

## Remerciements

*Avant d'exposer le contenu de ce travail Je tiens tout d'abord à remercier notre directeur de **ISST BEJA** ainsi que toute l'équipe pédagogique de l'Ecole Nationale Supérieure d'Ingénieurs de Tunis intervenant dans la formation en Génie Industriel.*

Je tiens à remercier vivement tous les responsables de **BDSI** (Béja) de leur chaleureux accueil et de leurs multitudes aides avec une grande sincérité et gratitude qui ont accepté de me consacrer de leur temps et n'ont pas hésité à me renseigner ou à me documenter ou à me montrer comment procéder pour maintenir la bonne marche du stage.

Mes vifs remerciements s'adressent spécialement à monsieur **Mr.Zouabi Zouhaier**, mon encadreur, pour son aide précieuse et la pertinence de ses conseils, fruits d'une grande expérience.

MERCI 

## Sommaire

<b>Introduction Générale.....</b>	<b>1</b>
<b>Chapitre 1 Cadre général du projet.....</b>	<b>2</b>
Introduction .....	3
I. Présentation générale de la BDSI .....	3
<b>II. Présentation du projet.....</b>	<b>4</b>
1. Critique de l'existant : .....	4
2. Solution proposée : .....	5
Conclusion :.....	5
<b>Chapitre 2 Spécification des besoins.....</b>	<b>6</b>
Introduction .....	7
<b>I. Besoins fonctionnels .....</b>	<b>7</b>
<b>II. Besoins non fonctionnels .....</b>	<b>7</b>
<b>III. Analyse des besoins .....</b>	<b>8</b>
1. Identification des acteurs .....	8
2. Diagramme de cas d'utilisation .....	8
3. Diagramme de séquence système : .....	9
Conclusion.....	13
<b>Chapitre 3 Conception .....</b>	<b>14</b>
Introduction .....	15
I. Présentation UML.....	15
<b>II. Conception détaillée.....</b>	<b>16</b>
1. Diagramme de classe .....	16
Conclusion :.....	16
<b>Chapitre 4 Réalisation et test.....</b>	<b>17</b>
Introduction .....	18
I. Environnement de développement .....	18
1. Environnement matériel .....	18
2. Environnement logiciel.....	18
II. Principales Interfaces graphiques .....	18
Conclusion .....	18
<b>Principales Interfaces graphiques .....</b>	<b>19</b>
<b>Conclusion Générale .....</b>	<b>22</b>



## Liste des figures

Figure 1 Diagramme de cas d'utilisation <<utilisateur>> -----	8
Figure 2 Diagramme de séquence « Ajout catégorie permis »-----	9
Figure 3 Diagramme de séquence << Modifier catégorie permis >>-----	11
Figure 4 Diagramme de séquence << Supprimer catégorie permis >> -----	12
Figure 5 Unified Modeling Language-----	15
Figure 6 Diagramme de classe-----	16

## Liste des tableaux

Tableau 1 Description des acteurs-----	8
Tableau 2 Description diagramme de séquence <<ajouter catégorie permis >> -----	10
Tableau 3 Description Diagramme de séquence << Modifier catégorie permis >> -----	11
Tableau 4 Description Diagramme de séquence << Supprimer catégorie permis >> -----	12

## Introduction Générale

De nos jours, les sociétés cherchent à automatiser et simplifier les tâches, les charger et à les rendre rapide fiable et ordonnées, pour cela, elles essayent d'informatiser toutes leurs activités.

A présent mon travail se situe dans le cadre du stage technicien deuxième année Technologie Informatique spécialité développement à l'Institut Supérieur des Etudes Technologiques BEJA.

J'ai effectué ce stage au sein de la société, il consiste à concevoir et à développer une application pour la gestion d'une auto-école.

Avec l'envergure de la BDSI, l'informatique est devenue une nécessité dans la vie de l'entreprise. Il faut savoir que chaque unité de la BDSI dispose d'un parc de véhicules. L'ensemble des données relatives à ce parc est géré localement, à un niveau régional et à un niveau local, et c'est dans ce sens qu'on a accentué mon travail. Le but de mon stage est de concevoir et développer une application permettant l'informatisation des différentes tâches de gestion de parc véhicule. Ce travail traite les parties importantes à savoir: Gérer les véhicules, ordonner les missions et les disponibilités.

Le rapport est divisé en quatre chapitres:

- ✓ Dans le premier chapitre la présentation de la société et ses activités.
- ✓ Dans le deuxième chapitre, j'ai présenté en premier temps le sujet, une étude de l'existant et par la suite une spécification détaillée des besoins fonctionnels et non fonctionnels de mon application.
- ✓ Dans le troisième chapitre, j'ai défini quelques généralités portant sur la méthode de travail et les outils mettant en évidence la réalisation d'application. J'ai commencé par présenter le langage de modélisation UML avec les principaux diagrammes utilisés pour l'application.
- ✓ Et dans le dernier chapitre la réalisation, prendre en compte l'étude de l'existant, les spécifications des besoins et la conception mentionnées dans le chapitre précédent, j'ai réalisé mes connaissances théoriques et pratiques par l'utilisation des langages et des logiciels.

## **Chapitre 1 Cadre général du projet**

## Introduction

Ce chapitre comprend, généralement, deux parties ; la présentation de la Boite de Développement et Services Informatiques et une étude de l'existant sur les modalités de travail actuelle.

### I. Présentation générale de la BDSI

B.D.S.I Boite de Développement et Services Informatiques, créée en 2016 est une société de développement informatiques spécialisée dans le développement des logiciels sur-mesure, des sites web, des applications web et des applications mobile.

La Boite de Développement et Services Informatiques offre différents services tels que :

- Développement informatique :
  - Développement des logiciels :
  - Développement des sites web :
  - Développement mobile
- Saisie de données :
  - Activités :
    - ✓ Réceptionner les documents à traiter, les vérifier et s'informer des consignes de délai, de nombre...
    - ✓ Saisir ou numériser les documents (factures, contrats, documents techniques, ...)
    - ✓ Contrôler la validité des informations lors de la saisie et corriger les erreurs.

Les différents types de saisie de données :

- ✓ saisie comptable
- ✓ saisie des bons de commande
- ✓ saisie des factures



- ✓ saisie des bulletins de salaire
- ✓ saisie des contrats
- ✓ saisie des courriers
- ✓ saisie d'annuaires
- ✓ saisie d'enquêtes
- ✓ saisie des documents techniques

- Traitement de données

Le traitement de données est réalisé directement sur Internet. L'équipe de BDSI prend en charge les documents transférés par serveur Ftp sécurisé à son plateau de traitement de données en Tunisie.

Les collaborateurs Seront formés à l'utilisation de l'interface du client afin de répondre à son rythme et à la qualité des informations requises.

Les différents types de traitement :

- ✓ Traitement Comptable
  - ✓ Traitement de Factures
  - ✓ Traitement de contrats
  - ✓ Traitement de bons de commande
  - ✓ Traitement de Courrier
  - ✓ Traitement de documents
  - ✓ Traitement de bases de données
  - ✓ Conception de bases de données
  - ✓ Contrôles de données
- Stockage de données
  - Référencement de sites web :
  - Hébergement des sites web
  - Compte mail professionnel
  - Création VPN
  - Maintenance des systèmes informatiques

## **II. Présentation du projet**

Cette partie comprend, généralement, deux parties ; le critique de l'existant et la solution proposée.

### **1. Critique de l'existant :**

Le suivi des inscriptions, de la formation des candidats est réalisé manuellement, en particulier utilisation de fiches de renseignement des candidats ainsi que des registres.

La planification des séances de formation se fait verbalement, en cas de changement de la date ou l'horaire de la séance suite par exemple à l'absence du moniteur ou un autre évènement la secrétaire doit contacter les candidats par téléphone pour signaler le changement.

Après un examen, le candidat doit se déplacer à l'auto-école ou contacter la secrétaire par téléphone pour connaître le résultat de l'examen, de même pour un moniteur s'il veut connaître les résultats de ces candidats.

Tous ces problèmes nous ont motivé pour développer une applications web qui permet d'une coté d'aider la secrétaire et le gérant de l'auto-école à la saisie, l'enregistrement et la diffusion de l'information sur les candidats, planifier les séances de formation et les examens, et d'autre coté offrir aux candidats un moyen pour consulter les informations en ligne sans se déplacer à l'auto-école.

## **2. Solution proposée :**

Développement d'une application web qui permettra la gestion d'une auto-école.

### **Conclusion :**

Dans ce chapitre j'ai eu recours à présenter la société où j'ai eu mon stage, par la suite j'ai critiqué l'existant et finalement j'ai eu recours à citer la solution proposée. Dans le chapitre qui suit je vais spécifier les besoins.

## **Chapitre 2 Spécification des besoins**

## Introduction

Dans ce chapitre, j'ai cité les différents besoins fonctionnels et non-fonctionnels de l'application de gestion d'une auto-école et ai identifié les messages échangés entre les différents acteurs du système.

C'est l'une des étapes les plus difficiles car les intervenant sont multiples (client, utilisateur, développeurs, etc.) et surtout le problème n'est pas encore formalisé. C'est le processus visant à établir quelles fonctionnalités le système doit fournir et à quelles contraintes il fera face.

- Préciser les contraintes de réalisation
- Identifier les services que le système devrait fournir en tenant compte des contraintes
- Réduire les coûts de développement du système. Eviter l'oubli, les allers-retours dans le cycle de vie
- Procurer les critères de validation et vérification

### I. Besoins fonctionnels

J'ai désiré réaliser une application gestion du parc véhicule qui répond aux besoins de l'entreprise et qui assure les principaux services demandés : □ Gestion des véhicules □ Ordre de mission et disponibilité

### II. Besoins non fonctionnels

La prise en compte des besoins non fonctionnels, telles que les contraintes liées au temps ou à la sûreté de fonctionnement, est délicate car les méthodes de développement sont généralement orientées vers la modélisation des besoins fonctionnels. Les besoins non fonctionnels liés au développement de plusieurs applications sont :

- ↗ La sécurité: les informations doit être sécurités
- ↗ La fiabilité: l'application doit être fiable
- ↗ La rapidité: temps d'exécutions
- ↗ La portabilité: l'application doit être portable
- ↗ Efficacité: l'application utilise les ressources utilement
- ↗ Facile à manipule
- ↗ Interface agréable

✍ Facile à maintenir

### III. Analyse des besoins

#### 1. Identification des acteurs

Acteurs	Description
Utilisateur	<ul style="list-style-type: none"><li>- Ajouter catégorie permis</li><li>- Modifier catégorie permis</li><li>- Supprimer catégorie permis</li></ul>

Tableau 1 Description des acteurs

#### 2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. Ce diagramme donne une vue du système dans son environnement extérieur et définit la relation entre l'utilisateur et les éléments que le système met en œuvre.

Ce diagramme montre les différentes fonctionnalités assurées par l'utilisateur. En effet, il peut gérer les catégories permis par l'ajout, la modification ou la suppression d'une catégorie permis.

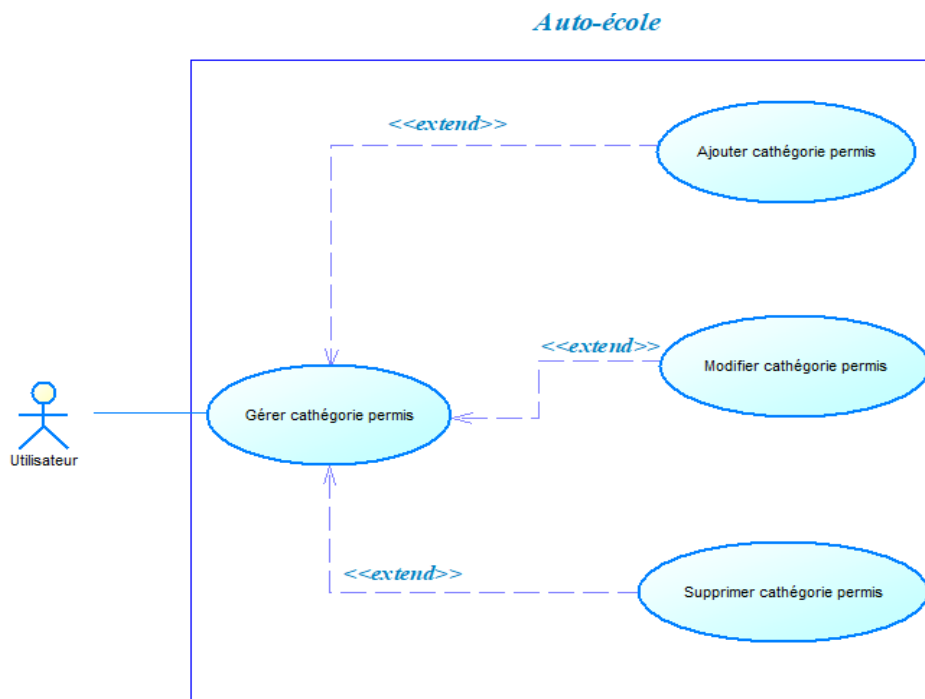


Figure 1 Diagramme de cas d'utilisation <<utilisateur>>

### 3. Diagramme de séquence système :

Un diagramme de séquence est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquence sont organisés en fonction du temps. Le temps s'écoule au fur et à mesure que vous parcourez la page. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence de messages.

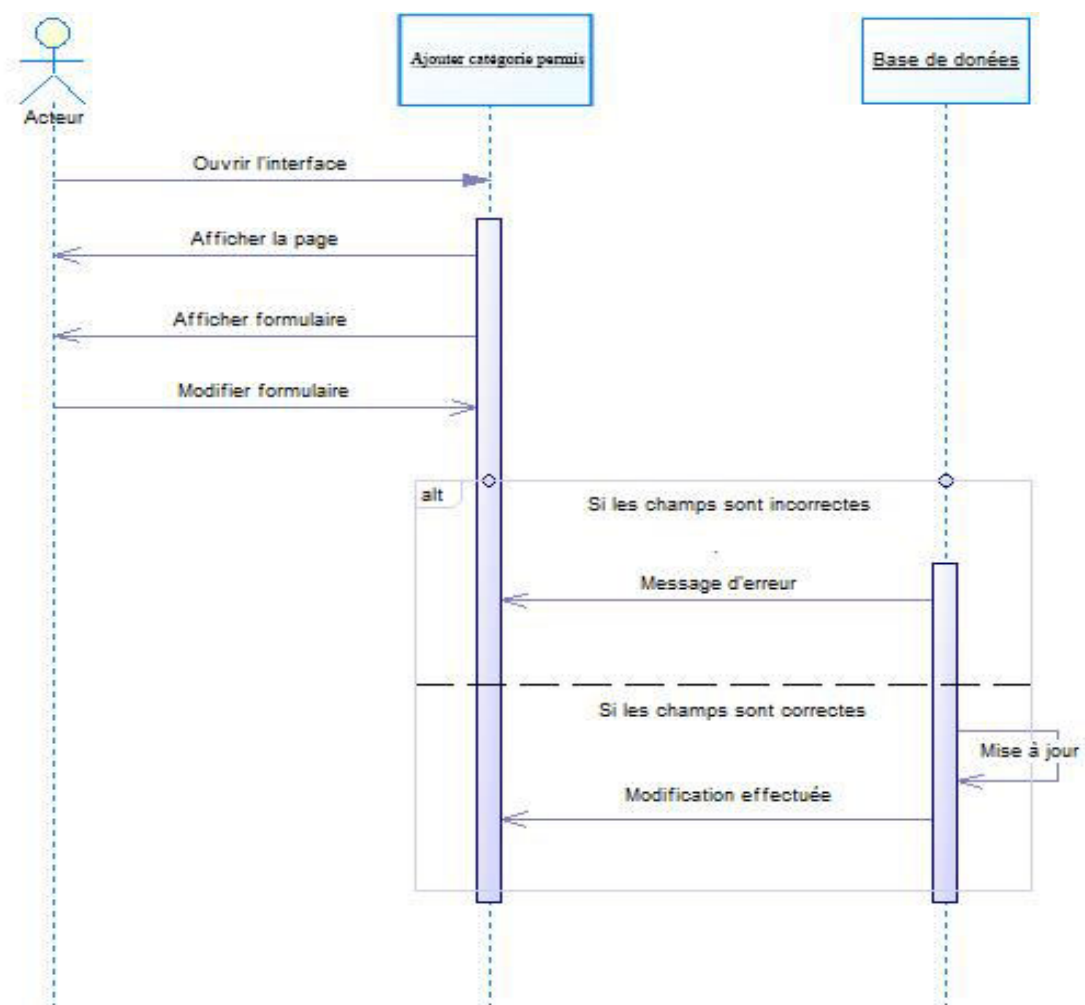
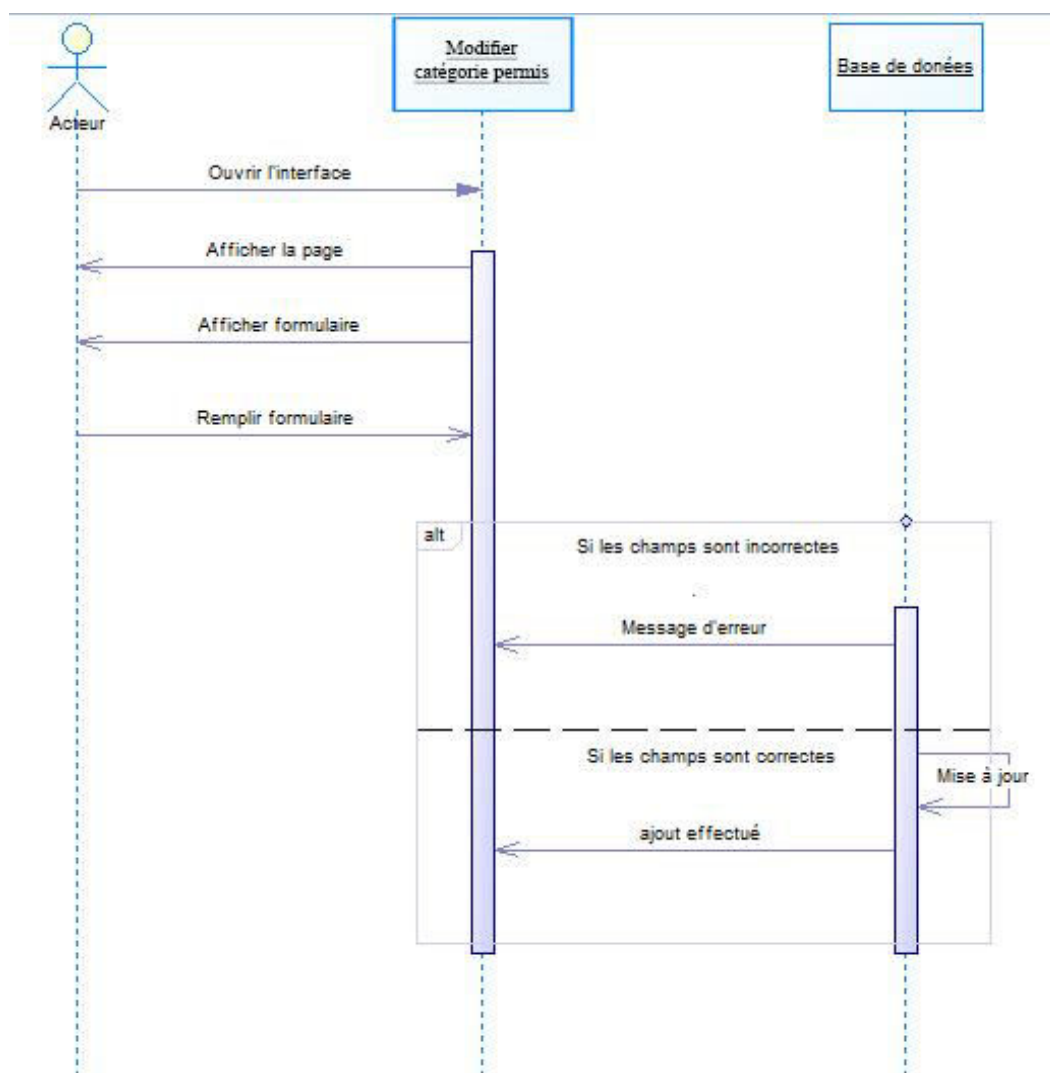


Figure 2 Diagramme de séquence « Ajout catégorie permis »

Titre	Diagramme d'ajout catégorie permis
But	Ajouter une nouvelle catégorie permise
Acteurs	Utilisateur
Scénario-nominal	<ul style="list-style-type: none"> <li>• L'utilisateur peut afficher et consulter une liste détaillée des catégories permis qui sont déjà créer.</li> <li>• L'utilisateur saisie les informations.</li> <li>• Le système vérifie les informations saisies.</li> <li>• Le système enregistre les informations.</li> <li>• L'utilisateur reçoit un message de succès.</li> </ul>

Tableau 2 Description diagramme de séquence <<ajouter catégorie permis >>



**Figure 3** Diagramme de séquence << Modifier catégorie permis >>

Titre	Diagramme de Modifier une Catégorie permis
But	Modifier une Catégorie permis
Acteurs	Utilisateur
Scénario-nominal	<ul style="list-style-type: none"><li>• L'utilisateur choisie une catégorie permise.</li><li>• L'utilisateur change les informations.</li><li>• Le système enregistre les nouvelles données</li><li>• L'utilisateur reçoit un message du succès.</li></ul>

**Tableau 3** Description Diagramme de séquence << Modifier catégorie permis >>



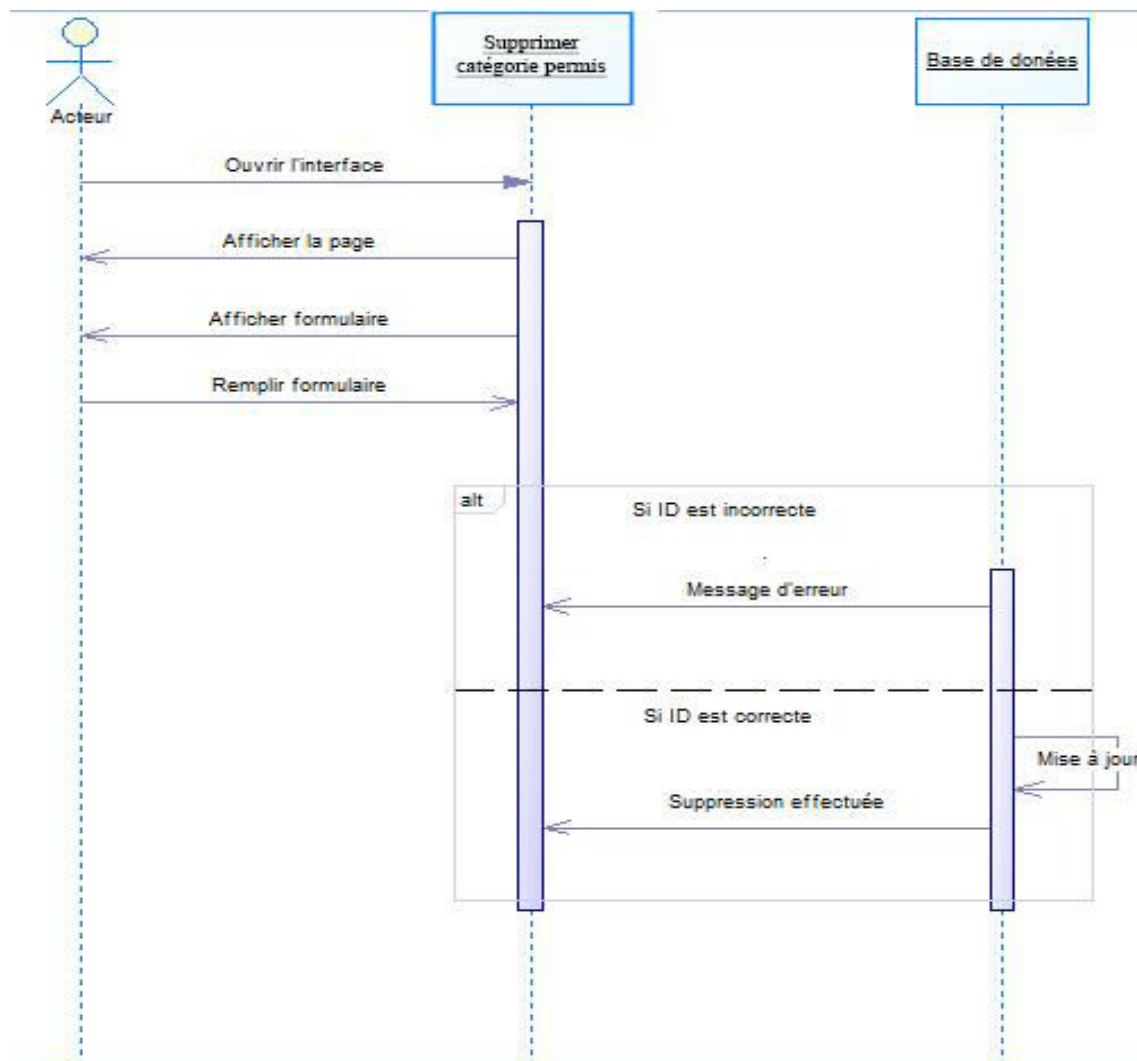


Figure 4 Diagramme de séquence << Supprimer catégorie permis >>

Titre	Diagramme de Supprimer une Catégorie permis
But	Supprimer une Catégorie permis
Acteurs	Utilisateur
Scénario-nominal	<ul style="list-style-type: none"> <li>• L'utilisateur choisie un questionnaire.</li> <li>• Le système supprime le questionnaire choisi.</li> <li>• L'utilisateur reçoit un message de succès</li> </ul>

Tableau 4 Description Diagramme de séquence << Supprimer catégorie permis >>

## Conclusion

Dans ce chapitre, nous avons pu concevoir un système de gestion des questionnaires en se basant sur les diagrammes du langage UML à savoir le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme de classes.

## **Chapitre 3 Conception**

## Introduction

Dans le cadre de ce chapitre, j'ai défini quelques généralités portant sur la méthode et les outils mettant en évidence la réalisation de notre projet. J'ai commencé par présenter le langage de modélisation UML avec les principaux diagrammes utilisés pour l'application

### I. Présentation UML

UML (Unified Modeling language ou Langage Unifier de Modélisation) est un langage graphique destiné à la modélisation de système et de processus. UML est un langage basé sur l'approche par objets, celle qui a d'abord conduit à la création des langages de programmation comme Java, C++ ou Small talk

UML est unifié car il provient de plusieurs notations qui l'ont précédé. Aujourd'hui, UML est promu par l'OMG (Object Management Group) un consortium de plus de 800 société et universités actives dans le domaine des technologies de l'objet.



Figure 5 Unified Modeling Language

Donc les trois diagrammes sont :

- Diagramme de cas d'utilisation
- Diagramme de classes
- Diagramme de séquences

## II. Conception détaillée

### 1. Diagramme de classe

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles. Les diagrammes de classes sont statiques: ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction.

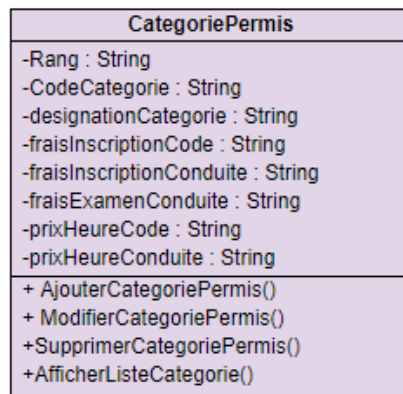


Figure 6 Diagramme de classe

### Conclusion :

Dans ce chapitre j'ai vu l'importance et l'utilité de la conception pour bien appliquer les informations trouvées dans l'étude de l'existant de notre application. Après j'ai réalisé la conception dans le chapitre suivant.

## **Chapitre 4 Réalisation et test**

## I. Introduction

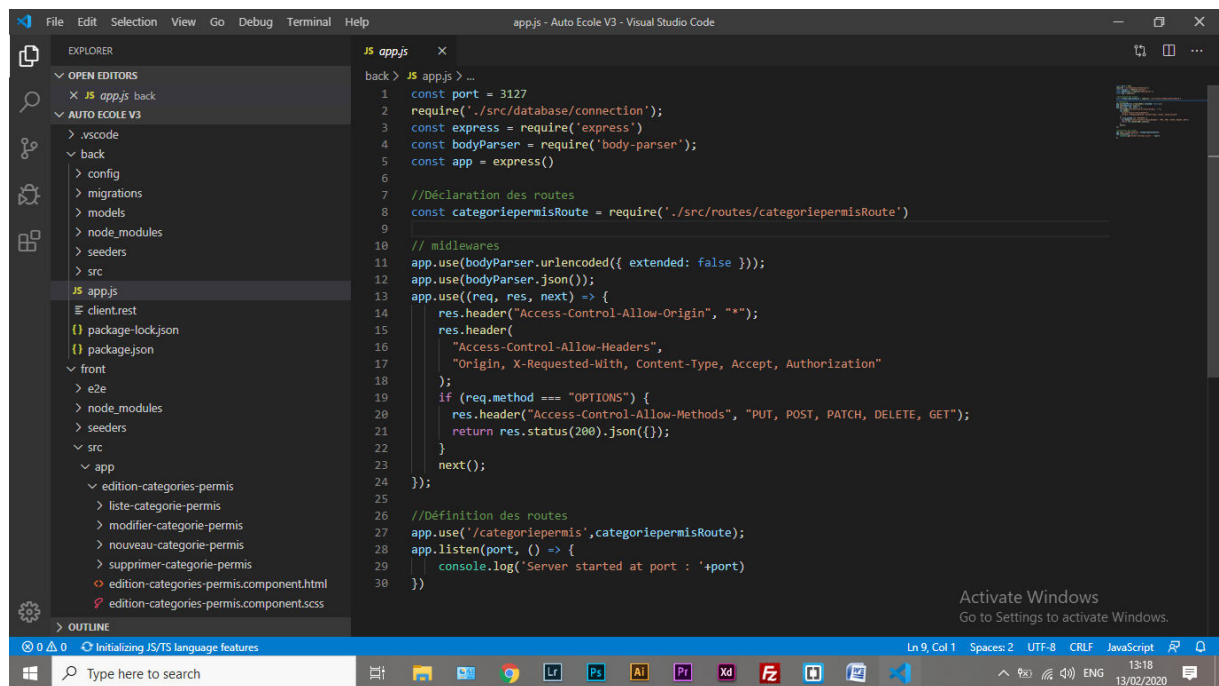
Après avoir achevé la partie de la définition des besoins et la partie de conception, on va entamer l'étape de la réalisation qui représente une partie très importante du temps consacré à ce projet. L'objet de ce chapitre, ici, n'est pas de décrire les lignes du code source les une après les autres. Cela serait fastidieux et profondément ennuyeux pour le lecteur. Il s'agit plutôt de présenter l'environnement de travail, les interfaces utilisateurs de l'application ainsi que les tests d'évaluation du produit final.

## II. Environnement de travail :

Dans la partie suivante, on présente les différents outils, des langages, des technologies, des plateformes et des frameworks utilisés pour développer cette application.

### 1. Visual Studio Code

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et possède un riche écosystème d'extensions pour d'autres langages (tels que C ++, C #, Java, Python, PHP, Go) et des runtimes (tels que .NET et Unity).



## 2. Node.js

**Node.js** est une plateforme logicielle libre en JavaScript orientée vers les applications réseau événementielles hautement concurrentes qui doivent pouvoir monter en charge.

Elle utilise la machine virtuelle V8, la librairie libuv pour sa boucle d'évènements, et implémente sous licence MIT les spécifications CommonJS.

Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js.

Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.

Node.js est utilisé notamment comme plateforme de serveur Web, elle est utilisée par Groupon, Vivaldi, SAP, LinkedIn, Microsoft, Yahoo!, Walmart, Rakuten, Sage et PayPa



## 3. MySQL

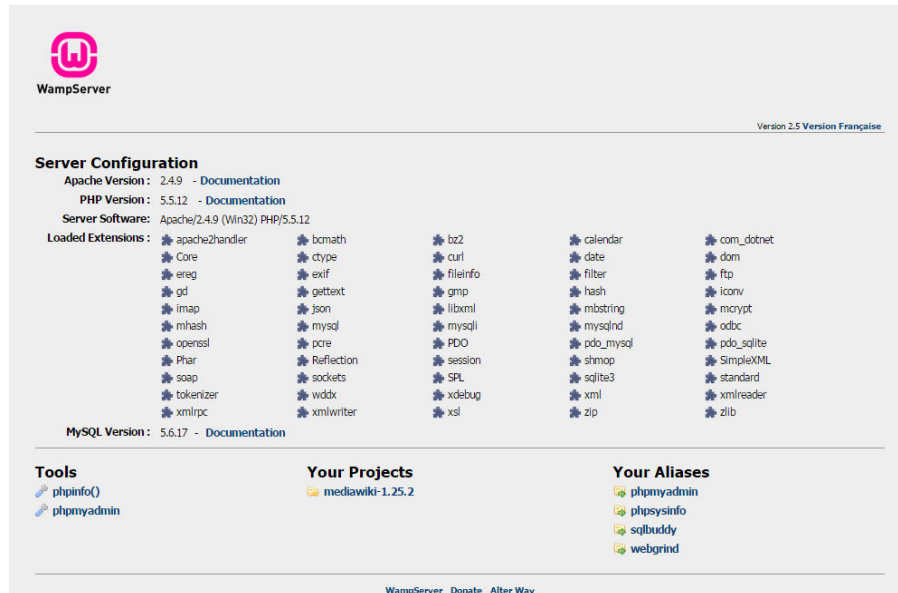
MySQL est un système de gestion de bases de données relationnelles (SGBDR) fonctionnant sous Windows et Linux. Il fait partie des logiciels de gestion de base des données les plus utilisées au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server. Cette application a besoin de plusieurs données pour bien tourner comme les données et prospects des clients, différents achats effectué, etc. Alors, on a décidé d'utiliser MySQL comme système de gestion de bases de données. Le choix est pris car :

- Solution très courante en hébergement public.
- Open-source, bien que les critères de licence soient de plus en plus difficile.
- Plusieurs de stockage adaptés aux différentes problématiques, configurable.
- C'est un SGBD relationnel.



## 4. WampServer

Pour gérer les données, on a pris le choix d'utiliser WampServer qui est une plateforme web de type WAMP. WampServer n'est pas un logiciel en soi un logiciel mais un environnement comprenant deux serveurs qui sont Apache et MySQL, un interpréteur de script PHP ainsi que phpMyAdmin pour l'administration web des bases MySQL.



Dans l'environnement du WampServer, on va utiliser que la partie phpMyAdmin pour administrer les bases des données MySQL. Il existe plusieurs façons d'accéder à sa base de données et d'y faire des modifications. On peut utiliser une ligne de commande (console), exécuter des requêtes SQL ou faire appel à un programme qui nous permet d'avoir rapidement une vue ensemble. Ici on va découvrir phpMyAdmin, un des outils les plus connus permettant de manipuler une base des données MySQL. PhpMyAdmin est l'une de plus célèbre interfaces pour gérer une base des données MySQL. Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances dans le domaine des bases des données, des nombreuses requêtes comme les créations de table des données, les insertions, les mises à jour, les suppressions les modifications de structure de la base des données. Ce système est très pratique pour sauvegarder une base des données sous forme de fichier .SQL et ainsi transférer facilement ses données. De plus celui-ci accepte la formulation de requête SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

## 5. Angular

**Angular** (communément appelé "**Angular 2+**" ou "**Angular v2 et plus**") est un cadriciel (Framework) côté client open source basé sur TypeScript dirigée par l'équipe du projet Angular à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète d'AngularJS, cadriciel construit par la même équipe.



AngularJS est fondé sur l'idée que la programmation déclarative doit être utilisée pour construire les interfaces utilisateurs et les composants logiciels de câblage, tandis que la programmation impérative excelle pour exprimer la logique métier<sup>3</sup>. La conception d'AngularJS est guidée par plusieurs objectifs :

- découpler les manipulations du DOM du logique métier. Cela améliore la testabilité du code ;
- considérer le test d'une application aussi important que l'écriture de l'application elle-même. La difficulté de la phase de test est considérablement impactée par la façon dont le code est structuré ;
- découpler les côtés client et serveur d'une application. Cela permet au développement logiciel des côtés client et serveur de progresser en parallèle, et permet la réutilisabilité de chacun des côtés ;
- guider les développeurs pendant toute la durée de la construction d'une application : de la conception de l'interface utilisateur, en passant par l'écriture du logique métier, jusqu'au test de l'application ;
- rendre les tâches faciles évidentes et les tâches difficiles possibles.

Une page web conçue avec AngularJS suit le patron d'architecture MVC (modèle-vue-contrôleur). Ce patron a pour avantage de proposer un couplage faible entre la présentation, les données, et les composants métier. Dans un langage web, cette séparation permet de diminuer l'importance des manipulations DOM et d'améliorer la testabilité du code.

Dans AngularJS, la partie « vue » est déclarée dans une version étendue du HTML traditionnel, qui comporte de nouvelles balises (*tags*) et attributs. Ce HTML étendu est utilisé pour déclarer une liaison de données bidirectionnelle entre les modèles et les vues. Ainsi, les données sont synchronisées automatiquement, et moins d'éléments ont besoin d'être définis en JavaScript. Les modèles sont composés de plusieurs couches appelés « scopes ». Les contrôleurs dans AngularJS permettent de prototyper des actions en code JavaScript natif.

En proposant l'injection de dépendances, AngularJS apporte aux applications web côté client les services traditionnellement apportés côté serveur, comme les contrôleurs de vues. En

conséquence, une bonne partie du fardeau supporté par le back-end est supprimée, ce qui conduit à des applications web beaucoup plus légères côté serveur.

Par ailleurs, AngularJS a la particularité d'utiliser des promesses (*promises*), reprises du langage Q. Une promesse est une opération asynchrone (elle ne bloque donc pas l'exécution du reste du programme) qui enverra son résultat final à une fonction de rappel dès que celui-ci sera connu.

La synchronisation automatique des données entre le modèle et la vue est permise par un mécanisme appelé *dirty-checking*. AngularJS exécute en permanence une boucle qui consiste à surveiller et à détecter des modifications sur un objet JavaScript.

AngularJS embarque un sous-ensemble de la bibliothèque open source jQuery appelé jQLite, mais peut aussi utiliser jQuery si elle est chargée.

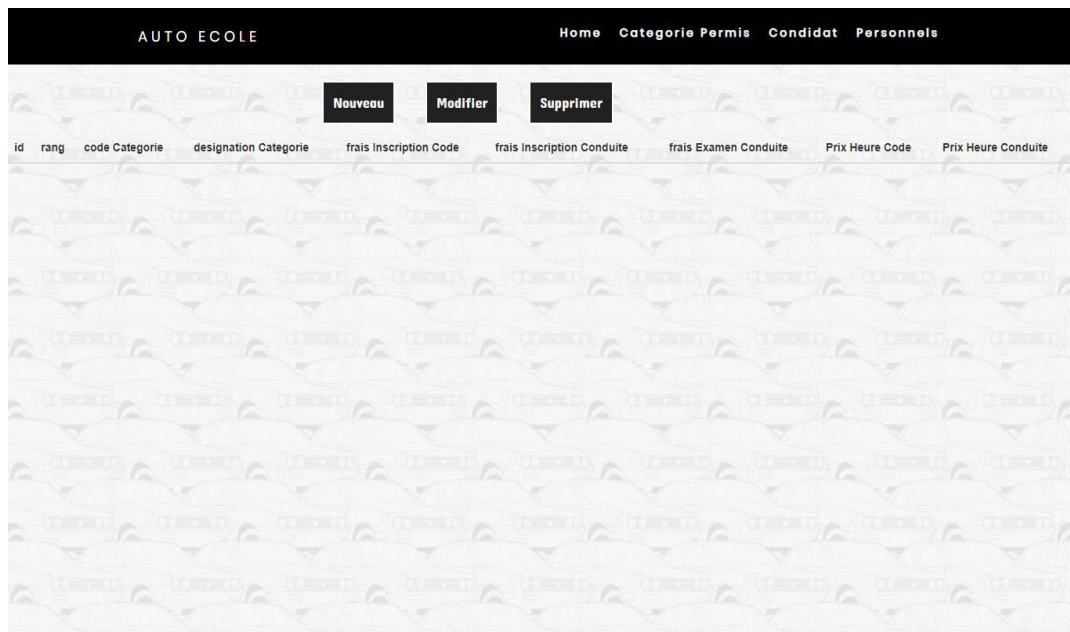
### III. Principales Interfaces graphiques

On présente quelques interfaces de l'application web. La figure ci-dessous est la page de bienvenue qui permet aux acteurs de l'application de s'identifier par leurs pseudonymes et leurs mots de passe en précisant leurs statuts soit un client visiteur, fournisseur ou l'administrateur qui a le rôle de gérer le contenu de l'application.



#### Menu principale de l'application

On oblige les visiteurs de cette application (fournisseurs et clients) à inscrire avant d'accéder, pour récupérer plus d'informations. Ceci, va assurer l'intégration de système business intelligence.



### La partie gestion de catégorie permis

Après l'authentification l'utilisateur, en affiche la liste des catégories permis avec trois boutons d'action pour ajouter un nouveau catégorie, Modifier un catégorie permis ou le supprimer.

### Ajouter un nouveau Catégorie permis

Si l'utilisateur cliqué sur nouveau un formulaire va apparaitre pour écrire les champs de la catégorie.

**AUTO ECOLE** Home Catégorie Permis Candidat Personnels

**Rang :**  
ajouter le Code personnel

**Code Catégorie :**  
ajouter le CIN

**Designation Catégorie :**  
ajouter la designation de Catégorie

**Frais Inscription Code :**  
ajouter le frais de l'Inscription Code

**Frais Inscription Conduite :**

**Frais Examen Conduite :**  
ajouter le frais de l'Examen Conduite

**Prix Heure de Code :**  
ajouter le Prix de l'Heure Code

**Prix Heure Conduite :**  
ajouter le Prix d'Heure Conduite

Modifier

Reset

### Modifier une Catégorie permis

Si l'utilisateur cliqué sur modifier un formulaire va apparaitre pour écrire les champs de la catégorie.

**AUTO ECOLE** Home Catégorie Permis Candidat Personnels

**Id Catégorie Permis**  
ajouter l'Id Catégorie Permis

Supprimer

Reset

### Supprimer une catégorie permis

Si l'utilisateur cliqué sur supprimer un formulaire va apparaitre pour écrire le champ id de la catégorie.

# Conclusion Générale

Durant mon stage j'ai conçu et réalisé un système d'information pour faciliter les tâches de gestion du parc véhicule.

J'ai utilisé l'outil UML pour le développement de mon application à savoir : Gérer la gestion des véhicules et l'ordre de mission.

Le travail a été réalisé en quatre parties : Dans la première partie j'ai présenté la société et ses activités. Dans la deuxième partie j'ai précisé les fonctionnements actuels, les spécificités d'application ainsi que la description des différentes fonctionnalités du système ciblent qui seront la base sur laquelle j'ai réalisé la conception de mon application. Dans la troisième partie j'ai vu l'importance et l'utilité du stage pour bien appliquer les informations trouvées en se basant sur l'existant afin de définir la bonne conception de mon application. Et dans la dernière partie j'ai vu que l'étude de l'existant permet en outre de remplir les obligations générales d'information, et de constituer un contrat permettant de limiter notre champs d'intervention.

Le système réalisé permet à la société d'éviter les tâches manuelles, il va permettre aux utilisateurs de se servir rapidement en éditant sur place et d'une manière automatique les documents administratifs.