Dédicaces

Au moment où s’achève le travail, il m’est agréable d’exprimer mes remerciements à tous ceux qui m’ont aidé.

Ma plus grande gratitude et tout mon amour **à ma mère et mon frère**, qui ont su me faire confiance, me soutenir et m’encourager au cours de ma vie.

**A tous mes amis** à tous qui nous aime et spécialement pour celui qui m’a toujours fourmi le soutien et le réconfort.

**A mon père** qui est décédé trop tôt, j'espère que tu es fier de ta petite fille.

Jrad Istabrak

Dédicaces

Toutes les lettres ne sauraient trouver les mots qu'il faut . . . Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance. . . Aussi, c'est tout simplement que je dédie ce projet de fin d'études....

A mes chers parents

Qui n'ont pas cessé de m'aider, de me soutenir et de m'encourager. En espérant qu’ils trouvent ici le résultat de nombreuses années de sacrifice et de privation pour m'aider à avancer dans la vie.

A mes adorables sœur et frère

Les bijoux de la famille à qui je donnerai tout, je vous souhaite un avenir plein de bonheur, de réussite et de sérénité.

À mes très chers amis

Zitouni Mohamed , Ben Aaicha Chaima et Haffar Neffa, un remerciement particulier et sincère pour tous vos efforts fournis. Vous avez toujours été présents. Que ce travail soit un témoignage de ma gratitude et mon profond respect. A tous mes amis , Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

Hmida Imen

Remerciements

Nous tenons à exprimer nos remerciements à tous ceux qui ont rendu ce travail possible. Leurs aides précieuses, leurs conseils fructueux et leurs encouragements, tout au long de l’élaboration de ce projet de fin d’études, nous a permis de le réaliser dans la meilleure considération.

Nous voulons rendre un hommage particulier à notre encadrante : **Mme. Haj Mohamed Hela p**our ses soutiens et ses conseils précieux.

A  **Abdessalem Bhouri Khila**, initiateur du besoin pour notre travail Nous voudrions lui exprimer nos remerciements pour son accueil et son temps partagé avec nous sous de nombreuses réunions. Nous avons beaucoup appris à travers le partage de son expertise et de ses qualités humaines. C’est grâce à sa confiance que nous avons pu mener à bien nos tâches.

 Aux membres du jury qui ont bien voulu nous honorer de leur présence d’évaluer notre travail. Un grand merci à toutes les personnes qui nous ont soutenues de près ou de loin au cours de la réalisation de ce projet.

Imen Hmida & Istabrak Jrad******

Table des matières

[Introduction générale 1](#_Toc44678255)

[Étude préliminaire 3](#_Toc44678256)

[1.1 Introduction 4](#_Toc44678257)

[1.2 Présentation de l’organisme d’accueil 4](#_Toc44678258)

[1.2.1 Organisations 4](#_Toc44678259)

[1.2.1.1 MaKIA Motors 4](#_Toc44678260)

[1.2.1.2 Modern Auto 5](#_Toc44678261)

[1.2.2 Services 5](#_Toc44678262)

[1.3 Cadre général du projet 5](#_Toc44678263)

[1.3.1 Etude et critique de l’existant 6](#_Toc44678264)

[1.3.2 Problématique 6](#_Toc44678265)

[1.3.3 Solutions proposées et objectifs globaux du projet 7](#_Toc44678266)

[1.4 Branche Fonctionnelle 7](#_Toc44678267)

[1.4.1 Besoins fonctionnels 7](#_Toc44678268)

[1.4.2 Besoins non fonctionnels 9](#_Toc44678269)

[1.5 Branche technique 9](#_Toc44678270)

[1.5.1 Architecture logicielle 9](#_Toc44678271)

[1.5.2 Langage de Développement 11](#_Toc44678272)

[1.5.3 Serveurs 12](#_Toc44678273)

[1.6 Conclusion 13](#_Toc44678274)

[Conception 14](#_Toc44678275)

[2.1 Introduction 15](#_Toc44678276)

[2.2 Méthode de conception 15](#_Toc44678277)

[2.2.1 Le langage UML 15](#_Toc44678278)

[2.2.2 Diagrammes d’UML 15](#_Toc44678279)

[2.2.3 Modèles d’UML utilisés 16](#_Toc44678280)

[2.3 Conception détaillée 16](#_Toc44678281)

[2.3.1 Identification des cas d’utilisations 17](#_Toc44678282)

[2.3.1.1 Diagramme de cas d’utilisation relatif aux acteurs 17](#_Toc44678283)

[2.3.1.2 Diagramme de cas d’utilisation général 17](#_Toc44678284)

[2.3.1.3 Diagramme de cas d’utilisation relatif à l’authentification 18](#_Toc44678285)

[La figure 5 montre le diagramme relatif au cas d’utilisation : s’authentifier. 18](#_Toc44678286)

[2.3.1.4 Diagramme de cas d’utilisation relatif au client 19](#_Toc44678287)

[2.3.1.5 Diagramme de cas d’utilisation relatif au magasinier 20](#_Toc44678288)

[2.3.1.6 Diagramme de cas d’utilisation relatif à l’administrateur : 21](#_Toc44678289)

[2.3.1.7 Diagramme de cas d’utilisation relatif à la gestion des pièces 22](#_Toc44678290)

[2.3.2 Diagramme des classes 23](#_Toc44678291)

[2.3.3 Diagrammes de séquence système 25](#_Toc44678292)

[2.3.3.1 Diagramme de séquence du scénario "Authentification" la phase d’identification 26](#_Toc44678293)

[2.3.3.2 Diagramme de séquence du scénario "Modifier Produit" pour un magasinier ( ou un administrateur )  : 27](#_Toc44678294)

[2.3.3.3 Diagramme de séquence du scénario "Passer Commande" 28](#_Toc44678295)

[2.4 Conclusion : 29](#_Toc44678296)

[Réalisation 30](#_Toc44678297)

[3.1 Introduction 31](#_Toc44678302)

[Bibliographie 41](#_Toc44678303)

Table des figures

[Figure 1 : Logo de la société MaKIA 4](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500699)

[Figure 2: Logo de la société Modern Auto 5](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500700)

[Figure 3: Relation entre les acteurs 18](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500701)

[Figure 4: Diagramme de cas d’utilisation : S’authentifier 19](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500702)

[Figure 5: Diagramme de cas d’utilisation relatif à l’administrateur 20](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500703)

[Figure 6: Diagramme de cas d’utilisation relatif au magasinier 21](#_Toc44500704)

[Figure 7: Diagramme de cas d’utilisation relatif au client 22](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500705)

[Figure 8: Diagramme de cas d’utilisation « Gérer les pièces » 23](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500706)

[Figure 9: Diagramme de cas d’utilisation 24](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500707)

[Figure 10: Diagramme des classes 26](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500708)

[Figure 11: Diagramme de séquence du scénario « Authentification » 28](#_Toc44500709)

[Figure 12: Diagramme de séquence du scénario « Modifier produit » 29](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500710)

[Figure 13: Diagramme de séquence du scénario « Passer Commande » 30](file:///C:\Users\ISTABRAK\Desktop\a.docx#_Toc44500711)

Liste des tableaux

[TABLE 1: Scénario « Authentification » 19](#_Toc44500712)

[TABLE 2: Sénario « Gérer les pièces » 24](#_Toc44500713)

[TABLE 3 : Tableau descriptif des classes 27](#_Toc44500714)

# Introduction générale

Des ventes de mains en mains, vers des ventes virtuelles, passent les priorités des opérations de ventes des biens et des services, ce qui nous rend obligés de donner plus d’importance à la vente électronique.

Les boutiques en ligne sont depuis des années, largement conseillés pour les sociétés qui se basent sur la vente des produits et même des services Ces types de sites web représentent un dispositif global fournissant aux clients un pont de passage à l’ensemble des informations, des produits, et des services à partir d’un portail unique en rapport avec son activité.

Les sites de vente en ligne permettent aux clients de profiter d’une foire virtuelle disponible est quotidiennement mise à jours sans la moindre contrainte, ce qui leur permettrai de ne jamais rater les coups de cœur, ainsi Une foire sans problèmes de distance géographique, ni d’horaire de travail ni de disponibilité de transport. D’une autre part ces sites offrent à la société de profiter de cette espace pour exposer ses produits à une plus large base de clientèle.

C’est dans ce cadre, que se situe notre projet de fin d’études dont le but est de mettre en place un site Web e-commerce de vente des pièces auto pour les sociétés KIA Motors et HYUNDAI Motorsqui vont à leur tour l’utiliser pour une société concerné.

Nous nous intéressons dans ce rapport à la description de différentes étapes de la réalisation de notre projet. Il est subdivisé en trois principaux chapitres :

Le premier chapitre intitulé "Étude préliminaire" nous présentons les deux sociétés, ensuite nous passerons à l’étude et à la critique de l’existant pour proposer une solution adéquate. Enfin, nous aborderons la compréhension du contexte du système etcela à travers la définition des besoins fonctionnels, les besoins non fonctionnels et techniques. Le deuxième chapitre intitulé "Conception" nous aborderons la conception détaillée par la présentation de modèle de l’application. Puis, nous détaillons la conception grâce aux diagrammes des cas d’utilisations, des classes et des séquences.

Dans le dernier chapitre intitulé "Réalisation", nous présentons l’environnement de développement matériel et logiciel que nous allons adopter. Puis,nous exposerons quelques interfaces des modules réalisés. Nous clôturerons le présent mémoire par une conclusion qui synthétise le travail et dresse les perspectives.

Chapitre 1

# Étude préliminaire

# Introduction

Ce premier chapitre définit le contexte général de notre travail, l’étape d’analyse et la spécification des besoins. Dans une première partie nous fournirons une courte présentation de la société MaKIA Motors et Alpha HYNDAI. Ensuite, nous détaillons le cadre de notre projet, puis nous allons blâmer l’existant pour délivrer leurs lacunes et proposer les axes de notre solution future. Dans la branche fonctionnelle, nous allons indiquer les besoins fonctionnels et non fonctionnels. Finalement nous terminons ce chapitre avec la branche technique, d’où seront indexés également les besoins majeurs.

# Présentation de l’organisme d’accueil

Puisque les deux sociétés MaKIA et Modern Auto ont le même manager, à notre demande de stage dans la société MaKIA, il nous a demandé de réaliser un site e-commercial de vente en ligne de pièces d’échange pour les deux maisons de voitures.

## Organisations

### 1.2.1.1 MaKIA Motors

La société MaKIA est l'agence agrée par CityCars dans la région de Monastir depuis 2015.CityCars est le représentant officiel et exclusif de Kia Motors en Tunisie.  
MaKia Monastir est une agence 3S (Services, Sells, Spare Parts) elle contient un showroom conforme à la charte de Kia Motors. L'agence comprend aussi un atelier de carrosserie, mécanique, services rapides et un Contoire de pièces de rechange.  
Alors les objectifs de MaKIA consistent à offrir de meilleurs services à ses clients à travers une équipe de professionnels spécialisés.

Figure 1 : Logo de la société MaKIA

### 1.2.1.2 Modern Auto

Modern Auto est l’agence de Alpha Hyundai Motors à Monastir sur la route côtière de Kheniss. L’agence est aussi 3S (showroom, Spare parts et service technique) et répartie entre les différents compartiments techniques, commerciaux et administratifs pour une offre complète de services.

Figure 2: Logo de la société Modern Auto

La partie showroom permet de voir la gamme Hyundai disponible sur le marché avec la possibilité d’acquérir la voiture adéquate au besoin sur place. L’agence compte également un service après-vente qui englobe les services techniques nécessaires pour l’entretien des véhicules des clients, un département technique, un espace de service rapide, un atelier, un département administratif pour faciliter le traitement des demandes des clients et l’organisation du service dans une logique de décentralisation au profit de l’agent agréé.

## Services

MaKIA et également Modern Auto propose un large éventail de services à ces clients:

—vente des véhicules

— diagnostique du véhicule.

— échange des pièces.

— vidange.

— contrôle et inspection

— conseil et accompagnement

— vente des véhicules

# 1.3 Cadre général du projet

Notre projet titré "**CREATION D’UN SITE WEB E-COMMERCIAL** " est élaboré dans le cadre de la perception du Projet de Fin d’études exhibé en vue de l’obtention de la licence fondamentale en informatique à la Faculté des sciences de Monastir pour l’année universitaire 2019/2020. Il a été réalisé au sein des sociétés MaKIA et Modern Auto.

## 1.3.1 Etude et critique de l’existant

Les différentes pièces autos ne sont pas exposées. En cas de panne, les techniciens remplacent la pièce en panne sur place. En effet, un nombre limité des clients qui résident proche des sociétés peuvent consulter les produits à vendre et savoir leur disponibilité ainsi que leur prix et leurs caractéristiques techniques, tout ça présente une entrave devant la commercialisation des produits. Or que la factorisation et la gestion de stock se fait manuellement. Ce que reflètent plusieurs inconvénients dont les plus importants sont :

—le client ne peut pas vérifier l’existence des pièces en stock, il doit appeler la société pour s’assurer.

—les pièces auto ne peuvent pas être vendues individuellement, ils peuvent être seulement remplacés dans les workshops de deux sociétés.

—Le magasiner doit revisiter le stock chaque fois un client l’appel pour demander des informations à propos une certaine pièce et ne peut pas savoir combien des pièces sont disponibles

—La procédure d’ajout de nouvelles pièces va être lente car le magasinier ne remarque pas immédiatement le manque.

—le responsable de deux sociétés trouve des difficultés en consultant les données séparément.

—difficulté de la factorisation manuelle puisque cette opération nécessite beaucoup d’attention.

## 1.3.2 Problématique

L’absence d’un site de vente limite l’expansion de commerces, c’est difficile de gérer les pièces tenant en considération les tâches divers à accomplir et le nombre important des pièces entrant et sortant. Parmi ces tâches, les tâches de facturation qui nécessite beaucoup d’attention et de temps pour une vérification manuelle. Donc pour garantir l’automatisation et la facilité des tâches, on peut mettre en place un site e-commercial qui permettra de gérer la pression commerciale.

## 1.3.3 Solutions proposées et objectifs globaux du projet

En constatant les insuffisances du système actuel et afin de les [adoucir](https://www.synonymes.com/synonyme.php?mot=adoucir), ce projet doit répondre à plusieurs objectifs qui sont :

— Assurer la simplicité du site tout en gardant la sécurité et la confidentialité des différentes données saisies.

— Concevoir des interfaces claires et simples à manipuler.

— La rapidité, la fiabilité et la facilité des traitements.

— Optimiser au maximum le temps d’accès et de manipulation des données.

— Assurer les différents services d’achat.

—Service d’exposition des pièces.

—Garantir un gain de temps pour les clients et les responsables de la société.

La solution que nous avons proposée est de créer un site Web dynamique de vente en ligne des pièces d’échange pour les voitures KIA et HUNDAI qui va traiter les achats, les commandes, les factures, les pièces, les catégories ainsi que la gestion des utilisateurs. Cette solution sera bien détailler dans la partie Analyse et spécification des besoins, conception et la partie réalisation.

# 1.4 Branche Fonctionnelle

L’analyse et la spécification des besoins est un aspect capital pour appréhender les fonctionnalités de notre site. De même, nous administrons dans ce qui suit une analyse des besoins fonctionnels et non-fonctionnels de notre site Web et nous proposons, de les indiquer à l’aide de diagrammes des cas d’utilisation.

## 1.4.1 Besoins fonctionnels

Nous révoquons qu’un besoin fonctionnel est un besoin caractérisant une action qu’un système doit être apte à effectuer sans épier aucune contrainte physique. C’est un besoin spécifiant un comportement d’entrée/sortie d’un système.

Notre objectif est de développer un site e-commerce qui facilite la manipulation des données aux administrateurs et magasiniers qui travaillent dans l’entreprise et aussi à leurs clients.

En effet, le site à développer doit assurer les fonctionnalités suivantes :

Si l’acteur est un administrateur il peut :

— S’authentifier.

— Gérer les comptes des utilisateurs.

— Gérer les produits.

— Gérer les commandes.

— Gérer les factures.

Si l’acteur est un magasinier il peut :

— S’authentifier.

— Gérer les produits.

— Gérer les commandes.

— Gérer les factures.

Si l’acteur est un client il peut :

— S’authentifier.

— passer des commandes.

— visualiser les produits.

— Consulter l’historique de ses achats.

## 1.4.2 Besoins non fonctionnels

En plus des besoins fonctionnels cités avant, le projet doit assurer certaines conditions non-fonctionnelles. Ces derniers représentent un ensemble de contraintes à respecter pour confirmer l’efficacité du projet.

Dans ce qui suit, nous invoquons les plus importants besoins non fonctionnels :

— **La sécurité** : le site devra assurer la sécurité des utilisateurs, d’où la nécessité de procéder l’authentification tout en assurant la confidentialité de leurs données.

— **Contraintes ergonomiques** : le site Web doit être simple et utilisable afin que l’utilisateur puisse l’exploiter sans se référer à des connaissances particulières, en d’autres mots, notre site doit être lisible et facile à manipuler par n’importe quel utilisateur.

— **Contraintes de disponibilité** : le site doit être toujours fonctionnel et disponible en permanence.

# 1.5 Branche technique

De ce qui précède, les besoins fonctionnels de notre système ayant été définis, on veut savoir quels outils allons-nous utiliser et surtout quelle architecture logicielle on veut choisir. C’est pourquoi nous définirons dans cette partie en premier lieu l’architecture logicielle, puis les langages de développement et nous terminerons par les serveurs utilisés.

## 1.5.1 Architecture logicielle

Nous devons savoir qu’il existe plusieurs types d’architectures. Parmi ces architectures, nous pouvons citer :

**L’architecture client-serveur :**C’est la description du fonctionnement coopératif entre le serveur et le client. Les services internet sont conçus selon cette architecture. Ainsi, chaque application est composée de logiciel serveur et logiciel client.

— On appelle logiciel serveur un programme qui offre un service sur le réseau.

— On appelle logiciel client un programme qui utilise le service offert par un serveur.

On reconnaît traditionnellement dans une application 3 modules : Données, Traitement et Présentation.

La répartition de ces 3 modules variera entre le client et le serveur, donc suivant les contraintes d’utilisation ou contraintes techniques on à des différents types d’architectures client-serveur :

— **Architecture 1-tiers** : les 3 couches applicatives s’exécutent sur la même machine, on parle d’informatique centralisé et on a contexte multiutilisateurs dans le cadre de site central (mainframe).

— **Architecture 2-tiers** : l’architecture à deux niveaux (aussi appelée architecture 2 tiers, tier signifiant rangée en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource au serveur qui la fournit à partir de ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service. Pour la présentation et les traitements sont sur le client, les données sur le serveur et contexte multiutilisateurs avec accès aux données centralisées

— **Architecture 3-tiers** : dans l’architecture à 3 niveaux (appelée architecture 3-tier) il existe un niveau intermédiaire, c’est-à-dire que l’on a généralement une architecture partagée entre :

— **Un client**, c’est-à-dire l’ordinateur demandeur de ressources, équipée d’une interface utilisateur (généralement un navigateur web) chargée de la présentation

— **Le serveur d’application** (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur.

— **Le serveur de données**, fournissant au serveur d’application les données dont il a besoin. Concernant le partage des 3 module on distingue que : pour la présentation est sur le client, les traitements sont pris par un serveur intermédiaire et les données sont sur un serveur de données.

— **Architecture n-tiers** : (N-tier architecture en anglais) l’architecture n-tiers a été pensée pour pallier aux limitations des architectures trois tiers et concevoir des applications puissantes et simples à maintenir. En fait, l’architecture n-tiers qualifie la distribution d’application entre de multiples services et non la multiplication des niveaux de services.

**Choix de l’architecture utilisée**

Suite à l'étude précédente, on choisit l’architecture client-serveur et plus précisément sur l’architecture 3-tiers :

— **Le client** : c’est un ordinateur personnel avec un navigateur web chargé de la représentation des interfaces de l’application qui sont destinées aux utilisateurs de la société.

— **Le serveur d’application (appelé middleware),** qui fournit la ressource, mais en faisant appel aux ressources d'un autre serveur. (language Node.js, web MVC Express for Node.js)

— **Le serveur de donnée** : sert à stocker les données et fournit au serveur d’application les données qui a besoin.(MongoDB)

Cette architecture offre plusieurs avantages :

**Couplage faible** : en cas de besoin de modification d’une fonctionnalité, nous ne risquons pas de modifier toute l’application.

**Modularité** : la réutilisation d’une fonctionnalité déjà développée ce qui nous fait gagner en terme de temps de développement et de maintenance.

## 1.5.2 Langage de Développement

Dans le développement de notre système, nous avons utilisé plusieurs langages tels que :

— **HTML5** : HTML5 est une évolution de la norme HTML dont le but avoué est de faciliter le développement d’interfaces utilisateur riches. HTML5 est beaucoup plus orienté applicatif que ses prédécesseurs et veut permettre de s’affranchir de plugins pour utiliser au maximum les technologies Web natives afin de construire une application riche. (1)

— **CSS3** : un langage qui permet de gérer la présentation d’une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML ou XML. (2)

— **JS** : JavaScript (qui est souvent abrégé en "JS") est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.

Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs

Web tels que node.js ou MongoDB CouchDB. (3)

## 1.5.3 Serveurs

**Serveur de base de données** :

**MongoDB**(de l'anglais [humongous](https://fr.wiktionary.org/wiki/humongous" \o "wikt:humongous) qui peut être traduit par « énorme ») est un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) [orienté documents](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_orient%C3%A9e_documents), [répartissable sur un nombre quelconque d'ordinateurs](https://fr.wikipedia.org/wiki/Scalability) et ne nécessitant pas de schéma prédéfini des données. Il est écrit en [C++](https://fr.wikipedia.org/wiki/C%2B%2B). Le serveur et les outils sont distribués sous [licence SSPL](https://fr.wikipedia.org/w/index.php?title=Server_Side_Public_License_(SSPL)&action=edit&redlink=1), les pilotes sous [licence Apache](https://fr.wikipedia.org/wiki/Licence_Apache) et la documentation sous [licence Creative Commons](https://fr.wikipedia.org/wiki/Licence_Creative_Commons)[5](https://fr.wikipedia.org/wiki/MongoDB#cite_note-licensing-5). Il fait partie de la mouvance [NoSQL](https://fr.wikipedia.org/wiki/NoSQL" \o "NoSQL). (4)

**Serveur d’application** :

**Node.js** est un environnement de serveur open source et gratuit qui fonctionne sur différentes plateformes et utilise JavaScript sur le serveur en avec une programmation asynchrone. Voici comment PHP ou ASP gère une demande de fichier:

1-Envoie la tâche au système de fichiers de l'ordinateur.

2-Attend que le système de fichiers s'ouvre et lit le fichier.

3-Renvoie le contenu au client.

4-Prêt à traiter la prochaine demande.

Or que Node.js exécute une programmation asynchrone à un seul thread, non bloquante, ce qui est très efficace en mémoire. Il peut aussi générer du contenu de page dynamique, manipuler les fichiers sur le serveur, collecter des données de formulaire et manipuler des données dans votre base de données

Les fichiers Node.js contiennent des tâches qui seront exécutées sur certains événements, typiquement une personne essayant d'accéder à un port du serveur. (5)

# 1.6 Conclusion

Dans ce premier chapitre nous avons présenté l’organisme d’accueil MaKIA et leurs services ainsi que le cadre général de notre projet. Par ailleurs, ce chapitre nous a autorisées de définir les différentes fonctionnalités que doit offrir notre site Web. Le chapitre suivant permet de mettre en évidence la conception.

Chapitre 2

# Conception

# 2.1 Introduction

Avant d’aborder le développement de notre projet, il est absolu de choisir la forme de conception ainsi la détaillée. Nous évoquons alors, le modèle affiliée pour comprendre comment nous allons acheminer la construction des fonctionnalités à développer.

À la fin, nous présentons des diagrammes de cas d’utilisations du système, le diagramme de classe et des diagrammes de séquence.

# 2.2 Méthode de conception

Le cycle de conception permet d’évoquer de manière non énigmatique, le plus fréquemment en utilisant un langage de modélisation, le fonctionnement futur du système, afin d’en faciliter la réalisation.

## 2.2.1 Le langage UML

Pour simplifier notre tâche nous avons recours au langage de modélisation unifié (UML : Unified Modelling Language) qui est une notation qui permet de modéliser un problème de façon standard. Ce langage qui est né de la fusion de plusieurs méthodes existantes auparavant est devenu une référence en termes de modélisation objet, UML est caractérisé par :

— C’est un langage formel et normalisé.

— Il permet le gain de précision, encourage l’utilisation d’outils et constitue à cet effet un gage de stabilité.

— Il cadre l’analyse et facilite la compréhension de représentations abstraites complexes.

**Son caractère polyvalent et sa souplesse en font un langage universel.**

— UML est un support de communication performant.

## 2.2.2 Diagrammes d’UML

UML s’articule autour de treize types de diagrammes, chacun d’eux étant dédié à la représentation des concepts particuliers d’un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes. (5)

**Sept diagrammes comportementaux** :

— Diagramme de cas d’utilisation.

— Diagramme d’états.

— Diagramme de séquence.

— Diagramme de vue d’ensemble des interactions.

— Diagramme de communication.

— Diagramme d’activité.

— Diagramme de temps.

**Six diagrammes structurels :**

— Diagramme d’objets.

— Diagramme de classes.

— Diagramme de packages.

— Diagramme de structure composite. Complexe.

— Diagramme de déploiement.

— Diagramme de composants.

## 2.2.3 Modèles d’UML utilisés

A cet effet on présente quelques diagrammes de modélisation, qu’on a jugé les plus importants pour la compréhension du fonctionnement du système :

— **Diagramme de cas d’utilisation :** il montre les interactions fonctionnelles entre les acteurs et le système à l’étude.

— **Diagramme de classes :** il montre les briques de base statiques : classes associations, interfaces, attributs, opérations, généralisations.

— **Diagramme de séquences :** il montre la séquence verticale des messages passés entre objets au sein d’une interaction.

# 2.3 Conception détaillée

Dans cette partie, nous présentons la conception de notre application, à travers des diagrammes de cas d’utilisations, le diagramme des classes du modèle de conception et quelques diagrammes de séquence.

## 2.3.1 Identification des cas d’utilisations

Un cas d’utilisation représente une unité discrète d’interaction entre un utilisateur

(Humain ou Machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d’utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d’utilisation. Ce diagramme permet d’identifier les possibilités d’interaction entre le système et les acteurs. Il représente toutes les fonctionnalités que le système doit fournir.

Dans notre cas les diagrammes des cas d’utilisations cités par la suite décrivent ce que peut faire un administrateur, un magasinier et un client lors de l’utilisation de notre site.

### 2.3.1.1 Diagramme de cas d’utilisation relatif aux acteurs

Avant de commencer, en tenant compte des fonctionnalités communes entre les trois

acteurs, une relation d’héritage s’impose. Cela est expliqué par la figure 3.

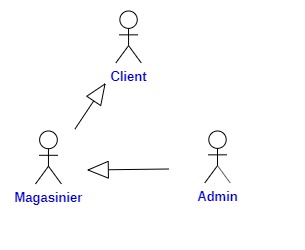


Figure 3: Relation entre les acteurs

### 2.3.1.2 Diagramme de cas d’utilisation général

La figure 9 présente l’ensemble des fonctionnalités offerte par l’application pour tous les utilisateurs. Ce Diagramme met en évidence les fonctionnalités générales attendues de notre application.

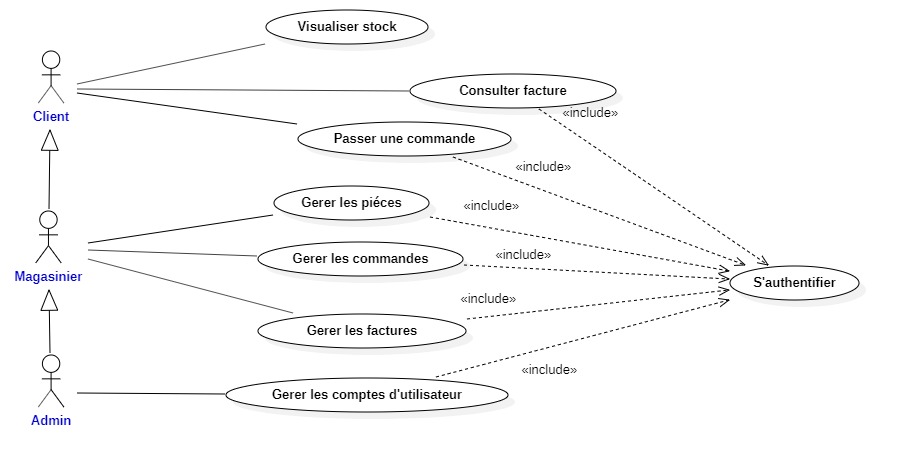


Figure 4: Diagramme des cas d’utilisation général

### 2.3.1.3 Diagramme de cas d’utilisation relatif à l’authentification

### La figure 5 montre le diagramme relatif au cas d’utilisation : s’authentifier.

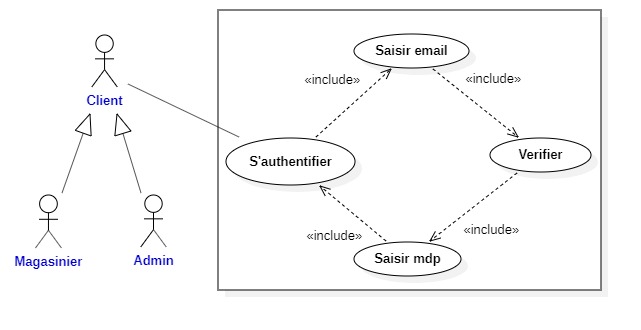


Figure 5: Diagramme de cas d’utilisation : S’authentifier

Le tableau 1 présente la description textuelle "s’authentifier".

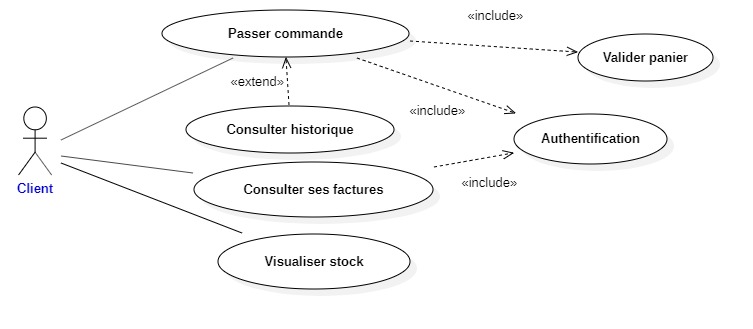
|  |  |
| --- | --- |
| Titre | Authentification |
| But | Notre système exige la phase d’authentification pour les différents utilisateurs afin de bénéficier des différents services du système.  Chaque utilisateur est authentifié par le couple "Email/mot de passe". |
| Acteurs | Administrateur, magasinier, client |
|  | Description des enchainements |
| Pré-condition | L’utilisateur doit avoir un compte. |
| Post-condition | L’utilisateur est authentifié. |
| Scénario principal | |
| 1. L’utilisateur saisit les paramètres de connexion (email/mot de passe) et confirme les paramètres déjà entrés. | |
| 2. Le système vérifie les paramètres email et mot de passe. | |
| 3. Le système affiche l’espace de l’utilisateur authentifié. | |
| Scénario(s) alternatif(s) | |
| 1. : Si les données saisies sont invalides, le système affiche un message d’erreur   et l’enchaînement reprend au point 1 de la séquence principale. | |

TABLE 1: Scénario « Authentification »

### 2.3.1.4 Diagramme de cas d’utilisation relatif au client

La figure 6 ci-dessous illustre le diagramme de cas d’utilisation relatif au client, il nous permet d’obtenir une vision globale du comportement fonctionnel de notre site web coté client

Figure 6: Diagramme de cas d’utilisation relatif au client



**Les cas d’utilisation :**

— **Passer une commande :** un client peut ajouter une commande après son authentification et la validation de son panier, il peut aussi consulter ses commandes.

— **Visualiser les pièces :** les pièces sont accessibles pour tous visiteurs de site, authentifiés ou pas.

— **Consulter ses factures :** le client peut aussi consulter ses factures après son authentification.

Dans la suite, nous présentons les raffinements de quelques cas d’utilisation pour les détailler.

### 2.3.1.5 Diagramme de cas d’utilisation relatif au magasinier

La figure 7 ci-dessous illustre le diagramme de cas d’utilisation relatif au magasinier,

il nous permet d’obtenir une vision globale du comportement fonctionnel de notre

site web coté magasinier.

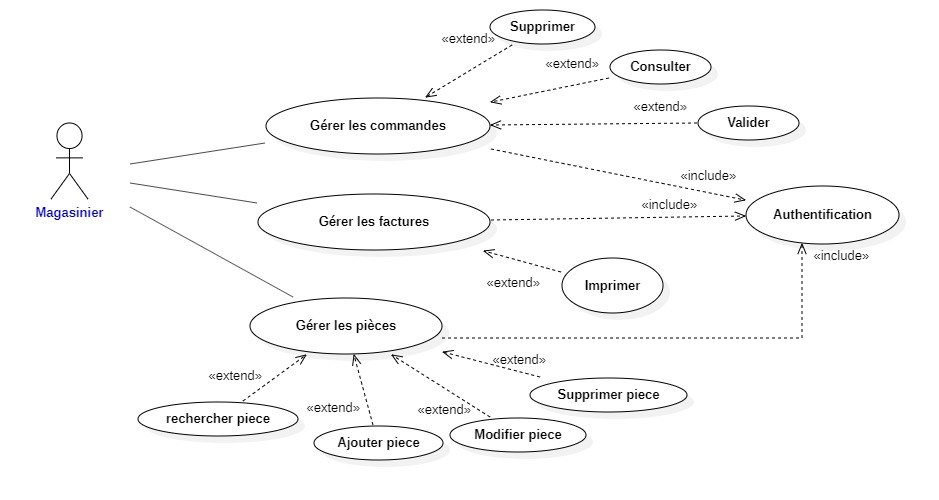


Figure 7: Diagramme de cas d’utilisation relatif au magasinier

**Pré-condition :**

Nous imposons que le magasinier n’a le droit d’accéder au système qu’après son authentification.

**Les cas d’utilisation :**

* Gérer les commandes **:** un magasinier peut consulter, valider ou bien supprimer une commande.
* Gérer les factures **:** un magasinier peut gérer les factures qui seront, par la suite

, affectées à un client.

* Gérer les pièces **:** le magasinier peut aussi ajouter, modifier, supprimer ou rechercher une pièce.
* Visualiser pièce : le magasinier peut voir les pièces existantes.

### 2.3.1.6 Diagramme de cas d’utilisation relatif à l’administrateur :

La figure 8 ci-dessous illustre le diagramme de cas d’utilisation relatif à l’administrateur, il nous permet d’obtenir une vision globale du comportement fonctionnel de notre site web coté administrateur.

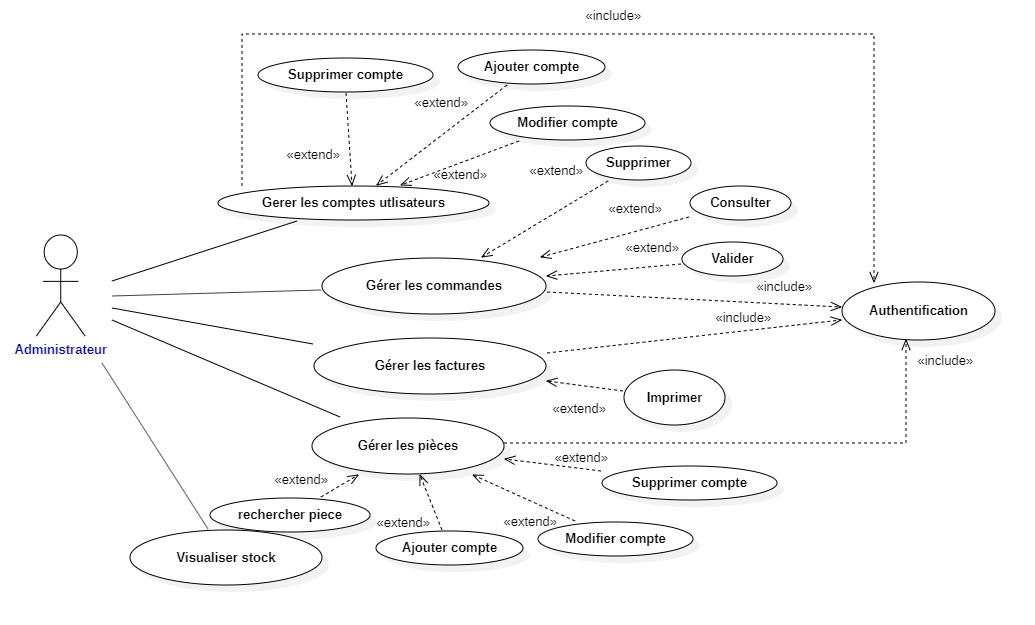


Figure 8: Diagramme de cas d’utilisation relatif à l’administrateur

**Pré-condition :**

Nous imposons que l’administrateur n’a le droit de faire ses taches qu’après son authentification.

**Le cas d’utilisation :**

* Gérer les comptes des utilisateurs : c’est une fonction privilégiée à l’administrateur. Il peut ajouter, modifier, rechercher ou supprimer un compte.
* Gérer les commandes **:** un administrateur peut supprimer, consulter ou bien valider une commande.
* Gérer les factures **:** l’administrateur peut gérer les factures qui seront, par la suite

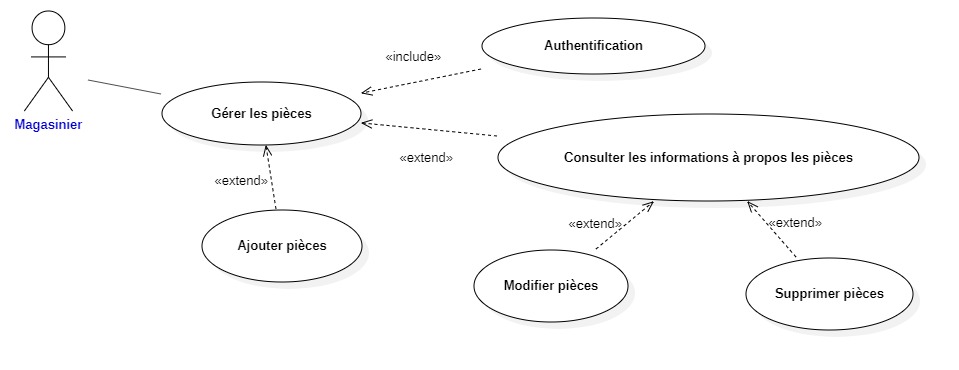
, affectées à un client.

* Gérer les pièces **:** l’administrateur peut aussi ajouter, modifier, supprimer ou rechercher une pièce.
* Visualiser stock : l’administrateur peut voir les pièces existantes.

### 2.3.1.7 Diagramme de cas d’utilisation relatif à la gestion des pièces

La figure 9 présente le diagramme de cas d’utilisation « Gérer les pièce »

Figure 9: Diagramme de cas d’utilisation « Gérer les pièces »



Le tableau 2.2 présente la description textuelle "Gérer les pièces".

|  |  |
| --- | --- |
| **Titre** | **Gérer les pièces** |
| **But** | Le système doit permettre au responsable d’ajouter d’autres pièces au site, de modifier leurs informations, de les supprimer ou de voir les disponibilités de ces derniers. |
| **Acteurs** | Magasinier, Administrateur |
|  | Description des enchainements |
| **Pré-condition** | Le responsable doit être authentifié. |
| **Post-condition** | Opération effectuée avec succès. |
| **Scénario principal** | |
| 1. Le responsable choisit l’option "pièces" du menu. | |
| 2. Le système affiche tous les pièces qui existent dans la base. | |
| 3. Le responsable choisit l’action qui veut effectuer, soit d’ajouter une pièce, supprimer une pièce, de modifier ces informations. | |
| 4. Si le responsable choisit d’ajouter une nouvelle pièce ou de modifier un autre existant. | |
| 5. le système affiche un formulaire respectivement vide ou contenant les données de la pièce. | |
| 6. Le magasinier remplit les champs puis il valide l’action. | |
| 7. Le système enregistre les nouvelles données ou les changements entrés et affiche un message de validation. | |

TABLE 2: Sénario « Gérer les pièces »

## 2.3.2 Diagramme des classes

Dans cette partie, nous allons présenter le diagramme de classe. Ce Diagramme exprime la structure statique du système en termes de classes et de relations entre ces classes. L’intérêt du diagramme de classes est de modéliser les entités du système d’information. La figure 10 met en relief le diagramme des classes de l’application en précisant dans chaque classe ses attributs et ses méthodes avec les relations qui existent entre elles. (Page suivante)

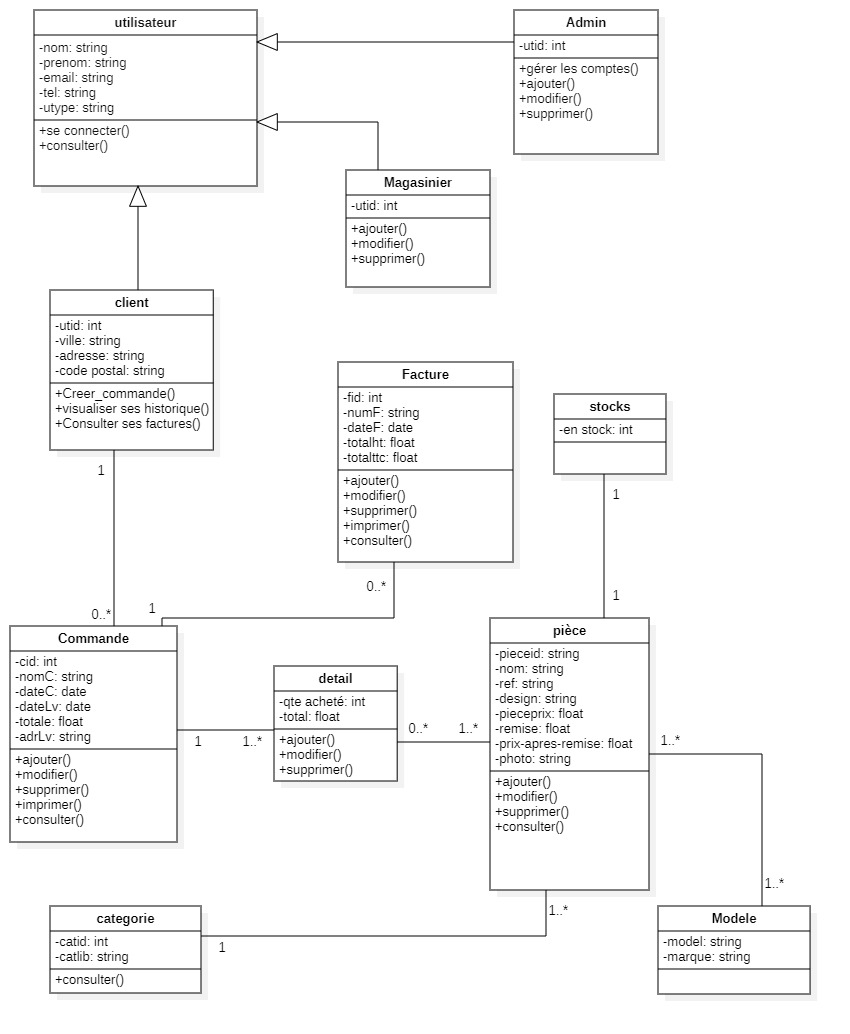


Figure 10: Diagramme des classes

Nous présenterons dans ce qui suit un tableau descriptif des classes les plus importantes de la Figure 10.

|  |  |
| --- | --- |
| Utilisateur | C’est l’utilisateur de l’application qui peut être un client un administrateur ou magasinier. |
| Facture | Cette classe représente la facture affectée à un client. cette entité a une date de facturation, un numéro unique et une totale. |
| Commande | Cette classe représente la commande dont le client va la remplir. Cette entité a une date de commande et un numéro unique et une totale. Cette classe est en relation avec Pièces c’est à dire les pièces commandés. |
| Modele | Chaque véhicule a sa marque et un modèle,  Un véhicule peut être affecté à un ou plusieurs clients.  Le client choisit son véhicule avant pour mieux filtrer les pièces. |
| Pièce | Classe pièce représente l’enregistrement dans laquelle on stock les données relatives aux pièces en stock. |

TABLE 3 : Tableau descriptif des classes

## 2.3.3 Diagrammes de séquence système

Nous rappelons que les diagrammes de séquences peuvent servir à illustrer un cas d’utilisation décrit précédemment. C’est un moyen semi-formel de capturer le comportement de tous les objets et acteurs impliqués dans un cas d’utilisation. Dans ce qui suit, quelques scénarios de l’site sont présentés.

### 2.3.3.1 Diagramme de séquence du scénario "Authentification" la phase d’identification

Est indispensable pour la suite de l’utilisation de l’outil qui se déroule selon les informations relatives à l’utilisateur (administrateur, magasinier ou client). Le diagramme de séquence système dans la figure 11 illustre plus en détails le scénario "Authentification".

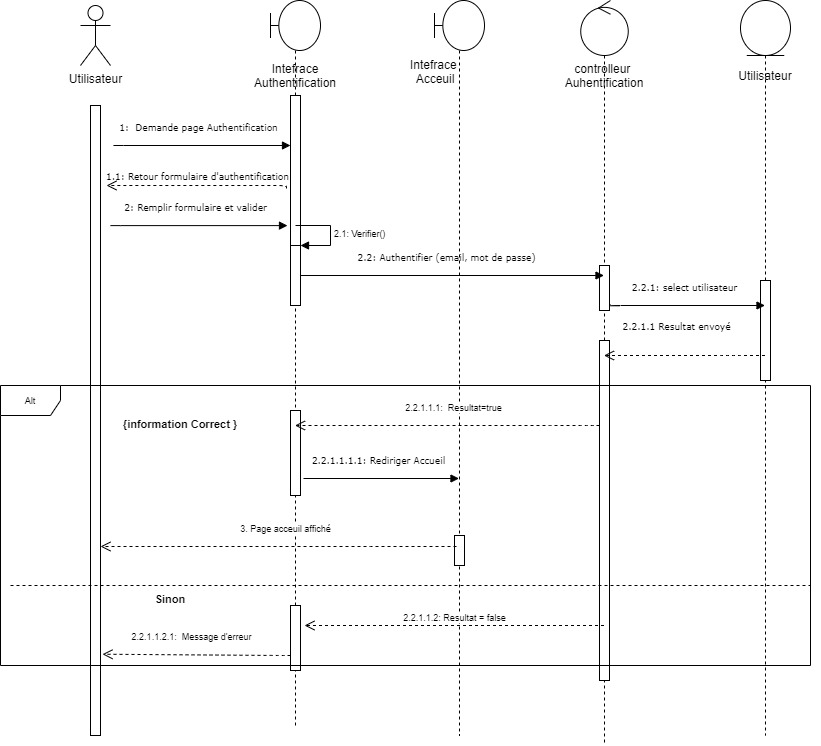


Figure 11: Diagramme de séquence du scénario « Authentification »

### 2.3.3.2 Diagramme de séquence du scénario "Modifier Produit" pour un magasinier (ou un administrateur) :

Le magasinier demande la liste des produits, le système à son tour affiche la liste des produits demandé. Il va sélectionner le produit à modifier, après la modification le système affiche soit un message de succès ou un message d’échec de modification suite à une erreur de saisie par exemple. Le diagramme de séquence dans la figure 12 illustre plus en détails le scénario “Modifier produit”.

Figure 12: Diagramme de séquence du scénario « Modifier produit »

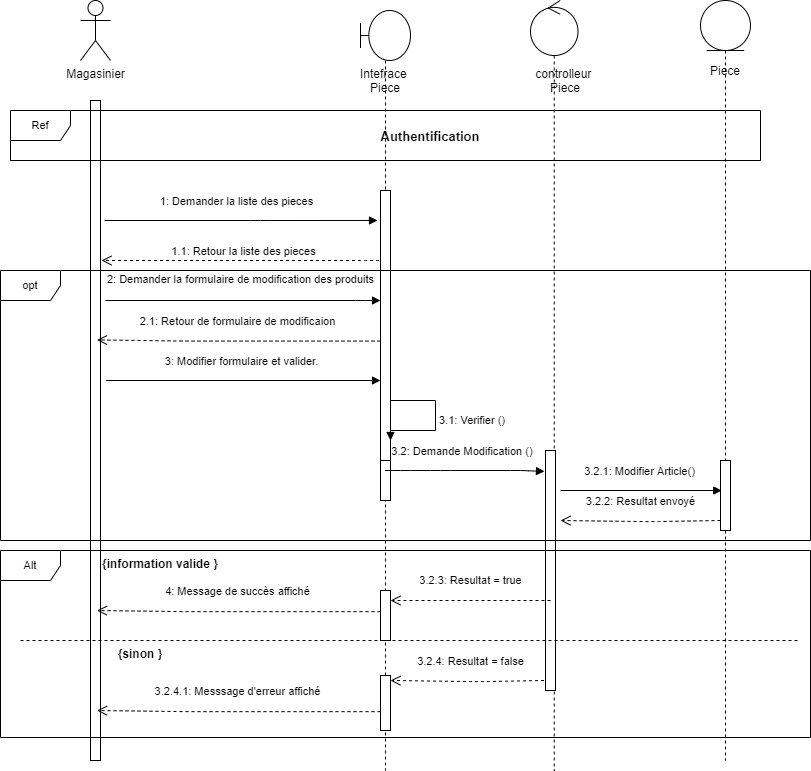


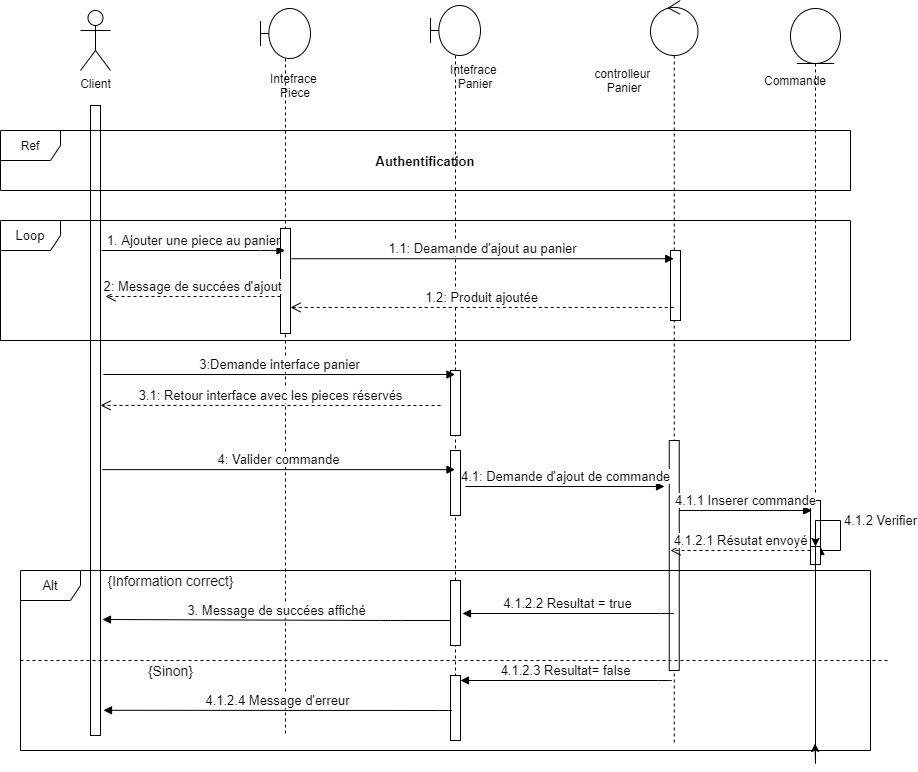
Figure 12: Diagramme de séquence du scénario « Modifier produit »

### 2.3.3.3 Diagramme de séquence du scénario "Passer Commande"

Après authentification, le client accède à la rubrique "Nos articles". Une liste des pièces sera affiché avec une bouton « plus de détails »,il choisit les pièces à commander, le système les enregistre dans le panier. Il valide le panier pour que le système avec la base de données puisse vérifier la validité des champs, un test doit être réalisé pour les produits. Si tous les champs sont correcte alors le système prend en charge les informations introduites et valide la commande sinon une erreur de saisie va être affichée.

Ce diagramme de séquence illustre plus en détails le scénario "Passer Commande".

Figure 13: Diagramme de séquence du scénario « Passer Commande »



# 2.4 Conclusion :

Ce chapitre nous a permis le formalisme de conception utilisé. Puis, une deuxième phase était consacrée à la conception, et donc l’élaboration de différents diagrammes.

Dans le troisième chapitre, nous présentons les étapes de réalisation de notre site ainsi que quelques interfaces représentatives de notre travail.

Chapitre 3

# Réalisation



# Introduction

L’étape de réalisation et de test représente également une étape fondamentale à entreprendre dans tout projet. Pour cela, nous présentons les différentes interfaces que nous avons réalisées et leurs descriptions seront aussi présentées tout au long de ce chapitre.

* 1. **Environnement et outils de travail**

Nous présentons dans ce volet l’environnement matériel mis à la disposition du projet, ainsi que l’environnement logiciel utilisé pour le développement et la mise en place de notre application.

* + 1. **Environnement matériel**

Pour la réalisation de ce projet, nous avons disposé d’un ordinateur ACER caractérisé par :

* Processeur : Intel® Core (TM) I5 -8250U CPU@ 1.60 GHz
* Mémoire : 4 Go de RAM
* Disque dur : 1 To
* Carte graphique : Intel® UHD Graphics 620
* Système d’exploitation : Windows 10

Et un ordinateur ASUS caractérisé par :

* Processeur : Intel® Core (TM) I5 -7200U CPU@ 2.50 GHz
* Mémoire : 4 Go de RAM
* Disque dur : 500 Go
* Carte graphique : Intel® UHD Graphics 620
* Système d’exploitation : Windows 10
  + 1. **Environnement logiciel**

Dans ce qui suit, nous présentons l’environnement logiciel utilisé pour mener à terme ce sujet.

1. **Framework utilisé**

* **React JS** (aussi appelé React.js ou ReactJS) **:** est une bibliothèque JavaScript libre développée par Facebook depuis 2003  Le but principal de cette bibliothèque est de faciliter la création d'[application web monopage](https://fr.wikipedia.org/wiki/Application_web_monopage), via la création de composants dépendant d'un état et générant une page (ou portion) [HTML](https://fr.wikipedia.org/wiki/HTML) à chaque changement d'état.
* **Node JS :** est une [plateforme logicielle](https://fr.wikipedia.org/wiki/Plate-forme_(informatique)) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre) en [JavaScript](https://fr.wikipedia.org/wiki/JavaScript) orientée vers les applications [réseau](https://fr.wikipedia.org/wiki/R%C3%A9seau_informatique) [événementielles](https://fr.wikipedia.org/wiki/Programmation_%C3%A9v%C3%A9nementielle) hautement [concurrentes](https://fr.wikipedia.org/wiki/Ing%C3%A9nierie_concourante) qui doivent pouvoir [monter en charge](https://fr.wikipedia.org/wiki/Scalability).
* **Mongo DB :** est un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) [orienté documents](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_orient%C3%A9e_documents), [répartissable sur un nombre quelconque d'ordinateurs](https://fr.wikipedia.org/wiki/Scalability) et ne nécessitant pas de schéma prédéfini des données. Il est écrit en [C++](https://fr.wikipedia.org/wiki/C%2B%2B).Il fait partie de la mouvance [NoSQL](https://fr.wikipedia.org/wiki/NoSQL" \o "NoSQL).
* **Express JS :** est un [Framework](https://fr.wikipedia.org/wiki/Framework) pour construire des [applications web](https://fr.wikipedia.org/wiki/Applications_web) basées sur [Node.js](https://fr.wikipedia.org/wiki/Node.js). C'est de fait le Framework standard pour le développement de serveur en [Node.js](https://fr.wikipedia.org/wiki/Node.js).

1. **Technologies**

* **Bootstrap** : est une [collection d'outils](https://fr.wikipedia.org/wiki/Framework) utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de [sites](https://fr.wikipedia.org/wiki/Site_web) et d'[applications web](https://fr.wikipedia.org/wiki/Application_web). C'est un ensemble qui contient des codes [HTML](https://fr.wikipedia.org/wiki/HTML) et [CSS](https://fr.wikipedia.org/wiki/Cascading_Style_Sheet), des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions [JavaScript](https://fr.wikipedia.org/wiki/JavaScript) en option

1. **Langage de programmation :**

* **JavaScript :** est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js. Le langage JavaScript permet des pages web interactives, et à ce titre est une partie essentielle des applications web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript dédié pour l'interpréter, indépendamment des considérations de sécurité qui peuvent se poser le cas échéant.

1. **Logiciels de développement :**

* **Visual Studio Code :** Visual Studio Code est un éditeur de code multiplateforme édité par Microsoft. Cet outil destiné aux développeurs supporte plusieurs dizaines de langages de programmation comme le HTML, C++, PHP, Javascript, Markdown, CSS, etc… Visual Studio Code intègre plusieurs outils facilitant la saisie de code par les développeurs comme la coloration syntaxique ou encore le système d'auto-complétions IntelliSense. En outre, l'outil permet aux développeurs de corriger leur code et de gérer les différentes versions de leurs fichiers de travail puisqu'un module de débogage est aussi de la partie.
* **HTML (HyperText Markup Language) :** C'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. HTML permet également d'architecturer sémantiquement 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 tels que des applets.
* **CSS (Cascading Style Sheets) :** Ce sont les feuilles de styles qui désignent l'ensemble d'attributs de caractères et de formats de paragraphes pouvant être appliqués en une seule opération à un ou plusieurs paragraphes, à un livre etc. De façon extensive, c'est l'ensemble de contraintes typographiques, de maquette, de couleur, de reliure etc.… qui définissent l'unité d'une collection littéraire, d'une revue périodique etc...

1. **Logiciel de Modélisation :**

* **UML** : "Unified Modeling Language" est un langage de modélisation graphique, il est aujourd’hui utilisé comme standard international, incontournable, stabilisé et industriel.

1. **Logiciel de Modélisation**

* **Star UML** : est un logiciels de modélisation [UML](https://fr.wikipedia.org/wiki/UML_(informatique)), qui a été "cédé comme [open source](https://fr.wikipedia.org/wiki/Open_source)" par son éditeur, à la fin de son exploitation commerciale (qui visiblement continue ...), sous une licence modifiée de GNU GPL.
* **Draw.io :** est une application gratuite en ligne, accessible via son navigateur (protocole http) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter.
  1. **Travail Réalisé :**

Dans cette partie, nous présentons notre travail en exposant des captures d’écran des interfaces les plus importantes de notre application.

* + 1. **Page d’accueil :**

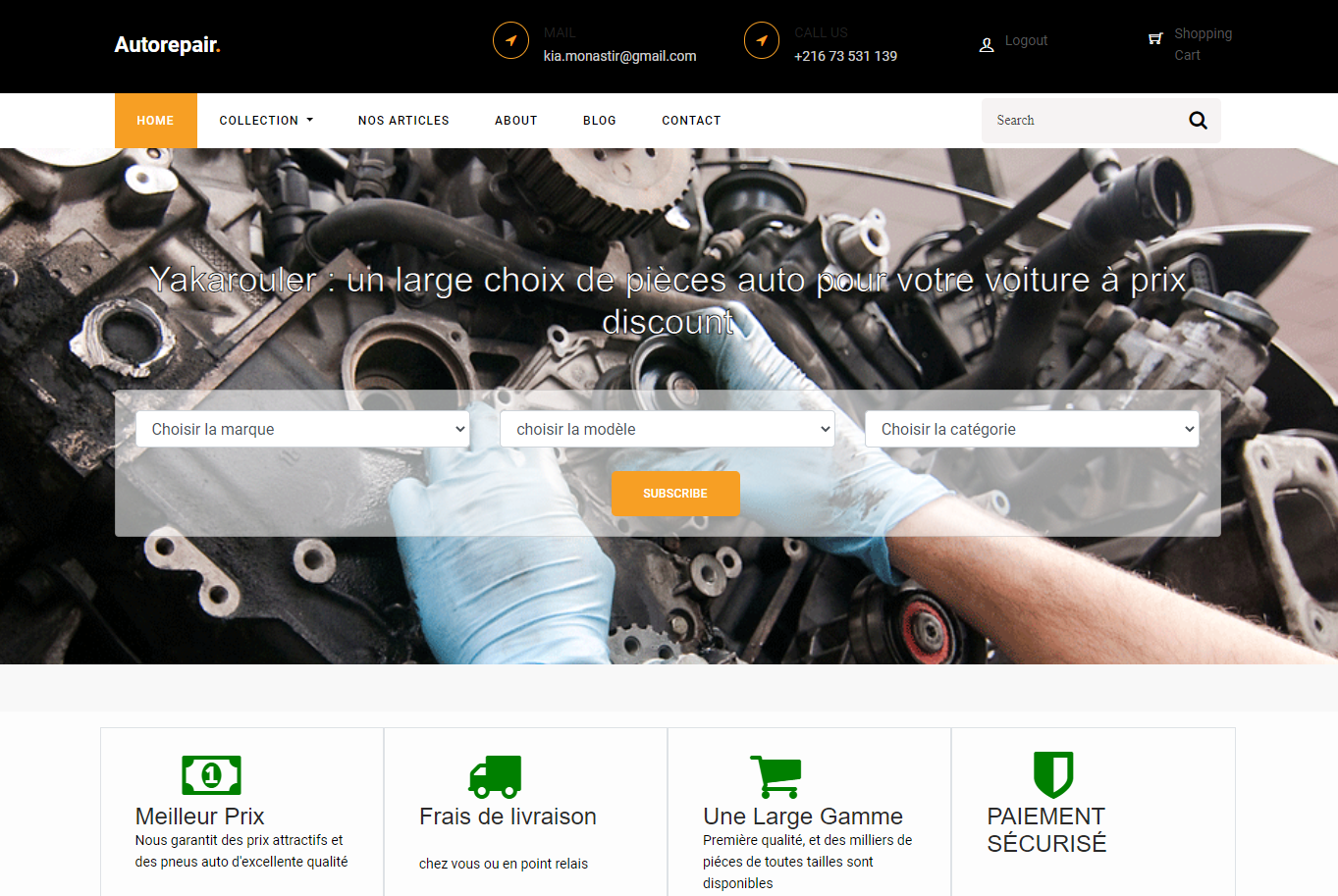


Figure 14: Interface d’accueil

L’interface d’accueil de l’application contient le menu principal du site Web. Des liens hypertextes donnent la possibilité à l’utilisateur d’accéder directement aux rubriques qui l’intéressent. Ces pages constituent également des menus qui peuvent être subdivisés en sous-menus.

Si l’utilisateur clique sur le lien « Nos articles » de l’interface d’accueil, il est dirigé vers une interface dont les listes des pièces existent. Aussi si l’utilisateur clique sur le lien « Se connecter», il est invité à s’authentifier s’il a un compte sinon il doit s’inscrire.

* + 1. **Interface d’authentification :**

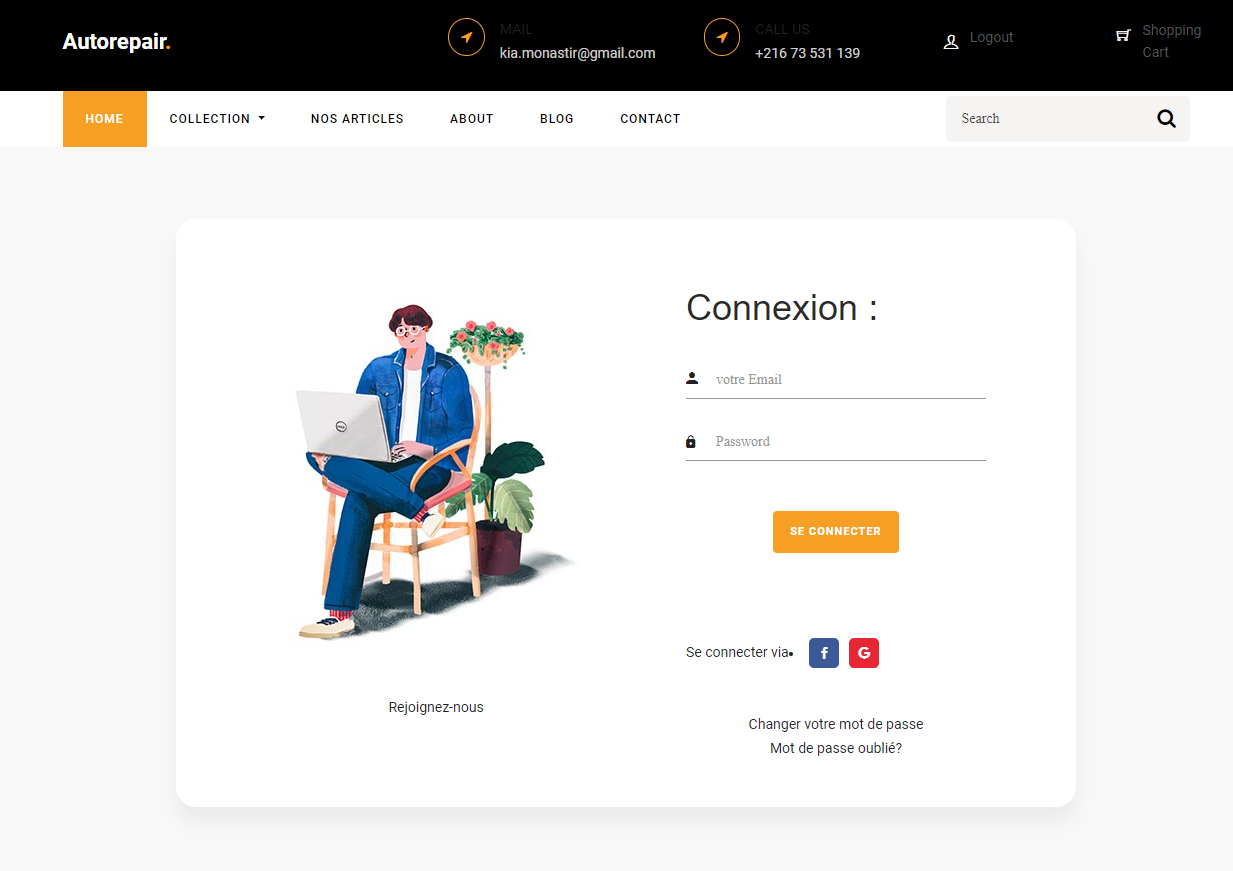
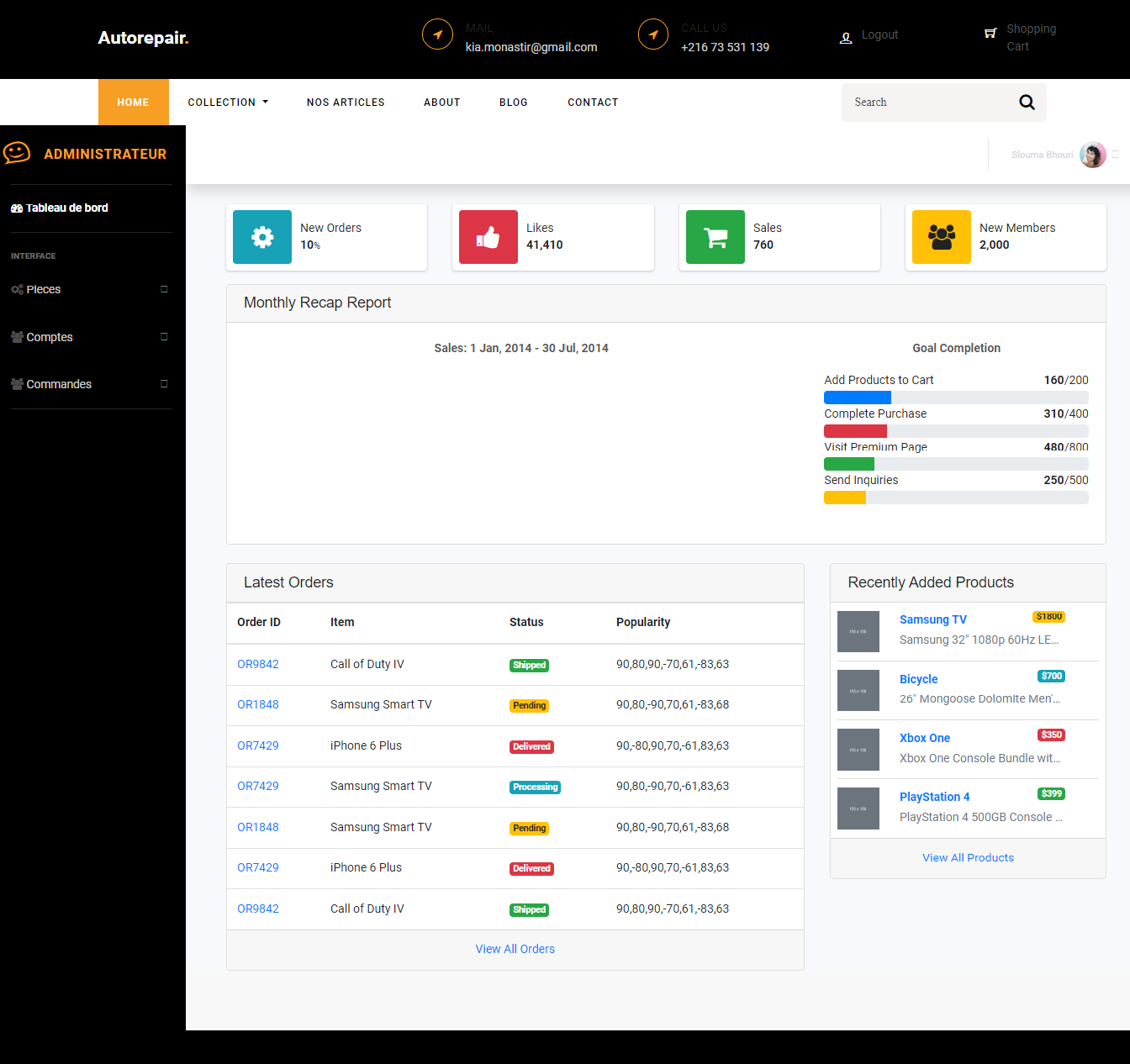


Figure 14: interface d’authentification

L’utilisateur doit s’authentifier pour pouvoir accéder à son espace. Dans cette interface, l'utilisateur remplit ces champs. S’ils sont corrects il sera dirigé vers la page d’accueil. S’il veut connecter à l’aide du « Facebook » ou « Google » il doit cliquer sur l’un des deux buttons. S'il a oublié son mot de passe, il peut le récupérer en cliquant sur "Mot de passe oublié ?". Dans ce cas, un lien de changement de mot de passe sera envoyé sur l'adresse mail saisie lors de l'inscription. S'il ne dispose pas d'un compte, il doit s'inscrire en cliquant sur "Rejoignez-nous".

* + 1. **Interface Administrateur :**

La figure 15 présente l’interface de l’administrateur qui contient 4 menus : tableau de bord, pièces (visualiser, ajouter, modifier) ,compte (ajouter des comptes, modifier , supprimer ) ou bien commande ( un tableau qui contient tous les commandes des clients , les supprimer ou les valider )



**Figure 15: interface d’espace administrateur**

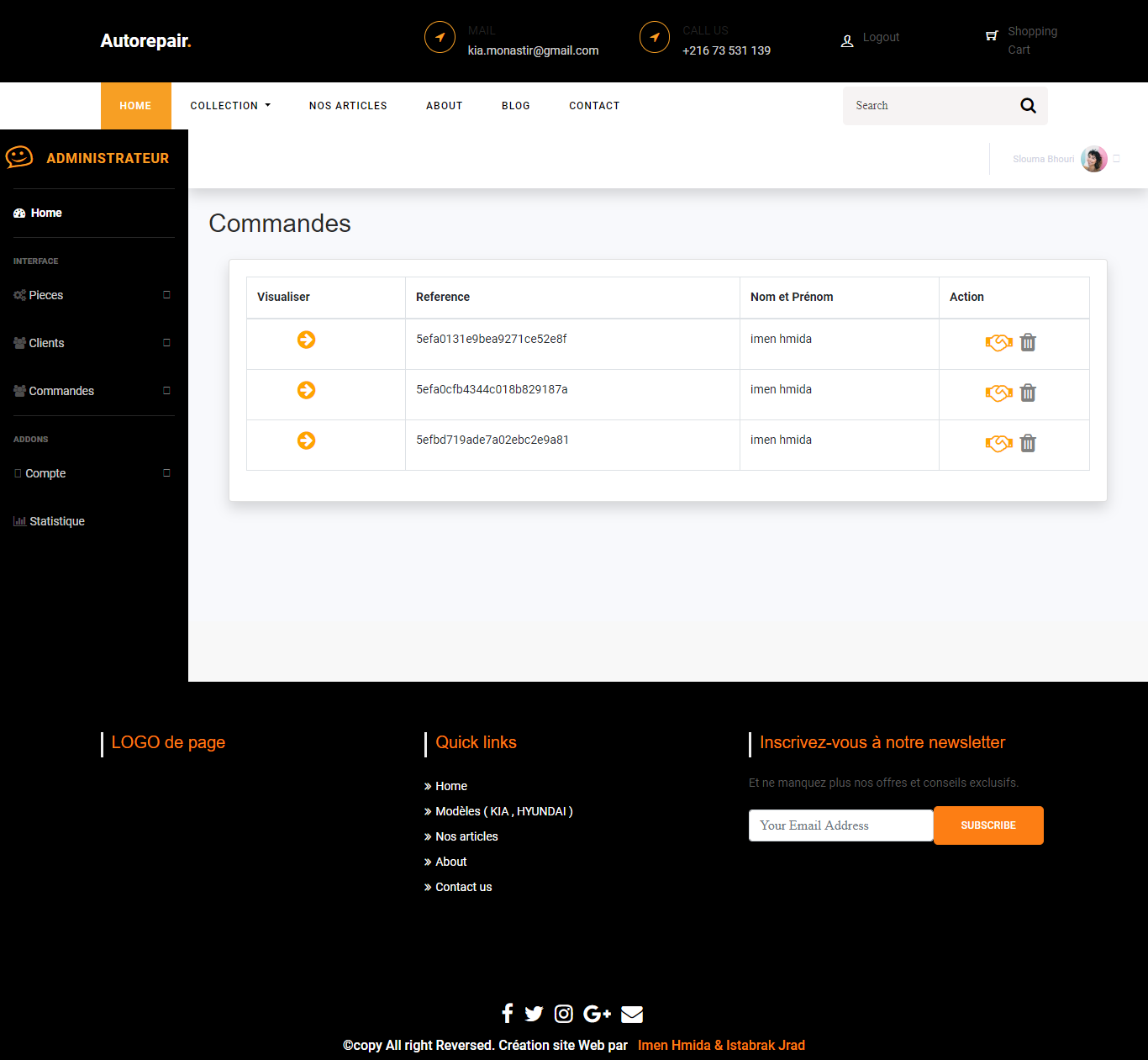
* + 1. **Interface des commandes :**

Figure 16: interface des commandes

L’interface illustre les différentes commandes, si l’utilisateur choisi une commande pour la consulter, il doit cliquer sur le bouton « Visualiser» pour les mieux détailler , les supprimer ou bien les valider.

* + 1. **Interface des bons de commande :**

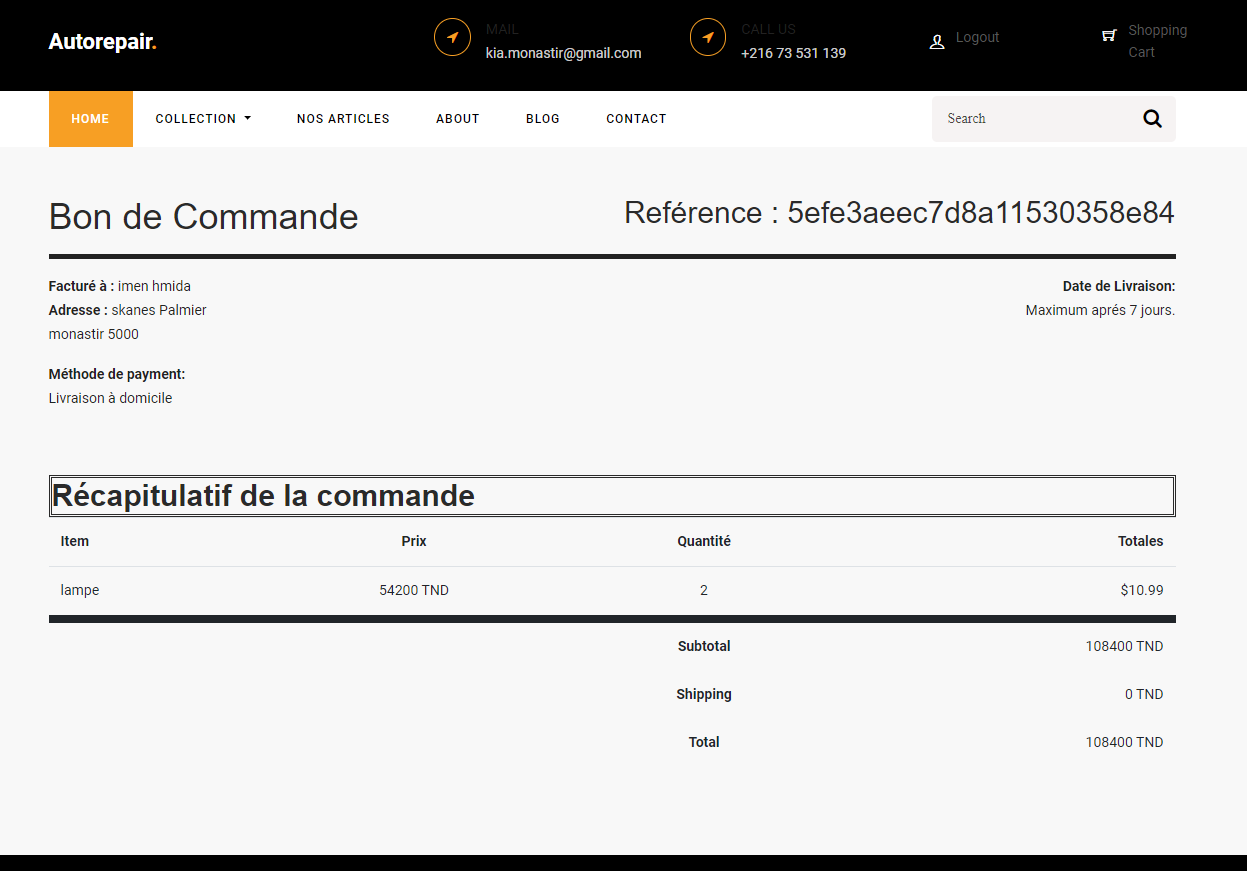


Figure 17: interface des bons de commande

L’interface illustre la bon de commande, sa référence, le nom du client, son adresse, les pièces commandés, la quantité acheté, le prix, le prix total de tous les achats, et le prix plus le shipping.

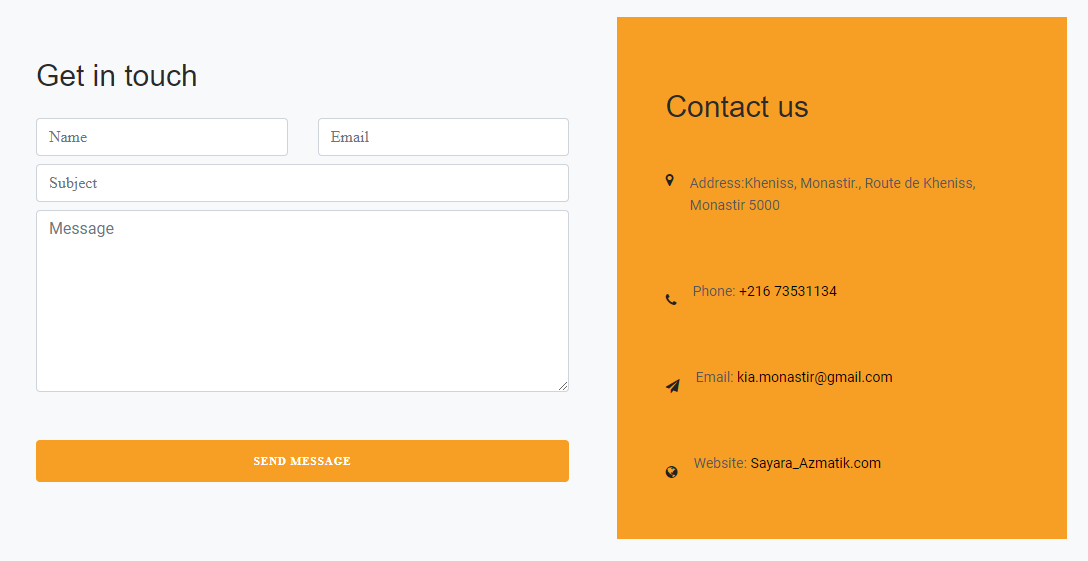
* + 1.  **Interface de contact :**

Figure 18: interface de contact

La figure 18 ci-dessous montre l'interface se formulaire de contact pour toute information supplémentaire ou réclamation.

* 1. **Conclusion :**

Ce chapitre a été consacré à la description de la partie réalisation de notre application. Quelques interfaces sont exposées pour expliquer le fonctionnement de notre projet. Maintenant après avoir terminé tous les chapitres nous allons résumer notre travail dans la conclusion générale.

# Conclusion générale

Pour conclure, j’ai effectué mon stage de fin d’études de (7Ot esm licence mte3ik) …….. en tant que stagiaire en développement informatique au sein de l’entreprise KIA et HYNDAI Monastir.

L’objectif de ce projet est de concevoir et de développer une application web et Mobile pour la gestion des ventes en ligne des pièces automobile et (chnawa e5er ya3ml el site) ……………………… qui va permettre une meilleure augmentation des revenus et l’expansion du marché, et elle va résoudre plusieurs problèmes de mobilité des acheteurs, suivie de vente de pièces pour le responsable Kia Monastir ainsi que la suivie de la quantité des pièces dans le magasin.

Nous avons bien précisé au début de notre rapport que nous envisageons comme première phase l’étude préliminaire qui nous a permis de spécifier les besoins fonctionnels et non fonctionnels . Ensuite, nous avons entamé les différents choix technologiques et techniques adoptés pour la réalisation de ce projet. Par la suite, nous avons présenté une conception bien détaillée à travers la représentation des différents types de diagrammes . Notre rapport est clôturé par une phase de réalisation qui comporte une exposition du notre travail. Lors de ce stage de 6 mois, nous avons pu mettre en pratique nos connaissances théoriques acquises durant notre formation dans les domaines que nous connaissons déjà tels que ReactJS et nodeJs , et nous nous confrontons aux difficultés du monde du travail. Cependant, notre projet ne manque pas de perspectives. En effet, on peut envisager d’améliorer ce site en ajoutant d’autre module comme ………………..

# Bibliographie

1. *HTML5.* [En ligne] http ://www.html5-css3.fr/html5/introduction-html5.

2. *CSS3.* [En ligne] http ://www.html5-css3.fr/css3/introduction-css3.

3. *JS.* [En ligne] https ://developer.mozilla.org/fr/docs/Web/JavaScript].

4. *MongoDB.* [En ligne] https://fr.wikipedia.org/wiki/MongoDB.

5. *w3schools.* [En ligne] https://www.w3schools.com/nodejs/nodejs\_intro.asp.

6. **Pascal Roques et Franck Vallée.** *UML 2 en action De l’analyse des besoins à la conception 4ième édition.* février 2007.