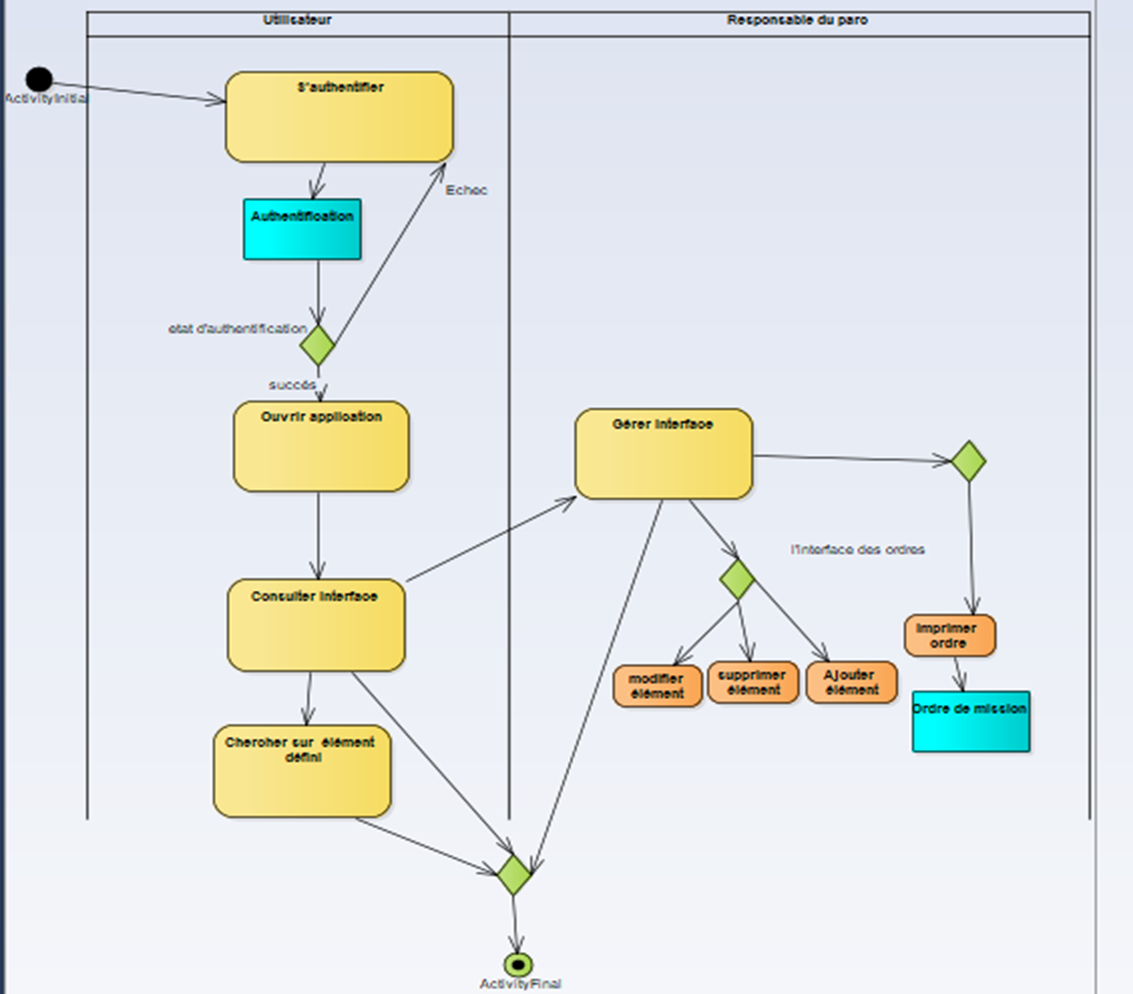


Figure 7 : diagramme d’activité de l’application

**3.3.4 Le diagramme de classes**

**Définition de diagramme de classe**

Il représente la structure statique d'un système. Il contient principalement les classes ainsi que leurs associations mais on peut aussi y trouver des objets. En pratique, l'intérêt majeur du diagramme de classes est de modéliser les entités du système d'information.

**Le diagramme de classes de l’application**

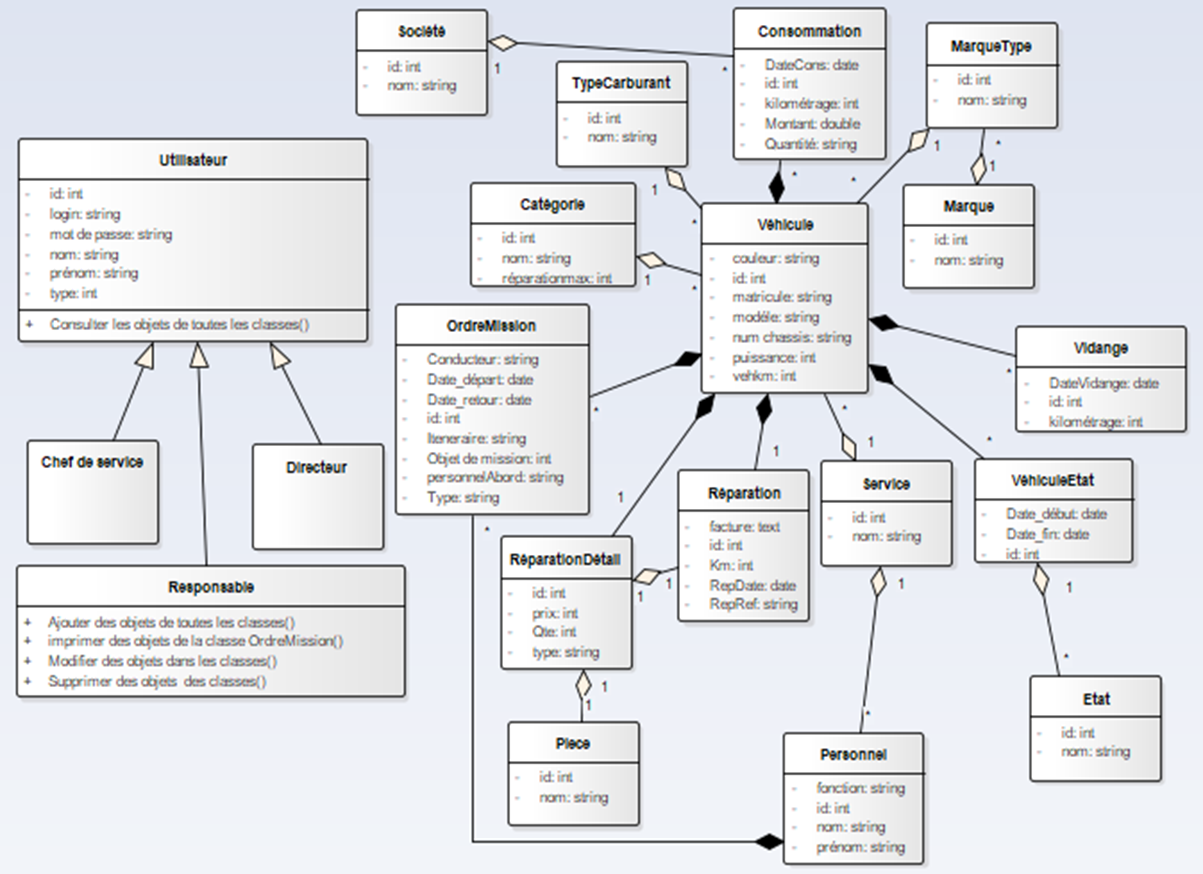


Figure 8 : diagramme de classe de l’application

**3.4 Modèle logique de données (MLD)**

**Définition du modèle logique**

Le modèle logique des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements. Ainsi, le modèle logique est dépendant du type de base de données utilisé.

**Le modèle logique de l’application**

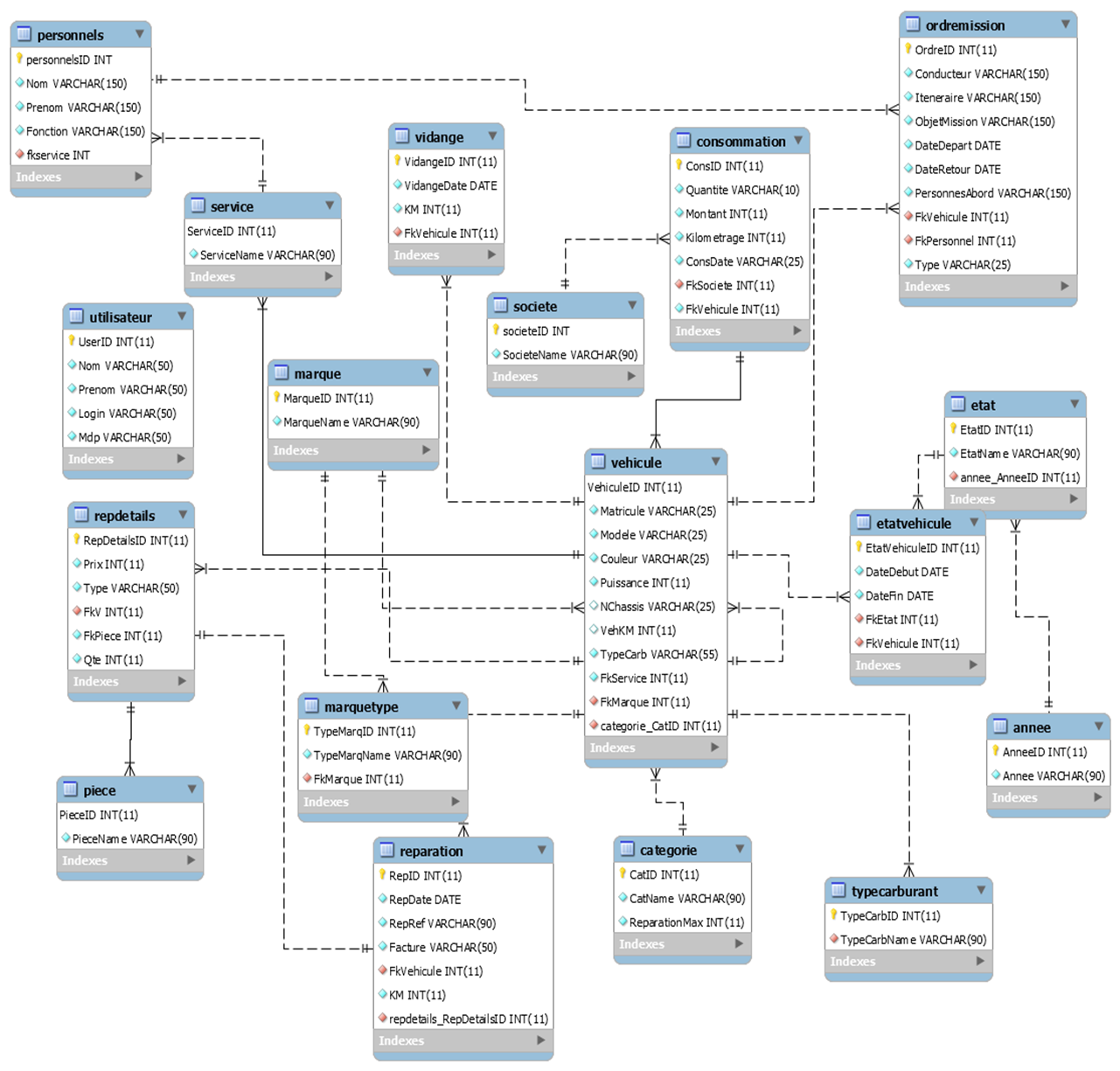


Figure 9 : le modèle logique de l’application

**Conclusion**

Au cours de ce chapitre, nous avons décrit les différents éléments de notre conception pour mettre en place notre application. Nous avons fixé le choix de l’architecture de notre système. Nous avons par la suite élaboré le diagramme de classe, le diagramme de cas d’utilisation, le diagramme de séquence, le diagramme d’activités et le modèle logique pour expliquer mieux les interfaces de notre application.

**Conception des interfaces**

**Outils Utilisés**

Figma

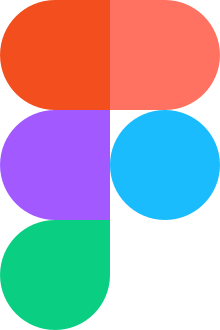
****

Figure 10 : logo Figma

**Figma** est un [éditeur de graphiques vectoriels](https://fr.wikipedia.org/wiki/%C3%89diteur_d%27image_vectorielle) et un outil de [prototypage](https://fr.wikipedia.org/wiki/Prototypage_logiciel). Il est principalement basé sur le web, avec des fonctionnalités hors ligne supplémentaires activées par des applications de bureau pour [Mac OSX](https://fr.wikipedia.org/wiki/MacOS) et [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows) (par exemple : vous pouvez utiliser des polices locales sur la version desktop). Les Figma Mirror companion apps pour [Android](https://fr.wikipedia.org/wiki/Android) et [iOS](https://fr.wikipedia.org/wiki/IOS) permettent de visualiser des prototypes Figma sur des appareils mobiles. L'ensemble des fonctionnalités de Figma est axé sur l'utilisation dans la conception de l'interface utilisateur et de l'expérience utilisateur, en mettant l'accent sur la collaboration en temps réel.

Grâce à cet outil nous avons créé des sketchs pour concevoir les interfaces adoptées pour notre application de gestion du parc automobile.



Figure 11 : la page accueil

C’est la première page de notre plateforme, il donne une idée générale sur son utilité grâce à la zone de description puis on y trouve un bouton pour s’authentifier et accéder à la page de des ordres de missions.

La troisième section comporte des images et des liens en bas qui permettent d’aller vers la page désirée.



Figure 12 : la page d’authentification

Après avoir cliqué sur le bouton s’authentifier dans le navbar ou dans la page d’accueil , la page d’authentification sera affichée .

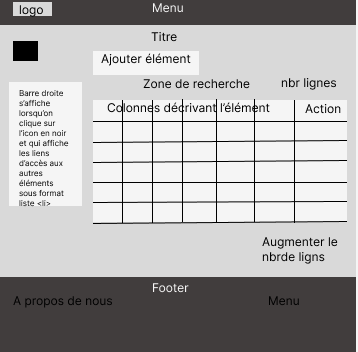


Figure 13 : la page de gestion

C’est le type des pages qui se présentent le plus, pour chaque élément à gérer on trouve une, cette page contient un data table pour gérer les lignes.

Pour la colonne Action on trouve deux icones une pour modifier et autre pour supprimer la ligne

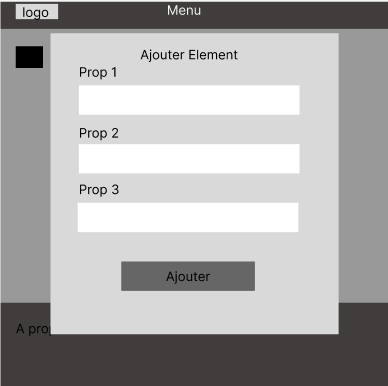


Figure 14 : Modal de l'ajout

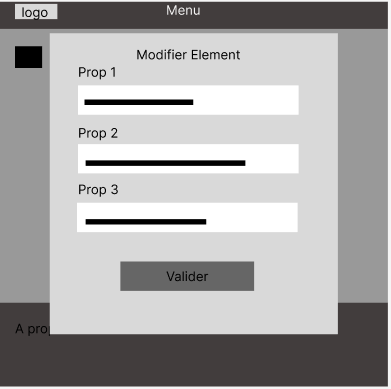


Figure 15 : Modal de la modification

L’ajout ou la modification d’un élément se fait par une fenêtre modale.