



المدرسة الوطنية للعلوم التطبيقية  
ⵜⴰⵎⴰⵔⵜ ⵜⴰⵏⵓⵔⴰⵢⵜ ⵜⴰⵖⴰⵏⵏⵜ ⵜⴰⵙⴰⵎⴰⵏⵜ  
Ecole Nationale des Sciences Appliquées

## Rapport de mini-projet

**Sujet :**

Gestion d'un parc automobile

**Encadré par :**

Mme. Hayat SEMLALI

**Réalisé par :**

Reda HAJJAMI

Mouad BHIH

Mohamed saad JMARI

Année universitaire 2020-2021

# Remerciements

Au début, nous adressons nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mini-projet ainsi à la réussite de cette année universitaire.

Aux membres du jury d'avoir accepté d'évaluer et de juger ce travail, nous voudrions témoigner notre vive reconnaissance pour leur participation au jury.

En guise de preuve de gratitude et de reconnaissance, nous tenons à remercier vivement notre encadrante **Mme Hayat SEMLALI**, qui a été toujours à l'écoute et très disponible tout au long de la réalisation de ce mini-projet, pour l'inspiration, l'aide et le temps qu'elle a bien voulu nous consacrer.

Finalement, on n'oublie pas d'adresser nos sincères remerciements à nos enseignants à l'ENSA Safi.

## Liste des acronymes

**MCD** : Modèle Conceptuel de Données.

**MLD** : Modèle Logique de Données.

**MPD** : Modèle Physique de Données.

**JVM** : Machine virtuelle java

**JFC** : Java foundation classes

**AWT**: Abstract window toolkit

**MVC**: Model Vue contrôleur

**FTP**: File transfer protocol

**CDDL**: Common development and distribution license

## Liste de figures

Figure 1: Diagramme de GANTT .....	4
Figure 2: Modèle conceptuel des données de notre projet .....	8
Figure 3: Modèle logique des données de notre projet.....	9
Figure 4: Modèle physique des données .....	10
Figure 5: Page login du responsable .....	14
Figure 6: Page accueil du responsable .....	15
Figure 7: Interface conducteur du responsable .....	15
Figure 8: Interface mission du responsable .....	16
Figure 9: Interface gestion de véhicules .....	16
Figure 10: Interface login de l'opérateur .....	17
Figure 11: Interface accueil pour operateur .....	18
Figure 12: Interface des conducteurs pour l'opérateur .....	18
Figure 13: Interface mission pour l'opérateur .....	19

# Sommaire

<b>Introduction Générale .....</b>	<b>1</b>
<b>Chapitre I : Présentation du projet .....</b>	<b>2</b>
I-Introduction .....	3
II – La gestion d’un parc automobile .....	3
III – Problématique .....	3
IV – Planning de la réalisation du projet .....	4
V – Présentation de la mission .....	4
VI – Conclusion .....	5
<b>Chapitre II : Conception du Système.....</b>	<b>6</b>
I – Introduction.....	7
II – Présentation de la méthode MERISE.....	7
II.1. Modèle conceptuel des données .....	7
II.2. Modèle logique des données .....	8
II.3. Modèle physique des données.....	10
III – Conclusion .....	10
<b>Chapitre III : Implémentation du système.....</b>	<b>11</b>
I-Introduction .....	12
II- Les outils et langages utilisés .....	12
II.1 – Java .....	12
II.2 – Java SWING.....	12
II.3 – MySQL.....	13
II.4 – XAMPP server .....	13
II.5 – NetBeans .....	13
II.6 – PowerAMC.....	13
III-Les Interfaces de l’application.....	14
1-Section responsable .....	14
2-Section operateur.....	17
IV-Conclusion.....	19
<b>Conclusion Générale.....</b>	<b>20</b>
<b>Webographie .....</b>	<b>21</b>

# Introduction Générale

Durant le deuxième semestre de notre 1<sup>ère</sup> année du cycle d'ingénieur en Génie Informatique à l'ENSA de SAFI, nous avons acquis le bagage nous permettant de réaliser notre projet informatique : **Application Desktop de la gestion d'un parc automobile d'une entreprise.**

Ce projet s'intéresse à la conception et à la mise en place d'une application Desktop qui informatise les opérations liées à la gestion des voitures, la gestion des conducteurs et l'affectation des voitures aux chauffeurs.

Dans le premier chapitre nous allons donner une représentation générale concernant notre projet. Le deuxième chapitre contient l'étude conceptuelle de notre application. Et dans le dernier chapitre nous allons présenter les interfaces de notre application avec les outils et les langages utilisées.

# **Chapitre I : Présentation du projet**

## I-Introduction

L'informatique est une discipline à la mode, très variée et très riche. Elle est devenue indispensable dans tous les domaines, vue les avantages majeures qu'elle offre. Elle rend le travail plus facile, plus précis et surtout bien géré et provoque une nouvelle révolution de l'organisation du travail.

L'informatique occupe évidemment une grande place dans le domaine de transport et en particulier, la gestion des parcs automobiles.

## II – La gestion d'un parc automobile

La gestion d'un parc de véhicules est un métier de la logistique, essentiellement, est en charge de la gestion administrative, logistique (disponibilité, gestion des capacités de transport ...) et technique (Achat de prestation de maintenance ou de transport) des véhicules d'une entreprise.

Pour chaque opération de transport, il est essentiel de définir : le lieu de départ, le lieu d'arrivée et les principaux points d'escales qui constituent l'itinéraire.

Les responsabilités du gestionnaire de parc automobile, Essentiellement administratives Planification du transport, Emission des ordres de missions, Gestion des heures de travail (pointage des chauffeurs), Emission des documents de transport, Contrôle, suivi et actualisation des documents des véhicules et des documents des Conducteurs, Suivi du contrôle technique des véhicules, Gestion des entrées et sorties des véhicules, ainsi que le contrôle des processus de gestion informatique de l'information

## III – Problématique

La gestion manuelle du Parc Automobile est difficile compte tenu de la diversité des tâches à accomplir et du nombre important de chauffeurs qui ont besoin de véhicules pour leurs missions à l'intérieur du pays.

En fait, parmi ces tâches, la tâche de gestion de la documentation (Certificat de visite, Attestation d'importation temporaire) qui nécessite beaucoup d'attention, et de temps pour une vérification manuelle et régulière des délais de validité.

En outre, le non informatisation de la gestion du Parc Automobile rend la circulation des informations très lente. En effet, certains ordres de livraisons venant de l'entreprise sont directement adressés au Chef du Parc qui ne les transmet pas toujours à temps aux chauffeurs pour traitement.



En plus, l'absence d'une base de données et le non archivage des documents papiers utilisés pour les différentes tâches rendent quasiment impossible l'établissement d'avoir des statistiques fiables.

Aussi, l'emploi du temps des chauffeurs est très dynamique ce qui occasionne de nombreuses modifications sur le tableau de planning entraînant ainsi des ratures sur celui-ci et par la même occasion son illisibilité. Par ailleurs, les responsables du parc sont souvent en déplacement (mission), ce qui retarde les mises à jour du tableau de planning.

Donc Pour garantir la livraison de leur produit en bonne condition, une entreprise peut mettre en place une application desktop lui permettra de gérer son parc automobile.

## IV – Planning de la réalisation du projet

La figure ci-dessous illustre le planning de réalisation des différentes taches de notre projet.

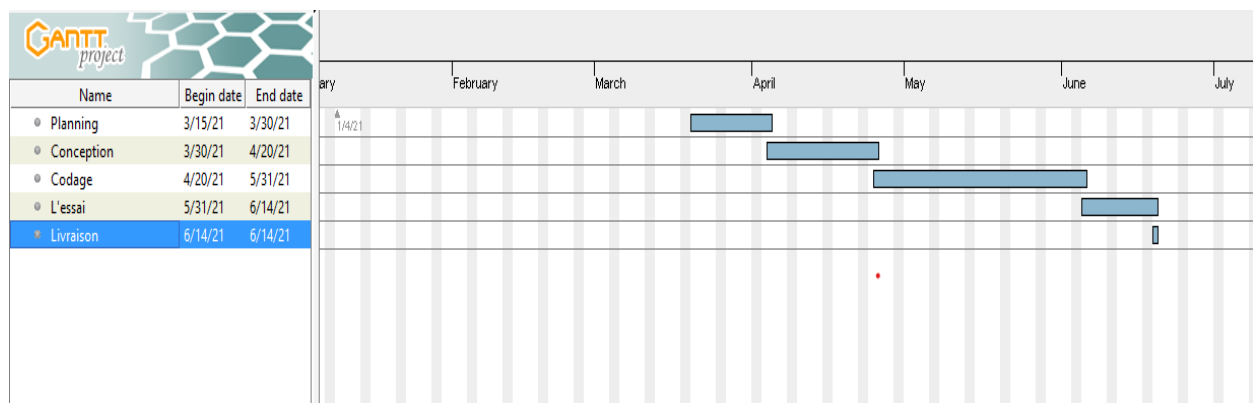


Figure 1: Diagramme de GANTT

## V – Présentation de la mission

Notre mission dans le cadre de ce projet est de créer une application permettant de gérer un parc d'automobile d'une entreprise.

Il s'agit de résoudre les problèmes rencontrés dans la gestion manuelle des ressources et prendre en compte les perspectives d'évolution.

Pour ce faire, notre travail consistera à mettre en place un système dont les fonctionnalités :

- Une meilleure répartition des véhicules entre les différents chauffeurs pour leurs missions ;

- Une bonne gestion du personnel du Parc automobile ;
- Un accès et une circulation des informations en temps réel ;
- Un suivi efficace des conducteurs ;
- L'archivage, la sécurité et la confidentialité des données ;
- La rapidité, la fiabilité et la facilité des traitements.

## VI – Conclusion

Dans ce chapitre nous avons présenté la gestion de parc automobile d'une manière générale ainsi que le contexte général de notre projet.

Dans le chapitre suivant, nous allons concevoir le système d'information pour la gestion de notre projet.

## **Chapitre II : Conception du Système**

## I – Introduction

Dans ce chapitre, nous allons présenter les différentes étapes de la conception de notre système. Nous allons faire un appel au Merise qui se constitue du MCD (Model conceptuel des données), le MLD (Model logique des données) et le MPD (Model physique des données).

## II – Présentation de la méthode MERISE

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information.

La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles : conceptuel, logique et physiques.

La séparation des données et des traitements assure une longévité au modèle.

### II.1. Modèle conceptuel des données

Le modèle conceptuel des données a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

La figure ci-dessous illustre le modèle conceptuel de notre projet :

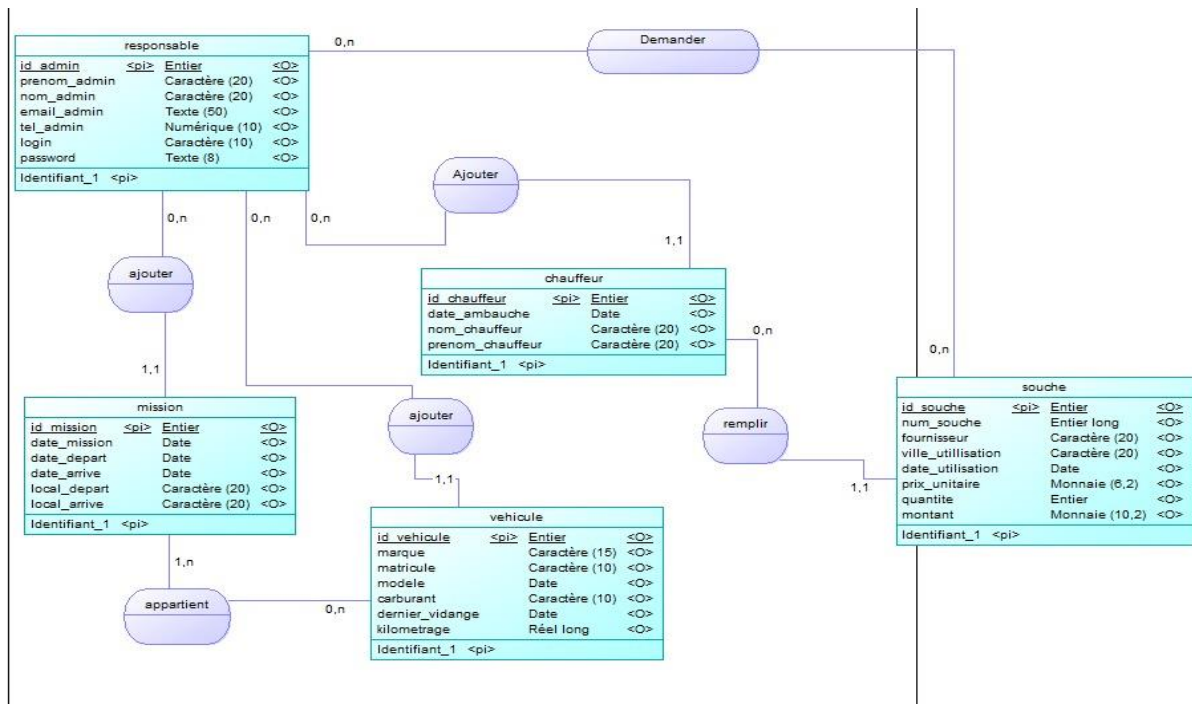


Figure 2: Modèle conceptuel des données de notre projet

## II.2. Modèle logique des données

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, ce modèle logique est dépendant du type de base de données utilisée.

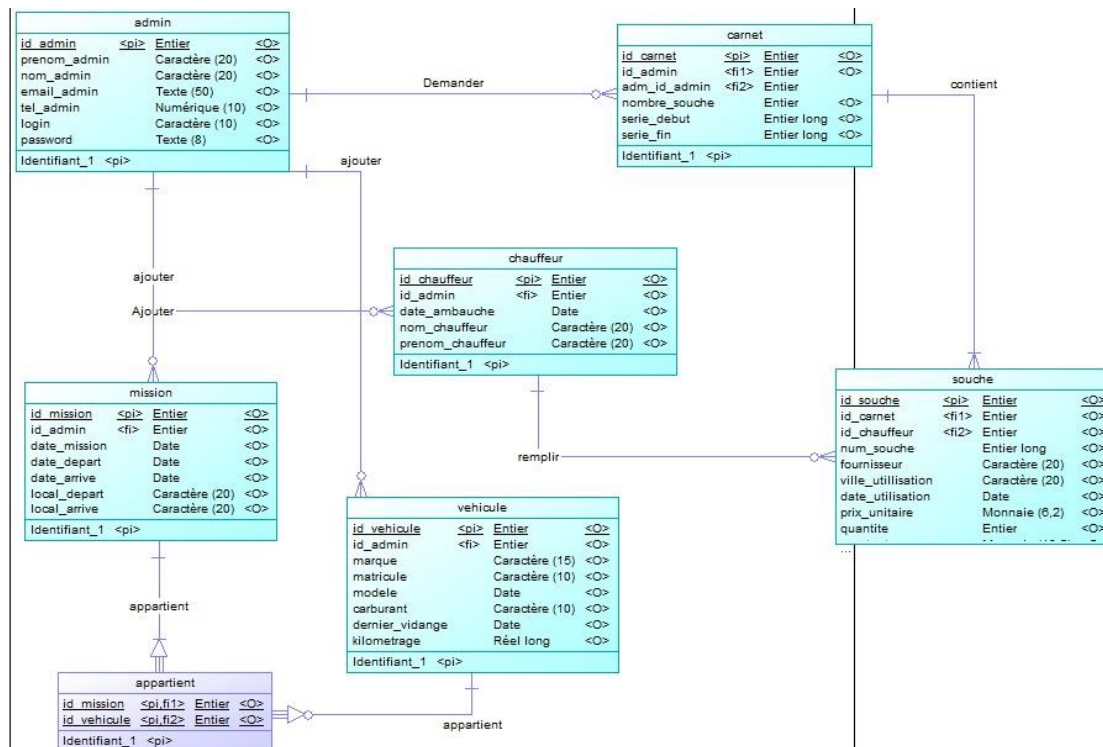


Figure 3: Modèle logique des données de notre projet

### II.3. Modèle physique des données

La dernière étape de la méthode merise consiste à transformer le Modèle Logique des données (MLD) en Modèle Physique (MPD), puis la génération d'un script qui sera implémenté par la suite dans la base des données.

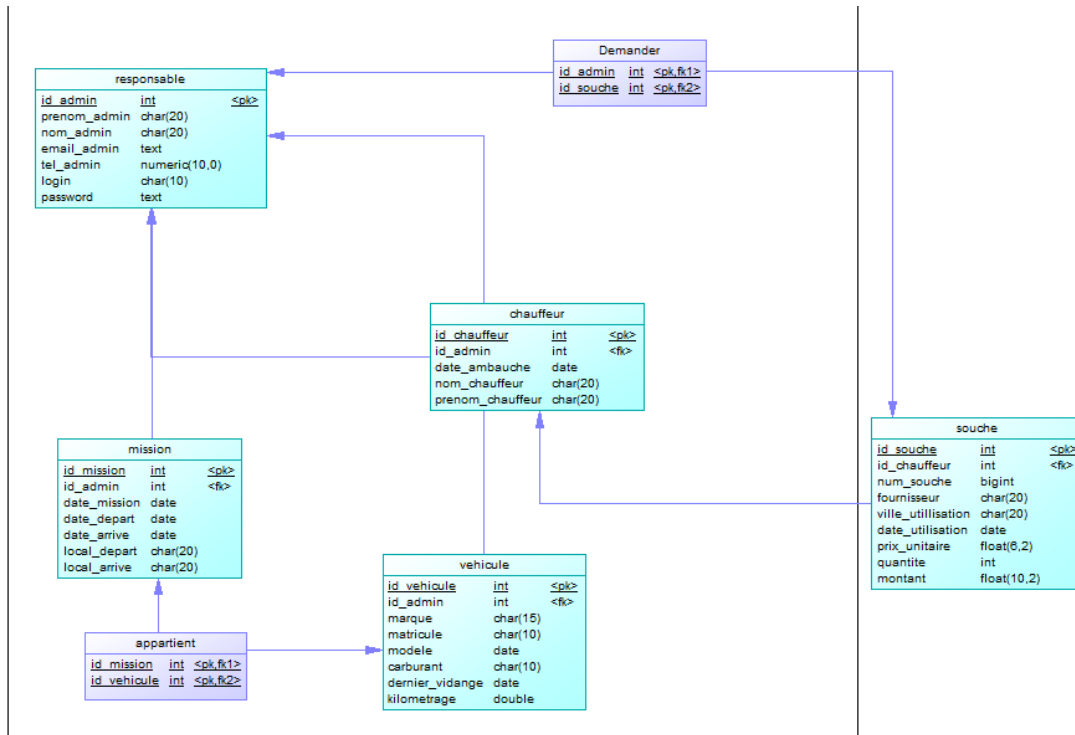


Figure 4: Modèle physique des données

## III – Conclusion

Dans ce deuxième chapitre, on a pu concevoir un système d'information pour la gestion de parc automobile en se basant sur la conception de la base de données Merise.

On présentera dans le chapitre suivant la réalisation du système de gestion de notre projet.

## **Chapitre III : Implémentation du système**



## I-Introduction

Nous présentons dans ce chapitre les outils de développement adoptés dans la réalisation de l'application desktop pour la gestion d'un parc automobile, ainsi que les captures des interfaces les plus importantes de notre application.

## II- Les outils et langages utilisés

### II.1 – Java

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.

### II.2 – Java SWING

Swing est une bibliothèque graphique pour le langage de programmation Java, faisant partie du package Java Foundation Classes (JFC), inclus dans J2SE. Swing constitue l'une des principales évolutions apportées par Java 2 par rapport aux versions antérieures.

Swing offre la possibilité de créer des interfaces graphiques identiques quel que soit le système d'exploitation sous-jacent, au prix de performances moindres qu'en utilisant Abstract Window Toolkit (AWT). Il utilise le principe Modèle-Vue-Contrôleur (MVC, les composants Swing jouent en fait le rôle de la vue au sens du MVC) et dispose de plusieurs choix d'apparence pour chacun des composants standards.



### II.3 – MySQL

Abréviation de "MyStructured Query Language" ou mon langage de requêtes structuré, est un système de gestion de bases de données relationnelles très employée sur le Web, souvent en association avec PHP[3]



### II.4 – XAMPP server

XAMPP [1] est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.



### II.5 – NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).



### II.6 – PowerAMC

PowerAMC [2] est un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées



### III-Les Interfaces de l'application

L'interface est le lieu où s'opère la communication avec le système informatique. Ainsi, il est fondamental de respecter les guides de styles et les principes ergonomiques de base afin d'accélérer la création du modèle mental de l'utilisateur et faciliter les tâches de ce dernier.

Notre application est divisée en deux sections principales, une section qui concerne l'administrateur ou bien le responsable, l'autre partie sa concerne l'opérateur.

#### 1-Section responsable

##### Interface d'authentification

Dans l'interface d'authentification de notre application, le responsable peut s'authentifier avec le nom et un mot de passe pour qu'il puisse accéder à l'interface d'accueil illustrée par la figure ci-dessous :

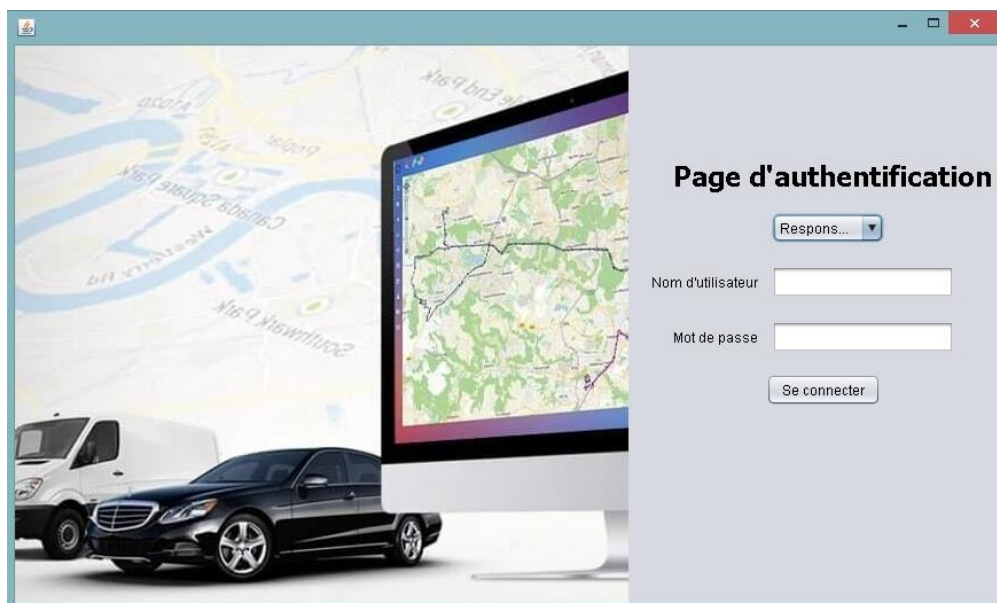


Figure 5: Page login du responsable

### Interface d'accueil

L'interface d'accueil contient un message de Bienvenue et une simple définition de l'application.

La figure ci-dessus présente un aperçu de cette interface :

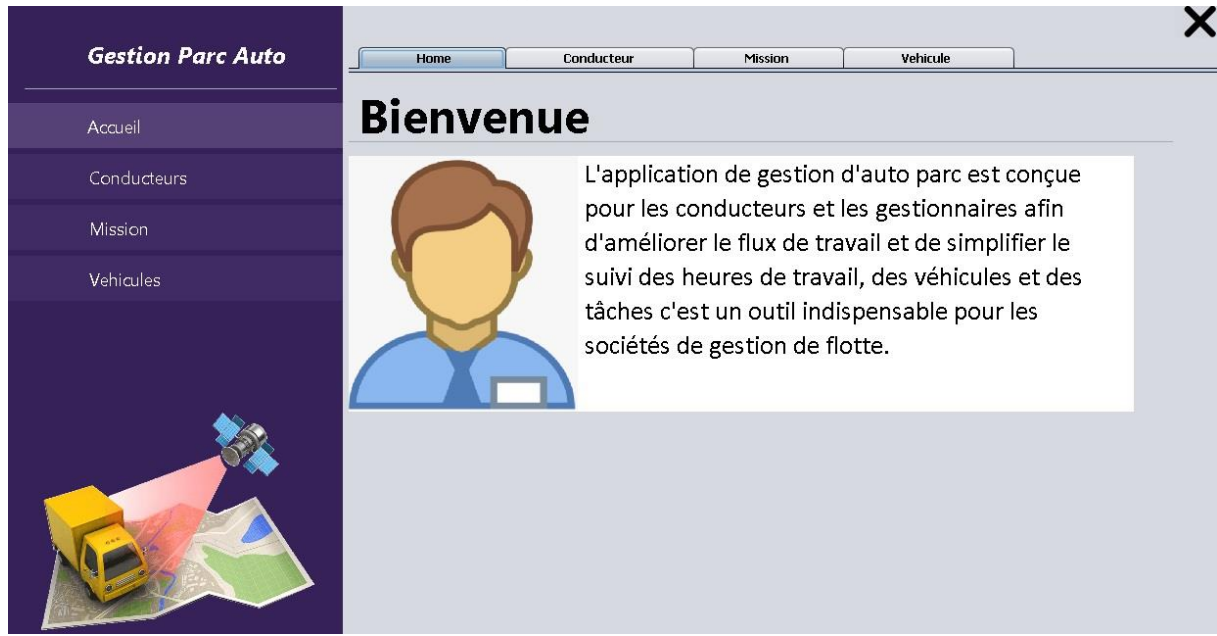


Figure 6: page accueil du responsable

### Page du conducteur

Le page conducteur c'est la partie où le responsable(admin) peut saisir les données qui concernent les nouveaux conducteurs dans la base de données et modifier ceux des conducteurs déjà existants.

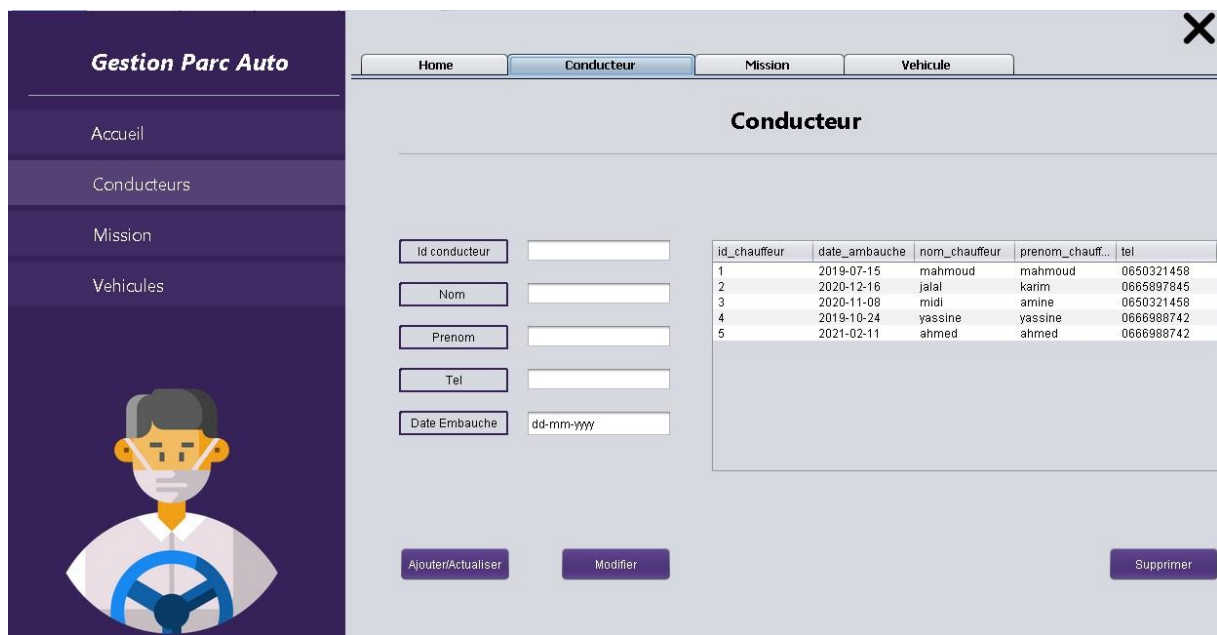


Figure 7: Interface conducteur du responsable

### Interface Mission

Dans l'interface Mission, le responsable peut consulter les missions ajoutées par l'opérateur.

La figure ci-dessous illustre un aperçu de cette interface :

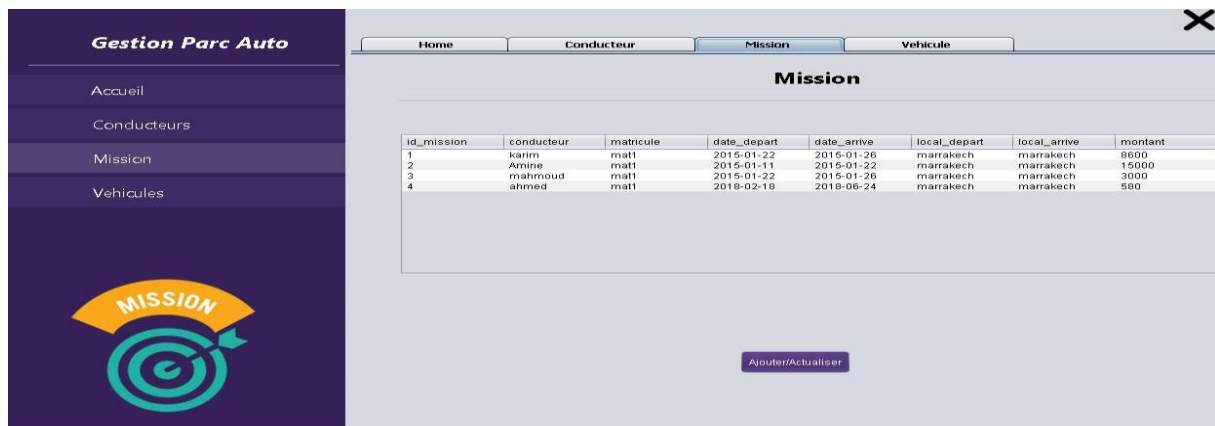


Figure 8: Interface mission du responsable

### Interface Gestion des Véhicules

Dans cette interface le responsable saisit les informations concernant les véhicules disponibles dans l'entreprise.



Figure 9: interface gestion de véhicules

## 2-Section opérateur

### Interface d'authentification

L'opérateur peut s'authentifier avec son nom d'utilisateur et son mot de passe pour qu'il puisse aussi accéder à l'interface d'accueil.

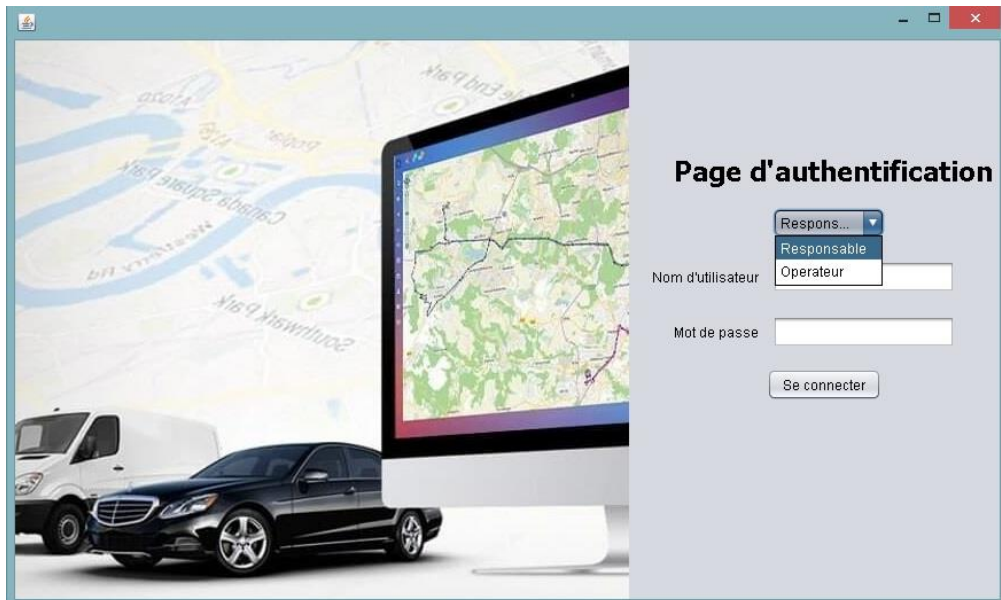


Figure 10: interface login de l'opérateur

### Interface Accueil

L'interface d'accueil contient un message de Bienvenue et une simple définition de l'application.

Depuis l'interface d'accueil, nous pouvons voir que l'opérateur peut accéder deux autres interfaces : Conducteurs et Missions.

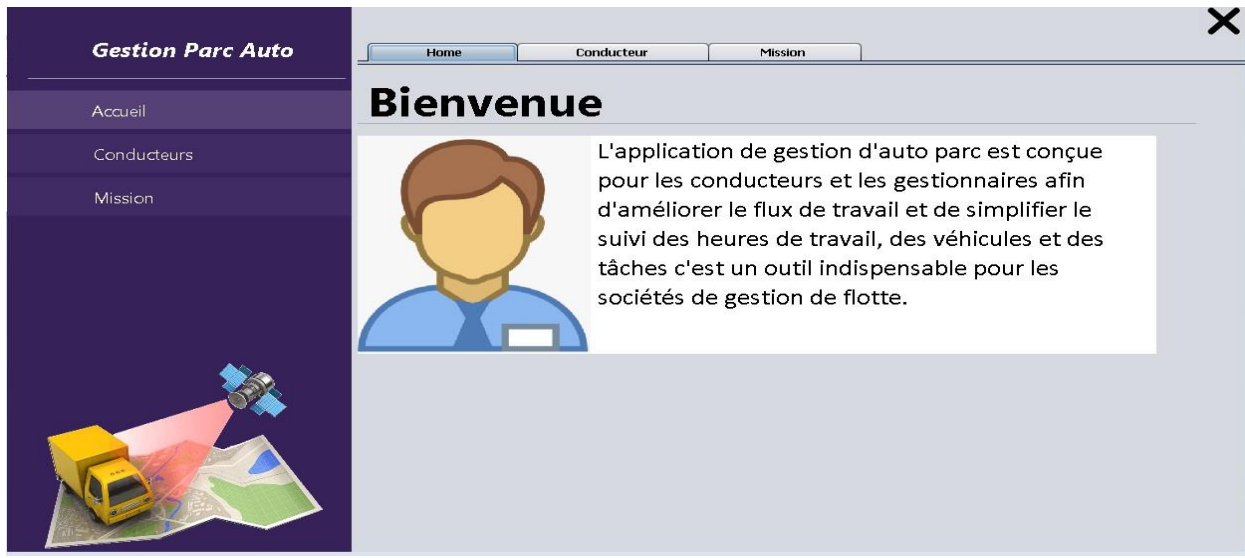


Figure 11: Interface accueil pour operateur

### Interface Conducteurs

L'opérateur a la possibilité d'ajouter les informations concernant un conducteur mais il ne peut pas consulter la liste des conducteurs.

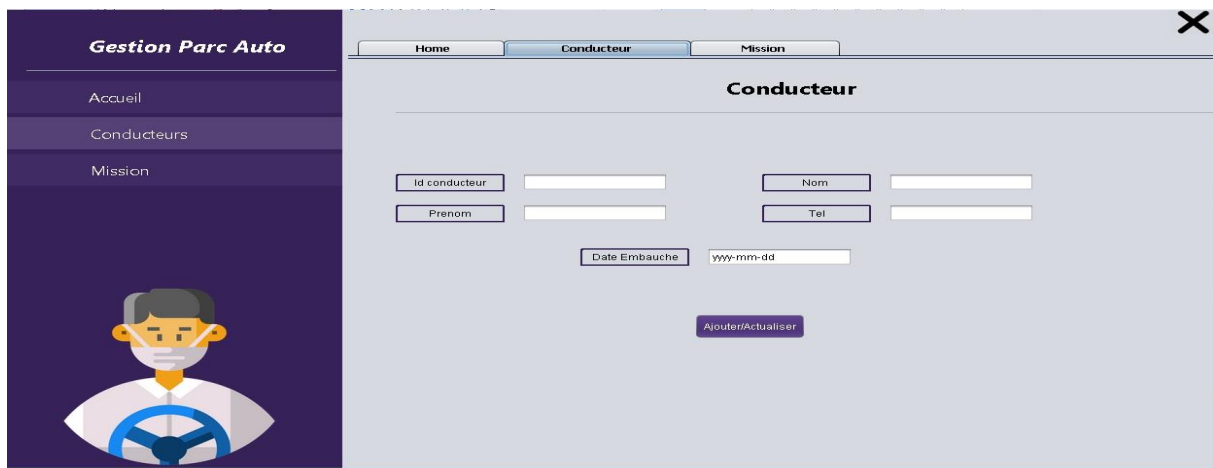


Figure 12: Interface des conducteurs pour l'opérateur

### Interface Mission

Dans cette interface, l'opérateur peut saisir les informations pour chaque mission mais sans consulter la liste des missions.

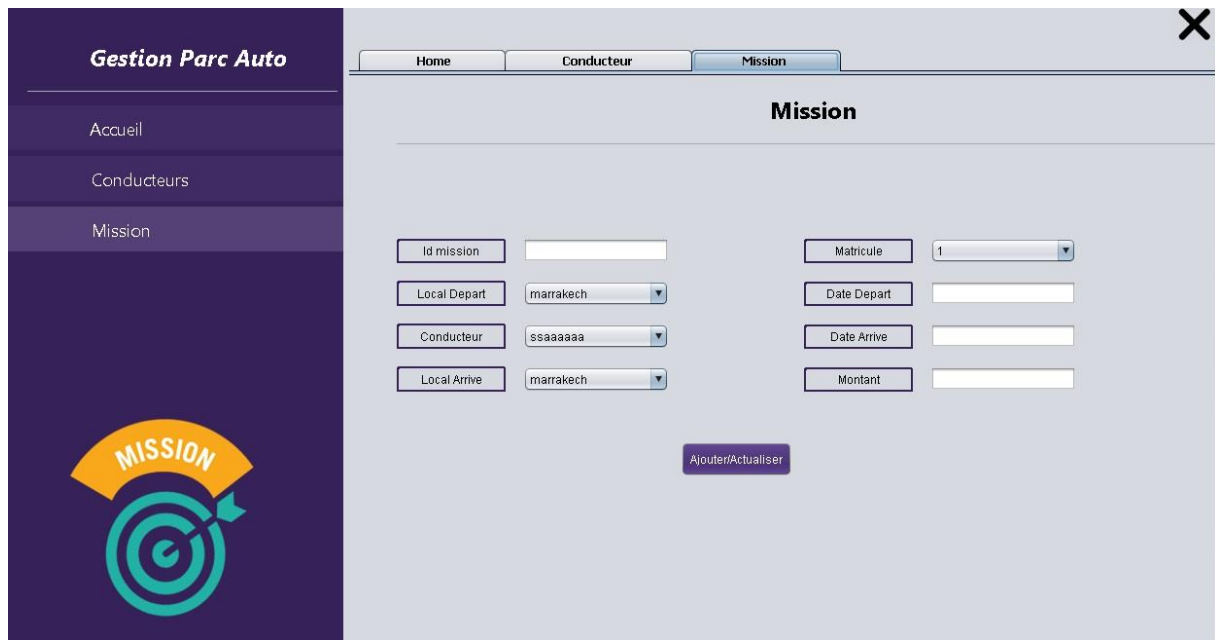


Figure 13: Interface mission pour l'opérateur

### IV-Conclusion

Dans ce chapitre, nous avons présenté les différents outils et langages utilisés pour le développement de notre application, ainsi que la partie réalisation de notre projet à travers les principales interfaces du projet.



## Conclusion Générale

L'objectif de ce mini-projet était de concevoir et développer une application Desktop pour la gestion du parc automobile.

Ce mini projet nous a donné la possibilité de découvrir de nouvelles approches de développement et d'utiliser de nouvelles technologies, telles que Java Swing.

Ce travail a représenté pour nous une véritable occasion pour approfondir nos connaissances tout en rassemblant nos acquis théoriques à l'environnement pratique en utilisant la méthode MERISE pour la conception comme langage de modélisation et Java Swing pour le développement des interfaces de notre application

## Webographie

[1] XAMPP (standard-du-web.com), consulté le 14/06/2021

[2] PowerAMC : définition et explications (techno-science.net), consulté le 14/06/2021

[3] MySQL — Wikipédia (wikipedia.org), consulté le 14/06/2021