**République algérienne démocratique et populaire**

**Ministère de l’enseignement supérieur et de la recherche scientifique**

**Université Abdelhamid Mehri – Constantine 2**

**Faculté des nouvelles technologies de l’information et de la communication (NTIC)**

**Département d’informatique fondamentale et ses applications (IFA)**

**MÉMOIRE DE MASTER**

*pour obtenir le diplôme de Master en Informatique*

**Option : Sciences et technologies de l’information et de la  
communication (STIC)**

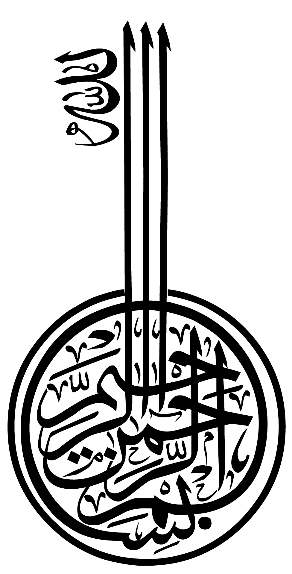
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Application Web Progressive pour la vente de pièces automobile**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

| **Réalisé par :**  HOUACHE Brahim  HADBOUN Daoud | **Sous la direction de :**  MOSTEFAI Sihem |
| --- | --- |

Juin 2022



|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Remerciements** |
|  | | | | |

Nous tenons avant toute chose, à remercier Allah, qui nous a aidé à achever notre projet et nous a inspiré patience et santé tout au long de nos années d’études.

Nous adressons également nos sincères remerciements et appréciations à notre encadrant

Dr SIHEM MOSTEFAI pour le temps qu’elle a consacré et pour les conseils et les précieuses informations qu’elle nous a prodigués avec intérêt et compréhension.

Nous tenons à remercier également les membres du jury pour avoir bien voulu examiner et juger ce travail.

Enfin, nous tenons à remercier tous nos enseignants de la faculté des NTICs, et tous les collègues de spécialité STIC que nous avons contacté durant ces cinq dernières années auprès desquels nous avons trouvé l’aide dont nous avons besoin.

|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Dédicaces** |
|  | | | | |

**Dédicace 1**

*Rien n'est aussi beau à offrir que le fruit d'un labeur qu'on dédie du fond du cœur à ceux*

*qu'on aime et qu'on remercie en exprimant la gratitude et la reconnaissance durant toute notre existence.*

*Je dédie ce modeste travail :*

*À Mes chers parents qui ont sacrifié leur vie pour ma réussite et qui m'ont soutenu psychologiquement. J’espère qu’un jour, je pourrai leur rendre un peu de ce qu’ils ont fait pour moi, que dieu leur prête bonheur et longue vie.*

*À mon frère et à mes et sœurs et toute ma famille.*

*À tous mes enseignants du primaire jusqu'à l'université.*

*À tous mes amis, à toute la promotion, à tous mes compatriotes.*

**Daoud**

**Dédicace 2**

*Toute l'encre du monde ne peut suffire à exprimer mes sentiments pour toi, ma chère mère, tu as toujours été mon école dans cette vie, j'ai appris de toi la confiance et surtout l'espoir et l'amour, Que Dieu ait pitié de ton âme et te place au paradis.*

*A mon cher père, tu as été mon plus grand soutien après le décès de ma mère, tu es et resteras pour moi ma référence, la lumière qui éclaire mon chemin. Tu m'as montré encouragement et soutien, Loué soit Dieu, qui m'a permis de réaliser ton rêve, mon père. Que Dieu ait pitié de votre âme et vous accorde le paradis, Comme j'aimerais que vous soyez ici avec moi en ce grand jour, mes chers parents.*

*A celle qui a réduit la douleur de la mort de ma mère, la femme de mon père,*

*À ma chère épouse qui n'a jamais cessé de me soutenir, tu as toujours voulu le meilleur de moi. Je remercie Dieu de t'avoir dans ma vie*

*A mes frères et sœurs qui ont toujours été mon soutien*

*À toute ma famille et mes amis , qui m'ont supporté et encouragé à suivre ma passion. C'est à vous que j'offre le fruit de mon travail.*

**Brahim**

**ملخص**

يعاني الكثير من الميكانيكيين وأصحاب السيارات في الجزائر من إيجاد قطع غيار السيارات وذلك بتنقلهم من متجر إلى آخر و أحيانا من مدينة إلى أخرى وهذا يؤرق هؤلاء الأشخاص و يكلفهم الكثير من المال و الوقت وبالرغم من توفر عدد قليل من مواقع الويب التي تعرض هذه القطع إلا أنها لا تفي بالغرض لان معظم هؤلاء الأشخاص لا يتصفحون المواقع بل يستعملون الهواتف النقالة وتطبيقاتها..

في هذه الأطروحة نقترح نظاما تحت مسمى « AutoPartDz » والذي يعتبر أحد الحلول للمشاكل المذكورة سابقا وهذا النظام عبارة عن تطبيق ويب وموقع ويب في نفس الوقت، يتميز تطبيقنا عن باقي التطبيقات أنه تطبيق ويب تقدمي.

نظامنا هذا موجه لأصحاب ورشات الميكانيك ومالكي السيارات وأصحاب متاجر بيع قطع غيار السيارات.

فبالنسبة للعملاء استعمال التطبيق يكون بدخولهم أول مرة للموقع وتنزيل ملف صغير الحجم لتثبيته في الهاتف ليصبح بعدها مثل التطبيقات الأصلية للهاتف وبعد ذلك يستطيع الدخول إلى التطبيق وتصفح المنتوجات وتقديم طلبياته وتقديم استفسارات عن المنتوجات ويستطيعون استعمال التطبيق حتى في وجود أنترنت ضعيفة أو انعدامها ذلك لأن التطبيق يدعم تقنية حفظ الصفحات وأيضا حفظ التنقلات داخل التطبيق.

أما بالنسبة لأصحاب المتاجر فقد خصصنا واجهة خاصة لهم حيث يستطيعون عرض سلعهم واستقبال طلبيات العملاء و رسائلهم ويقدم لهم التطبيق إحصائيات عامة تساعدهم على اتخاذ قرارات هامة في تجارتهم.

**الكلمات المفتاحية:** موقع ويب ، تطبيق ويب تقدمي ، قطع غيار السيارات ، التطبيقات الأصلية ، أنترنت ضعيفة ، حفظ cache التنقلات.

**Abstract**

Mechanics and car owners in Algeria suffer from finding auto parts by moving from one store to another, sometimes from one city to another, which causes inconveniences and costs them lots of money and time. Although there are a few websites that display these parts, it is not enough because people do not browse websites but use cell phones and their apps.

In this project, we propose a system called "AutoPartDz ", as a solutions to the problems mentioned before. It behaves like a web application and a mobile application at the same time, it is known as a Progressive Web App.

Our system targets mechanic shop owners, car owners and auto parts stores.

Customers can easily use the application by entering the site and downloading a small file to install in the phone, then it becomes like a native application which can run with a weak internet connection or even without internet. This is made possible because the application keeps pages and app navigation history in cache.

For the store admins, we have designed a special interface where they display their products and receive customer orders and messages. The application also provides general statistics which help make important business decisions in the future.

**Keywords:**, Website, Progressive Web App, auto parts, native applications, weak internet, navigation cache.

**Résumé**

De nombreux mécaniciens et propriétaires de voitures en Algérie peinent à trouver des pièces automobiles en se déplaçant d’un magasin à l’autre, parfois d’une ville à l’autre, ce qui dérange ces personnes et leur coûte beaucoup d’argent et de temps. Bien qu’il existe quelques sites Web qui affichent ces pièces, ce n'est pas suffisant car la plupart des gens ne naviguent pas sur des sites Web mais utilisent des téléphones portables et leurs applications.

Dans ce projet de fin d’étude, nous proposons un système appelé « AutoPartDz », comme solution aux problèmes mentionnés précédemment. Ce système fonctionne à la fois comme une application Web et une application mobile, appelée application Web progressive.

Notre système cible les propriétaires d'ateliers mécaniques, les propriétaires de voitures et les magasins de pièces automobiles.

Les clients peuvent facilement utiliser l’application en entrant sur le site et en téléchargeant un petit fichier à installer. L’application devient alors comme une application native qui fonctionne même en présence d’une connexion Internet faible ou inexistante, parce que l’application prend en charge la technologie de mise en cache des pages et l’historique de navigation.

Pour l'administrateur de magasin, nous avons conçu une interface spéciale lui permettant d’afficher ses produits, reçevoir les commandes des clients et leurs messages. L'application fournit aussi des statistiques générales qui aident à prendre des décisions commerciales importantes à l'avenir.

**Mots clés:** Site Web, Application Web Progressive, Pièces Automobiles, Application native, Internet faible, cache de navigation.

|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Table des matières** |
|  | | | | |

[**Remerciements i**](#_30j0zll)

[**Table des matières vii**](#_1fob9te)

[**Table des figures ix**](#_3znysh7)

[**Liste des tableaux x**](#_2et92p0)

[**Introduction générale 1**](#_3dy6vkm)

[**1**](#_17dp8vu) **PWA - État de l’art 4**

[1.1](#_3rdcrjn) Applications Web réactives et Applications natives 5

[1.1.1](#_26in1rg) Applications Web Réactives 5

[1.1.2](#_lnxbz9) Application Native 5

[1.2](#_35nkun2) Progressive web Application 6

[1.2.1](#_1ksv4uv) Définition 6

[1.2.2](#_44sinio) Les Caractéristique Des PWAs 7

[1.2.3](#_2jxsxqh) Comparaison 8

[1.2.4](#_2xcytpi) Aperçu des forces et faiblesses 10

[1.3](#_3whwml4) La vente en ligne 12

[Investissement des entreprises de vente en ligne dans les PWAs en 2021 12](#_2bn6wsx)

[Conclusion 13](#_3as4poj)

[**2**](#_1pxezwc) **Modélisation 14**

[2.1](#_49x2ik5) Modélisation UML du système proposé 15

[2.1.1](#_2p2csry) Elaboration du diagramme de cas d’utilisation 15

[2.1.2](#_2grqrue) Description des scénarios des cas d’utilisation 18

[2.1.3](#_19c6y18) Elaboration du diagramme de classe 20

[2.2](#_nmf14n) Structure d'une Progressive web app 21

[2.3](#_1mrcu09) Les technologies de PWA 22

[2.3.1](#_46r0co2) Manifeste de l'application Web 22

[2.3.2](#_2lwamvv) Service Workers 23

[Conclusion 25](#_3l18frh)

[**3**](#_206ipza) **Implémentation 26**

[3.1](#_4k668n3) Implémentation d’application web ‘AutoPartDZ’ 27

[Présentation du système proposé 27](#_2zbgiuw)

[3.2](#_1egqt2p) Description de l’implémentation de l’application ‘AutoPartDZ’ 28

[3.2.1](#_3ygebqi) Points d'entrée de l'application 28

[3.2.2](#_2r0uhxc) Types de comptes 33

[3.3](#_43ky6rz) Lighthouse (Outil de test PWA) 39

[Conclusion 40](#_xvir7l)

[**Conclusion générale 41**](#_3hv69ve)

[**Bibliographie 43**](#_1baon6m)

[**Acronymes 46**](#_3vac5uf)

[**Annexe 47**](#_2afmg28)

[Outils de développements 47](#_pkwqa1)

[Frameworks utilisés 48](#_39kk8xu)

[Langages de programmation utilisés 49](#_1opuj5n)

[Base de données 50](#_48pi1tg)

[Environnement de développement (IDE) 51](#_2nusc19)

[Autres outils utilisés 52](#_1302m92)

|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Table des figures** |
|  | | | | |

[Figure 1 : Capacités par rapport à la portée entre des application native, des applications Web et des applications Web progressives 10](#_4i7ojhp)

[Figure 2 : Part des entreprises de vente en ligne prévoyant d'investir dans des applications Web progressives (PWA) en Amérique du Nord et en Europe en 2021 12](#_qsh70q)

[Figure 3 : Acteurs du système 15](#_23ckvvd)

[Figure 4 : Diagramme de cas d’utilisation Administrateur 16](#_32hioqz)

[Figure 5 : Diagramme de cas d’utilisation Visiteur / Client 17](#_41mghml)

[Figure 6 : Diagramme de Classe 20](#_28h4qwu)

[Figure 7 : Application Web progressive 22](#_37m2jsg)

[Figure 8 : Un cycle de vie d’un Service Worker 24](#_111kx3o)

[Figure 9 : Interface Desktop du visiteur dans l’application ‘AutoPartDZ’ 29](#_2dlolyb)

[Figure 10 : Interface Desktop pour la création de nouveau compte Client 30](#_sqyw64)

[Figure 11 : Interface Mobile pour la création de nouveau compte Client 31](#_3cqmetx)

[Figure 12 : Interface d’authentification Desktop de l’application ‘ AutoPartDZ ‘ 32](#_1rvwp1q)

[Figure 13 : Interface d’authentification Mobile de l’application ‘ AutoPartDZ ‘ 32](#_4bvk7pj)

[Figure 14: Les Interfaces de la fonctionnalité passer la commande 33](#_1664s55)

[Figure 15: Les Interfaces de Client proposée par l’outil ‘AutoPartDZ’ 34](#_3q5sasy)

[Figure 16 : Dashboard d’un administrateur 35](#_25b2l0r)

[Figure 17 : La liste des utilisateurs (Administrateur + Client) 36](#_kgcv8k)

[Figure 18 : Interface d'ajout ou de modification d’un produit 37](#_34g0dwd)

[Figure 19 : Etape de traitement d’une commande 38](#_1jlao46)

[Figure 20 : Exemple de rapport Lighthouse 39](#_2iq8gzs)

|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Liste des tableaux** |
|  | | | | |

[Tableau 1 : Apporter le meilleur des deux mondes 9](#_1y810tw)

[Tableau 2 : Avantages et inconvénients des progressives web apps 10](#_1ci93xb)

[Tableau 3 : Fiche descriptive du cas d'utilisation « S’authentifier » 18](#_1v1yuxt)

[Tableau 4 : Fiche descriptive du cas d'utilisation « Commander » 19](#_2u6wntf)

|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Introduction générale** |
|  | | | | |

**Contexte général**

L'économie mondiale a subi des changements importants ces dernières années, et l'ouverture du marché et la rapidité des développements technologiques ont poussé les entreprises à se retrouver à un niveau de concurrence, de complexité et d'incertitude plus élevé qu'auparavant. Actuellement, la croissance et la rentabilité des entreprises doivent dépendre de la capacité à introduire ces nouvelles technologies émergentes et à adopter de nouvelles méthodes de transactions commerciales.

Les chiffres du commerce électronique dans le monde n'ont cessé d'augmenter depuis l'émergence de pionniers du commerce électronique tels que Amazon, eBay ou Yahoo! Au milieu des années 90, le développement du e-commerce a fortement contribué au développement des usages du web auprès du grand public, et l'usage de l'Internet mobile via les smartphones et les tablettes a abouti au statut du e-commerce de manière concluante chez les personnes.

En 2021, l'Union Internationale des Télécommunications estime que près de 4,9 milliards de personnes sont connectées à Internet [1], soit plus de la moitié de la population mondiale, Et pendant que ces personnes sont connectées, elles seront sur différents appareils avec des vitesses de connexion différentes et des conditions en constante évolution. Les développeurs web se retrouvent face au problème de ne pas pouvoir répondre à tous ces différents scénarios, cela semble très difficile. C'est là qu'interviennent les applications Web Progressives ou PWA (Progressive Web Applications), car elles nous aident en tant que développeurs à créer des sites Web plus rapides, plus efficaces et plus attrayants (comme Uber, Pinterest), auxquels des milliards de personnes dans le monde peuvent accéder [2].

Dans ce projet, nous allons développer une application Web Progressive qui aide les concessionnaires de pièces automobiles à vendre leurs produits confortablement et qui garantit un accès à un plus grand public, grâce à la technologie "PWA" qui améliorera l’accessibilité de l’application.

**Problématique et motivation**

Afin d'améliorer le commerce électronique, il est nécessaire de fournir un bon service à l'utilisateur en termes de fiabilité, de performances et de rapidité de service. Parmi les techniques qui permettent de doter les applications Web/mobile classiques de ces fonctionnalités, il y a les Applications Web Progressives (Progressive Web Application)

Dans le cadre de ce projet, nous nous efforcerons de trouver des solutions aux problèmes suivants :

Qu'est-ce que le Progressive Web Apps ?

Pourquoi créer des applications Web progressives ?

Quelle est la différence entre les applications natives, les PWAs et les sites Web traditionnels ?

Qu'est-ce qui fait une bonne Progressive Web App ?

Quels sont les avantages et les limites d'avoir PWA ?

Comment les applications Web progressives peuvent favoriser le succès des entreprises ?

Pourquoi les entreprises ont besoin d'une application Web progressive ?

Comment les PWA peuvent fonctionner avec le commerce électronique ?

La motivation principale de ce projet de fin d'études est donc, de contribuer à fournir une application web progressive ou PWA qui permet d'aider les propriétaires et les mécaniciens automobiles à faciliter le processus de recherche et d’achat des pièces détachées, même avec une connexion internet limitée, ils n'auront pas à aller au magasin de pièces détachées. L’autre motivation, c’est d'aider les revendeurs de pièces détachées automobiles à promouvoir leurs produits et à augmenter leurs profits.

**Plan du mémoire**

Ce mémoire est principalement structuré en trois chapitres :

* **Chapitre 1 :** PWA : Dans ce chapitre, nous présenterons le concept de PWA, définition de la PWA, puis nous soulignerons l'importance de la PWA et pourquoi nous devrions l'utiliser dans les sites Web, nous comparerons également entre les applications PWA et Application native et les sites Web traditionnels, Nous évoquerons également de la relation entre les PWA et les sites E-commerce, nous présenterons les avantages et inconvénients des PWA, et enfin montrerons l'importance des PWA dans les sites e-commerce.
* **Chapitre 2 :** Conception : ce chapitre s'intéresse à la modélisation de notre système. Il commence par les caractéristiques globales du projet. Nous utilisons les diagrammes UML comme formalisme de représentation des différents aspects de la conception de notre application, En suit la structure de PWA et les technologies du PWA : ce partie présente les exigences techniques nécessaires à la mise en place d’applications PWA, en particulier : HTTPS du site Web, le fichier Manifeste Web,les Service Workers pour une application web progressive.
* **Chapitre 3 :** Implémentation : ce chapitre s'intéresse à décrire des détails de l’implémentation de notre application à travers les outils et les langages de programmation qui ont été utilisés, les interfaces principales de l’application.

Enfin, nous clôturerons ce mémoire par une conclusion générale des travaux réalisés, qui en souligne les principaux aspects et en suggère quelques perspectives.

|  | | **Chapitre** | | |
| --- | --- | --- | --- | --- |
|  |  | | **1** |  |
| 1. **PWA - État de l’art** |
|  | | | | |

Aujourd'hui, un site Web fait partie intégrante de toute entreprise et, pour pouvoir être compétitives, les entreprises devraient envisager d'intégrer les dernières avancées en matière de développement Web dans leur présence en ligne, rendre l'expérience utilisateur aussi transparente que possible. Une de ces avancées est connue sous le nom de Progressive Application Web ou PWA qui a gagné en popularité au cours des deux dernières années. De nos jours, les entreprises ont souvent demandé à leurs développeurs Web de créer un site Web prenant en charge la fonctionnalité PWA. La connaissance de ce domaine devient donc une compétence essentielle pour les développeurs Web.

À l'heure actuelle, cette technologie fait l'objet de nombreuses discussions dans la communauté des développeurs Web. Certains disent qu'il n'est pas nécessaire de faire des PWAs car les technologies Web existantes sont déjà suffisantes pour faire des sites Web avec les mêmes capacités. Par exemple, ils soutiennent que la création d'un site Web réactif ou une application native suffit pour éviter d'utiliser des PWA. Alors que d'autres soutiennent que ce qui précède les technologies ne disposent pas des fonctionnalités disponibles dans les PWAs. Cependant, il est important de donner un aperçu sur ce que sont les PWA, les sites Web réactifs et les applications natives avant de se plonger dans l'un de ces arguments.

* 1. **Applications Web réactives et Applications natives**
     1. **Applications Web Réactives**

La conception Web réactive (Responsive Web Design) consiste à créer des sites Web qui peuvent s'adapter à n'importe quelle taille d'écran, que ce soit une tablette ou un mobile. De nos jours, il est difficile de trouver un site Web professionnel qui ne réponde pas à cette exigence, considérée comme incontournable. En fait, la plupart des utilisateurs ne passent pas beaucoup de temps sur des sites Web qui ne correspondent pas à la taille de l'écran de leur appareil.

Pour rendre un site Web réactif, il existe un ensemble commun d'outils tels que les requêtes multimédia CSS, les mises en page et les grilles flexibles et les images. Lorsque l'utilisateur passe de son ordinateur portable à un smartphone, le site Web s'adapte automatiquement à la taille de cet appareil.

* + 1. **Application Native**

Une application native est une application mobile créée spécialement pour les systèmes d'exploitation mobiles présents sur les smartphones et les tablettes. Ces applications sont des programmes qui sont construits pour fonctionner sur les différents systèmes d'exploitation qui sont maintenant utilisés par les tablettes et les smartphones (iOS d'Apple ou Android de Google ou Windows Phone de Microsoft).

Les applications natives offrent un haut niveau de performance et de fiabilité. Elles peuvent également utiliser les nombreuses fonctionnalités du téléphone, telles que l'appareil photo et GPS et le Stockage Interne. De plus, certains programmes peuvent être utilisés sans connexion en ligne. Ce type de programme est coûteux à produire car il est lié à un seul système d'exploitation, obligeant le développeur à générer des copies en double qui s'exécutent sur d'autres plateformes. Il occupe également un grand espace lorsqu'il est installé sur le téléphone, et son installation nécessite l'accès aux Stores fournis par le système d'exploitation du téléphone, et cela nécessite également une connexion Internet avec un débit fort, car le téléchargement est impossible avec un débit faible. Ici on constate que les applications Native manquent parfois de performance. Il y a un autre problème : c’est la compatibilité avec le système d'exploitation : par exemple, les applications qui fonctionnent sur la version d’Android Oreo 8.1 ne fonctionnent pas sur Android KitKat 4.4 .

* 1. **Progressive web Application**
     1. **Définition**

Le terme "PWA" est l’acronyme de Progressive Web Application. Il a été inventé en 2015 par le designer « Francis Berryman » et l'ingénieur Google Chrome Alex Russell [3].

Les PWA sont des applications Web qui utilisent une combinaison de technologies et de modèles d'applications Web et natives pour tirer parti des fonctionnalités des applications Web et natives. Les applications Web, par exemple, sont plus détectables que les applications natives par les moteurs de recherche. Visiter un site Web est plus facile et plus rapide que d'installer une application, et on peut facilement partager des applications Web en envoyant un lien par e-mail. D’autre part, les applications natives sont mieux intégrées au système d'exploitation et offrent donc une expérience plus transparente aux utilisateurs. Les utilisateurs préfèrent cliquer sur leurs icônes pour accéder rapidement à leurs applications préférées plutôt que d'y accéder via un navigateur, car les applications natives fonctionnent hors ligne.

Les PWA nous donnent la possibilité de créer des applications Web qui peuvent profiter de ces mêmes avantages.

Les PWA, cependant, fournissent tout cela et plus encore sans perdre aucune des fonctionnalités existantes qui font la grandeur du Web [4].

* + 1. **Les Caractéristique Des PWAs**

Les PWA ne sont pas créées à l'aide d'une seule technologie, comme nous l'avons dit précédemment. Ils représentent une nouvelle approche du développement d'applications Web qui inclut certains modèles, API et autres fonctionnalités. Il n'est pas toujours clair si une application Web est une PWA à première vue. Lorsqu'une application répond à des critères particuliers ou implémente un ensemble de certaines caractéristiques, telles que le travail hors ligne, l'installation, la simplicité de synchronisation et la possibilité de fournir des notifications...etc. elle est considérée comme une PWA.

De plus, il existe des outils pour mesurer le degré de complétude (en pourcentage) d'une application Web, comme Lighthouse [5].

Nous pouvons rendre une application plus progressive en incorporant de nombreux avantages techniques, ce qui se traduit par un score Lighthouse plus élevé. Cependant, ceci n'est qu'une estimation préliminaire, car pour être classée comme PWA, une application Web doit respecter un ensemble de caractéristiques fondamentales [6]. Il devrait être comme suit :

* **Progressive** : Une application Web progressive doit fonctionner sur n'importe quel appareil et se développer continuellement au fil du temps, en utilisant les fonctionnalités disponibles sur l'appareil et le navigateur de l'utilisateur.
* **Découverte** : Parce qu'une PWA est un site Web, elle doit être compatible avec les moteurs de recherche (SEO friendly). Il s'agit d'un avantage significatif par rapport aux applications natives, qui sont toujours à la traîne des pages Web en termes de résultats de recherche.
* **Liens :** Parce qu'une PWA est conçue comme un site Web, elle a autant d'URL que de pages.
* **Responsive**: L'interface utilisateur d'une PWA doit être compatible avec le facteur de forme et la taille de l'écran de l'appareil qui l’utilise (PC, tablette, smartphone). Pour une utilisation tactile.
* **Connectivité indépendante** : C’est notre caractéristique préférée. Une PWA devrait fonctionner dans les zones où la connectivité est faible ou hors ligne (stockage de données sur le smartphone), bien que certaines fonctionnalités puissent être limitées.
* **Installable :** Une PWA peut être installée sur l'écran d'accueil de l'appareil pour accéder rapidement. Elle occupera moins de mémoire sur l'appareil qu'une application native. Par exemple : Le Twitter PWA, a une taille inférieure à 1 Mo, contre plus de 100 Mo pour l'application Native.
* **Actualisée :** Lorsqu'un nouveau contenu est publié et qu'une connexion Internet est disponible, les nouvelles données sont chargées dans l'application en direct. En ce qui concerne les mises à jour, l'application se met à jour une fois qu'elle est chargée.
* **Sécurisée :** Étant donné qu'une PWA offre une expérience utilisateur plus personnelle et que les Services workers peuvent intercepter toutes les requêtes réseau, l'application doit être hébergée sur HTTPS pour éviter les attaques man-in-the-middle.
  + 1. **Comparaison**

Si vous pensez aux applications natives et aux applications Web en termes de capacités et de portée, les applications natives représentent les meilleures capacités, tandis que les applications Web représentent la meilleure portée. Alors, où se situent les Progressive Web Apps ?

Les applications Web progressives (PWA) sont conçues et améliorées avec des API modernes pour offrir des capacités, une fiabilité et une Instalabilité améliorées tout en atteignant n'importe qui, n'importe où, sur n'importe quel appareil avec une seule base de code.

* + - 1. **La différence entre PWA et les sites web responsive**

La principale différence entre un PWA et un site Web réactif est la fonctionnalité, il n'y a pas beaucoup de différence en termes d'interface utilisateur. Les PWA ont des pages Web, qui sont accessibles via une URL, comme n'importe quel autre site Web (réactif ou non). Cependant, les PWAs offrent plus de fonctionnalités d'applications natives que les sites Web réactifs manquent.

Les utilisateurs peuvent personnaliser la PWA en fonction de leurs besoins. Par exemple, ils peuvent activer les notifications push, accéder à la caméra et même ajouter un PWA à l'écran d'accueil de l'appareil, tout comme l'ajout d'une application native [7].

* + - 1. **La différence entre PWA et application native**

L'une des principales différences entre les deux est que les applications natives sont spécifiques à la plate-forme, c'est-à-dire qu'elles peuvent Fonctionner uniquement sur le système d'exploitation pour lequel elles ont été développées. Elles dépendent fortement des développeurs et des outils disponibles dans le système d'exploitation afin que les utilisateurs puissent profiter de leur fonctionnalité.

Les PWA dépendent principalement du navigateur qui y accède. Elles n'ont pas besoin d’être développées pour des plates-formes spécifiques car elles fonctionneront sur n'importe quel appareil moderne avec un navigateur. Elles se comportent comme des applications natives à bien des égards, mais ne contrôlent pas le matériel de l'appareil [7].

Tableau 1 : Apporter le meilleur des deux mondes [8]

|  | **les applications Web** | **les applications native** |
| --- | --- | --- |
| **Les PWAs** | **Responsabilité**  **Accessible par défaut**  **Omniprésent**  **Facile à déployer**  **Facile à mettre à jour**  **Tout le monde peut poster** | **Assistance hors ligne**  **Haute performance**  **Intégration de l'appareil**  **Expérience autonome**  **Icône installée**  **Riche et fiable** |

Lorsque vous pensez aux applications de plate-forme représentent les meilleures fonctionnalités et les applications Web représentent la meilleure portée. Les applications Web progressives sont situées à l'intersection de la fonctionnalité des applications de plate-forme et de la portée des applications Web. L'application Web progressive comprend des fonctionnalités des deux mondes [8] .

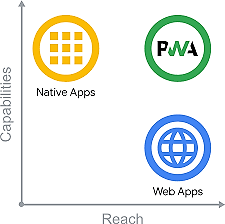


Figure 1 : Capacités par rapport à la portée entre des application native, des applications Web et des applications Web progressives [9]

* + 1. **Aperçu des forces et faiblesses**

Tableau 2 : Avantages et inconvénients des progressives web apps [10]

| **Les Avantages des PWAs** | **Les Limites des PWAs** |
| --- | --- |
| Les PWA fonctionnent généralement **plus rapidement** que les sites Web.. | Il n'y a **aucun avantage pour le trafic de l'App Store :** Répertorié comme un avantage. Il s'agit d'une épée à double tranchant, car de nombreux utilisateurs ne recherchent que des applications dans l'App Store. |
| Ils sont accessibles **hors ligne**. | Ils **consomment beaucoup de batterie :** En raison du code beaucoup plus complexe, les PWA consomment plus d'énergie que les applications natives. Par conséquent, le dispositif est très difficile à interpréter. |
| Ils sont **réactifs** et fonctionnent donc avec tous les systèmes d'exploitation et formats d'appareils électroniques.. | Ils n'ont **pas accès à toutes les fonctionnalités du téléphone** **:** Cela peut être considéré comme la principale faiblesse des PWA par rapport aux applications natives. |
| Les PWA **fonctionnent comme des applications natives :** L'icône apparaîtra sur votre téléphone comme n'importe quelle autre application, vous pouvez envoyer des notifications push et accéder à certaines fonctionnalités du téléphone | Certaines fonctionnalités ne sont pas **compatibles avec iOS :** La plupart des utilisateurs utilisent des appareils avec le système d'exploitation Android, mais les PWA ne sont pas optimisés pour la compatibilité iOS, ce qui est une faiblesse majeure. |
| **Les PWA sont détectées par les moteurs de recherche** et se distinguent par les algorithmes de Google. |  |
| Vous pouvez envoyer des notifications pour **attirer l'attention** de l'utilisateur. |  |
| **Aucune installation depuis l'App Store n'est requise**. C'est un plus pour les utilisateurs et les propriétaires. Le premier est un avantage car vous n'êtes pas obligé de passer par l'app store, et le second est que vous n'êtes pas obligé de suivre les règles de l'app store. |  |
| **Il prend moins de place sur la mémoire** de votre appareil car vous n'avez pas à le télécharger. |  |

* 1. **La vente en ligne**

**Investissement des entreprises de vente en ligne dans les PWAs en 2021**

Dans une enquête d'octobre 2020 auprès des décideurs du commerce électronique en Europe et en Amérique du Nord, 9 % des personnes interrogées ont déclaré que leurs entreprises prévoyaient d'investir dans des applications Web progressives (PWA) en 2021. De plus, 8 % des entreprises l'ont déjà fait et 28 % sont en train d’étudier la possibilité d’introduire les PWA en 2021. Les PWA permettent aux consommateurs d'acheter des produits sur plusieurs appareils, connexions et même systèmes d'exploitation [11]. Par ce fait, elles touchent plus de consommateurs et de clients que les applications web classiques ou les applications natives.

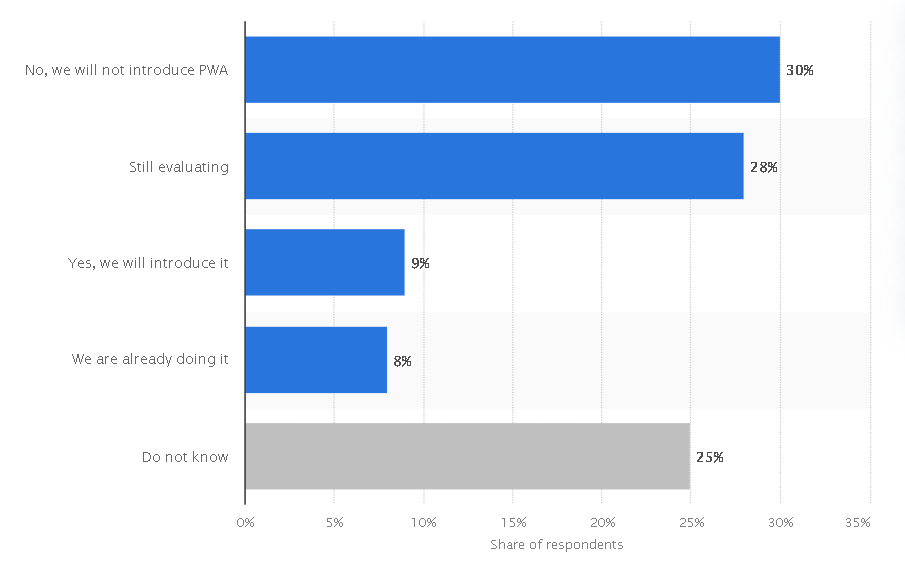


Figure 2 : Part des entreprises de vente en ligne prévoyant d'investir dans des applications Web progressives (PWA) en Amérique du Nord et en Europe en 2021 [11]

**Conclusion**

Ce chapitre permis d’introduire la notion d’applications Web progressives ou PWA. Nous avons commencé par une introduction aux applications natives et aux sites Web réactifs. Après cela, nous avons défini le concept de PWA avec un bref historique, différentes branches, avantages et inconvénients, et quelques domaines d'application. A la fin de ce chapitre, nous avons souligné que les entreprises de e-commerce sont fortement intéressées par les PWAs, afin d’acceder au maximum de consommateurs.

Dans le chapitre suivant, nous présenterons la conception et la modélisation de notre système.

|  | | **Chapitre** | | |
| --- | --- | --- | --- | --- |
|  |  | | **2** |  |
| 1. **Modélisation** |
|  | | | | |

Dans le processus de développement d'un système informatique, il doit y avoir des phases et des étapes clairement définies pour organiser le travail et obtenir un produit de qualité qui répond aux objectifs dans les plus brefs délais. L'une des étapes les plus importantes est la conception et la modélisation du système~~,~~ Nous présentons dans ce chapitre la modélisation de notre système à travers différents diagrammes UML. Les PWAs utilisent les technologies web de base : HTML, CSS et JavaScript. Elles se distinguent par l’intégration de plusieurs caractéristiques. Les applications Web progressives ne sont pas une technologie ou un concept unique, mais plutôt une variété de ceux-ci utilisés ensemble pour fournir une expérience utilisateur améliorée.

Le chapitre est organisé comme suit : la modélisation UML avec les diagrammes de cas d'utilisation, les scénarios possibles et les diagrammes de classes. Ensuit dans la deuxième partie nous présentons différents aspects des PWAs. Concernant Structure d'une Progressive web app, Manifest et Service Worker. Enfin, nous terminons ce chapitre par une conclusion.

* 1. **Modélisation UML du système proposé**

Le langage de modélisation unifié (UML) est un langage de modélisation à usage général. L'objectif principal d'UML est de définir une manière standard de visualiser la manière dont les systèmes sont conçus. Il est très similaire aux plans utilisés dans d'autres domaines de l'ingénierie.

UML n'est pas un langage de programmation, mais un langage de visualisation. Nous utilisons des diagrammes UML pour représenter le comportement et la structure du système. UML aide les ingénieurs logiciels, les hommes d'affaires et les architectes système à modéliser, concevoir et analyser. L'Object Management Group (OMG) a adopté le langage de modélisation unifié comme norme en 1997. Il est géré par OMG depuis lors. L'Organisation internationale de normalisation (ISO) a publié UML en tant que norme approuvée en 2005. UML a été révisé et révisé régulièrement au fil des ans [12].

* + 1. **Elaboration du diagramme de cas d’utilisation**

Un diagramme de cas d'utilisation décrit un système du point de vue de l'utilisateur final et représente les tâches que les utilisateurs et le système peuvent effectuer. Il se compose d'acteurs (personnes et systèmes externes pouvant interagir avec l'intérieur du système), de cas d'utilisation et des relations entre les acteurs et les cas d'usage [13].

* + - 1. **Identification des acteurs**

Les diagrammes de cas d'utilisation utilisent le concept d'acteurs, un acteur est un groupe d'utilisateurs dans le système qui partagent la même fonctionnalité, notre système proposé a trois (03) types d'acteurs comme indiqué.

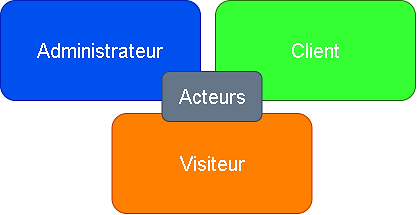


Figure 3 : Acteurs du système

* + - 1. **Diagramme de cas d’utilisation**

**Administrateur**

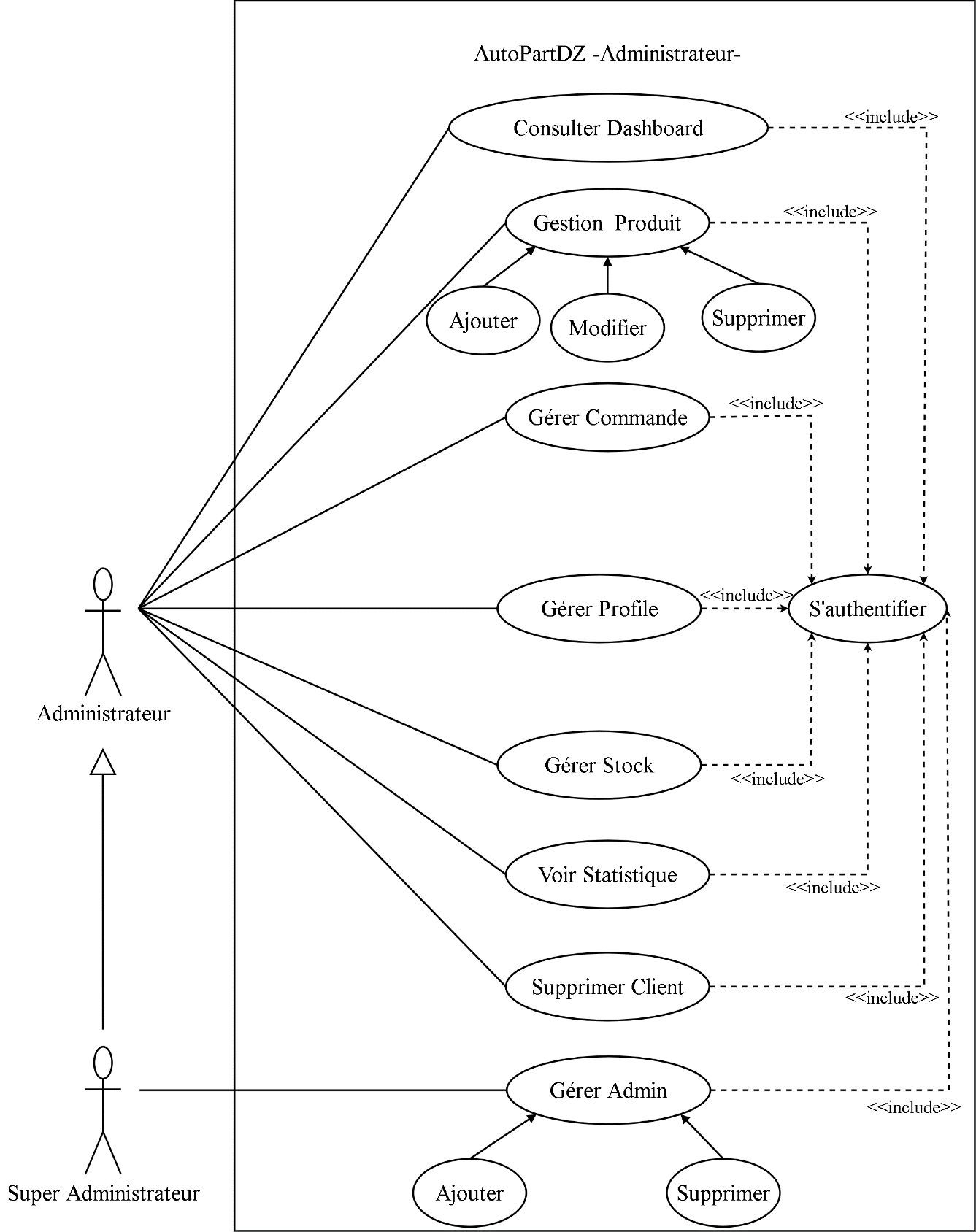


Figure 4 : Diagramme de cas d’utilisation Administrateur

**Client / Visiteur**

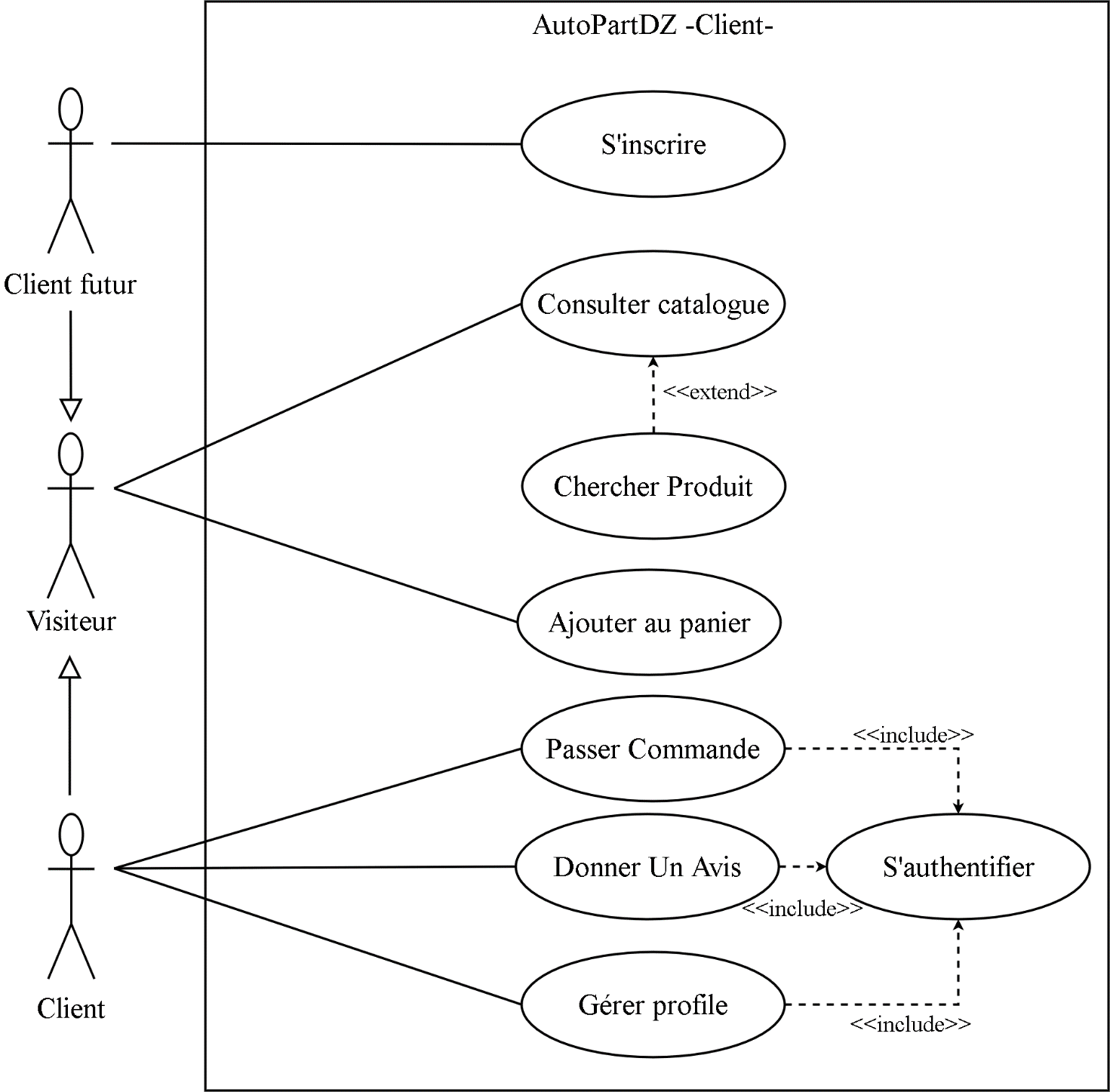


Figure 5 : Diagramme de cas d’utilisation Visiteur / Client

* + 1. **Description des scénarios des cas d’utilisation**

Les scénarios de cas d'utilisation expriment le flux de fonctionnalités et la séquence temporelle de ses opérations. Pour illustrer cela, nous présentons les cas d'utilisation les plus importants de notre système dans cette section et fournissons des tableaux de description pour chaque cas d'utilisation. Les cas décrits dans cette section sont :

* + - 1. **L’authentification**

Tableau 3 : Fiche descriptive du cas d'utilisation « S’authentifier »

| Cas d’utilisation | S’authentifier |
| --- | --- |
| Objectif | Permettre d'identifier un acteur et lui donne l'accès à son  compte et à ses différentes tâches. |
| Acteur | Administrateur, Client. |
| Préconditions | L’acteur doit posséder un compte dans le système |
| Postconditions | Accéder à la plateforme et aux différentes fonctionnalités. |
| Scenario nominal | 1. L’acteur accède au système.  2. Le système affiche la page d’authentification.  3. L’acteur entre l'email et le mot de passe.  4. L’acteur clique sur le bouton « Sign In ».  5. Le système vérifie les informations de connexion.  6. Le système affiche la session. |
| Scenario alternatif | 5.1. Les informations remplis sont invalides :  1. Le système affiche un message d’erreur.  2. Le scenario reprend du point 3. |
| Scenario exceptionnel | / |

* + - 1. **Passer commande**

Tableau 4 : Fiche descriptive du cas d'utilisation « Commander »

| Cas d’utilisation | Commander |
| --- | --- |
| Objectif | Le client passe une commande . |
| Acteur | Client. |
| Préconditions | Le client s’authentifier dans le system. |
| Postconditions | / |
| Scenario nominal | 1. Le système demande au client de saisir son identifiant et son mot de passe.  2. Le client saisit son identifiant et son mot de passe et valide.  3.Le system affiche le catalogue des produits.  4. Le client choisit les produits.  5.Le system ajouter les produits dans le panier.  6.Le client click sur  (passer la commande).  7. Le système demande au client de saisir les informations de livraison.  8. Le client remplit le formulaire de livraison et valide.  9. Le système demande au client de choisir un mode de paiement.  10. Le client choisit un mode de paiement et valide.  11. Le système affiche un récapitulatif de la commande, indique le montant total de la livraison.  12. Le client valide la commande.  13. Le système informe le client que la transaction s'est effectuée correctement. |
| Scenario alternatif | 6.1. Le client clicque sur (panier) :  1.Le system affiche les produits dans le panier.  2. Le scénario reprend à l'étape 6.  11.1. Nouvelle adresse de livraison :  1. Le client saisit une nouvelle adresse de livraison et valide  2. Le scénario reprend à l'étape 7. |
| Scenario exceptionnel | / |

* + 1. **Elaboration du diagramme de classe**

Un diagramme de classes, composé d'un ensemble de propriétés et de fonctions, définit un système d'un point de vue de conception statique et fournit un plan montrant comment construire le système [13]. La figure 6 illustre le diagramme de classes.

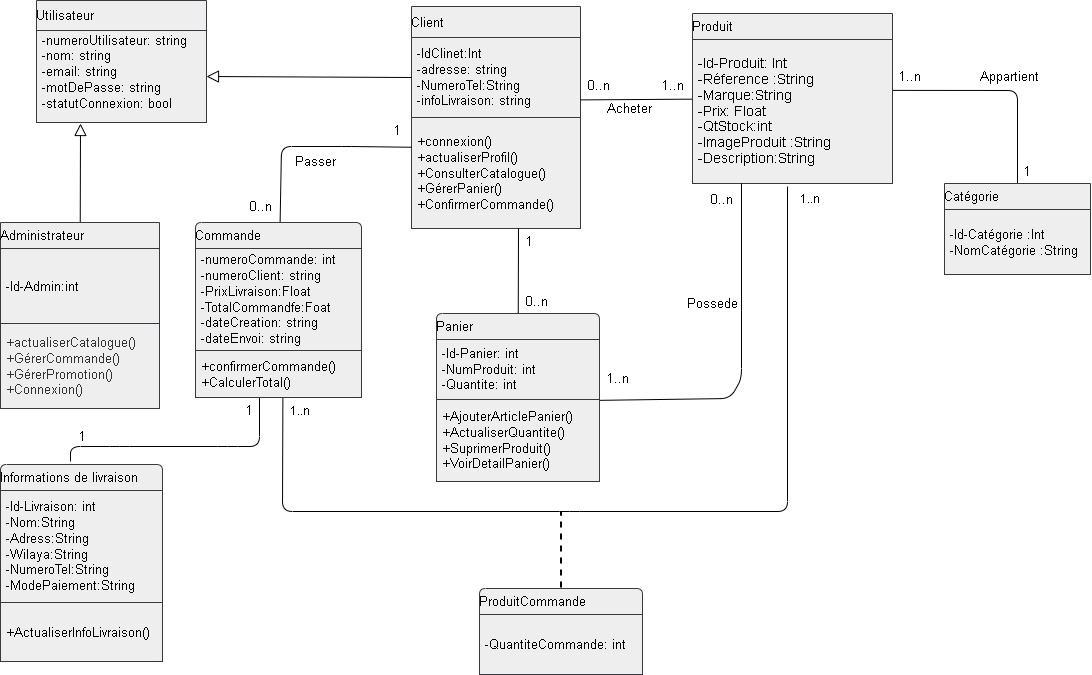


Figure 6 : Diagramme de Classe

* 1. **Structure d'une Progressive web app**

Maintenant que nous avons pris connaissance des principes théoriques sur lesquelles sont bâties les PWAs, penchons-nous sur la structure recommandée d'une vraie application.

Le but d'une PWA est de vous offrir la même expérience utilisateur qu'une application mobile native. PWA ne sont pas des sites Web au sens traditionnel, mais ce sont néanmoins des sites Web. Les technologies utilisées pour construire des PWA sont les suivantes [2] :

* HTML5, CSS3 et JavaScript sont utilisés pour créer l'interface utilisateur qui est la partie statique d'une PWA. Cette interface utilisateur, connue sous le nom de shell d'application, est mise en cache sur le mobile de l'utilisateur. Périphérique afin qu'il puisse être chargé sans aucun retard pouvant survenir en raison du mauvais connexion réseau.
* Le côté client utilise des appels de service Web pour le chargement et le rendu du contenu, comme c'est le cas dans les applications natives. Cela permet d'éviter le scénario classique dans lequel, à chaque demande, l'interface utilisateur et les données sont rassemblées côté serveur et envoyées au client en une seule fois.
* Un service worker est l'une des technologies les plus importantes d'une PWA. Le service worker est un code JavaScript exécuté en arrière-plan par le navigateur. Ce code ne dépend pas d'une page Web qui donne accès à des fonctionnalités qui ne nécessitent pas de page Web ou d'interaction avec un utilisateur.

Des frameworks tels que Angular et React peuvent être utilisés pour développer des PWA. Ces frameworks ont des service workers intégrés qui rendent inutile leur reconstruction pour chaque PWA.

La figure 7 montre l'architecture d'une PWA.

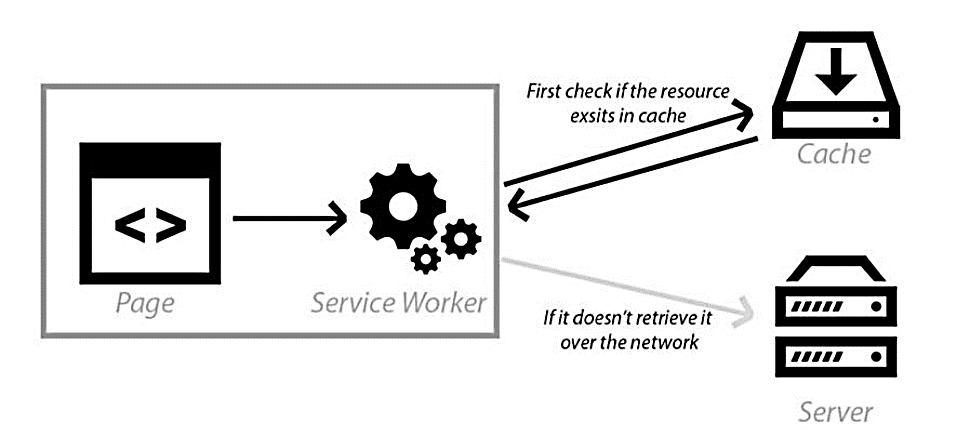


Figure 7 : Application Web progressive [2]

* 1. **Les technologies de PWA**

Les PWAs utilisent les technologies web de base : HTML, CSS et JavaScript. Elles se distinguent par l’intégration de plusieurs caractéristiques.

Les applications Web progressives ne sont pas une technologie ou un concept unique, mais plutôt une variété de ceux-ci utilisés ensemble pour fournir une expérience utilisateur améliorée. Dans ce chapitre nous présentons différents aspects des PWAs. Concernant Manifest et Service Worker.

* + 1. **Manifeste de l'application Web**

**Définition**

Le manifeste de l'application Web est un fichier JSON avec l’extension .Webmanifest qui permet aux développeurs de consolider les informations relatives aux PWA. Seul ce fichier, s'il est lié, permet au navigateur de faire la différence entre un site web traditionnel et une Progressive Web App. Ce fichier vous permet de spécifier des métadonnées pour la PWA, telles que l'icône, le nom, la catégorie et la description, ainsi que certains comportements spécifiques, tels que la façon dont la PWA est affichée dans l'environnement hôte, comment gérer un mode hors ligne et un réseau avec des coupures, traiter ou non les notifications etc. [14].

* + - 1. **Déployer un manifeste**

Les manifestes des applications Web sont déployés dans vos pages HTML en utilisant une balise lien (link) dans l'entête (head) de votre document : [15]

| HTML | |  |
| --- | --- | --- |
| 1 | <link rel="manifest" href="/manifest.webmanifest"> | |

* + - 1. **Exemple de manifeste**

Au minimum, un fichier manifeste doit contenir les informations suivantes : [16]

| JSON | |  |
| --- | --- | --- |
| 1  2  3  4  5 | {  "name": " Mon exemple de PWA",  "lang": "fr -FR",  "start\_url": "/"  } | |

* + 1. **Service Workers**

**Définition**

La plupart des caractéristiques et concepts qui séparent une PWA d'une application Web standard est disponible via Service Workers.

Un Service Worker est un script JavaScript qui effectue des opérations en arrière-plan séparé du reste du site Web car il ne peut pas interagir avec le DOM.

Il agit comme un proxy réseau au niveau de l'application, permettant aux développeurs d'intercepter entièrement requêtes réseau et contrôle tels que la mise en cache des données et des actifs pour hors ligne disponibilité, synchronisation en arrière-plan (par exemple, récupération de données pendant que l'application est dans l’arrière-plan) et l'enregistrement pour les notifications push [17].

* + - 1. **Événements de Service Worker**

Les techniciens de service sont axés sur les événements. Le processus d'installation et d'activation déclenche les événements d'installation et d'activation correspondants auxquels l'agent de service peut répondre. Il existe également des événements de message, où les techniciens de maintenance peuvent recevoir des informations d'autres scripts, et des événements fonctionnels tels que l'obtention, l'envoi et la synchronisation.

Pour vérifier les techniciens de service, accédez à la section des techniciens de service dans les outils de développement de votre navigateur. Le processus est différent dans chaque navigateur prenant en charge les service workers [18].

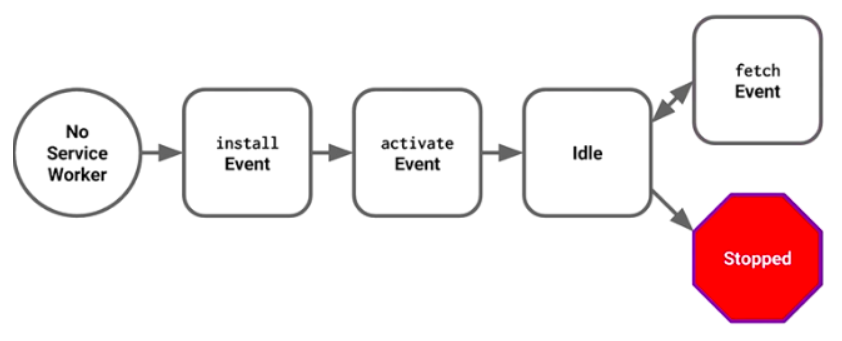


Figure 8 : Un cycle de vie d’un Service Worker [19]

**Conclusion**

Dans ce chapitre, nous modélisons notre système proposé nommé "AutoPartDZ". Nous enrichissons ensuite la modélisation de notre système en proposant une modélisation UML, pour laquelle les schémas suivants sont développés :

* Diagramme de cas d'utilisation
* Scénarios pour les cas d'utilisation les plus importants
* et Diagrammes de classes

Ensuite, nous avons présenté la structure d'une Progressive web app et les technologies des PWAs telles que Manifest et et Service Worker.

Nous allons commencer à l'étape suivante qui consiste en l’implémentation. Elle sera abordée dans le prochain chapitre ainsi que la validation et l’expérimentation.

|  | | **Chapitre** | | |
| --- | --- | --- | --- | --- |
|  |  | | **3** |  |
| 1. **Implémentation** |
|  | | | | |

Après avoir décrit la partie du technologies de PWA, ce chapitre sera consacré à la présentation de l’implémentation et l’expérimentation de ce dernier. L’objectif de ce chapitre sera donc de présenter l'application Web que nous avons mis en place.

L’organisation retenue pour ce chapitre est la suivante : D'abord, nous décrivons les outils de développement et les langages de programmation utilisés. Ensuite, nous présentons le mécanisme d'implémentation et le principe de fonctionnement de chaque fonctionnalité proposée par notre application. Enfin, nous illustrons la phase de validation et d'expérimentation .

* 1. **Implémentation d’application web ‘AutoPartDZ’**

**Présentation du système proposé**

L’application web progressive que nous proposons est nommée « AutoPartDZ», cette nomination a été inspirée des deux mots anglais « Auto » qui signifie les automobiles et « Parts » qui signifie les Pièces. ‘AutoPartDZ’ est donc une application web qui permet aux utilisateurs d’acheter des pièces automobiles. L'objectif est de générer de nouveaux E-Commerce en utilisant le PWA, Nous estimons que le travail réalisé dans le cadre de ce projet de fin d’étude pourra être une alternative aux méthodes classiques de E-Commerce. Grâce à ses caractéristiques d’accessibilité, de disponibilités, de rapidité et d’efficacité, Nous espérons que l’outil ‘AutoPartDZ’ sera également d’une utilité pour aider à la fois les fournisseurs et les clients en la rendant rapide et facile pour tout le monde.

* 1. **Description de l’implémentation de l’application ‘AutoPartDZ’**

Cette section est dédiée à l'implémentation et à l'introduction des fonctionnalités fournies par l'outil ‘AutoPartDZ’, chaque fonctionnalité sera expliquée en détail avec des captures d'écran pour mieux illustrer cette fonctionnalité et faciliter sa compréhension.

Nous avons conçu l’outil ‘AutoPartDZ’, pour qu’il fonctionne en deux environnement desktop et mobile, L’application web progressive représente le coté client, c’est elle qui est chargée d’accompagner l’utilisateur depuis la création de son compte jusqu’à la phase d’effectuer une commande. Tandis que le côté administrateur représente le panneau d’administration d'application " AutoPartDZ ", c’est dans la plateforme que les différentes ressources sont gérées par les administrateurs.

* + 1. **Points d'entrée de l'application**

Un utilisateur qui souhaite exploiter l’outil que nous proposons dispose de trois points d'entrée possibles :

1. **Interface Visiteur**

Le système ‘AutoPartDZ’ permet à un utilisateur même s'il ne possède pas de compte d'accéder à un nombre limité de fonctionnalités. Il peut notamment consulter la liste des pièces automobiles dans le système et au panier pour avoir une idée de la commande et du prix total. Il peut également effectuer une recherche pour trouver une pièce automobile spécifique. Il a également la possibilité de sélectionner des pièces automobiles et de les ajouter à ce panier parmi celles proposées par l'application "AutoPartDZ" et de créer une liste d'achat. Le visiteur ne pourra pas utiliser les options permettant de passer commande et sera donc limité à utiliser l'application uniquement pour consulter des pièces. Il ne peut pas non plus bénéficier de la communication avec le vendeur ou de l'expression d'un avis sur les pièces automobiles. Une fois qu’il terminé, sa commande s’affiche dans le panier, Il peut revenir à l’écran d’accueil ou faire l'authentification pour suivre les étapes suivantes. L’interface du visiteur proposée dans l’application ‘AutoPartDZ’ se présente l’illustre la figure 9.

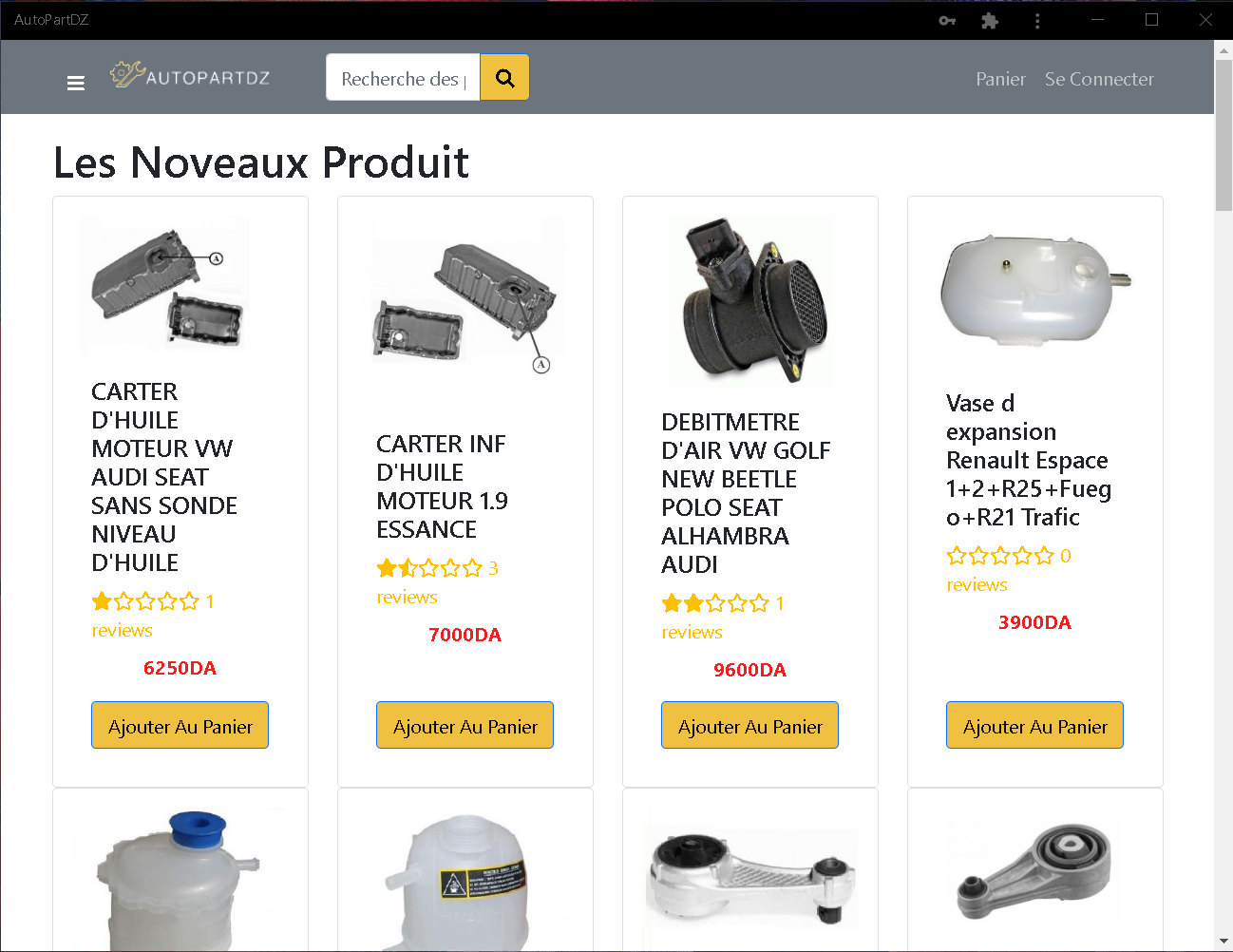


Figure 9 : Interface Desktop du visiteur dans l’application ‘AutoPartDZ’

1. **Création d’un nouveau compte**

" AutoPartDZ " permet à un visiteur de créer un nouveau compte de type client. Après que le visiteur ait saisi les informations nécessaires et appuyé sur le bouton « Register », le système vérifie que certaines informations obligatoires telles que le nom, l'email et le mot de passe ne sont pas vides, il vérifie également que le format de l'email est correct. En cas de manque d'informations, ou d’informations incorrectes, un message d'erreur s'affiche, sinon le compte sera créé et rajouté au système. Quand il s’agit d’un compte de type client. La figure 12 montre l’interface Desktop pour la création d’un nouveau compte, et la figure 13 illustre la même fonctionnalité pour l’interface mobile.

"AutoPartDZ" permet aux visiteurs de créer un nouveau compte de type client. Une fois que le visiteur a saisi les informations nécessaires et appuyé sur le bouton

« Registre », le système vérifie que certaines informations requises telles que le nom, l'e-mail et le mot de passe sont non vides et que l'e-mail est correctement formaté. Si les informations sont manquantes ou incorrectes, un message d'erreur s'affichera, sinon un compte sera créé et ajouté au système. Lorsqu'il s'agit d'un compte de type client. La figure 11 montre l'interface de Desktop pour créer un nouveau compte, et la figure 11 montre la même fonctionnalité pour l'interface mobile.

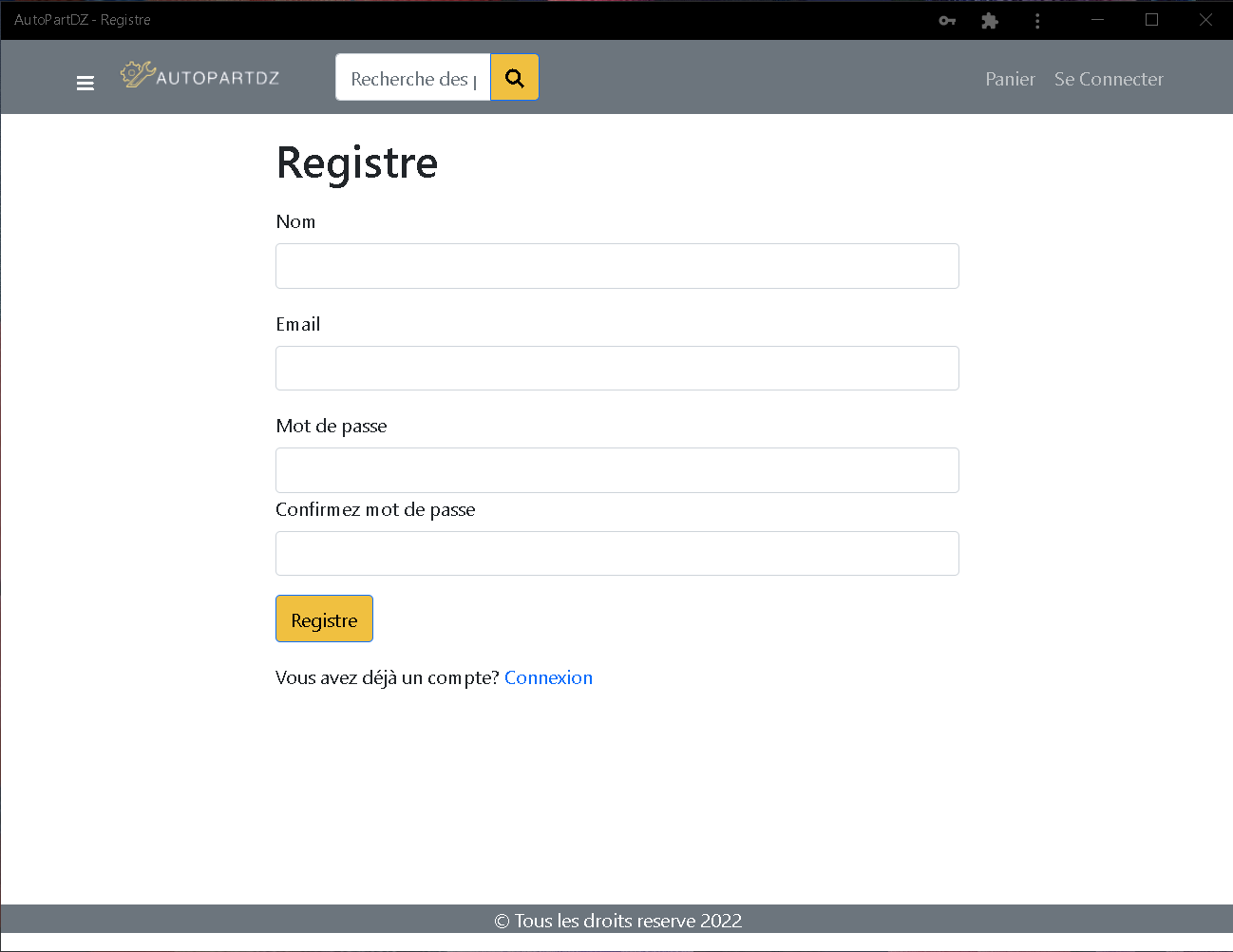


Figure 10 : Interface Desktop pour la création de nouveau compte Client



Figure 11 : Interface Mobile pour la création de nouveau compte Client

1. **Authentification**

Cette fonctionnalité est destinée aux utilisateurs ayant déjà un compte sur la plateforme " AutoPartDZ ". Une fois que l'utilisateur a saisi l'email et le mot de passe et appuyé sur « Se Connecter », le système procède à une vérification du format de l'e-mail, et récupère les données du compte, dans le cas où le compte existe. Sinon, si le compte n'existe pas ou que le mot de passe est incorrect, il affichera un message d'erreur. Les figures 12 et 13 représentent respectivement les interfaces d’authentification desktop et mobile de l’application " AutoPartDZ ".

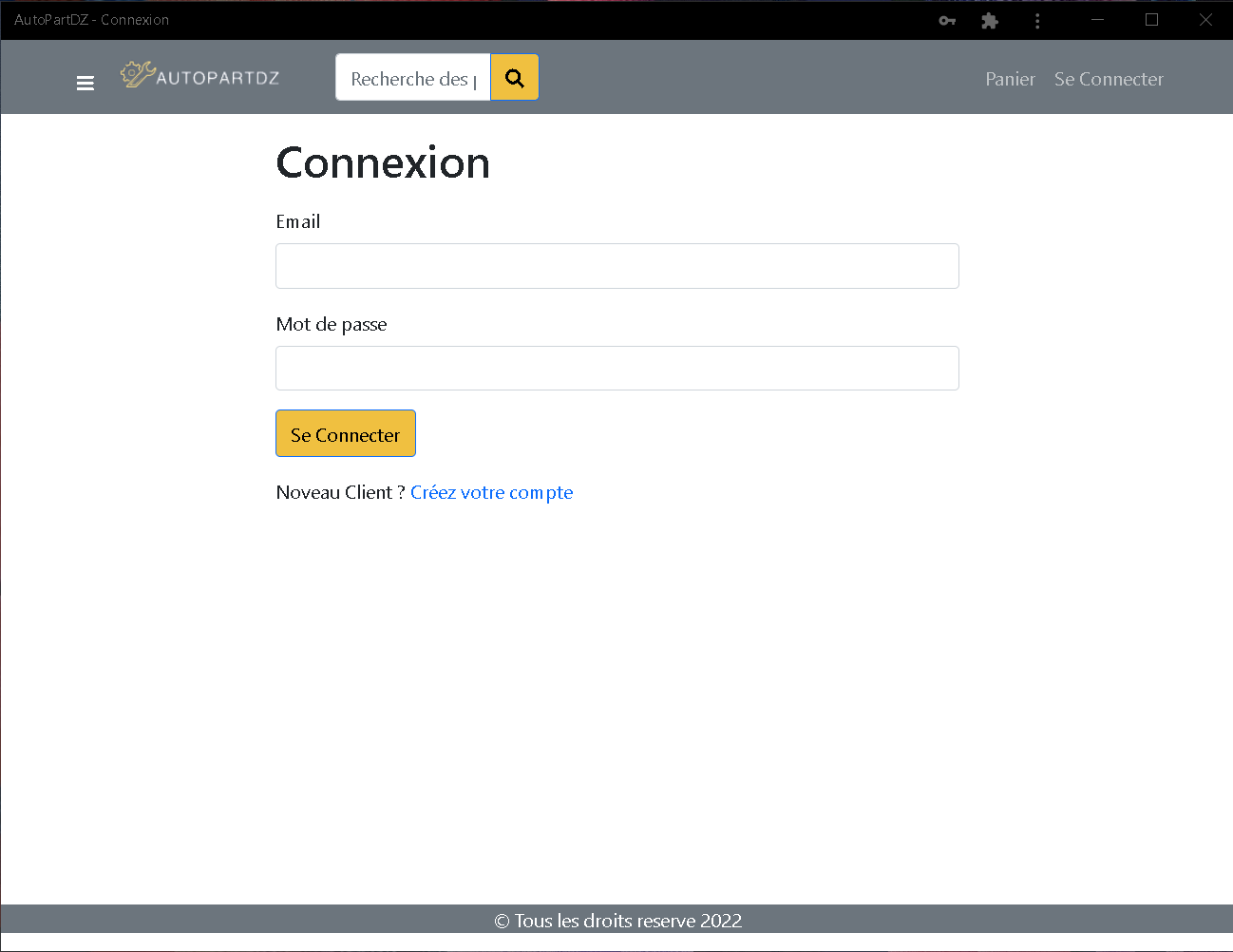


Figure 12 : Interface d’authentification Desktop de l’application ‘ AutoPartDZ ‘



Figure 13 : Interface d’authentification Mobile de l’application ‘ AutoPartDZ ‘

* + 1. **Types de comptes**

Une fois l'utilisateur authentifié, et selon le type de son compte le système affichera l'interface adéquate. Les deux types de comptes existants dans l’outil ‘AutoPartDZ’ sont :

1. **Compte Client**

Un Client peut effectuer toutes les fonctionnalités d'un visiteur, D’autre part Lorsqu'un client passe une commande, sa commande sera stockée dans l'espace prévu «Historique des commandes » dans son compte , afin que lorsqu'il le souhaite, il puisse revenir sur l'historique des commandes qu'il a faites.

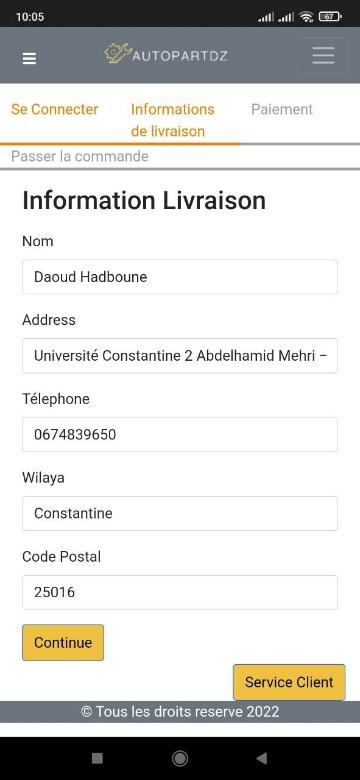
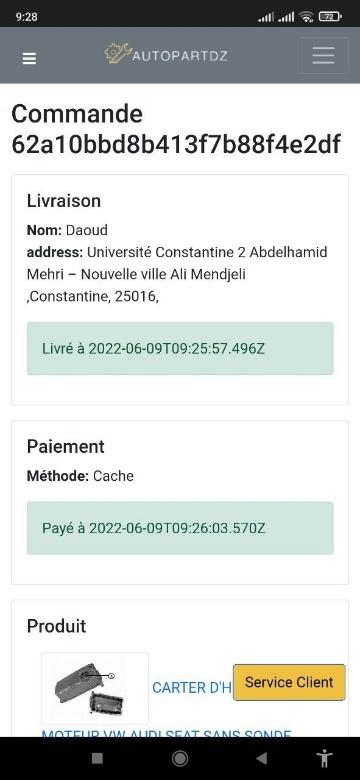
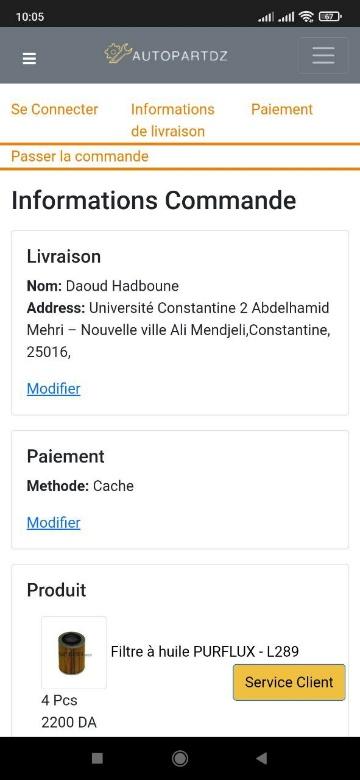


Figure 14: Les Interfaces de la fonctionnalité passer la commande

Le client peut également accéder aux détails pour voir l'état de sa commande, ce dernier peut également modifier son profil pour changer son nom ou son e-mail. De plus, il y a un avantage pour le client dans Autopartdz, c'est que le client peut se renseigner sur un produit, en envoyant des messages instantanés à l'administrateur, Dans Autopartdz, le client peut évaluer les produits en cliquant sur l'une des options « de mauvais à excellent », et il peut également écrire son avis sur ce produit. La figure 16 donne un aperçu de l’interface proposée par l’outil ‘AutoPartDZ’ à un Client. La figure 15 illustre quelques interfaces de compte Client.

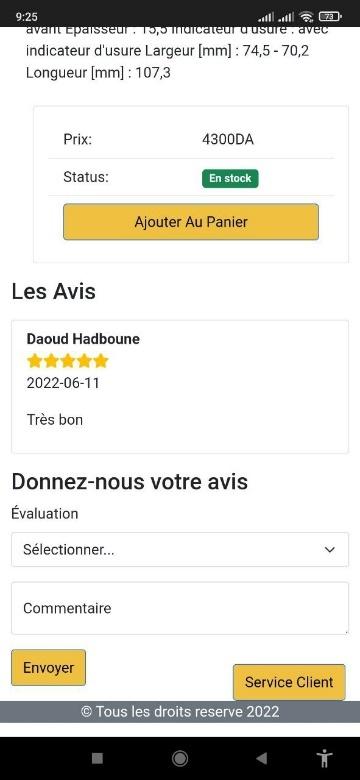
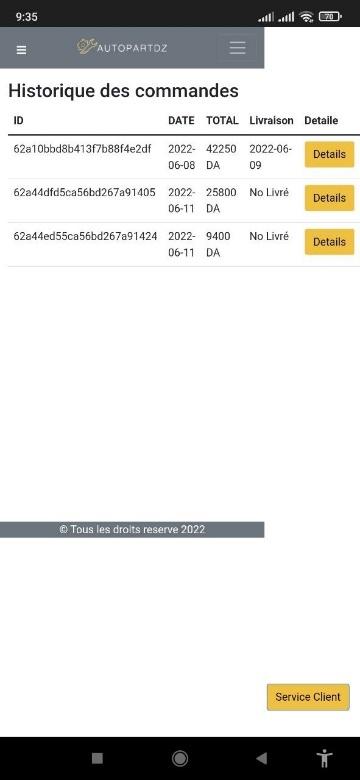


Figure 15: Les Interfaces de Client proposée par l’outil ‘AutoPartDZ’

1. **Compte Administrateur / Super Administrateur**

Le compte administrateur permet de gérer tout type de ressources dans l’outil ‘AutoPartDZ’. L’administrateur a accès à un Dashboard dans lequel sont présentées des statistiques générales sur le système tel que « nombre des commandes , montant total des commande , nombres d’utilisateurs , courbe des ventes ..etc » .

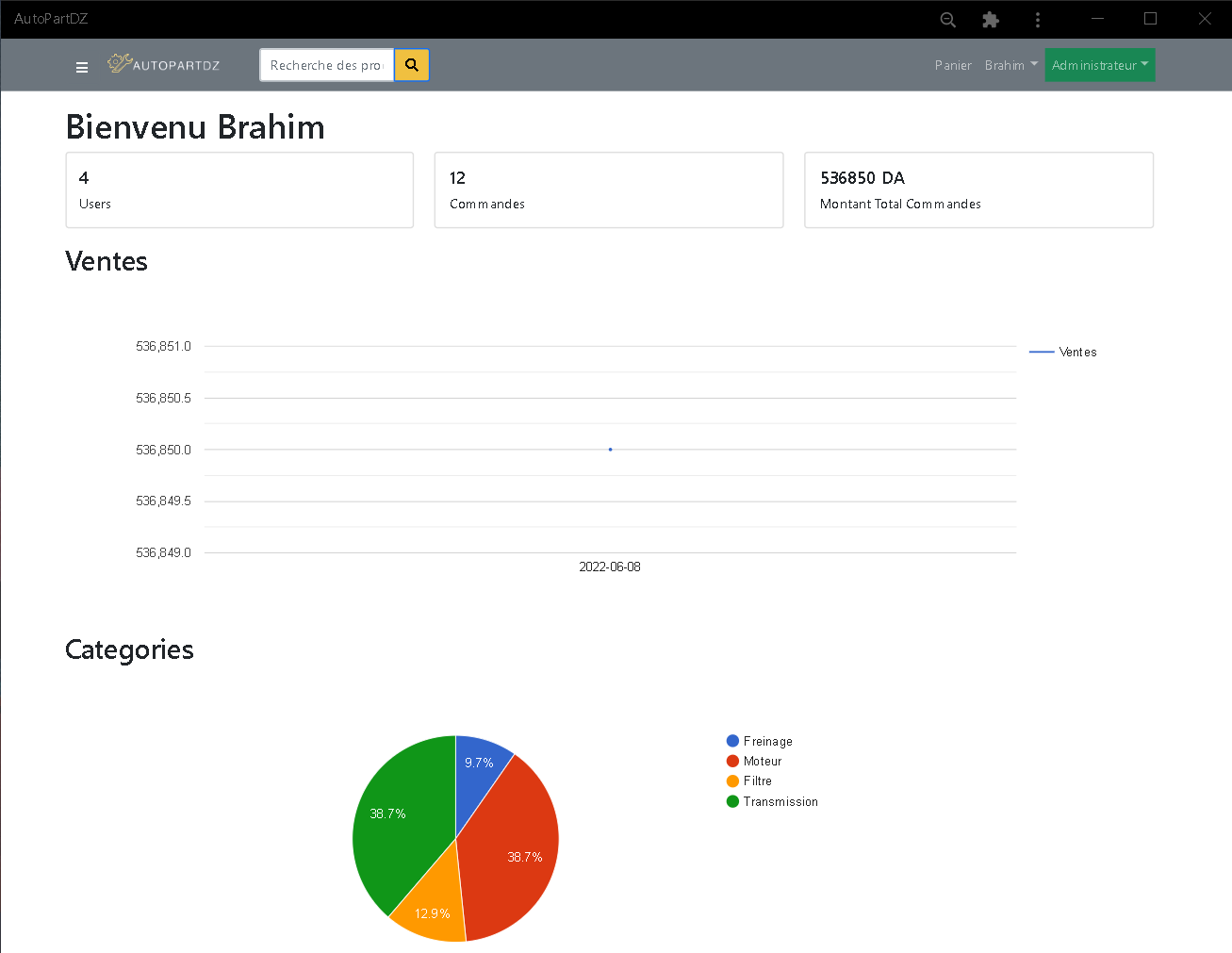


Figure 16 : Dashboard d’un administrateur

L'administrateur peut ajouter, modifier et supprimer des produits, comme il peut aussi consulter la liste des Clients dans laquelle il peut supprimer, modifier les informations de Client. Ce dernier peut accéder à la liste des commandes pour faire le traitement de ces commandes « livrer commande ,payer commande » et aussi supprimer commande. L'administrateur peut accéder à l'interface service client pour lire et répondre aux messages des clients.

Le Super administrateur a une propriété sur les administrateurs qui est la possibilité de nommer des nouveaux administrateurs parmi les clients, Le premier administrateur a une propriété supplémentaire sur les autres administrateurs qui est la possibilité de les supprimer.

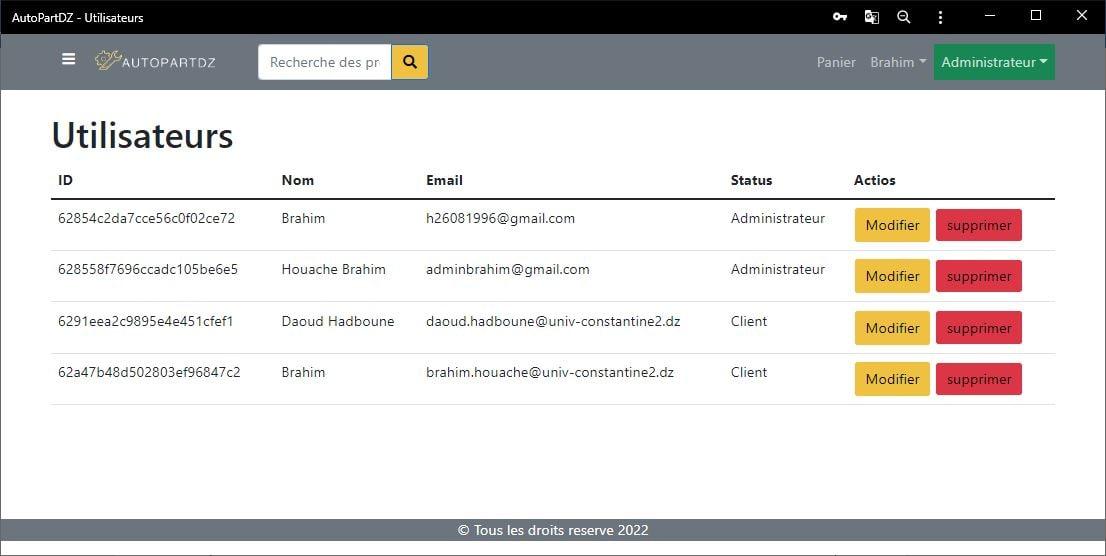


Figure 17 : La liste des utilisateurs (Administrateur + Client)

* **Ajouter / Modifier/supprimer Produit**

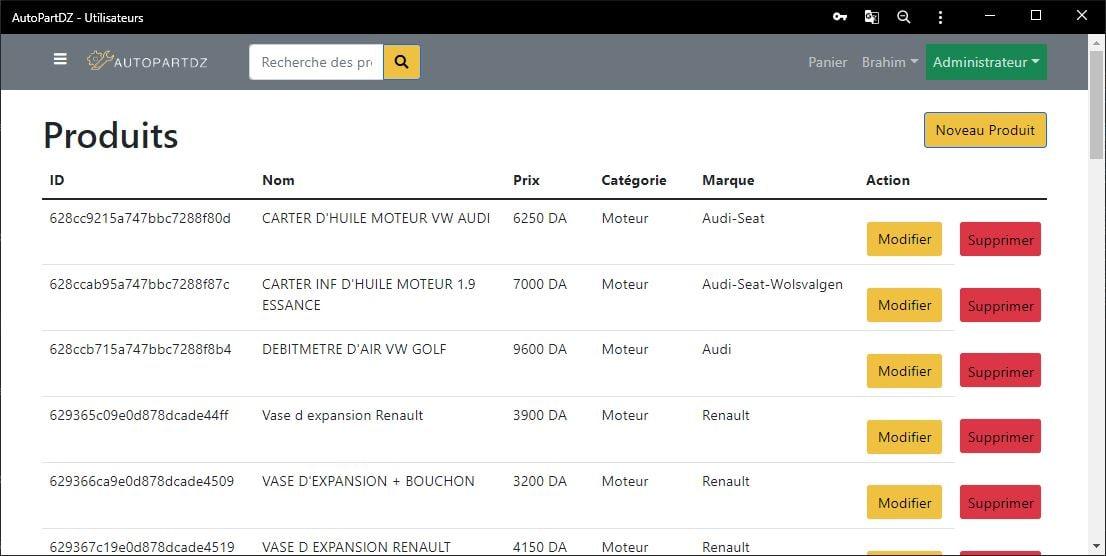
Pour ajouter ou modifier un produit l’administrateur accéde à la liste des produits et choisit ajouter ou modifier un produit , puis le formulaire d'information apparaît et l'administrateur entre les informations du produit (nom du produit, numéro de série, prix,description, quantité et images du produit) il peut également choisir plusieurs images de produits , après, il appuie sur confirmer les informations. Et pour supprimer un produit, l'administrateur doit appuyer sur le bouton supprimer et ainsi le produit sera supprimé. Lorsque le produit est supprimé, il ne sera pas retiré des commandes passées avant. La figure 19 présente l’interface d'ajout ou de modification d’un produit.

Figure 18 : Interface d'ajout ou de modification d’un produit

* **Traitement Commande**

Le traitement des commandes est l'une des tâches de l’administrateur, où il peut aller à la liste des commandes pour voir les commandes qu'il a, et lorsqu'il choisit l'une des commandes de la liste, les détails de la commande lui apparaissent, il peut appuyer sur “livrer la commande”, le statut de la livraison passera de Non-livré à Livré avec heure et date de livraison. D'autre part, le client recevra un email contenant le détail de la commande et le montant total de la commande en plus du prix de livraison et l'informemant également que sa commande lui a été envoyée et qu'il la recevra dans les plus brefs délais. En raison de la difficulté d'ajouter le paiement électronique à l'application en Algérie, cette fonctionnalité n'est pas disponible dans "AutoPartDZ", où il n'y a qu'un seul mode de paiement, qui est en espèces. Afin que le statut de paiement ne reste pas toujours impayé, l'administrateur peut appuyer sur le bouton de commande payée pour modifier le statut de paiement pour lui et le client après que la commande ait été payée par le client.

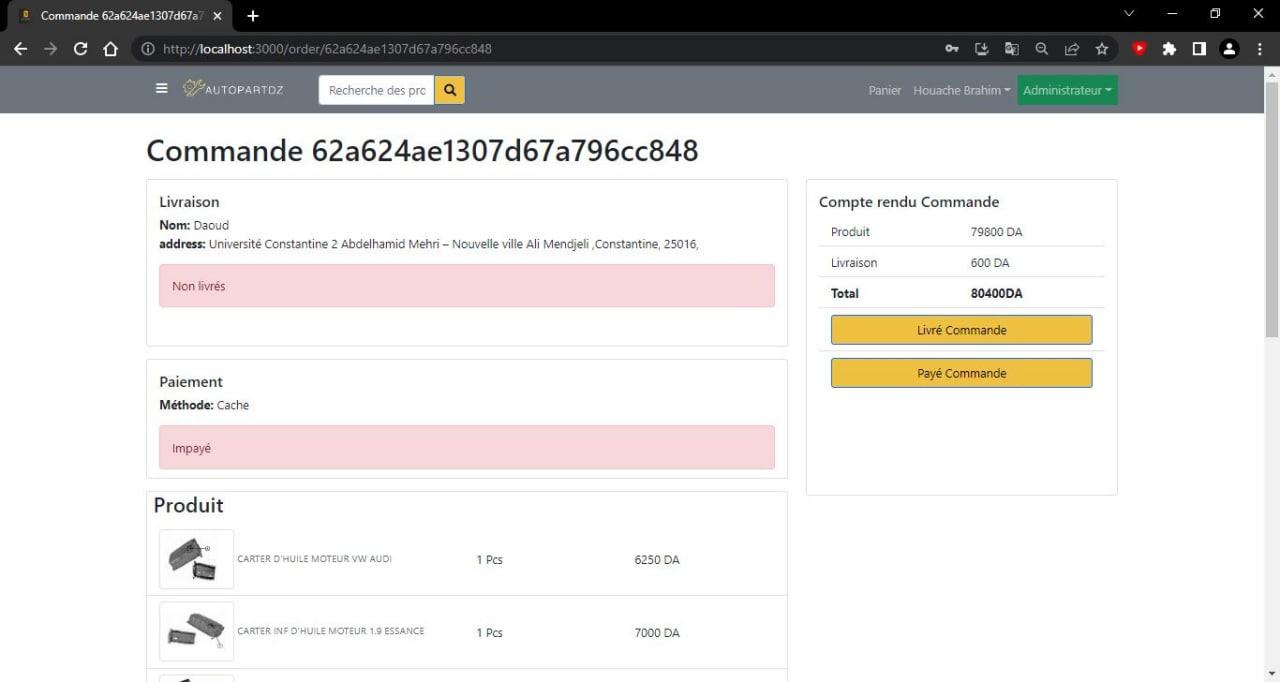


Figure 19 : Étapes de traitement d’une commande

* 1. **Lighthouse (Outil de test PWA)**

Lighthouse est un outil open source automatisé permettant d'améliorer les performances, la qualité et l'exactitude de vos applications Web.

Lors de l'audit d'une page, Lighthouse exécute une série de tests sur la page, puis génère un rapport sur les performances de la page. À partir de là, vous pouvez utiliser les tests défaillants comme indicateurs de ce que vous pouvez faire pour améliorer votre application [20].

Bien qu'il ne s'agisse pas d'une technologie ou d'un concept, l'extension Lighthouse Chrome [17] est un produit Google officiel ciblant les développeurs PWA. L'extension du Lighthouse aidera à comparer les sites Web par rapport à certaines mesures et objectifs que Google évalue comme important, en particulier pour les PWA. L'extension fonctionne même sur sites Web non-PWA, elle est conçue pour aider à optimiser les sites Web pour un meilleur rendu, vitesse, temps de la première interaction possible avec l'utilisateur et conformité générale avec L'objectif de Google pour les PWAs et l'avenir de l'expérience Web (mobile). Lighthouse générera un rapport détaillé sur l'état du site Web testé, fournira des ressources sur la façon d’optimiser le code et les actifs, et donnera une note globale, comme on le voit sur la figure 20 par exemple [17].

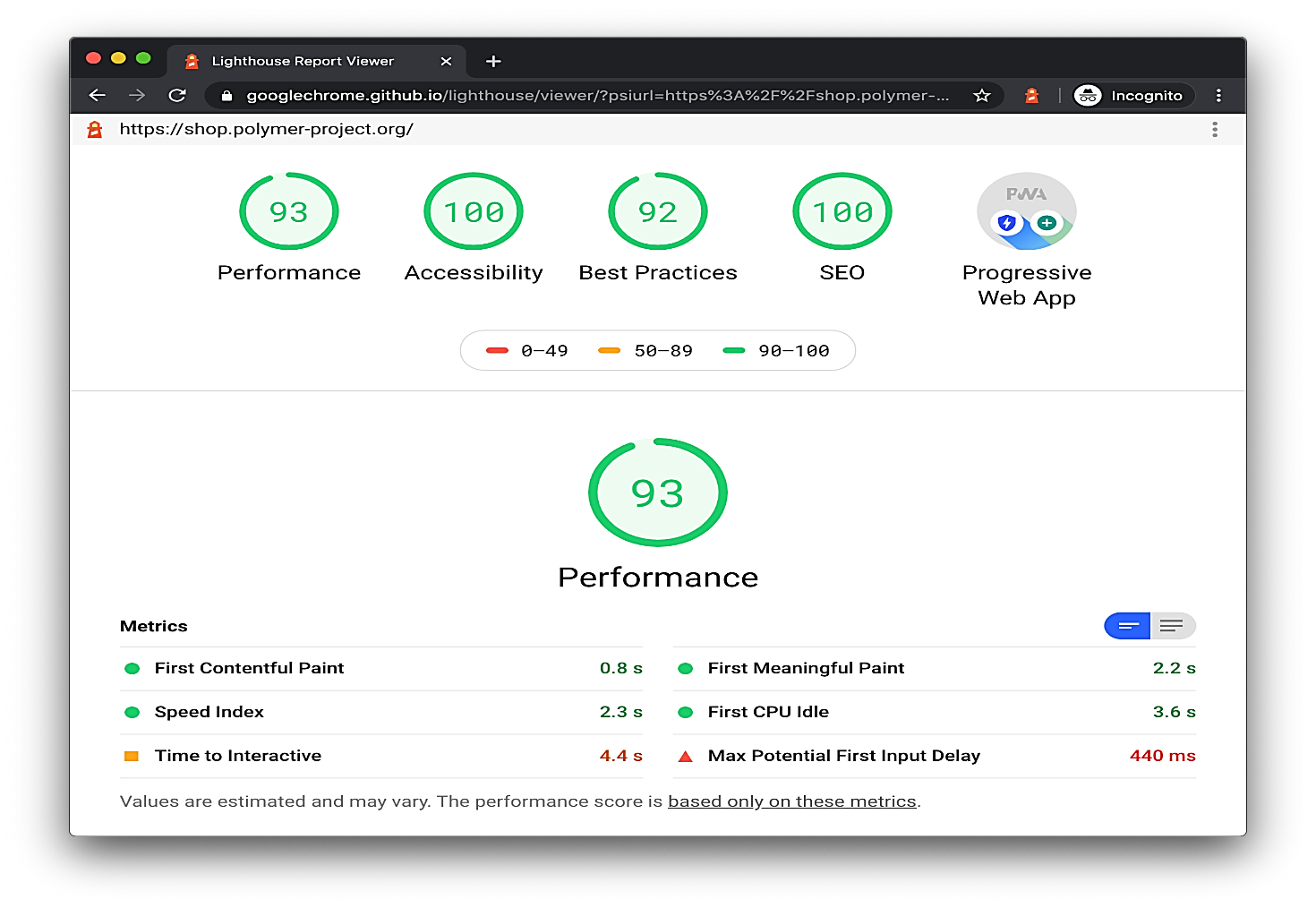


Figure 20 : Exemple de rapport Lighthouse [5]

**Conclusion**

Dans ce chapitre, nous avons présenté l’implémentation du système que nous proposons. A cet effet, nous avons décrit les outils que nous avons utilisés pour développer de l’outil ‘AutoPartDZ’ et nous avons expliqué en détail le fonctionnement de chaque partie de notre système, à travers des descriptions et des captures d'écran. Après cela, nous avons essayé de valider et d'expérimenter notre système avec l’outil Lighthouse.

|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Conclusion générale** |
|  | | | | |

Ce projet de fin d’étude s'est porté sur la conception, la modélisation et la réalisation d'une application web progressive que nous avons nommée « AutoPartDZ ». ‘AutoPartDZ’ est une application web progressive développée pour apporter des solutions dans le domaine de la vente en ligne, le e-commerce, plus spécifiquement celui des pièces automobile.

Ce mémoire a été structuré en quatre chapitres. Le premier chapitre contient l'état de l'art dans les différents domaines qui sous-tendent notre projet, notamment les applications web progressive et le e-commerce. Le deuxième chapitre a été dédié à la modélisation de notre projet. Nous avons aussi enrichi cette architecture par une modélisation UML, où nous avons élaboré les diagrammes de cas d’utilisations,les scénarios possibles ainsi que le diagramme de classes. Apres nous avons expliqué les différentes techniques utilisées pour démontrer une bonne application web progressive afin de faciliter l’implémentation.  Le chapitre 3 est consacré aux étapes de la mise en œuvre de ‘AutoPartDZ’ puis nous avons présenté les différentes parties d' application avec une capture de l’interface représentant chaque partie. Et pour finir, nous avons essayé d'expérimenter et de valider notre système en utilisant un scénario d’exécution de différentes fonctionnalités, comme nous avons testé notre application web avec l’outil Lighthouse.

**Perspectives**

Notre mémoire montre que les PWAs présentent de nombreux avantages qui les rendent idéales pour de nombreux utilisateurs. Ils sont probablement plus rentables que d'autres solutions, les PWAs ne sont peut-être pas la meilleure solution pour certaines entreprises à l'heure actuelle, mais elles ont le potentiel de le devenir bientôt.

L' application web progressive que nous avons développée peut-être enrichie de nombreuses fonctionnalités avancées. Parmi les fonctionnalités que nous proposons pour améliorer l'utilisabilité d’application ‘AutoPartDZ’ nous citons :

* Ajouter la HTTPS qui permet de protéger l'intégrité ainsi que la confidentialité des données lors du transfert des données échangées à l'utilisateur.
* Enrichir l'ajout de Manifest dans notre application web afin de faciliter l'expérience utilisateur et assurer une augmentation du nombre de visiteurs sur notre application e-commerce.
* Intégrer les techniques de Service Worker dans notre application web, pour pouvoir assurer son exécution avec de bonnes ou de mauvaises conditions de connexion Internet, sur mobile ou sur ordinateur.

Enfin, nous espérons que nous avons utilisé efficacement les connaissances que nous avons acquises et l'expérience accumulée tout au long des cinq années d'université, et nous espérons également que nous avons accompli de manière appropriée nos derniers travaux au sein de notre université.

|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Bibliographie** |
|  | | | | |

| [1] | «2,9 milliards de personnes ne sont toujours pas connectées,» UIT, 30 11 2021. [En ligne]. Available: https://www.itu.int/fr/mediacentre/Pages/PR-2021-11-29-FactsFigures.aspx. [Accès le 03 03 2022]. |
| --- | --- |
| [2] | Dean Alan Hume , Addy Osmani, Progressive Web Apps, SHELTER ISLAND: MANNING, 2017. |
| [3] | «Progressive web app (PWA),» Techtarget, 2017. [En ligne]. Available: https://www.techtarget.com/whatis/definition/progressive-web-app-PWA. [Accès le 23 03 2022]. |
| [4] | «Introduction to progressive web apps,» MDN contributors, 01 05 2022. [En ligne]. Available: https://developer.mozilla.org/en-US/docs/Web/Progressive\_web\_apps/Introduction. [Accès le 26 05 2022]. |
| [5] | «Lighthouse,» Google Developers, 2017. [En ligne]. Available: https://developers.google.com/web/tools/lighthouse/. [Accès le 23 03 2022]. |
| [6] | Sam Richard, Pete LePage., «Une Progressive Web App, c'est quoi ? - Définition, cas d'utilisation,» Progressive-web-apps, 2017. [En ligne]. Available: https://www.progressive-web-apps.fr/definition-progressive-web-apps-pwa. [Accès le 23 03 2022]. |
| [7] | V. Magomadov, «Exploring the role of progressive web applications in modern web development,» *Journal of Physics Conference Series,* vol. 1679, n° %1022043, pp. 1-4, 2020. |
| [8] | «Progressive Web Apps,» Google Developers, 2019. [En ligne]. Available: https://web.dev/learn/pwa/progressive-web-apps/. [Accès le 26 05 2022]. |
| [9] | Sam Richard, Pete LePage., «What are Progressive Web Apps?,» Google, 24 02 2020. [En ligne]. Available: https://web.dev/what-are-pwas/. [Accès le 22 05 2022]. |
| [10] | «Progressive web apps : qu’est-ce que c’est et comment ça fonctionne ?,» Yeeply, 2021. [En ligne]. Available: https://fr.yeeply.com/blog/progressive-web-apps/. [Accès le 20 05 2022]. |
| [11] | «Share of e-commerce companies planning on investing in progressive web apps (PWA) in North America and Europe in 2021,» Statista, 2021. [En ligne]. Available: https://www.statista.com/statistics/1174534/investing-pwa-ecommerce-companies-worldwide/. [Accès le 22 05 2022]. |
| [12] | «Unified Modeling Language (UML) | An Introduction,» Geeksforgeeks, 01 04 2019. [En ligne]. Available: https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/. [Accès le 24 05 2022]. |
| [13] | K. S. J. Erickson, «Unified Modeling Language: The Teen Years and Growing Pains,» chez *Human Interface and the Management of Information. Information and Interaction Design*, Las Vegas, NV, USA, Springer, 2013, pp. 295-304. |
| [14] | Pete LePage, François Beaufort, Thomas Steiner, «Add a web app manifest,» Google Developers, 21 10 2021. [En ligne]. Available: https://web.dev/add-manifest/. [Accès le 29 05 2022]. |
| [15] | «Web app manifests,» MDN contributors, 22 05 2022. [En ligne]. Available: https://developer.mozilla.org/en-US/docs/Web/Manifest. [Accès le 29 05 2022]. |
| [16] | «Web app manifests,» Microsoft, 25 01 2022. [En ligne]. Available: https://docs.microsoft.com/en-gb/microsoft-edge/progressive-web-apps-chromium/how-to/web-app-manifests. [Accès le 29 05 2022]. |
| [17] | Andreas Biørn-Hansen, Tim A. Majchrzak, Tor-Morten Grønli., «Progressive Web Apps for the Unified Development of Mobile Application,» pp. 65-86, 2018. |
| [18] | «Introduction to Service Worker,» Google Developers, 01 05 2019. [En ligne]. Available: https://developers.google.com/web/ilt/pwa/introduction-to-service-worker. [Accès le 24 05 2022]. |
| [19] | «Introduction to Progressive Web App Architectures,» Google Developers, 25 02 2021. [En ligne]. Available: https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures. [Accès le 26 05 2022]. |
| [20] | «Lighthouse,» Google Chrome, 2019. [En ligne]. Available: https://chrome.google.com/webstore/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk?hl=en. [Accès le 22 05 2022]. |
| [21] | «React (JavaScript library),» Wikipedia, 11 05 2022. [En ligne]. Available: https://en.wikipedia.org/wiki/React\_(JavaScript\_library). [Accès le 25 05 2022]. |
| [22] | «Node.js - Introduction,» Tutorialspoin, 2022. [En ligne]. Available: https://www.tutorialspoint.com/nodejs/nodejs\_introduction.htm. [Accès le 25 05 2022]. |
| [23] | «Express4.18.0 Fast, unopinionated, minimalist web framework for Node.js,» expressjs, 2017. [En ligne]. Available: https://expressjs.com/. [Accès le 25 05 2022]. |
| [24] | «JavaScript,» Wikipedia, 23 05 2022. [En ligne]. Available: https://en.wikipedia.org/wiki/JavaScript. [Accès le 25 05 2022]. |
| [25] | «MongoDB,» Wikipedia, 19 05 2022. [En ligne]. Available: https://en.wikipedia.org/wiki/MongoDB. [Accès le 25 05 2022]. |
| [26] | «Documentation for Visual Studio Code,» Microsoft, 14 04 2016. [En ligne]. Available: https://code.visualstudio.com/docs. [Accès le 14 05 2022]. |
| [27] | «What is Google Charts,» Reflectivedata, 12 03 2020. [En ligne]. Available: https://reflectivedata.com/dictionary/google-charts/. [Accès le 25 05 2022]. |
| [28] | W. contributors, «GitHub,» Wikipedia, 18 05 2022. [En ligne]. Available: https://en.wikipedia.org/wiki/GitHub. [Accès le 24 05 2022]. |
| [29] | «diagrams.net,» Google Workspace Marketplace, [En ligne]. Available: https://workspace.google.com/marketplace/app/diagramsnet/671128082532. [Accès le 24 05 2022]. |
| [30] | «Use Service Workers to manage network requests and push notifications,» Microsoft, 03 01 2022. [En ligne]. Available: https://docs.microsoft.com/en-us/microsoft-edge/progressive-web-apps-chromium/how-to/service-workers. [Accès le 24 05 2022]. |

|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Acronymes** |
|  | | | | |

| UML | Unified Modeling Language |
| --- | --- |
| PWA | Progressive Web Application |
| SEO | Search Engine Optimization |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| HTTPS | Hypertext Transfer Protocol Secure |

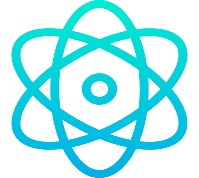
|  | |  | | |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
| **Annexe** |
|  | | | | |

**Outils de développements**

La mise en œuvre de notre système nécessite l'utilisation de différents outils de développement et de langages de programmation. Pour cela, une vaste gamme de choix s’offre à nous. Notre but est de sélectionner parmi les outils disponibles, ceux que nous pensons pouvoir nous fournir l’aide nécessaire afin de réaliser notre système d’une manière optimale. Dans ce qui suit, nous présentons les outils que nous avons utilisés pour la mise en place de l’outil ‘AutoPartDZ’ tout en motivant nos choix.

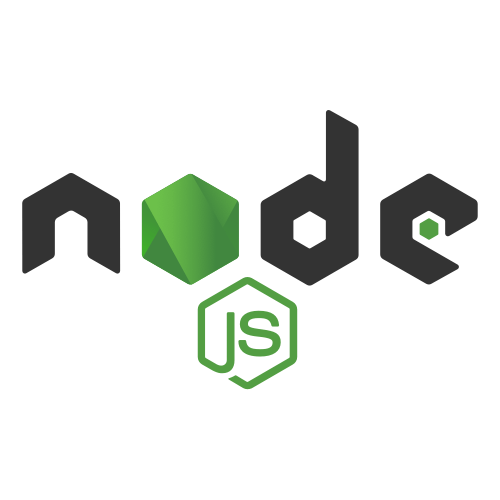
**Frameworks utilisés**

Un Framework est un ensemble de bibliothèques modulables qui travaillent dans un cadre commun, leur but est d’éviter la réécriture de code pour des problèmes similaires déjà résolus. Dans ce contexte, les Frameworks que nous avons utilisés pour la réalisation de notre projet sont :



**React**

React (également connu sous le nom de React.js ou ReactJS) est une bibliothèque JavaScript frontale open source gratuite permettant de créer des interfaces utilisateur basées sur des composants d'interface utilisateur. Il est maintenu par la communauté de Meta (anciennement Facebook) et des développeurs individuels et corporatifs. React peut être utilisé comme base pour développer des applications d'une page, mobiles ou rendues par un serveur à l'aide de frameworks tels que Next.js. Cependant, étant donné que React consiste à gérer l'état et à représenter cet état dans le DOM, la création d'applications React nécessite généralement l'utilisation de bibliothèques supplémentaires pour le routage et certaines fonctionnalités côté client. [21]



**NodeJS**

Node.js est une plateforme côté serveur basée sur le moteur JavaScript de Google Chrome (moteur V8). Node.js a été développé par Ryan Dahl en 2009.

Node.js est un environnement d'exécution multiplateforme open source pour le développement d'applications côté serveur et côté réseau. Les applications Node.js sont écrites en JavaScript et s'exécutent dans l'environnement d'exécution Node.js sous OS X, Microsoft Windows et Linux.

Node.js fournit également une vaste bibliothèque de divers modules JavaScript qui facilitent grandement le développement d'applications Web avec Node.js. [22]



**ExpressJS**

Express est un cadre d'application Web Node.js minimal et flexible qui fournit un ensemble de fonctionnalités robustes pour les applications Web et mobiles.

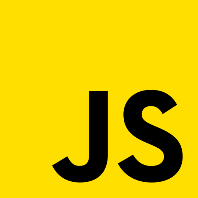
Avec une variété de méthodes utilitaires HTTP et d'intergiciels prêts à l'emploi, vous pouvez créer rapidement et facilement des API robustes.

Express fournit une fine couche de fonctionnalités d'application Web de base sans obscurcir les fonctionnalités Node.js que vous connaissez et aimez.

De nombreux frameworks populaires sont basés sur Express. [23]

**Langages de programmation utilisés**

Dans cette partie, nous présentons les différents langages de programmation que nous avons utilisé. Il s’agit d’outil permettant de formuler des algorithmes et de produire des programmes informatiques qui appliquent ces algorithmes.



**JavaScript**

JavaScript (JS) est un langage de programmation léger, interprété ou compilé juste à temps avec des fonctionnalités de premier ordre. Il est surtout connu comme langage de script pour les pages Web, mais il est également utilisé dans de nombreux environnements autres que les navigateurs. B.Node.js, Apache CouchDB, Adobe Acrobat. JavaScript est un langage dynamique, basé sur des prototypes, à plusieurs paradigmes et à un seul thread qui prend en charge les styles orientés objet, impératifs et déclaratifs (tels que la programmation fonctionnelle). En savoir plus sur JavaScript [24].

**Base de données**

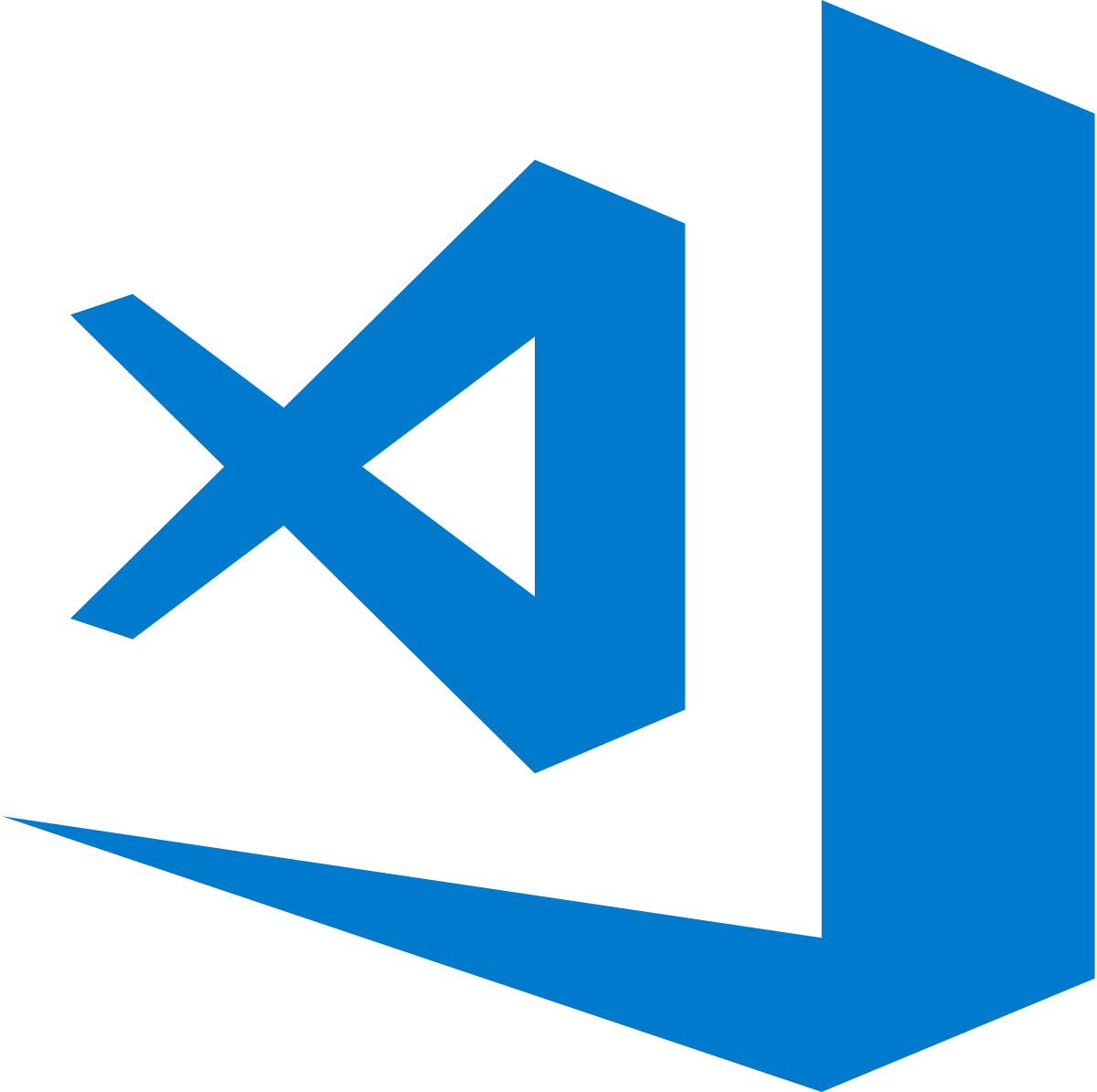
La majorité des systèmes informatique nécessitent un espace de stockage pour enregistrer et stocker les données et les informations il existe aujourd’hui plusieurs types de bases de données selon leurs structures, le type de données et leurs manières de stocker ces dernières. Dans notre cas nous avons utilisé ‘Mongo DB’.

**MongoDB**

MongoDB est un programme de base de données multiplateforme orienté document disponible en code source. Classé comme programme de base de données NoSQL, MongoDB utilise une documentation de type JSON avec un schéma facultatif. MongoDB a été développé par MongoDB Inc. et est distribué sous licence publique côté serveur (SSPL). [25]

**Environnement de développement (IDE)**

Dans cette partie, nous présentons les différents IDE (Integrated Development Environment) ou (Environnement de Développement Intégré) que nous avons utilisé, Un IDE est un logiciel permettant de créer des applications qui combinent des outils de développement communs dans une seule interface utilisateur graphique (GUI).



**Visual Studio Code**

Visual Studio Code est un éditeur de code source proposé par Microsoft disponible pour Windows, MacOs et Linux. Il est créé avec un support intégré pour Java Script, Type Script et Node.js, mais il dispose d'un riche écosystème d'extensions pour d'autres langages tels que C++, C #, Java, Python, PHP…, et bien sûr ce dont nous avons besoin Flutter & Dart.

Vs Code est très populaire parmi les développeurs, il est open source, facile à installer et à utiliser, très riche en extensions, le VS Code marketplace contient des centaines d'extensions qui nous permettant d'ajouter de nouveaux langages ou d'ajouter des outils pour faciliter le processus de développement [26].

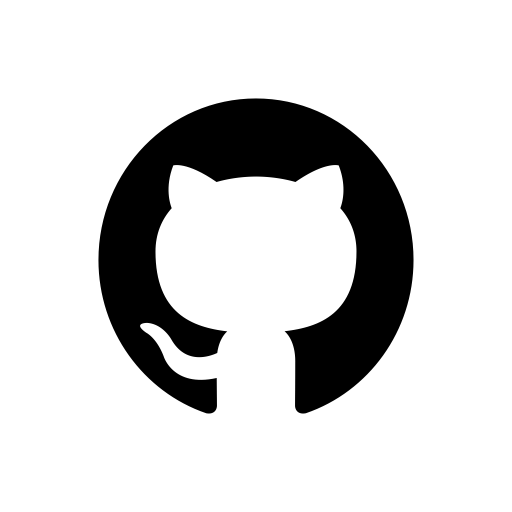
**Autres outils utilisés**

Il existe également d'autres outils que nous avons utilisés pour développer l’outil ‘AutoPartDZ’ notamment :

**Google Charts**

Google Charts est une bibliothèque de visualisation de données basée sur JavaScript, facile à utiliser et offrant une variété d'options de personnalisation.

Vous pouvez exécuter Google Charts n'importe où sur le Web. [27]



**GitHub**

GitHub est un fournisseur d'hébergement Internet pour la distribution et la gestion des codes sources à l'aide de Git. Il fournit ainsi un contrôle d'accès et autres fonctionnalités de collaboration telles que le suivi des bogues, la gestion des tâches, l'intégration continue et les wikis pour chaque projet.

Nous avons utilisé GitHub pour enregistrer le code source que nous avons développée, GitHub est gratuit et largement utilisé par les développeurs, Vs Code a l'extension GitHub qui nous permet de mettre à jour le code en seulement deux étapes [28].



**Draw.io**

Draw.io est un éditeur de diagramme en ligne entièrement gratuit construit par Seibert Media, qui offre de nombreuses possibilités pour créer des diagrammes tels que UML, relation d'entité, diagrammes de réseau et plus encore [29]. Nous avons utilisé Drow.io pour créer les différentes architectures et diagrammes qui existent dans ce mémoire.