

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid - Tlemcen  
Faculté des Sciences  
Département d'informatique

Mémoire de fin d'études pour l'obtention du diplôme de Master en  
Informatique

*Option : Génie Logiciel (G.L)*

*Thème*

**Conception et développement d'une  
Application mobile de vente assistée.**

**Réalisé par :**

- LACHACHI Abdelkrim
- BENMANSOUR Rayene

**Présenté le 26 Juin 2023 devant le jury composé de :**

- ILES Nawel (Présidente)
- HALFAOUI Amal (Encadrante)
- BENMANSOUR Asma (Examinatrice)
- SEMMOUD Abderezzak (Maître de stage)

## **Remerciements**

*Tout d'abord, nous remercions ALLAH tout puissant, de nous avoir accordé le courage, la force, la patience et le savoir tout au long de nos parcours universitaires, nous nous sommes reconnaissants pour les bénédictions qui nous ont été accordés dans nos recherches et la réalisation de ce mémoire qui est le fruit de la grâce divine.*

*Nous tenons à exprimer notre profonde gratitude à **Madame Halfaoui Amal**, pour l'aide, le soutien constant et les conseils qu'elle a fournis, pour sa présence d'une valeur inestimable et ses remarques pertinents qui ont enrichi nos réflexions afin de pouvoir approfondir nos recherches et savoirs. Nous sommes honorés d'avoir pu bénéficier de son expertise et de son évaluation profonde décidément judicieux dans la réussite de ce travail.*

*L'expression de nos vifs remerciements vont également à **Madame Iles** et **Madame Benmansour** pour leurs participations et leurs intérêts dans l'évaluation de notre travail.*

*Nous tenons également à remercier **Monsieur Semmoud** le responsable de **DATAMASTER** pour sa confiance, ses orientations et son soutien qui ont été d'une importance capitale pour le bon déroulement de notre projet. Nous sommes reconnaissants de l'opportunité qui nous a été offerte de travailler au sein de son entreprise et d'acquérir une expérience précieuse.*

*Nous remercions tous les enseignants de notre département d'informatique, pour leurs efforts fournis, et leurs savoirs transmis.*

*Un grand merci à nos parents pour leurs amours, leurs conseils ainsi que leurs soutiens indéfectibles et leurs encouragements, à la fois morales et économiques, qui nous ont permis de réaliser les études que nous voulions et par conséquent ce mémoire.*

*Et enfin nous adressons nos remerciements et reconnaissances envers nos amis et collègues qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.*

## *Dédicaces*

*A mes chers parents, qui ont toujours été mes piliers et ma source de soutien inconditionnel. Votre amour, vos encouragements et vos sacrifices ont été les fondations solides sur lesquels j'ai pu construire mon parcours académique. Je vous suis profondément reconnaissant pour tout ce que vous avez fait pour moi.*

*À ma famille, qui a été un soutien constant tout au long de ce voyage. Votre présence, vos encouragements et vos conseils précieux m'ont donné la force et la confiance nécessaires pour persévérer. Je vous suis redevable de toujours être là pour moi, peu importe les circonstances.*

*À mes amis, qui ont été mes compagnons fidèles, mes confidents et mes sources de joie pendant cette période. Vos encouragements, votre soutien et vos moments de partage ont allégé le fardeau des études et m'ont permis de trouver un équilibre.*

*À mon binôme et mon cousin **Abdelkrim**, avec qui j'ai partagé cette aventure académique. Ta collaboration, ton dévouement et ton travail d'équipe ont été inestimables. Nous avons relevé les défis ensemble, surmonté les obstacles et accompli ce projet grâce à notre synergie.*

*A tous ceux j'aime et ceux qui m'aiment.*

***BENMANSOUR Rayene***

# *Dédicaces*

*À Dieu, tout d'abord, je rends grâce pour la force, la sagesse et les opportunités qui m'ont été accordées tout au long de mon parcours académique.*

*Je dédie humblement ce travail à la mémoire de mon cher cousin **Mohamed Abderrahman**, dont le départ prématuré a laissé un vide dans nos vies, mais dont la présence et l'inspiration continuent de briller dans nos cœurs.*

*À mes parents, mes piliers inébranlables, qui ont toujours cru en moi et m'ont soutenu de façon inconditionnelle, je vous suis profondément reconnaissant pour votre amour et votre encouragement constants.*

*À mon frère et mes deux sœurs, qui sont ma source constante de motivation et de bonheur, vous êtes les véritables joyaux de ma vie.*

*À mes grands-parents, qui ont nourri mon âme de principes intemporels et ont enveloppé mon cœur de tendresse inconditionnelle, je vous suis infiniment reconnaissant pour votre amour et votre soutien inépuisable.*

*À mon binôme et cousin **Rayene**, nous avons parcouru ensemble ce chemin académique, et je ne saurais exprimer ma gratitude pour ton amitié, ta patience et ton soutien sans faille.*

*Enfin, à tous ceux qui m'ont soutenu de près ou de loin, je vous adresse ma plus profonde reconnaissance.*

**LACHACHI Abdelkrim**

# *Table des matières*

<b>Introduction Générale.....</b>	<b>9</b>
I.1 Introduction générale .....	1
I.2 Contexte globale.....	1
I.3 Problématique .....	3
I.4 Objectifs .....	3
I.5 Environnement de stage .....	4
I.6 Présentation de l'organisme d'accueil.....	4
I.7 Présentation des précédents projets de DATAMASTER .....	4
I.8 Plan de mémoire.....	5
<b>Étude et analyse du système de vente assistée.....</b>	<b>6</b>
II.1 Introduction.....	5
II.2 Présentation du projet à réaliser.....	5
II.1 Spécification des besoins et des exigences système .....	6
II.5.1 Identification des acteurs .....	8
II.5.2 Exigences fonctionnelles .....	9
II.5.3 Exigences non-fonctionnels .....	10
II.5.4 Exigences reportées.....	11
II.2 Conclusion .....	12
<b>Conception du système de vente assistée.....</b>	<b>13</b>
III.1 Introduction .....	14
III.2 Outils utilisés .....	14
III.2.1 PowerPoint .....	14
III.2.2 StarUML .....	14
III.2.3 Teamgantt.....	15
III.3 Répartition des tâches (diagramme de Gantt).....	15
III.4 Modélisation des fonctionnalités .....	16
III.4.1 Diagrammes de cas d'utilisation « chargé (e) de rayon ».....	16
III.4.1.2 Diagramme de cas d'utilisation d'un « Fournisseur ».....	17
III.4.1.3 Diagramme de cas d'utilisation d'un « Client » .....	18
III.4.2 Diagrammes de séquence .....	19
III.4.2.2 Diagramme de séquence « ajout ou modification d'un client ».....	22

III.4.3	Diagrammes de classe .....	24
III.5	Conclusion.....	28
<b>Réalisation de l'application «Vente Assistée » .....</b>		<b>29</b>
IV.1	Introduction .....	29
IV.2	Outils et Technologies utilisés .....	29
IV.2.1	Les outils utilisés .....	29
IV.2.2	Les outils et technologies utilisés coté serveur (Backend).....	31
IV.2.3	Les outils et technologies utilisés coté client (Frontend) .....	32
IV.3	Architecture du système .....	32
IV.3.1	L'architecture du backend .....	32
IV.3.2	L'architecture du frontend.....	34
IV.3.3	Autres technologies utilisées.....	34
IV.3.4	Tests et performance .....	35
IV.4	Déploiement de l'application .....	36
IV.5	Défis et obstacles rencontrés.....	36
IV.6	L'application « vente assistée » .....	36
IV.6.1	L'interface d'authentification.....	37
IV.6.2	L'interface 'Dashboard' (Tableau de bord).....	39
IV.6.3	L'interface 'Produits' .....	42
IV.6.4	L'interface d'un produit sélectionné.....	44
IV.6.5	L'interface 'Tickets' .....	45
IV.6.6	L'interface 'Commandes' .....	47
IV.6.7	L'interface 'Clients' .....	48
IV.6.8	L'interface 'Stock' .....	50
IV.6.9	L'interface 'Historique' .....	51
IV.6.10	L'interface 'Paramètres' .....	52
IV.7	Conclusion.....	54
<b>Conclusion générale .....</b>		<b>55</b>
<b>Bibliographie.....</b>		<b>57</b>

# *Liste des figures*

<b>Figure I : Quelques Statistiques de 'Google Forms' .....</b>	<b>8</b>
<b>Figure II: Logos des outils utilisés .....</b>	<b>14</b>
<b>Figure III : diagramme de Gantt.....</b>	<b>15</b>
<b>Figure IV: Diagramme de cas d'utilisation du vendeur .....</b>	<b>16</b>
<b>Figure V: Diagramme de cas d'utilisation du fournisseur.....</b>	<b>17</b>
<b>Figure VI: Diagramme de cas d'utilisation du client.....</b>	<b>18</b>
<b>Figure VII: Diagramme de séquence "lancer une commande" pour un vendeur.....</b>	<b>21</b>
<b>Figure VIII: Diagramme de séquence "lancer une commande" pour un fournisseur ...</b>	<b>24</b>
<b>Figure IX: Diagramme de classe.....</b>	<b>25</b>
<b>Figure X: Modèle logique de données (MLD).....</b>	<b>27</b>
<b>Figure XI: Liste des outils et technologies utilisés .....</b>	<b>29</b>
<b>Figure XII : Architecture de l'application .....</b>	<b>35</b>
<b>Figure XIII : Interface d'authentification (taille importante) .....</b>	<b>37</b>
<b>Figure XIV : Interface d'authentification (taille réduite et smartphone).....</b>	<b>38</b>
<b>Figure XV : Interface 'Dashboard' (taille importante).....</b>	<b>40</b>
<b>Figure XVI : Interface 'Dashboard' (taille réduite et smartphone).....</b>	<b>40</b>
<b>Figure XVII: Interface du 'Side Menu' .....</b>	<b>41</b>
<b>Figure XVIII : Interface 'Produits' (List View).....</b>	<b>43</b>
<b>Figure XIX : Interface 'Produits' (Grid View) .....</b>	<b>43</b>
<b>Figure XX : Interface 'Produit sélectionné' .....</b>	<b>44</b>
<b>Figure XXI : Interface 'Tickets' .....</b>	<b>46</b>
<b>Figure XXII : Interface 'Tickets' (Version2) .....</b>	<b>46</b>
<b>Figure XXIII : Interface 'Commandes' .....</b>	<b>47</b>
<b>Figure XXIV : Interface 'Clients' .....</b>	<b>49</b>
<b>Figure XXV : Interface 'Stock' .....</b>	<b>50</b>
<b>Figure XXVI : Interface 'Historique' .....</b>	<b>52</b>
<b>Figure XXVII : Interface 'Paramètres' .....</b>	<b>53</b>

# Liste des abréviations

**SDK** Software Development Kit

**API** Application Programming Interface

**BDD** Base De Données

**APK** Android Package Kit

**REST** Representational State Transfer

**UML** Unified Modeling Language

**MLD** Modèle Logique de Données

**IDE** Integrated Development Environment

**XML** eXtensible Markup Language

**GIT** Global Information Tracker

**URL** Uniform Resource Locator

**MVC** Model-View-Controller

**MVT** Model-View-Template



# **Introduction Générale**

## **I.1 Introduction générale**

Avec l'impact des applications mobiles et l'avènement d'une ère de technologie, l'industrie des marchandises et produits a connu une évolution constante. Les consommateurs expriment des exigences croissantes, ce qui pose de nouveaux défis aux entreprises. Dans ce contexte, la vente de produits et leur commercialisation jouent un rôle essentiel dans la performance économique et commerciale des entreprises, contribuant directement à leur rentabilité, leur croissance et leur compétitivité sur le marché mondial.

L'émergence de l'ère numérique a révolutionné les pratiques commerciales, ouvrant de nouvelles opportunités pour répondre aux besoins changeants des clients. Les avancées technologiques, conjuguées à l'évolution des comportements d'achat des consommateurs, ont conduit à l'émergence de nouvelles stratégies et de solutions innovantes dans le domaine de la vente et de la commercialisation des produits.

Cependant, malgré ces avancées, la vente traditionnelle ou classique rencontre des limites dans sa capacité à répondre de manière efficace aux exigences croissantes des clients. C'est ici que la vente assistée entre en jeu pour résoudre ce problème. En proposant une approche novatrice, la vente assistée exploite les technologies avancées, telles que les applications mobiles, pour améliorer l'expérience client, optimiser les processus de vente et renforcer la compétitivité des entreprises.

Au fil de ce mémoire, nous explorerons en détail les différentes dimensions de la vente assistée, en analysant ses avantages, ses fonctionnalités clés et son impact sur l'industrie des marchandises et produits. Nous aborderons également les aspects techniques, les défis potentiels et les recommandations pour une mise en œuvre réussie de cette solution.

En somme, en intégrant les applications mobiles et les avancées technologiques, la vente assistée se présente comme une réponse efficace aux défis auxquels sont confrontées les entreprises dans l'industrie des marchandises et produits. Elle offre des opportunités pour améliorer l'expérience client et renforcer la compétitivité globale.

## **I.2 Contexte globale**

L'industrie des marchandises et produits est un pilier de l'économie mondiale, englobant la production, la distribution et la vente de biens physiques. Cette industrie est extrêmement diversifiée et comprend une multitude de secteurs tels que l'agriculture, la production manufacturière, la distribution en gros et la vente au détail assez spécifiquement.

Le commerce de vente en gros et au détail, quant à lui, englobe la commercialisation de ces marchandises auprès des consommateurs car c'est un secteur dynamique qui est en constante évolution, il s'adapte aux changements des comportements des consommateurs, des avancées technologiques et des pressions environnementales.

Aujourd'hui la vente des produits et leurs commercialisation (gros et détail) est une étape cruciale et incisive sur les revenus financés d'une entreprise, sa croissance et sa performance commerciale et son chiffre d'affaires permettant le recouvrement des coûts de production et les dépenses de marketing.

Pour cela plusieurs entreprises phares dans l'industrie des marchandises et produits doivent s'adapter pour répondre aux demandes changeantes du marché et des clients et maintenir leur compétitivité et leur rentabilité face aux pressions externes telles que la concurrence accrue, les coûts de production élevés, les normes environnementales et les exigences réglementaires. Tout en incluant au processus de vente et de commercialisation de produits une solution technologique et innovante qui répond aux différents problèmes engendrés tels que la non-satisfaction des clients en raison du faible nombre de réseaux de distribution et de vente, manque de produits et d'assistance, ainsi que les temps d'attentes et aussi l'absence de la personnalisation des produits et leurs disponibilités en temps réel.

Dans ce contexte, notre projet de fin d'études se situe au croisement de ces enjeux. Nous nous proposons de développer une application de vente assistée, intégrant une solution technologique et innovante, pour résoudre ces problèmes spécifiques. Cette application vise à améliorer l'expérience client en offrant une assistance personnalisée, des recommandations ciblées, et en facilitant l'accès aux produits en temps réel.

En travaillant en collaboration avec l'entreprise "DATAMASTER", spécialisée dans le développement de solutions technologiques, nous nous efforçons de concevoir une application de vente assistée adaptée au marché algérien, répondant aux besoins spécifiques des clients et contribuant à la compétitivité des entreprises.

Selon un rapport de [State of Retail Report](#), 88% des consommateurs affirment qu'il est plus probable qu'ils achètent lorsqu'ils reçoivent des conseils de la part des employés du magasin et pour cette raison la vente assistée qui va être l'objet de notre étude constitue une stratégie optimisée de point de vente pour accompagner le client plus précisément la gestion de commandes personnalisées basées sur les achats des clients et leurs besoins.

### I.3 Problématique

Actuellement en Algérie la vente de produits et leurs commercialisations se fait encore de façon classique ce qui engendre des conséquences dévastatrices sur l'évolution économique et commerciale d'une entreprise, et cela entraîne :

- Un commerce non-lucratif.
- Expérience client limitée : les clients ne peuvent pas être aidés pour trouver les produits dont ils ont besoin ainsi leurs problèmes de remplacement ou de remboursement.
- Manque de produits et d'assistance : les clients n'auront pas accès à l'intégralité des produits en raison de leurs placements en rayons et leurs quantités et aussi en absence d'assistance qui semble selon des rapports être primordiales dans leurs choix de produits.
- Absence du **UpSelling**<sup>1</sup> et **Cross-Selling**<sup>2</sup> : les clients ne peuvent pas voir ce qui pourra être recommandé pour eux en raison d'absence d'historique de leurs achats et les vendeurs ne peuvent pas leur proposer des produits supplémentaires ou complémentaires qui peuvent intéresser les clients et augmenter leurs achats.
- Temps d'attente considérable : le client peut attendre jusqu'à 30min pour être satisfait en raison du nombre réduit du personnel et de son efficacité.

### I.4 Objectifs

L'objectif de ce projet est de pouvoir mettre en œuvre une solution qui satisfait les besoins et les exigences d'une large clientèle grâce à une application de vente assistée qui se résume en plusieurs points :

- Amélioration de l'expérience client : les vendeurs ou fournisseurs peuvent offrir une expérience de haute qualité en aidant les clients à trouver les produits dont ils ont besoin, en résolvant les problèmes et en répondant à leurs questions.
- Large gamme de produits : un catalogue de produits qui touche la totalité des produits avec une description de chaque produit.
- **UpSelling** et **Cross-Selling** : les vendeurs peuvent proposer des produits supplémentaires ou complémentaires qui peuvent intéresser les clients, ce qui peut augmenter le panier moyen.
- Historique des commandes et achats : cela permettra aux vendeurs de mieux comprendre le client afin de lui recommander les produits souvent procurés par les clients.

<sup>1</sup> Le UpSelling (dite aussi la montée en gamme) est une stratégie de vente qui consiste à proposer au client le produit le plus cher de la même catégorie (le meilleur produit, le plus récent...) [1].

<sup>2</sup> Le Cross-Selling (dite aussi la vente additionnelle) est une technique de vente qui consiste à proposer au client un produit supplémentaire adapté au produit sélectionné [1].

- Multigestion des clients : les vendeurs peuvent gérer plusieurs clients à la fois grâce au système de précommande.
- Achat et livraison immédiate : le client reçoit son panier de produits à la seconde où il donne son bon de commande au caissier qui ce dernier sera délivré par le vendeur.

## **I.5 Environnement de stage**

Nous avons effectué notre stage de fin d'études au sein de l'entreprise de communication et de développement « DATAMASTER », une jeune entreprise spécialisée dans le domaine de l'informatique créée le 1<sup>er</sup> juin 2015. Son objectif principal est de fournir des solutions informatiques à ses clients. La société est située à Les Dahlias, El Kiffane, Tlemcen.

## **I.6 Présentation de l'organisme d'accueil**

Fondée en 2015, DATAMASTER est une jeune entreprise spécialisée dans le développement et le support informatique. Son siège social se trouve à Les Dahlias, El Kiffane, Tlemcen. L'objectif de DATAMASTER est de comprendre les besoins spécifiques de chaque client et de leur proposer des solutions informatiques personnalisées afin de les accompagner jusqu'à l'aboutissement de leur projet.

Les solutions proposées par DATAMASTER sont le fruit d'une collaboration étroite avec ses clients, afin de s'assurer que les logiciels développés répondent à leurs besoins exacts. L'entreprise se distingue par son approche axée sur les résultats, qui vise à offrir à ses clients un gain de temps et d'argent en leur fournissant des solutions efficaces.

L'entreprise assure également la maintenance de ses logiciels, que ce soit à distance ou sur site, afin d'assurer un fonctionnement optimal et une assistance en cas de besoin, elle respecte aussi la charte graphique de ses clients en proposant des designs personnalisés pour leurs logiciels.

L'entreprise est assez réputée notamment grâce à son activité hors territoire national et aussi pour la qualité de leurs solutions.

## **I.7 Présentation des précédents projets de DATAMASTER**

- Conception et développement d'une application web pour la gestion des documents numériques.
- Développement d'un module de gestion de réapprovisionnement automatique.
- Conception et réalisation d'un gestionnaire intelligent des alertes dans un Système ERP.

- Conception et réalisation d'une application web pour la gestion de livraison.
- Conception et réalisation d'une application web pour la prise de rendez-vous et la gestion des ressources.
- Réalisation d'une application mobile pour un système d'information géographique communal.

## **I.8 Plan de mémoire**

Notre travail est constitué de trois chapitres :

**Le 1<sup>er</sup> chapitre** consiste à mettre au point un ensemble de définitions et de présentations de plusieurs concepts et outils dans le contexte de notre étude. Notamment aussi la présentation de l'organisme d'accueil et leurs projets réalisés au cours de leur existence et nous clôturerons le chapitre avec un tableau de fonctionnalités permettant d'illustrer notre projet ainsi sa gestion.

**Le 2<sup>ème</sup> chapitre** consiste à bien illustrer le déroulement du système avec la spécification des besoins et exigences, notamment aussi la conception et la modélisation de ce dernier avec les différents diagrammes.

**Le 3<sup>ème</sup> chapitre** quant à lui représente l'architecture de notre système à développer et les différents outils utilisés durant la phase de conception et de développement, ainsi que les résultats obtenus des deux phases et on clôtura avec une représentation graphique du système et du travail réalisé.

# **Chapitre I :**

# **Étude et analyse du système**

# **de vente assistée**

## II.1 Introduction

Dans ce chapitre, nous allons présenter l'organisme d'accueil, les projets réalisés au sein de cet organisme, ainsi qu'une présentation du projet à réaliser. Nous examinerons également les besoins et exigences du système que nous avons conçus, en mettant l'accent sur les fonctionnalités clés et les attentes des utilisateurs.

Nous allons aussi définir les différents besoins et exigences que comporte notre système que ça soit fonctionnelles ou non-fonctionnelles. Les exigences concernent tous les utilisateurs de l'application et ils ont été rédigées et ajoutées en suivant la méthode agile **SCRUM**<sup>3</sup> « de nouvelles exigences sont apparues tout au long du développement du projet, d'autres sont modifiées et d'autres sont supprimées carrément ».

L'objectif de ce chapitre est de fournir une vision d'ensemble de notre projet, en mettant en évidence l'importance de répondre aux besoins croissants de l'industrie et des clients grâce à une approche fiable et adaptée.

## II.2 Présentation du projet à réaliser

Notre application de vente assistée visera à améliorer l'expérience d'achat en magasin. Elle proposera une interface utilisateur conviviale adaptée aux différentes fonctionnalités. Les utilisateurs seront soit des « Vendeurs en rayon » chargés de prendre en charge les commandes internes pour une clientèle large et exigeante, soit des « fournisseurs ». L'application leur offrira un catalogue de produits filtré selon divers critères tels que le prix, la catégorie, les promotions, etc. Chaque produit sera accompagné d'une fiche détaillée comprenant des images et une description. Ils pourront également surveiller l'état du stock, éviter les doublons de commandes et s'assurer que les produits sont constamment mis à jour. Le fournisseur utilisera une fonctionnalité similaire, mais avec une clientèle différente et un mode de paiement spécifique. En revanche, le vendeur conclura la commande en générant un bon.

C'est une application qui offre aussi la possibilité d'utiliser les outils d'analyses de données et l'historique des activités de la journée pour suivre les performances de vente, les tendances du marché et les préférences des clients qui peuvent être utilisées pour améliorer les stratégies de vente. Elle prend également en charge le mode déconnecté pour permettre aux utilisateurs de satisfaire des commandes sans accès à Internet.

<sup>3</sup> Scrum est une méthode de développement agile orientée projet informatique dont les ressources sont régulièrement actualisées. La méthode Scrum tire son nom du monde du rugby et elle considérée comme la méthode la plus utilisée grâce à des feedbacks réguliers [2, 3].



Elle permet aussi aux utilisateurs de pouvoir suivre les informations des clients incluant leurs historiques de commandes et d'anciens achats pour améliorer et de personnaliser l'expérience d'achat de chaque client.

En résumé, notre application de vente assistée offre une expérience d'achat en magasin fluide, efficace et personnalisée pour une large clientèle, tout en offrant des fonctionnalités pour améliorer la gestion des entreprises et la stratégie de vente.

## **II.1 Spécification des besoins et des exigences système**

En ce qui concerne cette section, elle décrit toutes les fonctionnalités qui ont été mises en place sur l'application et qui ont été délivrées par l'entreprise dont laquelle on effectue notre stage, par contre nous avons essayé de les parfaire et de les compléter en s'appuyant sur les avis de quelques proches et d'autres personnes ayant l'habitude des achats classiques en leurs proposant un questionnaire de **GOOGLE FORMS**<sup>4</sup> (comme l'indique la figure ci-dessous) et cela a été percutant dans l'évolution de l'application.

<sup>4</sup> Google forms est un logiciel d'administration d'enquête inclus dans la suite Web gratuite Google Docs Editors proposée par Google [4].

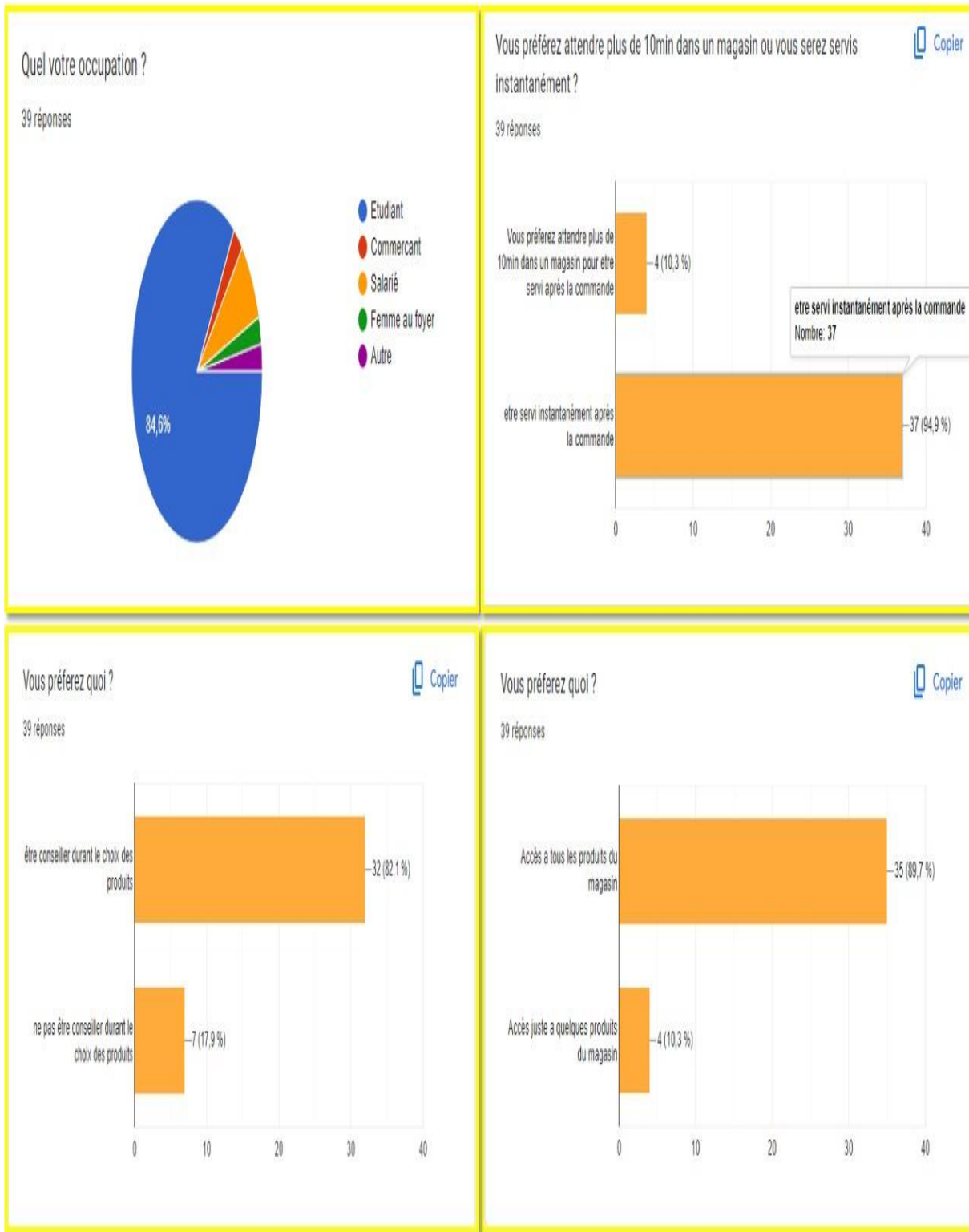


Figure 1 : Quelques Statistiques de 'Google Forms'

## II.5.1 Identification des acteurs

Une entité externe (utilisateur humain, dispositif matériel ou autre système) est dite acteur qui aura des interactions avec le système.

Les acteurs humains :

### ❖ **Le client (e)**

Il n'a aucun accès à l'application et il peut effectuer les opérations suivantes :

- Passer une commande auprès d'un des deux autres acteurs.
- Passer une précommande auprès d'un des autres acteurs en cas de non-disponibilité des produits désirés.
- Faire une réclamation auprès d'un supérieur en raison de maltraitance ou négligence d'un des deux autres acteurs.
- Possibilité de faire un retour en cas d'endommagement de produit, ou ne correspond pas à la description ou a été livré par erreur.
- Possibilité de garder l'anonymat durant la passation de commande.
- Conseiller l'un des deux autres acteurs d'approvisionner ou d'acquérir l'un des produits non-disponibles désirés par le client.

### ❖ **Le Vendeur en rayon « Chargé (e) de rayon »**

Une fois authentifié, il pourra effectuer les opérations suivantes :

- Satisfaire la commande du client en lui attribuant à la fin un bon de commande qui permettra au dernier de pouvoir acquérir ses produits au près du caissier après les avoir payés.
- Voir l'analyse des ventes, des commandes satisfaites et des produits du marché, des produits vendus et celles qui sont en pré-rupture.
- Accès à l'ensemble des commandes et des précommandes enregistrées au paravent.
- Avoir accès aux données et aux achats des clients.
- Attribuer des privilèges aux clients (VIP, Passager...).
- Accès au stock pour pouvoir alerter le gestionnaire de stock en cas de rupture d'un des produits.
- Accès à l'historique qui lui permettra d'avoir une vue globale sur l'ensemble de ses activités journaliers.
- Accès à ces propres informations et possibilité de les modifier.

### ❖ Fournisseur

Une fois authentifié, il pourra effectuer les opérations suivantes :

- Satisfaire la commande du client en lui attribuant à la fin un bon de commande ainsi que les produits après avoir effectué le paiement soit en espèces ou par carte de crédit.
- Voir l'analyse des ventes, des commandes satisfaites et des produits du marché, des produits vendus et celles qui sont en pré-rupture.
- Accès à l'ensemble des commandes et des précommandes enregistrées au paravent.
- Avoir accès aux données et aux achats des clients.
- Attribuer des privilèges aux clients (VIP, Passager...).
- Accès au stock qui lui permettra d'alerter le gestionnaire de stock en cas de rupture d'un des produits.
- Accès à l'historique des activités journaliers.
- Accès à ses propres informations et possibilité de les modifier.

## II.5.2 Exigences fonctionnelles

Dans cette partie nous allons présenter les besoins fonctionnels liés au bon déroulement de l'application et auxquelles doit répondre notre système et qui sont illustrer ci-dessous :

1. Interface utilisateur conviviale : une interface facile à utiliser pour les clients, avec des fonctionnalités intuitives pour parcourir les produits, ajouter des articles a un ticket (bon de commande).
2. Recherche et filtrage de produits : la possibilité de rechercher des produits en utilisant des termes de recherche pertinents, ainsi que la capacité de filtrer les produits en fonction de critères tels que le prix, la catégorie, etc...
3. Recommandations de produits : une fonctionnalité qui recommande des produits en fonction des préférences du client, des achats précédents et des tendances du marché.
4. Caractéristiques de produit détaillées : une fiche produit détaillée pour chaque produit, incluant des images, une description, ainsi que des informations sur les options de personnalisation (promotion, réduction...).

5. Options de paiement : différentes options de paiement, y compris la carte de crédit, le virement bancaire et le paiement en espèces.
6. Analyse des données de vente : des outils d'analyse de données pour suivre les performances de vente, les tendances du marché et les préférences des clients pour aider à améliorer les stratégies de vente.
7. Mode déconnecté : la capacité de continuer à utiliser l'application en mode déconnecté pour permettre aux clients de commander des produits sans accès à internet.
8. Gestion de stock : la capacité de suivre le stock de produits pour éviter les commandes en double et s'assurer que les produits disponibles sont correctement mis à jour.
9. Gestion des clients : la capacité de suivre les informations des clients, incluant les informations de contact, les achats précédents et les préférences qui semble être utile pour les entreprises pour pouvoir augmenter leurs chiffres d'affaires.
10. Suivi de commandes : la capacité de suivre les commandes, et d'enregistrer l'historique des commandes pour chaque client.
11. Gestion des propositions de commandes : la possibilité de suivre les propositions de commande aux clients, de les enregistrer dans le système et de les traiter.

### **II.5.3 Exigences non-fonctionnels**

Ce sont des exigences qui semblent être facultatifs et à la fois primordiales pour l'amélioration du système en spécifiant des critères de jugement de la qualité du produit qui sont détaillés ci-dessous :

#### **1. Sécurité**

- L'application doit être sécurisée en garantissant la protection des informations des clients et des utilisateurs, des transactions financières et des données de l'entreprise en les sauvegardant les données de manière régulière.
- L'application doit régulièrement se mettre à jour avec les derniers correctifs pour s'assurer que les vulnérabilités sont corrigées.
- L'application doit avoir une méthode d'authentification sécurisée pour permettre aux utilisateurs de s'identifier, en mettant en place des rôles et des permissions pour contrôler l'accès aux fonctionnalités de l'application.

## 2. Disponibilité

- L'application doit être disponible en tout temps pour permettre aux utilisateurs et aux clients de l'utiliser à tout moment.

## 3. Ergonomie

- L'application doit être ergonomique et facile à utiliser pour les utilisateurs, avec une interface conviviale et intuitive.

## 4. Fiabilité

- L'application ne doit contenir aucun bug ou erreur qui pourrait compromettre l'expérience utilisateur.
- Le système doit être précis ces fonctionnalités même avec une faible connexion.

## 5. Performance

- L'application doit être capable de gérer un grand nombre d'utilisateurs et de requêtes, transactions en même temps, sans ralentir ni perdre de données afin de garantir une expérience utilisateur optimale et d'éviter les pertes de temps inutiles pour les clients.
- L'application doit garantir le concept du multi commandes.

## 6. Maintenabilité

- L'application doit être facile à maintenir et à mettre à jour, afin de résoudre rapidement les problèmes et d'ajouter de nouvelles fonctionnalités.

## 7. Évolutivité

- L'application doit être capable de s'adapter à la croissance de l'entreprise et de répondre aux besoins futurs des utilisateurs.

### II.5.4 Exigences reportées

Les exigences ci-dessous sont des fonctionnalités qui pourraient ne pas être implémentés pour multiple raison et vont pas être disponibles dans la première version :

- Paiement en ligne et le virement bancaire.
- La recommandation de produits selon les anciens achats du client et ses préférences.
- La gestion des propositions de commandes. (Voir exigences fonctionnels).
- Mode Déconnecté en raison d'utilisation d'API nécessitant internet.

## II.2 Conclusion

En conclusion, ce chapitre nous a permis de présenter les différentes parties prenantes de notre projet, ainsi que les objectifs et les besoins de notre application de vente assistée. Nous avons souligné l'importance d'une interface conviviale, d'une recherche et d'un filtrage efficaces, ainsi que d'un mode déconnecté pour améliorer l'expérience d'achat en magasin pour les clients. De plus, nous avons identifié les acteurs en interaction avec le système et recueilli leurs différentes exigences auxquelles l'application doit répondre.

Dans les chapitres suivants, nous aborderons en détail les étapes de conception et de réalisation de ce projet, en mettant l'accent sur les multiples fonctionnalités et leurs modélisations. Cela nous permettra de développer une vision complète du système, en prenant en compte les besoins des utilisateurs et en concevant une solution adaptée.

# **Chapitre II :**

## **Conception du système de vente assistée**

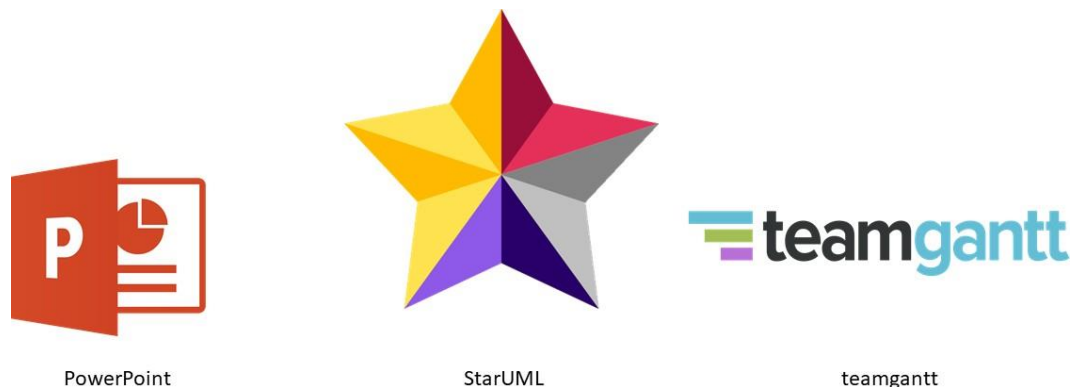


### III.1 Introduction

Dans ce chapitre, nous allons modéliser les différentes fonctionnalités réalisées par les acteurs en interaction avec l'application en utilisant les multiples diagrammes UML<sup>5</sup> dont nous avons besoin.

Notons que la réalisation de ces tâches et celles d'avant ont été réalisées suivant des sprints et des délais qui vont être illustrés grâce à un diagramme de GANTT<sup>6</sup>.

### III.2 Outils utilisés



*Figure II : Logos des outils utilisés*

#### III.2.1 PowerPoint

PowerPoint fait partie de la suite bureautique Microsoft Office, il permet de réaliser des présentations sous forme de diapositives, d'images..., il fonctionne sous Windows, Linux et MacOs et c'est l'un des logiciels les plus utilisés au monde [7].

#### III.2.2 StarUML

C'est un outil de génie logiciel dédié à la modélisation UML et édité par la société coréenne MKLabs. Il est multiplateforme et fonctionne sous Windows, Linux et MacOs [8].

<sup>5</sup> UML le langage de modélisation unifié, c'est une synthèse de langages de modélisation objet antérieurs à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines de développement logiciel et de conception orientée objet [5].

<sup>6</sup> Gantt est un diagramme de gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes tâches qui constituent un projet [6].

### III.2.3 Teamgantt

C'est un logiciel de collaboration et de planification de projet pour les équipes. Il combine diagrammes de Gantt intuitifs, gestion des tâches, communication au niveau des tâches, partage de fichiers et suivi de temps [9].

### III.3 Répartition des tâches (diagramme de Gantt)

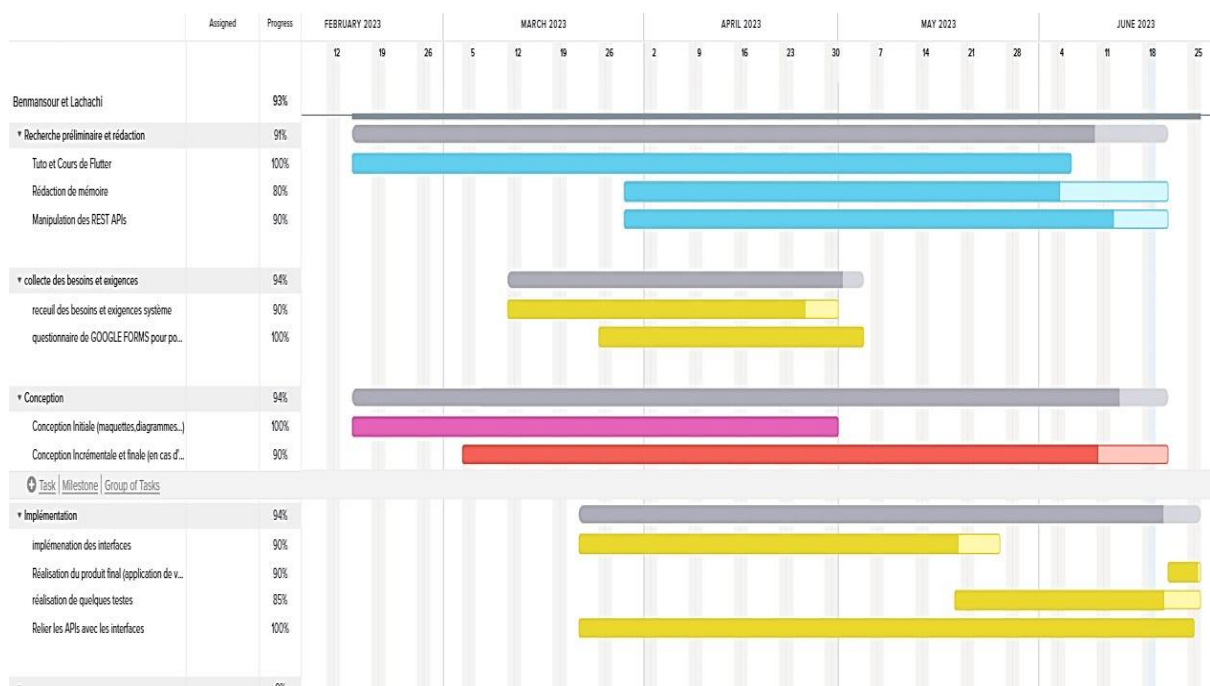


Figure III : diagramme de Gantt

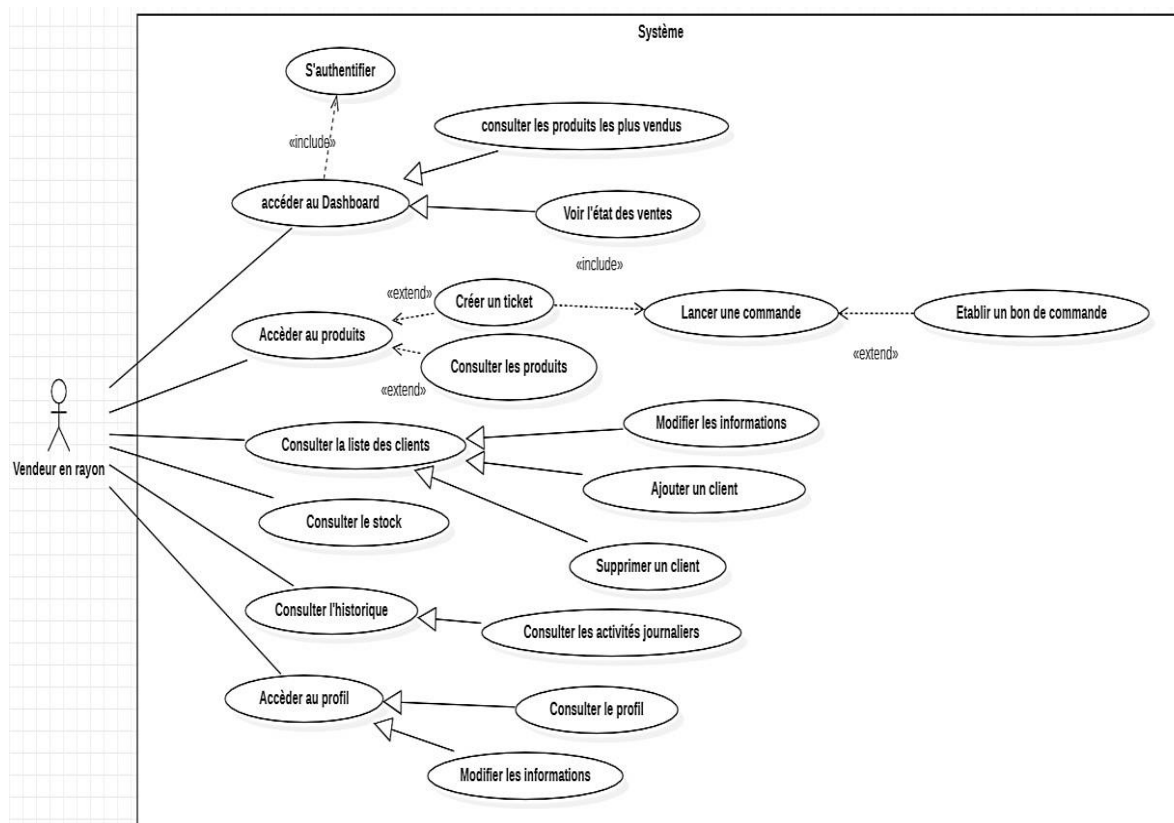
### III.4 Modélisation des fonctionnalités

Dans cette section nous allons modéliser les fonctionnalités majeures à travers les différents diagrammes.

#### III.4.1 Diagrammes de cas d'utilisation

##### III.4.1.1 Diagramme de cas d'utilisation « vendeur en rayon » ou « chargé (e) de rayon »

Ci-dessous la figure IV qui représente les multiples fonctionnalités du « Vendeur en rayon » ou bien « Chargé (e) de rayon ».



*Figure IV : Diagramme de cas d'utilisation du vendeur*

Après avoir effectué l'authentification qui est une étape indispensable et cruciale pour pouvoir passer aux autres fonctionnalités, le vendeur fera l'objet d'intermédiaire entre le système et le client en accomplissant les tâches désirées par ce dernier ainsi que les siennes pour faciliter la communication et les transactions entre le système et le client.

Donc le vendeur peut accéder au Dashboard pour pouvoir consulter les différents statistiques qui reflète l'état de ses ventes et qui offre une liste détaillée des produits les plus vendus et aussi de celles qui sont sur le point d'épuiser. Il pourra ainsi accéder aux différents produits disponibles dans le stock pour pouvoir soit satisfaire l'un des commandes désirées par un client ou juste consulter les produits.

Le vendeur accède à la liste des clients qui pourra ainsi les consulter, ajouter un client, supprimer un client ou modifier ses informations.

Il pourra consulter le stock ainsi que l'historique afin de voir l'ensemble de ses activités journalières (ajout, suppression d'un produit ou d'un client ...).

Et à la fin il pourra accéder à son profil pour consulter ses coordonnées ou de pouvoir les modifier.

### III.4.1.2 Diagramme de cas d'utilisation d'un « Fournisseur »

Ci-dessous dans la figure les multiples fonctionnalités d'un fournisseur

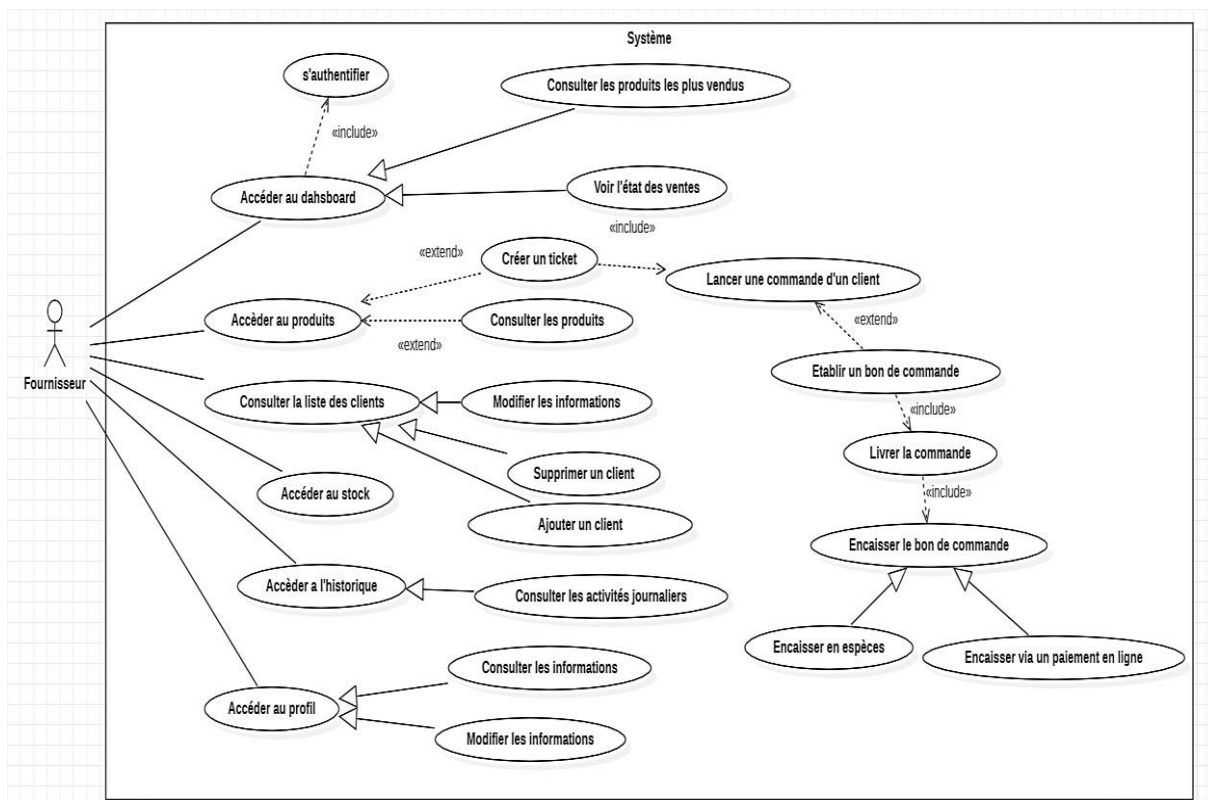


Figure V : Diagramme de cas d'utilisation du fournisseur

Une fois l'authentification a été effectué, le fournisseur joue le rôle d'intermédiaire entre le client et le système. Sa principale responsabilité est d'accomplir les tâches demandées par le client, en plus de celles qui sont spécifiques à sa fonction. Ceci permet de faciliter les échanges et les transactions entre les deux parties.

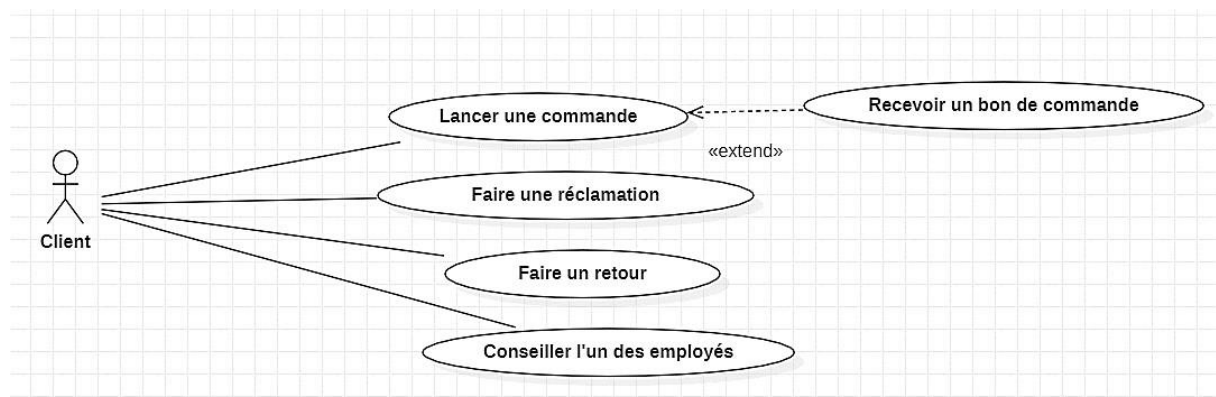
En utilisant le tableau de bord, le fournisseur peut avoir accès à des statistiques détaillées sur les ventes ainsi que sur les produits disponibles dans le stock. Il peut ainsi voir les produits les plus populaires et ceux qui risquent de s'épuiser bientôt. Cette fonctionnalité permet au fournisseur de mieux répondre aux demandes du client et de mieux satisfaire la commande qui lui sera livré après avoir reçu le bon de commande et acquitter le paiement.

Le fournisseur peut également accéder à la liste des clients pour consulter, ajouter, supprimer ou modifier leurs informations. Il peut effectivement accéder aux données de stock et a l'historique pour voir ses activités quotidiennes telles que l'ajout ou la suppression d'un client....

Le fournisseur peut consulter et modifier ses propres coordonnées en accédant à son profil.

### III.4.1.3 Diagramme de cas d'utilisation d'un « Client »

Ci-dessous dans la figure les principales fonctionnalités d'un client



**Figure VI : Diagramme de cas d'utilisation du client**

Avec un nonaccès à l'application le client pourra lancer une commande auprès de l'un des deux autres acteurs qui sera en cas concrétisation validé par un bon de commande. Il pourra aussi faire une réclamation pour revendiquer ses droits en cas de violation ou de négligence de la part des employés auprès de leurs supérieurs.

Le client pourra aussi faire un retour dans le cas d'un endommagement du produit désiré ou dans le cas de la non-correspondance du produit par rapport à la qualité ou à la marque.

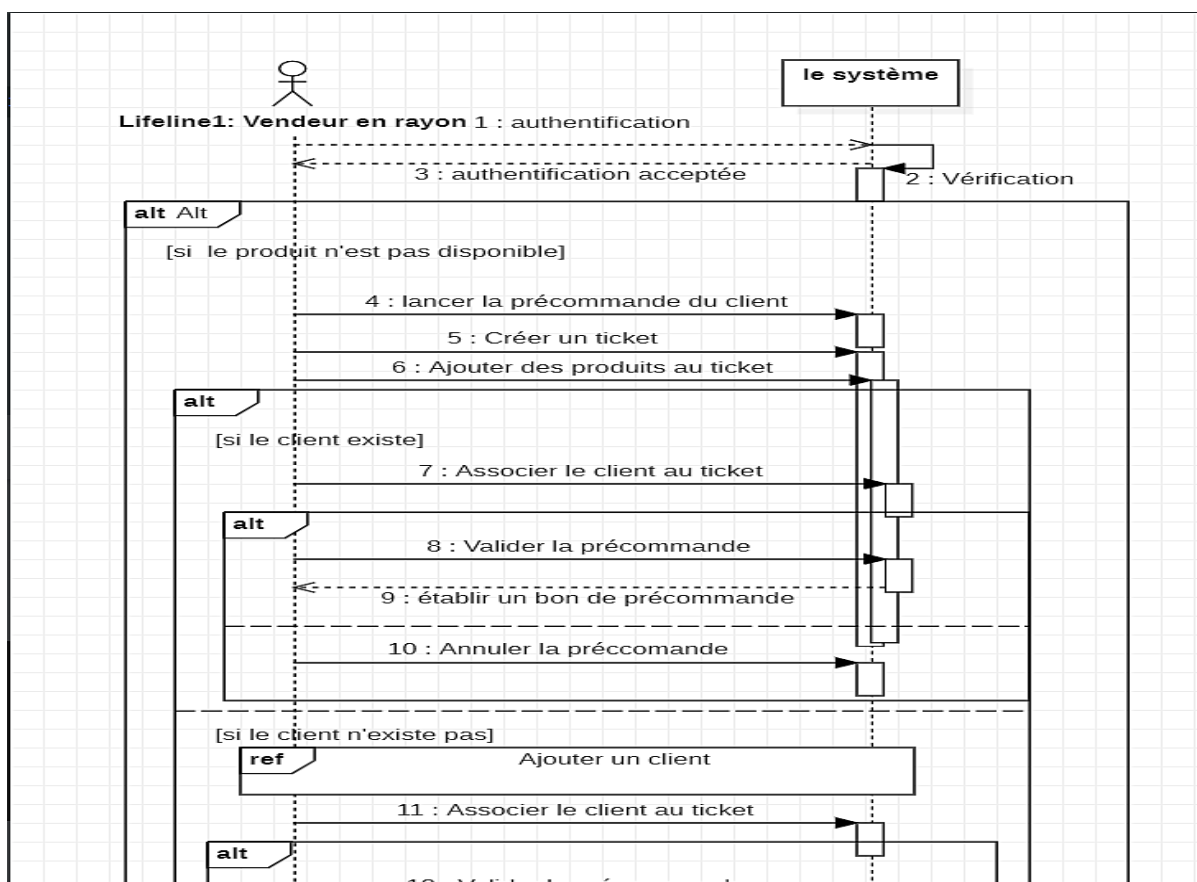
Le client pourra conseiller les employés d'acquérir un des produits non disponibles ou de qualité supérieur selon ses besoins afin d'avoir une large variété de choix.

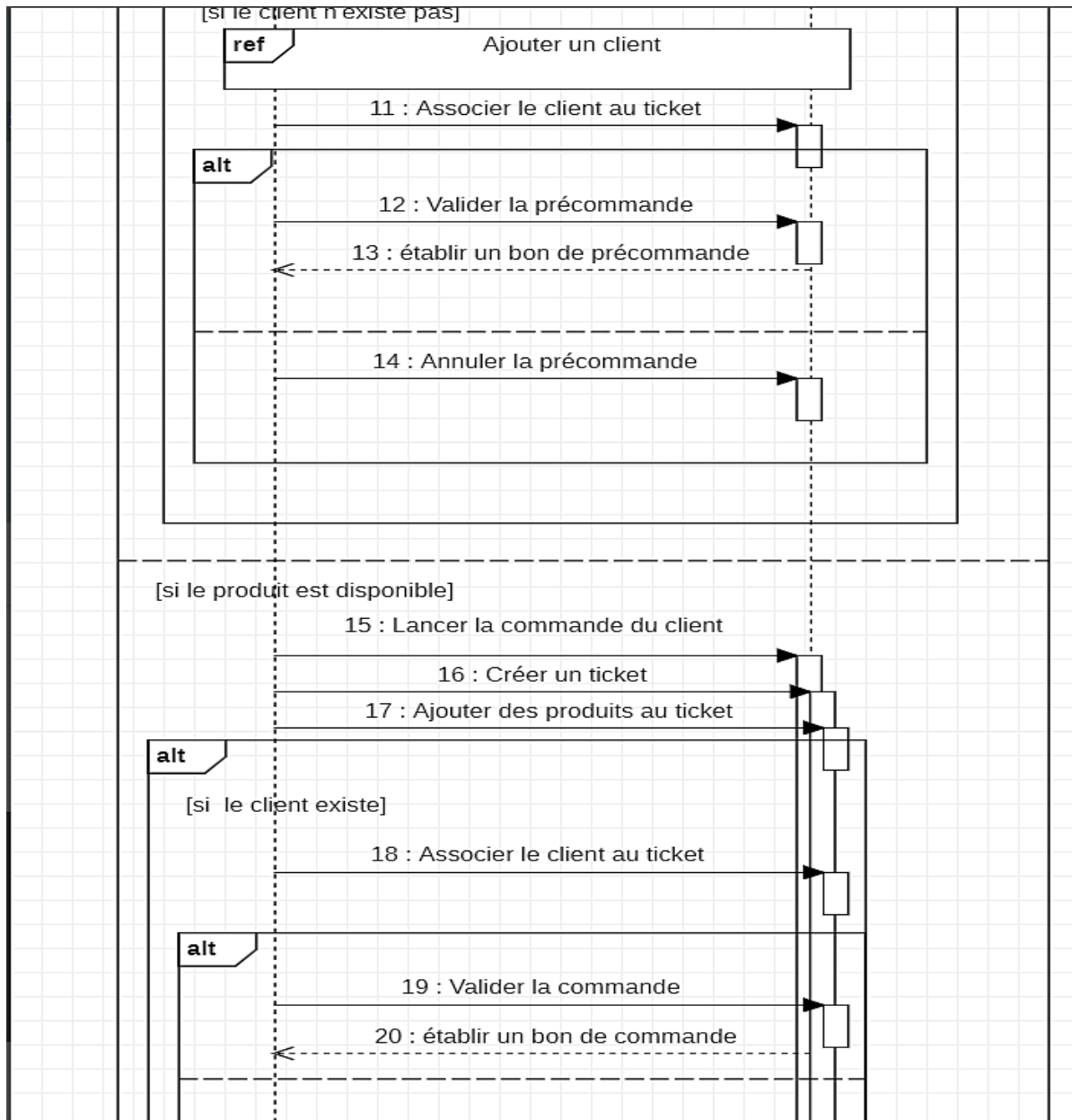
## III.4.2 Diagrammes de séquence

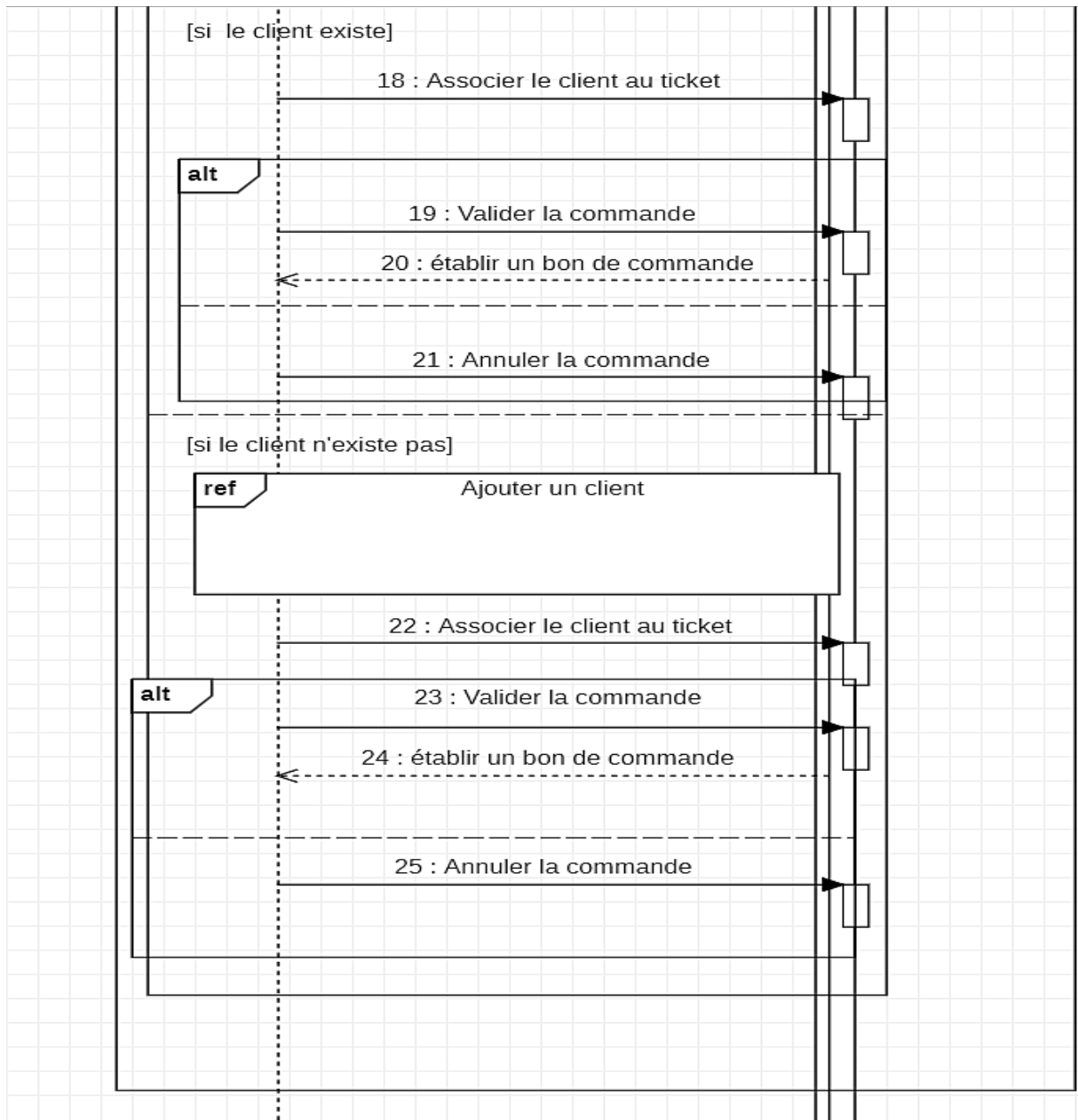
### III.4.2.1 Diagramme de séquence « Lancer une commande »

Ci-dessous la figure qui représente les différentes étapes d'un déroulement de commande faite par un vendeur en rayon.

Remarque : en raison de longévité du diagramme nous l'avons décomposé en 3 figures.





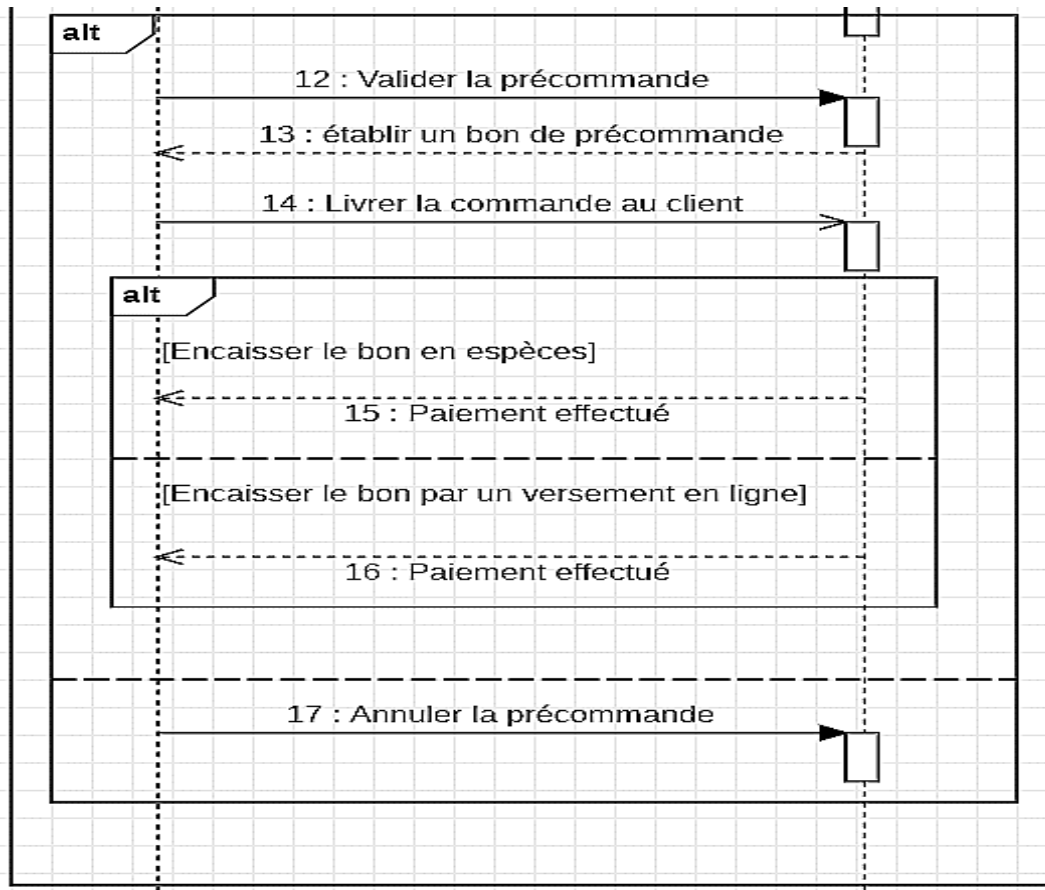


**Figure VII: Diagramme de séquence "lancer une commande" pour un vendeur**

Après que le vendeur ait saisi ses coordonnées d'authentification (identifiant, mot de passe) avec succès. Cet utilisateur aura accès aux fonctionnalités, en particulier à l'interface des produits, où il pourra créer un ticket pour le client en cas de non-disponibilité des produits. Ce ticket sera associé aux produits souhaités par le client, et sera considéré comme une précommande. En revanche dans le cas où les produits sont disponibles il répétera les mêmes étapes qui amèneront à un bon de commande qui sera remis au client et sera considérée comme une commande et sera livré instantanément par un chargé de stock ou magasinier et cela dans

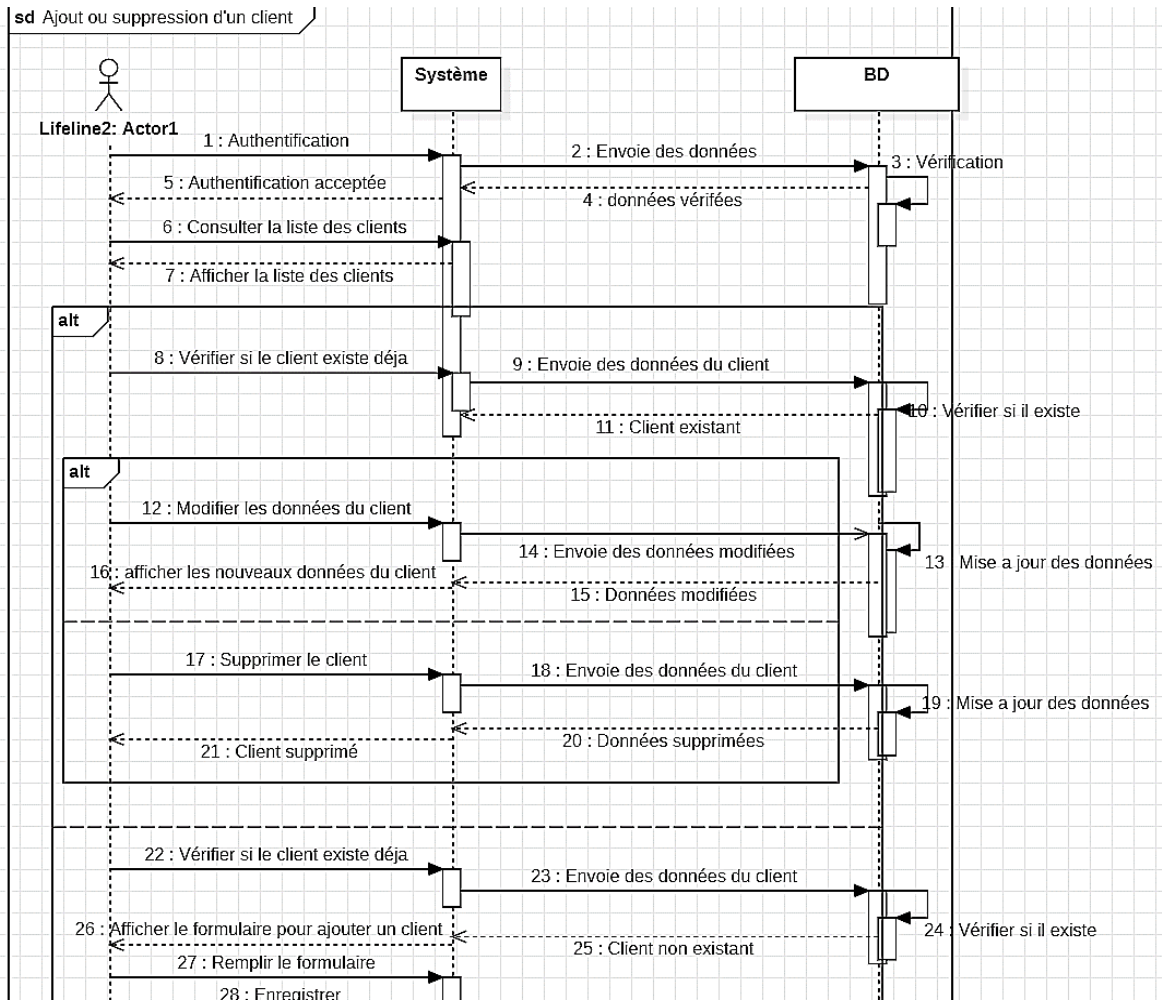


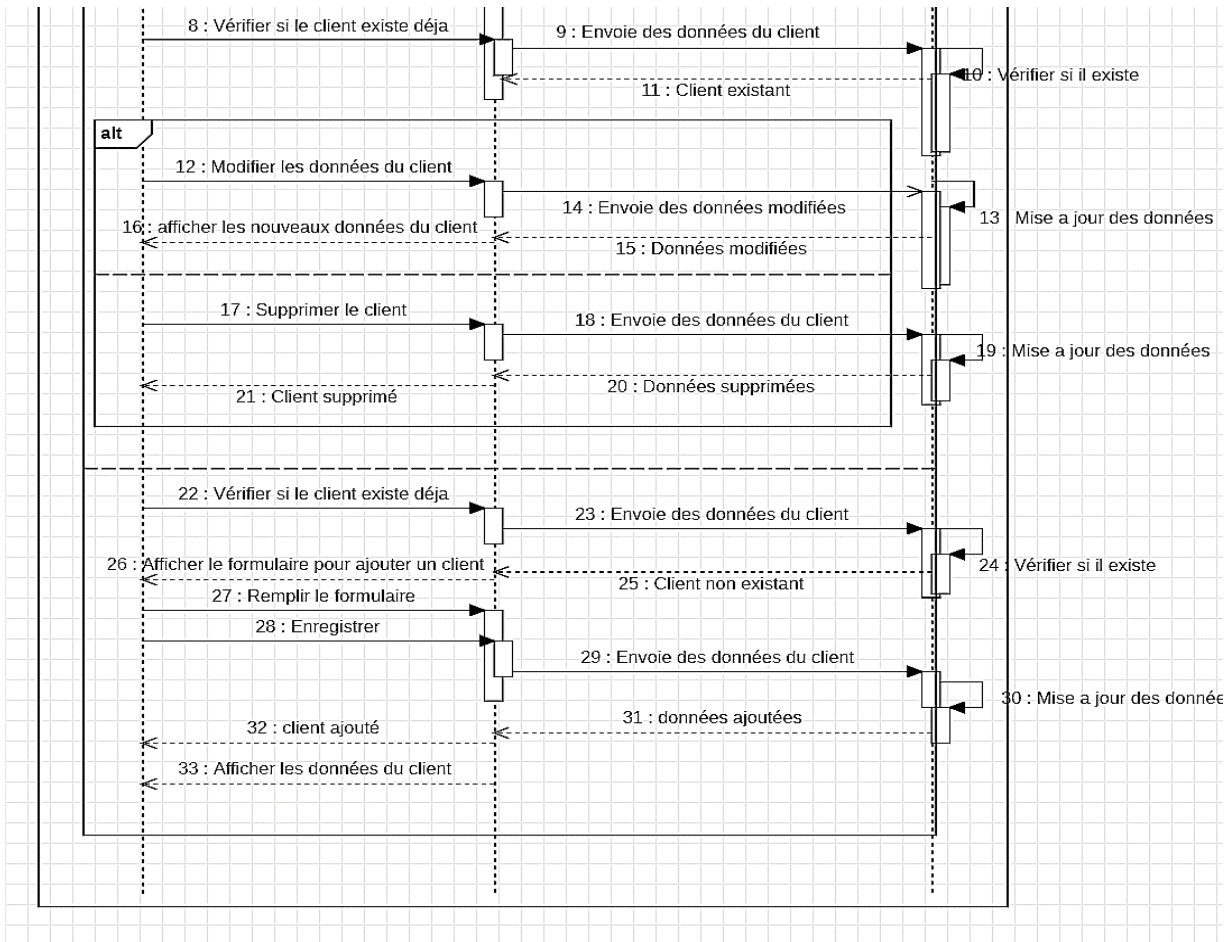
le cas d'un vendeur en rayon. Par contre si le fournisseur prendra le relais et lancera la commande du client, il effectuera aussi le même processus différant après la remise du bon de commande qui a dans ce cas c'est lui qui prendra en charge la livraison du produit après avoir été payé d'une des deux manières disponibles soit en espèces ou par un versement en ligne. (Cette section est illustrée dans la figure suivante)



### III.4.2.2 Diagramme de séquence « ajout ou modification d'un client »

Ci-dessous la figure qui illustre le processus d'ajout ou de modification des données d'un client réalisé par un utilisateur.



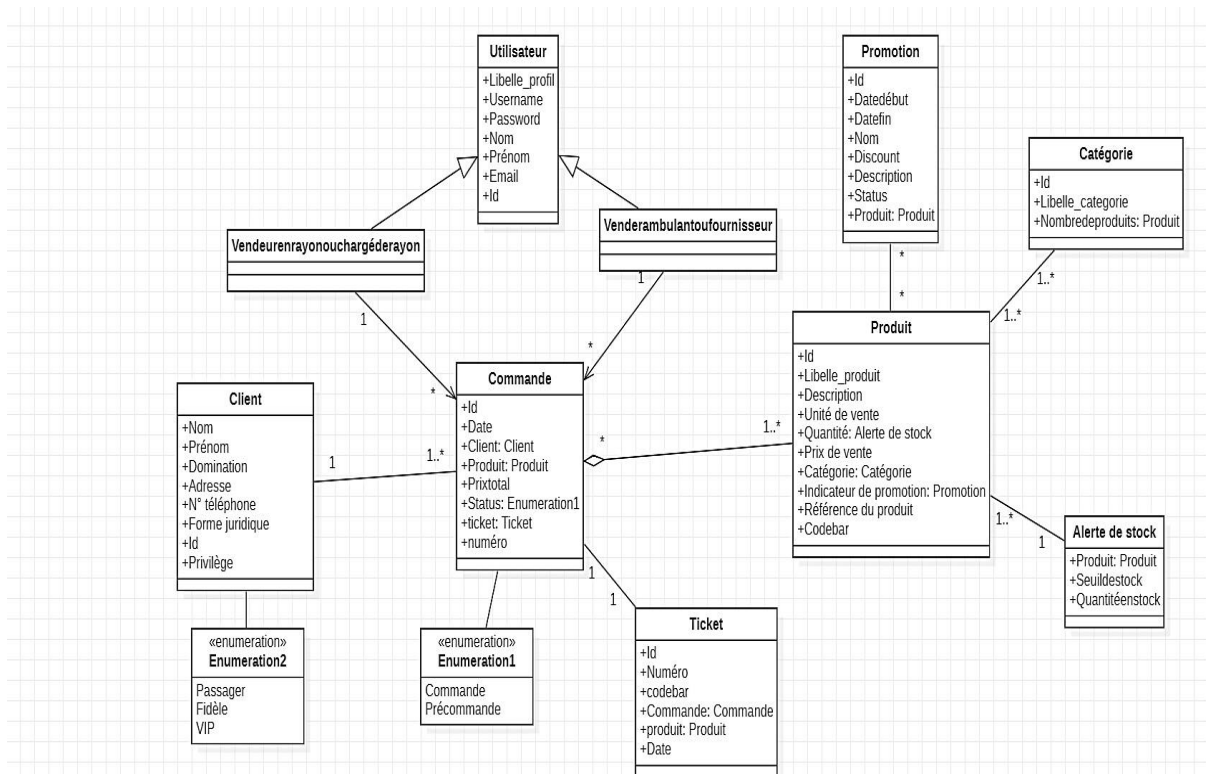


**Figure VIII : Diagramme de séquence "lancer une commande" pour un fournisseur**

Après avoir effectué l’authentification l’utilisateur peut consulter la liste des clients, en premier lieux il va vérifier si le client est déjà existant ou il pourra soit modifier les informations de ce dernier ou bien supprimer ces données carrément. Dans l’autre cas ou le client n’est pas existant, un formulaire apparaîtra ou il remplit les données de client, il pourra ensuite enregistrer ses informations qui vont être effectivement stockés dans la base donnée afin qu’il puisse les consulter ou les modifier ultérieurement.

### III.4.3 Diagrammes de classe

Ci-dessous la figure qui représente les multiples entités et les relations entre eux.



**Figure IX: Diagramme de classe**

D'après l'étude du système existant et des différents diagrammes nous constatons la présence de trois acteurs comme suit :

- **Vendeur en rayon et fournisseur** tous définis par un username, password, nom, prénom, l'email et l'id.
- **Client** qui est défini par un nom, un prénom, une adresse, une forme juridique, un numéro de téléphone, un id et des privilèges et une domination.

Et ce qui concerne les classes (entités) nous avons réussi à extraire 7 classes et 2 énumérations comme suit :

- Classe commande : qui possède un id, une date de création, un prix total, des produits et un client et un statut pour pouvoir différencier le type de commande (commande ou précommande) représenté par une énumération (énumération1), et elle sera déclenché au moment de la création d'un ticket.
- Classe Ticket : représente les détails d'une commande en ajoutant aussi un numéro de ticket, un code bar, et sa date de création, et il sera remis en cas de passation de commande pour permettre au client de la recevoir.

- Classe client : elle contient toutes les informations nécessaires pour définir un client avec un id, un prénom, une adresse...et aussi des privilèges pour les catégoriser soit VIP, passager....
- Classe produit : elle contient de multiples détails liés à un produit, un id, un libelle, une description...une quantité, la catégorie auquel il appartient et un indicateur de promotion et de remise.
- Classe catégorie : elle englobe une large liste de produits avec un id, un nom, et le nombre exact de produits dans chaque catégorie.
- Classe promotion : elle permet au client de savoir le taux de réduction sur un produit exact selon ses privilèges avec une description et le délai de validité de l'offre.
- Classe alerte de stock : ou bien stock, elle permettra aux utilisateurs de pouvoir connaître la quantité restante de chaque produit avec un seuil fixe.

Ces données et ces classes ont été définies grâce aux différents diagrammes de cas d'utilisation et un modèle logique de données (MLD<sup>7</sup>) de la version « Desktop » qui nous été transmis par les développeurs de l'entreprise subissant des modifications selon nos besoins et les exigences.

Ci-dessous la figure du MLD pour une compréhension assez claire et distincte du système.

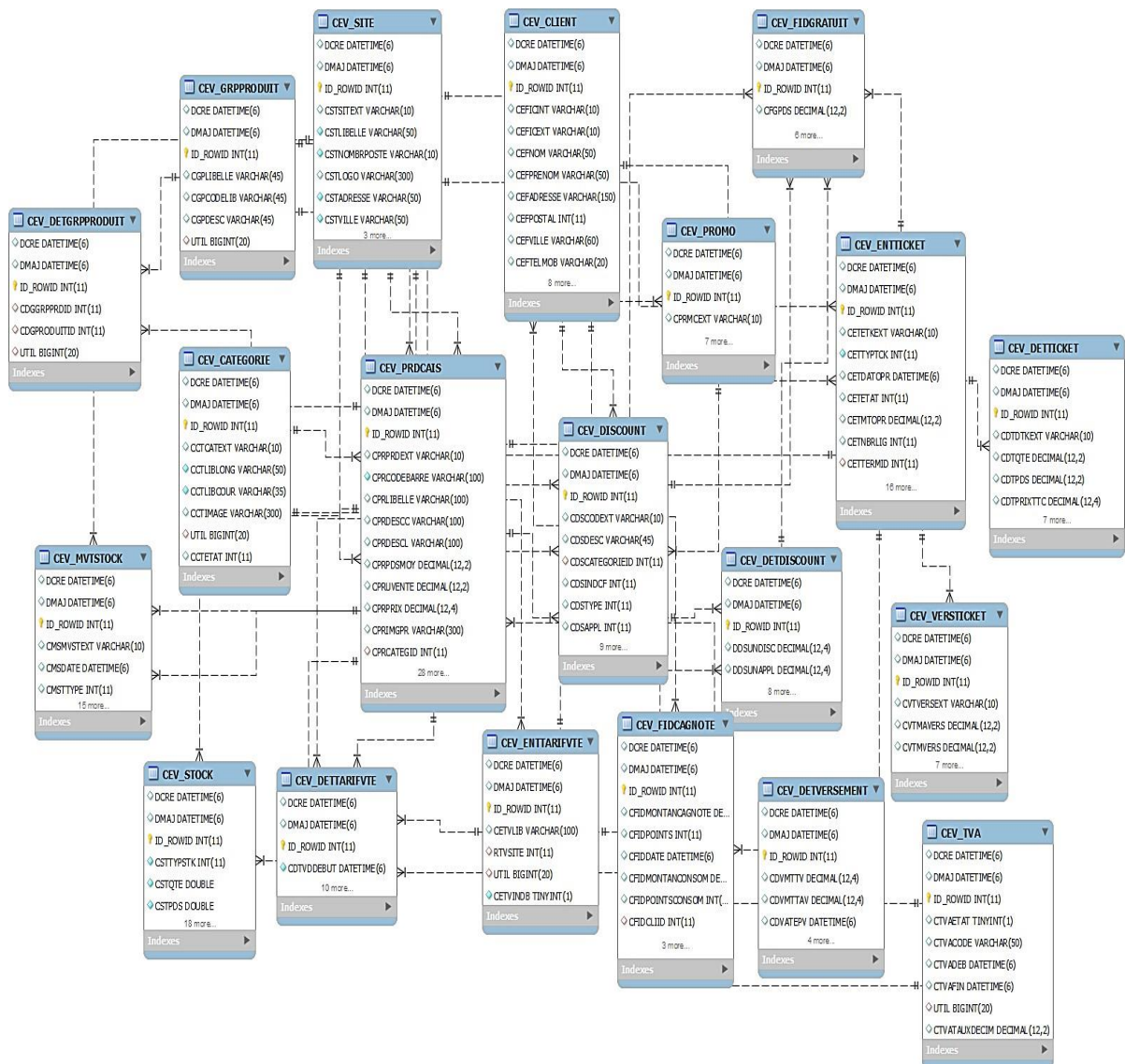


Figure X: Modèle logique de données (MLD)

<sup>7</sup> MLD le modèle logique de données établit la structure des éléments et les relations entre elles. Il est indépendant de la base de données physique qui détaille la manière dont les données seront mises en œuvre. Le MLD va plus loin que les éléments de la modélisation conceptuelle des données en leur ajoutant des informations supplémentaires [10].

### **III.5 Conclusion**

Ce chapitre nous a permis de définir les différents outils utilisés durant le cycle de conception, ainsi que la répartition des tâches et la gestion du projet. Et de définir les multiples fonctionnalités de l'application à travers les diagrammes de modélisation.

Dans le chapitre suivant, nous aborderons les étapes de la phase de développement, en mettant l'accent sur les bonnes pratiques et les méthodes qui permettent d'assurer la qualité et la fiabilité du code ainsi que l'application. Et on clôtura le chapitre avec une simulation détaillée des multiples rubriques de l'application.

**Chapitre III :**  
**Réalisation de l'application**  
**«Vente Assistée »**



## IV.1 Introduction

Dans ce dernier chapitre nous allons tout d'abord commencer par la description de notre environnement de travail, ainsi que les différents langages et technologies utilisés au cours de l'implémentation et conception du système.

Par la suite, nous allons aussi illustrer le fonctionnement de notre système grâce aux différentes interfaces de l'application.

## IV.2 Outils et Technologies utilisés



*Figure XI : Liste des outils et technologies utilisés*

### IV.2.1 Les outils utilisés

#### ➤ Visual Code Studio

Visual Code Studio est un éditeur de code simplifié, qui est gratuit et développé en open source par Microsoft. Il fonctionne sous Windows, mac OS et Linux. Il fournit aux développeurs à la fois un environnement de développement intégré [11].

Il propose de nombreuses fonctionnalités, notamment la prise en charge de centaines de langages de programmation, telles que C, DART, C++, PYTHON, HTML, CSS, JSON... ainsi que d'autres fonctionnalités comme le débogueur, la complétion du code.

➤ **Android Studio**

Android Studio est un environnement de développement intégré (IDE) officiel pour le développement d'application Android, basé sur le logiciel JetBrains IntelliJ IDEA<sup>8</sup>. Il utilise un système de construction basé sur Gradle, des émulateurs permettant de lancer une application Android virtuel sur un ordinateur, des modèles de code JAVA/Kotlin et d'autres fichiers de configuration XML et l'intégration GitHub [13].

➤ **GitHub**

GitHub est un service d'hébergement Open-Source utilisant le logiciel de gestion de versions Git<sup>9</sup>, permettant aux programmeurs et aux développeurs de partager le code informatique de leurs projets afin de travailler dessus de façon collaborative. On peut le considérer comme un Cloud dédié au code.

Il permet aussi de contrôler les différentes versions d'un système permettant aux utilisateurs de modifier le code sans toutefois impacter directement le système ou l'expérience des utilisateurs actuels [15].

➤ **Swagger**

Swagger est un langage de description d'interface permettant de décrire des API REST exprimées à l'aide de JSON. Il est utilisé avec toute une série d'outils logiciels open source pour concevoir, créer, documenter et utiliser des services web.

Il permet aussi de générer du code afin de pouvoir tester les APIs [16].

➤ **MySQL Workbench**

MySQL Workbench est un outil de conception visuelle base de données disponibles sous Windows, Mac et Linux [17] qui intègre le développement de logiciels, l'administration des bases de données, la conception, la création et la maintenance de bases de données pour le système de BDD MySQL [18].

<sup>8</sup>JetBrains est un éditeur de logiciels pour développeurs parmi eux IntelliJ IDEA est un environnement de programmation spécial ou un environnement de développement intégré (IDE) largement destiné à JAVA. Cet environnement est utilisé notamment pour le développement de programmes [12].

<sup>9</sup>Git est un système de contrôle de versions et de développement inventé en 2005 qui aide une équipe de développeurs à gérer les changements apportés au code source au fil de temps en utilisant des lignes de commandes [14].

## IV.2.2 Les outils et technologies utilisés coté serveur (Backend)

### ➤ Django

Django est un Framework python open-source publié sous licence BSD consacré au développement web 2.0 considéré comme le Framework web le plus perfectionné et orienté pour la production des projets solides rapidement.

Il s'inspire du modèle MVT<sup>10</sup> (variation du modèle MVC) c'est-à-dire la structure du Framework sépare les données (Model, Vue/Template, Controller) avec un moteur de template efficace souple et facile à utiliser, un routeur qui permet de diriger les actions en fonction des URL et une API permet de fournir des informations sur le projet sans pouvoir passer par le cas SQL [20].

### ➤ MySQL

MySQL (Structured Query Language) est un serveur de gestion BDD relationnels sous licence GNU. Il la plupart des temps intégré dans la suite des logiciels LAMP qui comprend un système d'exploitation (Linux), un serveur web (Apache) et un langage de script (PHP).

Il distribué sous double licence GPL et propriétaire et il est peut-être utilisé sur de nombreux systèmes d'exploitation (Windows, Mac OS, etc.). Il supporte les langages informatique SQL et SQL/PSM [21].

Nous soulignons que le Backend a été développé par les développeurs de DATAMASTER en utilisant les technologies et les langages cités ci-dessus.

<sup>10</sup>Le MVT (Model View Template) est un patron de conception logiciel. Il s'agit d'une collection de trois composants importants qui aident à générer la base de données et elle aussi une couche d'accès aux données qui gère les données [19].

### IV.2.3 Les outils et technologies utilisés coté client (Frontend)

#### ➤ Flutter

Flutter est un kit de développement logiciel (SDK) d'interface utilisateur open-source créé par Google. Il est utilisé pour développer des applications pour Android, Ios, linux, Mac, Windows et le web à partir d'une seule base de code [22].

Il est composé d'une plateforme DART, un moteur Flutter, une bibliothèque foundation et les widgets spécifiques à la conception [23].

### IV.3 Architecture du système

Dans cette section, nous allons voir de manière globale l'architecture de notre application en citant trois aspects :

- L'architecture du Backend conçue grâce au Framework Django.
- L'architecture du Frontend conçue grâce SDK Flutter.
- L'architecture des APIs permettant de relier le Backend et le Frontend.

#### IV.3.1 L'architecture du backend

Composée de :

- **Modèle (Model)** : le modèle représente la structure des données de l'application. Il définit la logique métier et les règles de validation des données. Dans Django, les modèles sont généralement définis à l'aide de classes Python qui héritent de la classe 'django.db.models.Model'. Les modèles sont responsables de l'interactions avec la base de données et de la gestion des opérations de lecture, d'écriture et de manipulation des données.
- **Contrôleur (Controller)** : Dans Django, le contrôleur est généralement représenté par les vues (views). Les vues traitent les requêtes HTTP, interagissent avec les modèles pour récupérer les données nécessaires, et génèrent les réponses correspondantes.
- **Vue (View)** : la vue est responsable de la logique de traitement des requêtes http et de la génération des réponses correspondantes. Ils sont généralement des fonctions ou des classes qui prennent en entrée une requête http et renvoient une réponses http.

Ils interagissent avec les modèles pour récupérer les données nécessaires et utilisent les templates pour générer les réponses http ou les formats des données souhaités tels que JSON.

- **Template (Template) :** les templates sont des fichiers qui définissent la présentation et la structure des réponses HTML générées par les vues. Ils permettent de séparer la logique de présentation du code de la vue. Elle utilise généralement la syntaxe Django Template Language (DTL) pour incorporer des variables, des boucles, et des conditions.
- **URLs :** les URLs conf de Django permettant de mapper les URL entrants aux vues correspondants. Elles définissent les règles de routage pour diriger les requêtes vers les vues appropriées.
- **Middleware :** le middleware de Django est une couche intermédiaire qui peut traiter les requêtes et réponses avant qu'elles n'atteignent les vues. Il est utilisé pour effectuer des opérations communes, telles que l'authentification, la gestion des sessions, la compression, etc.
- **Serializers :** les sérialiseurs de Django permettent de convertir les objets Python en représentation de données telles que JSON, XML, etc., et vice versa. Ils sont souvent utilisés pour sérialiser et désérialiser les données lors des opérations d'API.

## IV.3.2 L'architecture du frontend

- **Modèle (Model) :** le modèle représente les données de l'application, ainsi que la logique qui leur est associé. Cela peut inclure la gestion des API, la manipulation des données, la persistance des données, etc. Le modèle est généralement indépendant de l'interface utilisateur et peut être partagé entre différentes parties de l'application.
- **Vue (View) :** la vue est responsable de l'affichage de l'interface utilisateur à l'aide de widgets. Cependant, plutôt que d'interagir directement avec le modèle, la vue va interagir avec le VueModèle (ViewModel) pour obtenir les données nécessaires, elle va également écouter les changements dans le VueModèle et mettre à jour l'interface utilisateur en conséquence.
- **VueModèle (ViewModel) :** la VueModèle agit comme une couche intermédiaire entre la vue et le modèle. Il contient l'état de l'interface utilisateur ainsi que la logique métier associé. Le VueModèle est responsable de la récupération des données à partir du modèle, de leur transformation si nécessaire, et de les fournir à la vue via des objets observables. Il utilise le package Provider pour gérer l'injection de dépendances et la gestion global de l'application.
- **Provider :** le package Provider est utilisé pour gérer l'état global de l'application et pour fournir les instances de VueModèle aux vues qui en ont besoin. Il permet de partager facilement des données entre les différentes parties de l'application sans nécessiter de dépendances directes.
- **Widgets :** un widget est un élément logiciel qui représente une partie spécifique de l'interface utilisateur. Il peut s'agir d'une visualisation d'informations, d'un composant interactif ou d'un conteneur pour d'autres widgets et on distingue deux types : Stateless et Stateful.

## IV.3.3 Autres technologies utilisées

- **API REST (Representational State Transfer) :** Une API REST (également appelée API RESTful) est une interface de programmation d'application (API ou API web) qui respecte les contraintes du style d'architecture REST et permet d'interagir avec les services web RESTful. Ça été créée par l'informaticien Roy Fielding [24].

#### IV.3.4 Tests et performance

- **Performance** : en ce qui concerne la performance nous avons opté pour des widgets simples Stateless quand il est statique et Stateful quand -il change en évitant la complexité. On a aussi optimisé le rendu en utilisant des widgets telles que les listes ou les grilles (ListView, GridView), ainsi que l'utilisation de quelques packages pour faciliter l'accès et rendre l'application plus ergonomique et responsive.
- **Tests** : cette étape a été faite selon les différentes fonctionnalités, premièrement en ce qui concerne les APIs ils ont été testés grâce au logiciel **Swagger** pour pouvoir savoir si les APIs répondent aux différentes requêtes que ça soit complexes ou basiques. Deuxièmement, nous avons testé la disposition des différents éléments et widgets de l'interface dans l'écran de l'émulateur et donc cela permettra de percevoir tout emplacement ou comportement non désiré et de les corriger.

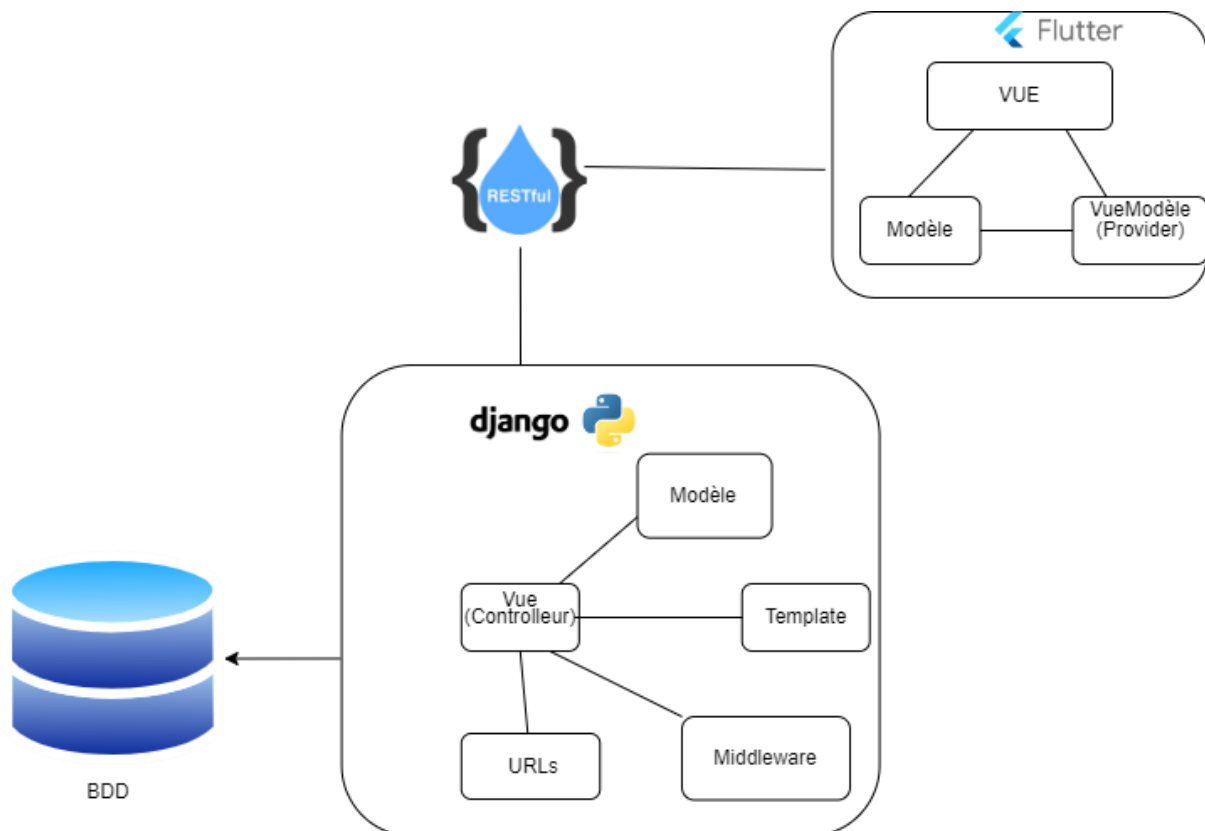


Figure XII : Architecture de l'application

## IV.4 Déploiement de l'application

En ce qui concerne le déploiement de l'application il suffit de générer un APK selon des exigences et des conditions :

- Pour générer un APK pour les appareils Android 64 bits uniquement '**flutter build apk --split-per-abi**'.
- Pour générer un APK avec le profil de release (optimisé pour les performances) '**flutter build apk --release**'.
- Pour générer un APK avec un nom spécifique '**flutter build apk --output=« nom\_du\_fichier ». apk**'.

Ces APKs peuvent être installés sur n'importe quel appareil selon ses caractéristiques et dans notre cas nous avons opté pour l'APK release idéale pour une tablette qui fera l'objet d'un outil de simulation.

Vue que notre projet a été réalisé au sein d'une entreprise donc l'application ne sera pas uploadée dans PlayStore<sup>11</sup>.

## IV.5 Défis et obstacles rencontrés

Comme quasiment tous les projets, nous avons eu à faire plusieurs contraintes, parmi eux :

- La prise en main du Framework 'Flutter' prenait un peu de temps.
- Entrave du temps qui nous a limités dans la réalisation de certaines tâches et qui a restreint notre capacité à approfondir certaines parties du projet.
- L'adaptation des APIs à nos besoins a été contraignante et nous a exigé de faire des changements radicaux.
- Des difficultés rencontrées dans l'adaptation et la gestion de la responsivité de l'application sur différentes tailles d'écrans et dispositifs, ce qui a nécessité des efforts supplémentaires pour assurer une expérience utilisateur cohérente.

## IV.6 L'application « vente assistée »

Dans cette section, nous allons évoquer les différentes fonctionnalités de l'application à travers des captures d'images de chaque interface permettant de bien illustrer le fonctionnement de l'application et pour cela, nous allons les présenter suivant un processus ou un cas de commande d'un client au pré de l'un des deux utilisateurs 'vendeur' ou 'fournisseur' et d'autres cas de façon générale.

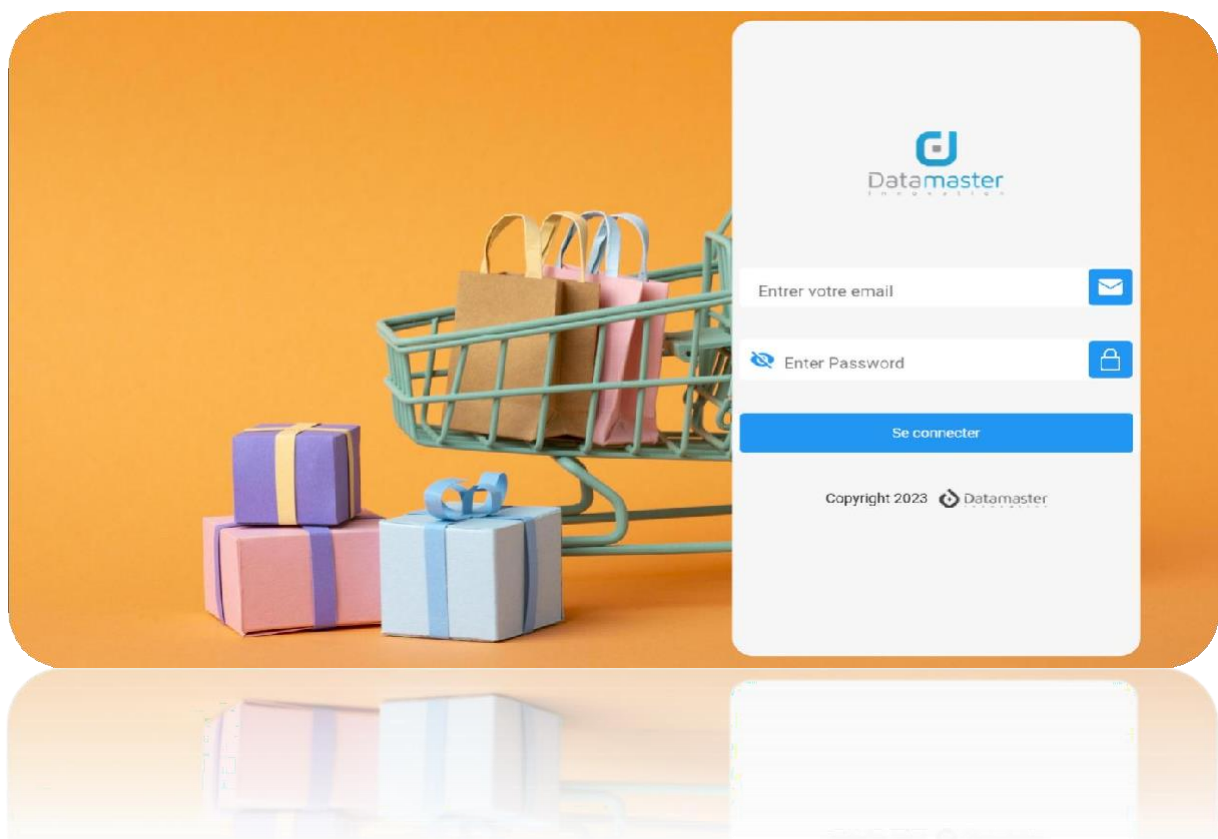
<sup>11</sup>Le Play Store est le magasin virtuel dans lequel vous pourrez télécharger des nouvelles applications votre smartphone et tablette Android, elle permet aux développeurs de pouvoir publier leurs applications afin qu'elles soient disponibles au téléchargement par d'autres utilisateurs [25].



### IV.6.1 L'interface d'authentification

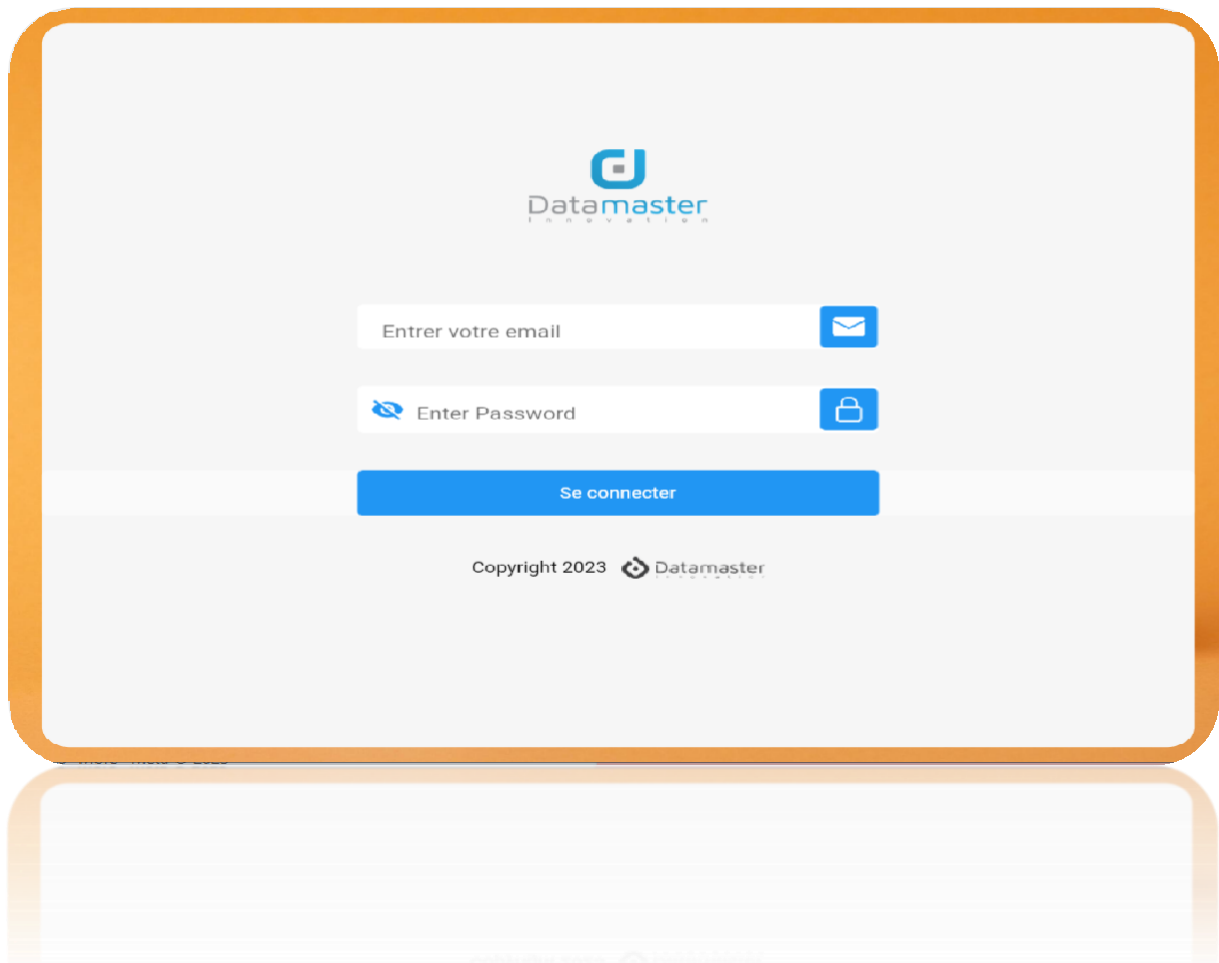
Les deux figures ci-dessous illustrent l'interface utilisateur de la page d'authentification de l'application, mettant en évidence ses éléments clés. Elle permet aux utilisateurs de se connecter à leur compte en saisissant leur adresse e-mail et leur mot de passe, qui leur sont attribués par l'administrateur. Il est important de noter que seule l'adresse e-mail peut être modifiée par l'utilisateur.

Une fois connectés, les utilisateurs ont accès au tableau de bord principal, où ils peuvent visualiser un aperçu personnalisé de leurs activités, des notifications importantes et des mises à jour récentes. De plus, la page d'authentification offre des liens vers différentes rubriques de l'application, telles que la gestion des produits, les commandes, les paramètres du compte, etc.



*Figure XIII : Interface d'authentification (taille importante)*

Pour garantir la sécurité des informations sensibles, des mesures de protection supplémentaires sont mises en place, telles que le chiffrement des mots de passe et des protocoles d'authentification sécurisés.



**Figure XIV : Interface d'authentification (taille réduite et smartphone)**

L'objectif de cette représentation est de créer une interface conviviale et sécurisée, permettant aux utilisateurs d'accéder facilement aux fonctionnalités de l'application et de gérer leurs données de manière confidentielle. Pour atteindre cet objectif, l'interface met en avant une image en arrière-plan attrayante. Le formulaire d'authentification, qui se présente sous la forme d'un effet flottant, offre une expérience visuelle agréable. En haut de la page se trouve le logo de l'entreprise, renforçant ainsi l'identification de la marque. Au centre de la page, deux champs de saisie identiques sont présents, accompagnés d'une icône dans chacune de leurs extrémités droites, pour faciliter la compréhension des utilisateurs. Juste en dessous, le bouton de

connexion, de dimensions similaires, se démarque par une couleur bleu clair, qui attire le regard des utilisateurs et constitue l'élément phare de l'application.

#### **IV.6.2 L'interface 'Dashboard' (Tableau de bord)**

Les figures XV et XVI illustrent l'interface du "Dashboard", qui offre une présentation globale des performances et des données essentielles. Cette interface intègre divers éléments visuels visant à faciliter la gestion et la prise de décision. En haut de la page se trouvent trois cercles, représentant respectivement l'image de profil de l'utilisateur, les notifications importantes et les tickets ou demandes de support en attente.

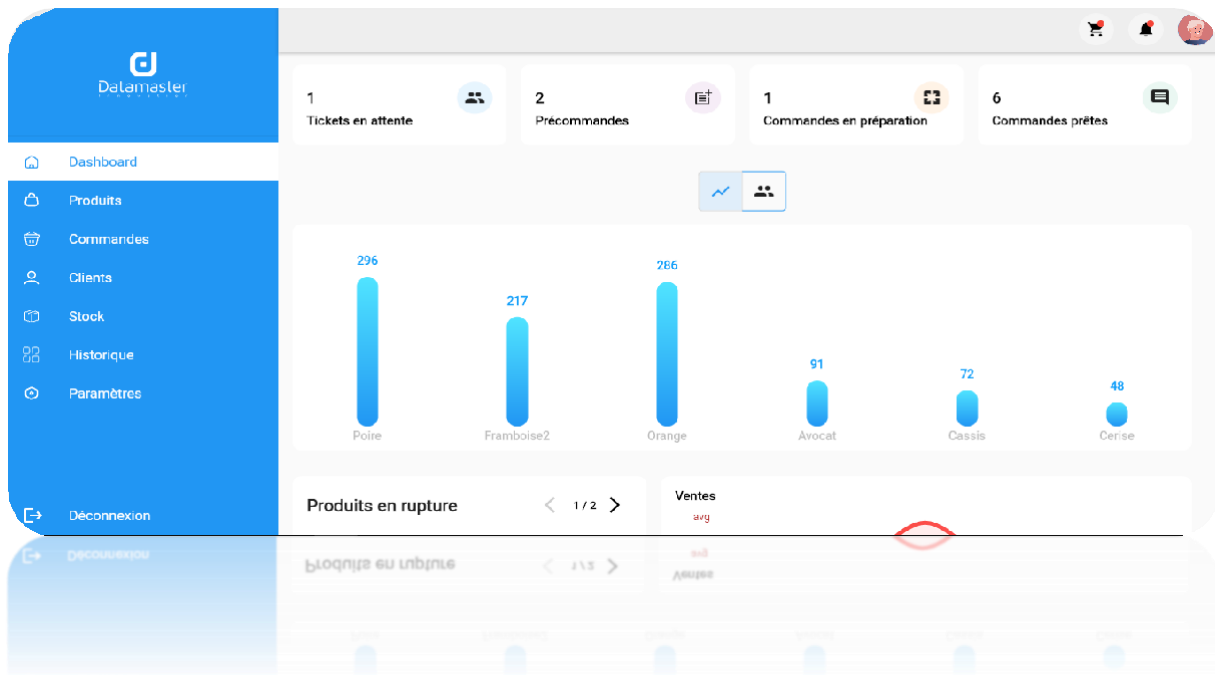
Au centre de la page, vous trouverez quatre étiquettes dynamiques, chacune représentant une valeur spécifique pertinente pour l'activité de l'application. Ces étiquettes fournissent une vue rapide des indicateurs clés tels que le nombre total de produits, le nombre de commandes traitées, le nombre de catégories, et ainsi de suite.

Juste en dessous des étiquettes, un graphique à barres est affiché, présentant les dix produits les plus vendus. Ce graphique permet de visualiser rapidement les performances des produits et d'identifier les articles les plus populaires.

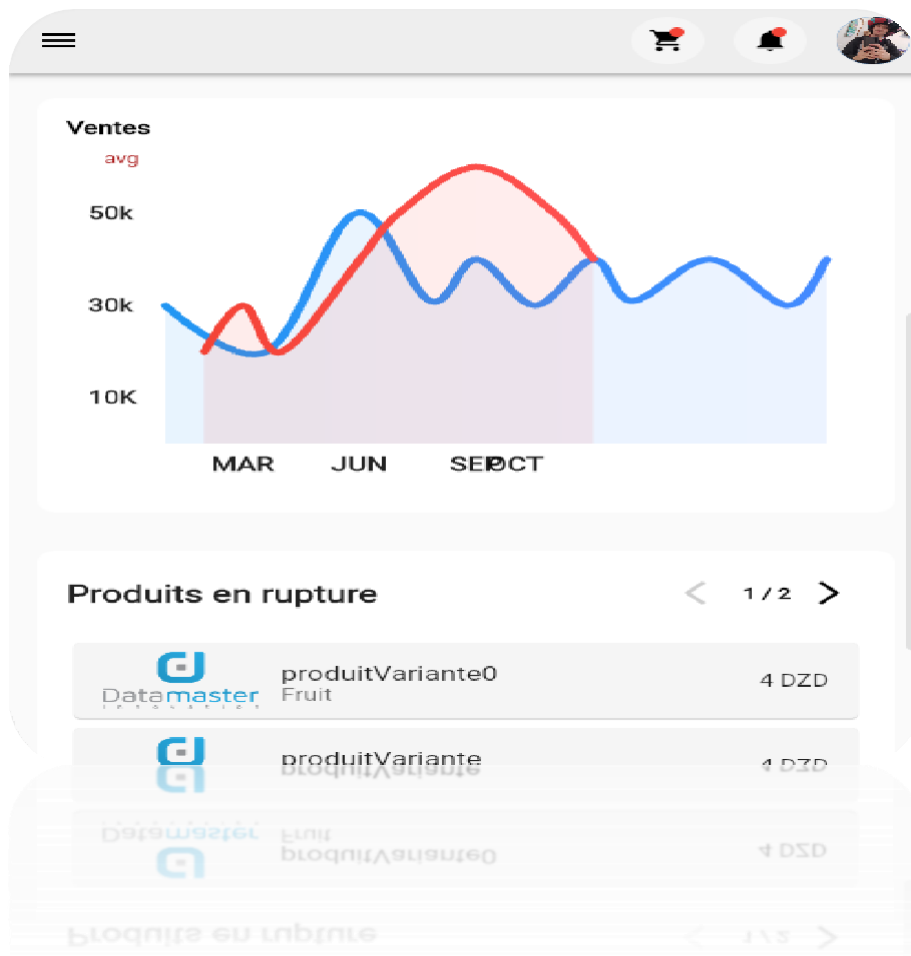
Un bouton de basculement permet également de passer du graphique à une liste détaillée des dix produits les plus vendus. Cette liste fournit des informations supplémentaires sur chaque produit, telles que le nom, le prix, le nombre de ventes, et ainsi de suite.

En descendant davantage sur la page, une autre liste met en évidence les produits qui sont en rupture de stock. Cette liste permet aux utilisateurs de prendre des mesures appropriées, comme le réapprovisionnement des produits épuisés.

Enfin, deux graphiques en courbes sont présentés pour représenter les tendances des commandes et des précommandes. Ces courbes permettent de suivre l'évolution des commandes passées et des précommandes au fil du temps, offrant ainsi une vue d'ensemble de l'activité commerciale.



*Figure XV : Interface 'Dashboard' (taille importante)*



*Figure XVI : Interface 'Dashboard' (taille réduite et smartphone)*

Cette présentation de la page Dashboard vise à fournir aux utilisateurs un aperçu visuel clair et complet des données et des indicateurs essentiels pour une gestion efficace de leur activité. Elle met en valeur les informations importantes, tout en facilitant la navigation et l'interprétation des données.

Parallèlement à la page Dashboard, l'application est dotée d'un Side Menu comme l'indique la figure ci-dessous qui est accessible depuis n'importe quelle section. Le Side Menu est un élément de navigation présent de manière constante dans l'interface. Il permet aux utilisateurs de passer rapidement entre les différentes sections de l'application.

Lorsque l'utilisateur se trouve sur la page Dashboard, l'option correspondante du Side Menu, intitulée "Dashboard", est mise en évidence pour indiquer sa position actuelle. Cette mise en évidence est réalisée en utilisant une couleur blanche ou une autre indication visuelle pour distinguer l'option sélectionnée des autres.

Le Side Menu offre une navigation intuitive et facilite l'accès à d'autres fonctionnalités clés de l'application, telles que les rapports, les paramètres, les analyses, etc. En cliquant sur chaque option du Side Menu, l'utilisateur est dirigé vers la section correspondante de manière fluide et rapide.

L'inclusion du Side Menu dans l'application assure une expérience de navigation cohérente et facilite la transition entre les différentes fonctionnalités, permettant ainsi aux utilisateurs de naviguer aisément et de profiter pleinement des capacités offertes par l'application.



**Figure XVII : Interface du 'Side Menu (menu latéral)'**

### IV.6.3 L'interface 'Produits'

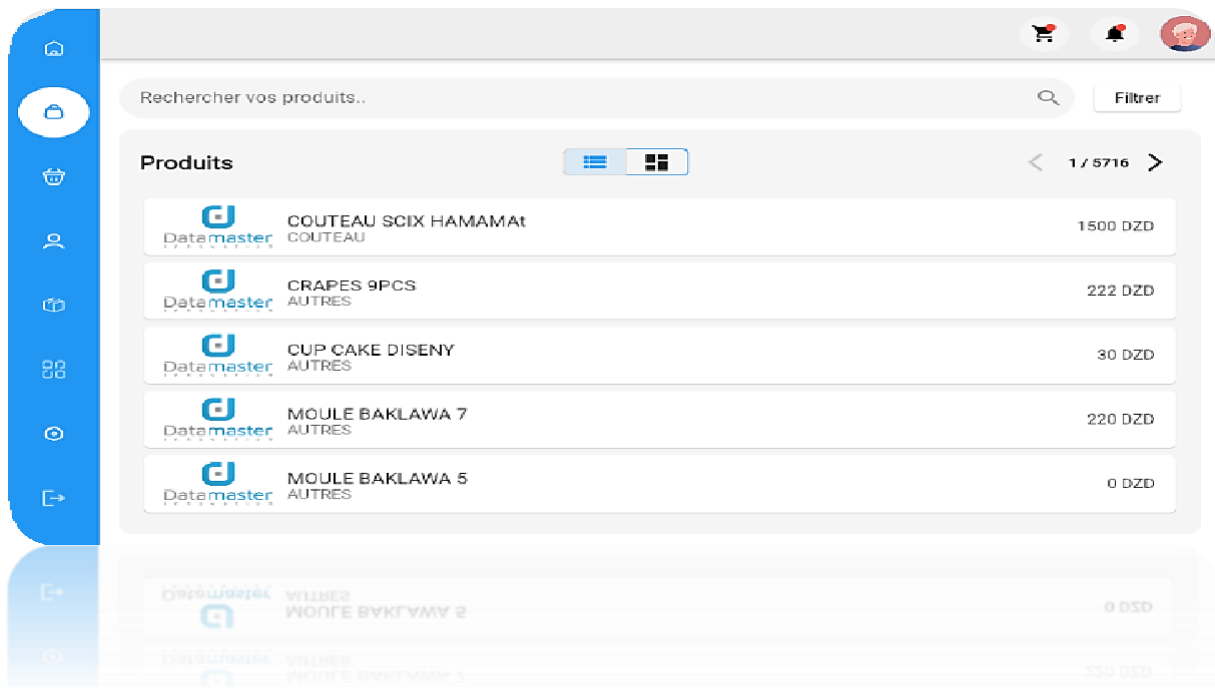
Les figures XVIII et XIX illustrent encore une fois la page « Produits » qui est accessible depuis le menu latéral de l'application, offrant aux utilisateurs un moyen pratique de consulter et de parcourir la liste des produits disponibles. Voici une description détaillée de cette page :

En haut de la page, un champ de recherche permet aux utilisateurs de rechercher des produits spécifiques en saisissant des mots-clés pertinents. À côté de ce champ de recherche, se trouve un bouton qui, lorsqu'il est cliqué, fait apparaître une fenêtre contextuelle offrant des options de filtrage supplémentaires. Les utilisateurs peuvent choisir une catégorie spécifique ou appliquer d'autres critères de filtrage pour affiner les résultats de recherche.

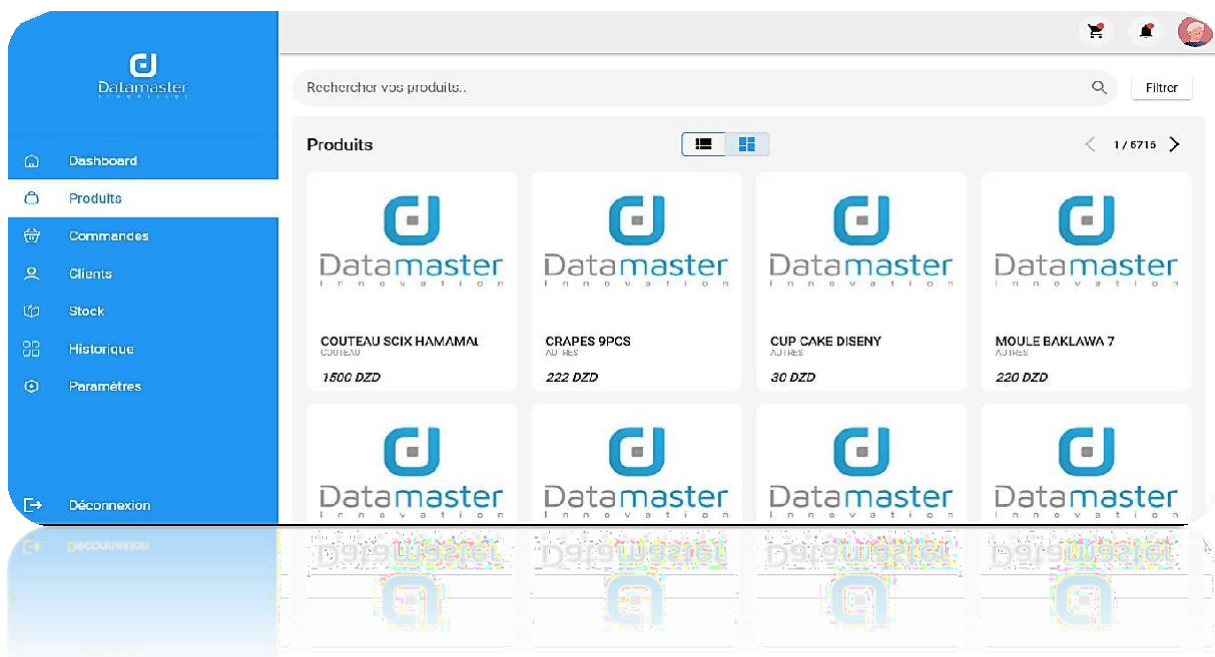
Juste en dessous de la barre de recherche et de filtrage, se trouve un bouton permettant de basculer entre deux vues différentes pour afficher les produits. Les utilisateurs peuvent choisir entre une vue en liste et une vue en cartes. La première vue en liste (figure XVIII) présente chaque produit avec une miniature de l'image, accompagnée de son prix et de sa catégorie, permettant aux utilisateurs de parcourir rapidement les produits et d'obtenir des informations essentielles. La deuxième vue en cartes (figure XIX) met en avant une image plus grande du produit, mettant en valeur ses caractéristiques distinctives, ainsi que le prix et la catégorie, offrant ainsi aux utilisateurs une expérience visuelle plus immersive.

À droite des options de vues, vous trouverez des icônes de flèches pour la pagination. Ces icônes sont accompagnées du nombre de pages disponibles, permettant aux utilisateurs de naviguer entre les différentes pages de produits. Cette fonctionnalité de pagination facilite la consultation des produits lorsque leur nombre dépasse la capacité d'affichage d'une seule page.

En utilisant les options de filtrage, les différentes vues d'affichage et la pagination, les utilisateurs peuvent personnaliser leur expérience de navigation, trouver facilement les produits qui correspondent à leurs préférences et explorer l'ensemble de la liste de produits de manière efficace.



*Figure XVIII : Interface 'Produits' (List View)*



*Figure XIX : Interface 'Produits' (Grid View)*

Cette présentation de la page "Produits" vise à offrir aux utilisateurs une expérience conviviale et flexible pour explorer et sélectionner les produits. Elle met en avant les

fonctionnalités de recherche, de filtrage et les deux vues d'affichage différentes pour faciliter la découverte des produits.

#### IV.6.4 L'interface d'un produit sélectionné

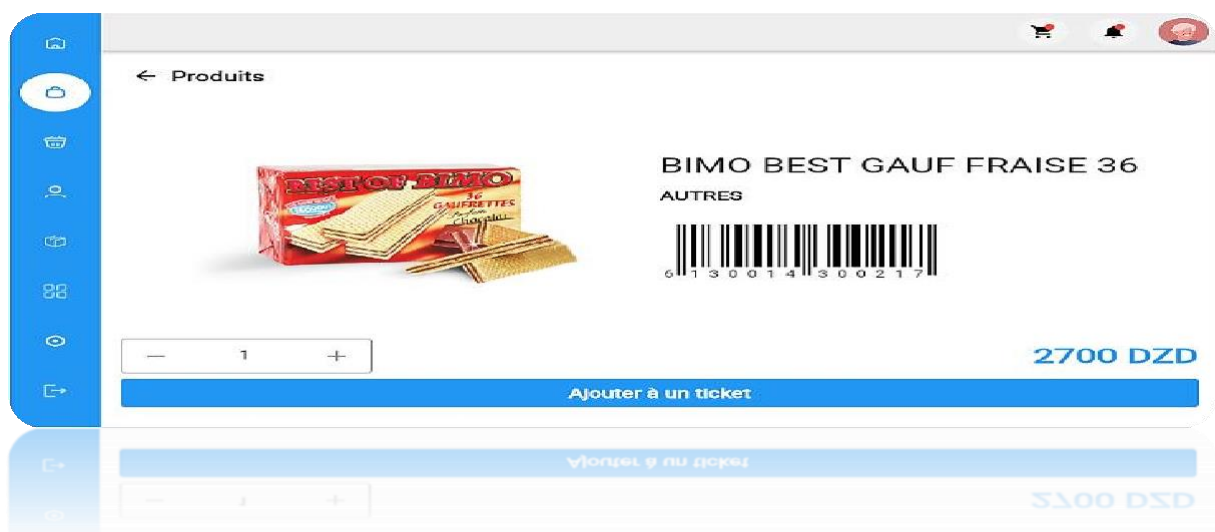
Cette figure montre l'interface d'une des produits sélectionnés, offrant un aperçu visuel détaillé lorsqu'un utilisateur clique dessus dans l'une des deux vues disponibles (liste ou carte). La page du produit est organisée de manière claire et conviviale, offrant aux utilisateurs toutes les informations nécessaires sur le produit sélectionné.

Au sommet de la page, une photo de grande taille du produit est mise en évidence, permettant aux utilisateurs de visualiser les détails et les caractéristiques du produit de manière plus immersive.

Juste en dessous de la photo, les informations clés du produit sont présentées, telles que son nom, sa description, sa catégorie et son prix. Ces détails permettent aux utilisateurs de se familiariser rapidement avec le produit.

En dessous des informations, deux icônes représentant un signe plus (+) et un signe moins (-) sont affichées. Ces icônes permettent aux utilisateurs de régler la quantité du produit souhaitée en augmentant ou en diminuant la valeur.

En bas de la page, un bouton bleu intitulé "Ajouter au ticket" est présent. Lorsque les utilisateurs cliquent sur ce bouton, le produit sélectionné est ajouté au ticket en cours. De plus, ce bouton redirige les utilisateurs vers la page du ticket, où ils peuvent visualiser et gérer les produits ajoutés.



*Figure XX : Interface 'Produit sélectionné'*



Cette représentation visuelle de la page du produit permet aux utilisateurs de naviguer facilement, d'obtenir les informations essentielles sur le produit et de procéder à l'ajout au ticket en quelques étapes simples qui permet aux utilisateurs de passer rapidement à l'étape suivante de leur expérience d'achat. Cette conception ergonomique de la page du produit favorise l'engagement des utilisateurs et facilite leurs interactions avec l'application, améliorant ainsi leur satisfaction globale.

#### **IV.6.5 L'interface 'Tickets'**

Les figures XXI et XXII illustrent la page 'Ticket' de l'application qui offre aux utilisateurs un moyen pratique de gérer les transactions et d'associer des clients aux produits sélectionnés. Voici une description détaillée de cette page :

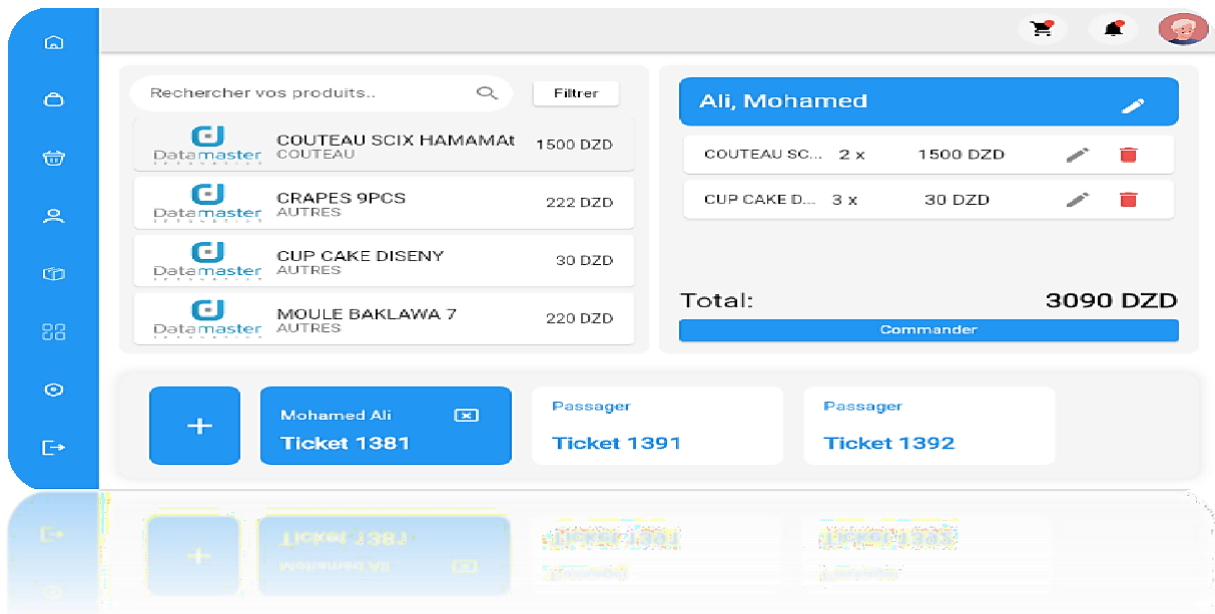
En haut à gauche de la page se trouve une liste des produits disponibles. Cette liste est accompagnée d'une barre de recherche et d'un bouton de filtrage, permettant aux utilisateurs de trouver rapidement les produits souhaités. Lorsque les utilisateurs cliquent sur un produit dans la liste, une fenêtre contextuelle s'affiche, leur permettant de définir la quantité souhaitée du produit à l'aide d'un champ d'entrée. Un bouton "Ajouter au ticket" est également présent dans la fenêtre contextuelle.

La zone en haut à droite de la page est réservée à la gestion des tickets. Initialement, cette zone affiche une liste déroulante des clients enregistrés. Cependant, lorsque les utilisateurs choisissent un produit, la fenêtre contextuelle se déplace vers cette zone et se transforme en une icône représentant le produit sélectionné. Les utilisateurs peuvent alors cliquer sur cette icône pour associer le produit au client de leur choix.

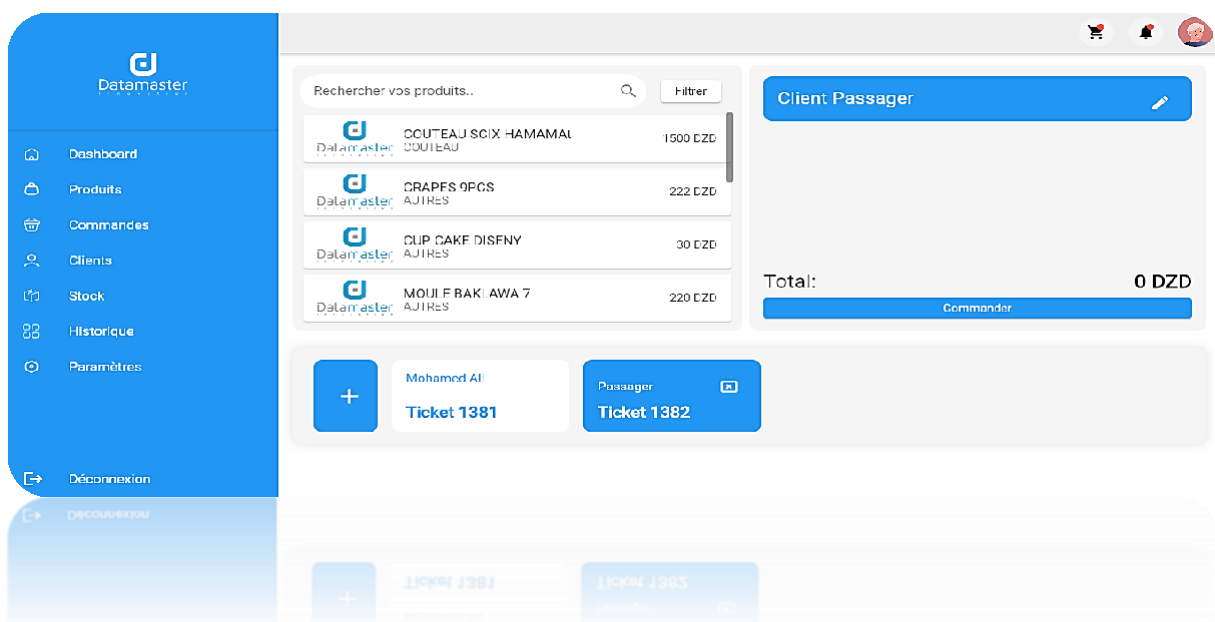
Sous la zone de gestion des tickets, deux boutons sont disponibles : "Commande" et "Précommande" pour les vendeurs, et "Commande et Vente" pour les fournisseurs. Les utilisateurs peuvent sélectionner l'option correspondant à leur besoin et finaliser la transaction. Cela générera automatiquement un bon de commande ou une précommande en fonction de l'option choisie.

Cette présentation de la page "Tickets" offre aux utilisateurs une interface conviviale et pratique pour gérer les transactions et associer des clients aux produits sélectionnés. La possibilité de déplacer la fenêtre contextuelle vers la zone de gestion des tickets facilite le

processus d'association des produits aux clients. Les boutons clairs permettent aux utilisateurs de conclure rapidement la transaction en générant les documents appropriés.



*Figure XXI: Interface 'Tickets'*



*Figure XXII: Interface 'Tickets' (Version2)*

On opter pour le choix de ces widgets et leurs emplacements en vue de rendre les interfaces de notre application plus rapprochées de celle de la version Desktop qui existe déjà.

## IV.6.6 L'interface 'Commandes'

La figure ci-dessous illustre la page 'Commandes' de l'application offre aux utilisateurs un moyen pratique de visualiser, filtrer et gérer les commandes et précommandes. Voici une description détaillée de cette page :

En haut de la page se trouve une barre de recherche et un bouton de filtrage, permettant aux utilisateurs de rechercher des commandes spécifiques ou d'appliquer des critères de filtrage pour affiner les résultats. Cette fonctionnalité facilite la recherche rapide et la gestion des commandes en fonction de différents critères tels que la date, le client, etc.

Juste en dessous de la barre de recherche et du bouton de filtrage, se trouve une liste des commandes. Cette liste affiche les informations essentielles de chaque commande, telles que le numéro de commande, la date, le client, etc. Des flèches de pagination sont disponibles en bas de la liste pour permettre aux utilisateurs de naviguer facilement entre les différentes pages de commandes.

Sur la même ligne que les informations de chaque commande, deux icônes sont présentes. Ces icônes permettent aux utilisateurs de visualiser les détails de la commande ou de la précommande. En cliquant sur l'icône correspondante, une fenêtre contextuelle s'affiche, affichant en détail toutes les informations liées à la commande ou à la précommande, telles que la liste des produits, les quantités, les prix, etc.

En cliquant sur le deuxième icône correspondant qui est disponible pour transformer une précommande en commande. Cette fonctionnalité permet aux utilisateurs de confirmer et finaliser la précommande en la convertissant en une commande complète. Cela permet de garantir une gestion fluide du processus de commande.

ID	Client	Type	Etat		
1381	M Ali	Ticket			
1384	Passager	Commande			
1385	Passager	Commande			
1386	Passager	Commande			
1387	S Benadla	Commande			
1388	Passager	Précommande			

Figure XXIII : Interface 'Commandes'

Cette présentation de la page "Commandes" offre aux utilisateurs une interface claire et organisée pour visualiser, filtrer et gérer les commandes et précommandes. La barre de recherche et le bouton de filtrage facilitent la recherche rapide, tandis que la liste des commandes avec les flèches de pagination permet une navigation aisée. Les icônes des détails de la commande ou précommande et le bouton de transformation de précommande en commande offrent une gestion efficace des transactions.

En ce qui concerne le design de la liste des commandes, celui-ci est conçu de manière moderne et attrayante. Les informations de chaque commande sont présentées de manière claire et concise, avec une disposition soignée et une esthétique élégante. Les couleurs et les éléments visuels sont choisis de manière à faciliter la lisibilité et la compréhension des informations. De plus, la liste est conçue pour être réactive, offrant une expérience utilisateur fluide, que ce soit sur un ordinateur de bureau ou sur des appareils mobiles. Ce design moderne ajoute une touche professionnelle et agréable à l'ensemble de l'interface de la page "Commandes", offrant aux utilisateurs une expérience visuelle plaisante tout en facilitant la gestion efficace des transactions.

#### **IV.6.7 L'interface 'Clients'**

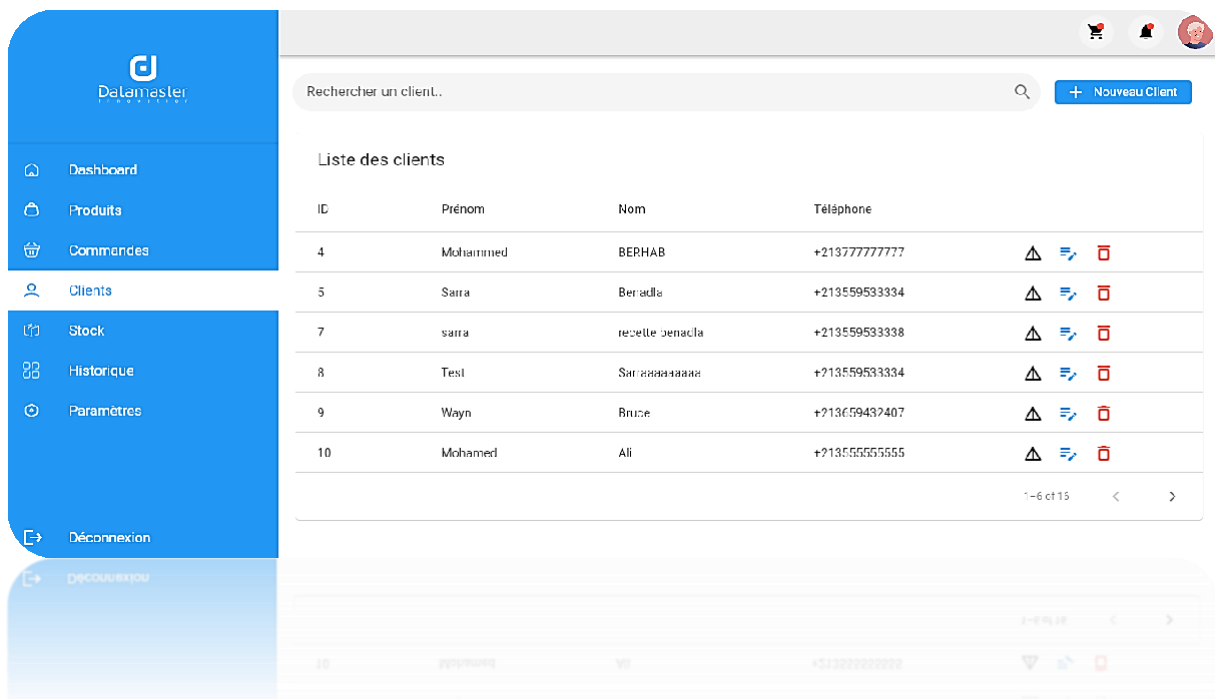
Cette figure représente la page 'Clients' qui offre aux utilisateurs un moyen pratique de gérer et de visualiser les informations relatives aux clients. Voici une présentation détaillée de cette page :

En haut de la page, on retrouve une barre de recherche qui permet aux utilisateurs de trouver rapidement un client spécifique en saisissant des mots-clés pertinents. À côté de cette barre de recherche se trouve un bouton "Ajouter un nouveau client", qui déclenche l'apparition d'une fenêtre contextuelle. Cette fenêtre contextuelle contient un formulaire à remplir avec les informations du nouveau client, permettant ainsi aux utilisateurs d'ajouter facilement de nouveaux clients à la base de données.

Au centre de la page, se trouve une liste des clients, présentée dans un format similaire à celui des commandes. Chaque client est représenté par un ensemble d'informations telles que son nom, son adresse, son numéro de téléphone, etc. Cette liste est conçue de manière à faciliter la navigation et la consultation des informations des clients. Les utilisateurs peuvent parcourir la liste et trouver rapidement les détails spécifiques d'un client donné.

Trois icônes sont associées à chaque client de la liste. La première icône permet de supprimer le client, offrant aux utilisateurs la possibilité de retirer un client de la base de données si nécessaire. La deuxième icône permet de modifier les informations du client. Lorsque l'utilisateur clique sur cette icône, un formulaire de modification s'affiche, lui permettant de mettre à jour les informations du client. Enfin, la troisième icône permet de visualiser les détails complets d'un client sur une page séparée, offrant une vue détaillée et exhaustive des informations relatives à ce client.

En bas de la liste des clients, les flèches de pagination facilitent la navigation entre les différentes pages de clients, permettant aux utilisateurs de parcourir facilement l'ensemble de la liste.



**Figure XXIV : Interface 'Clients'**

Cette présentation de la page "Clients" vise à offrir aux utilisateurs une expérience fluide et intuitive pour la gestion des informations clients. La combinaison de la barre de recherche, des icônes fonctionnelles et de la pagination facilite la recherche, la modification et la visualisation des détails des clients, contribuant ainsi à une gestion efficace de la base de données clients.

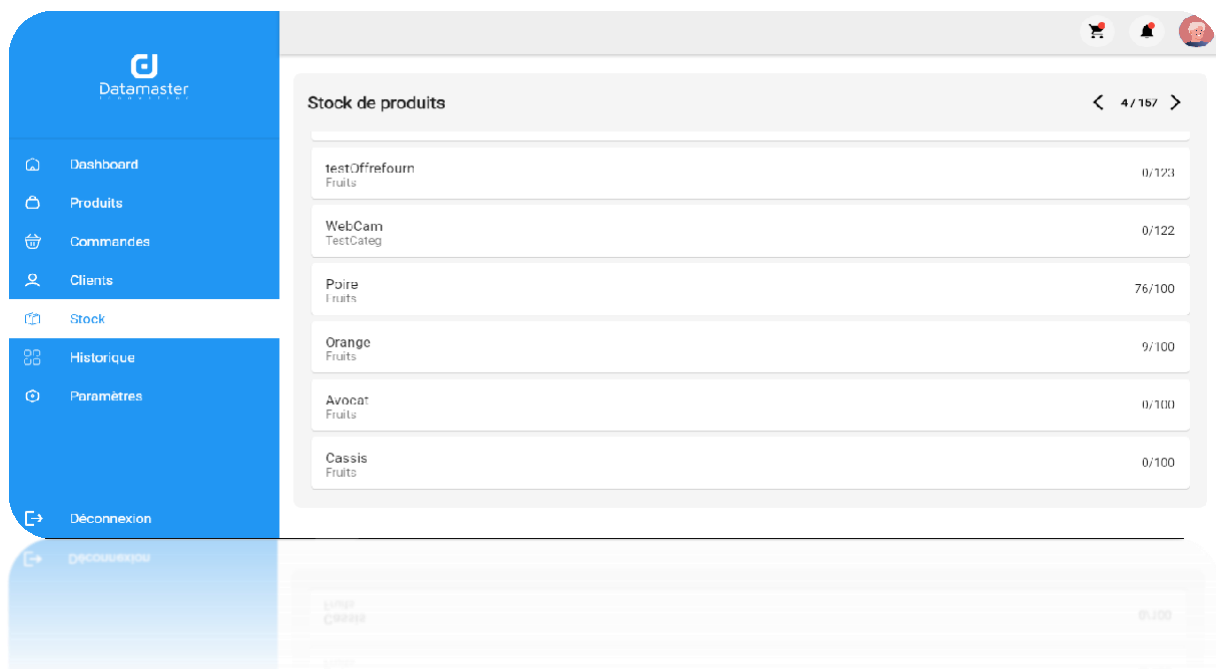
## IV.6.8 L'interface 'Stock'

La figure ci-dessous illustre la page 'Stock' qui est une interface conçue pour permettre aux utilisateurs de gérer et de suivre les stocks des produits. Voici une présentation détaillée de cette page :

En haut de la page, les utilisateurs trouveront une barre de recherche qui leur permet de rechercher rapidement des produits spécifiques en saisissant des mots-clés pertinents. À côté de cette barre de recherche, se trouve un bouton "Filtrer" qui permet d'appliquer des critères de filtrage supplémentaires pour affiner les résultats de recherche.

Juste en dessous de la barre de recherche et du bouton de filtrage, se trouve une liste de produits. Chaque produit est représenté par une petite image, accompagnée de son nom et de sa catégorie. Sur l'extrémité droite de chaque ligne, une valeur en rouge est affichée pour indiquer la quantité disponible en stock pour ce produit.

La page "Stock" comporte également une fonctionnalité de pagination située au-dessus de la première ligne de la liste des produits. Cette pagination permet aux utilisateurs de naviguer entre les différentes pages de produits, en affichant un nombre limité de produits par page. Cela facilite la visualisation et la gestion des stocks, en évitant une surcharge d'informations à l'écran.



*Figure XXV : Interface 'Stock'*

Cette présentation claire et concise permet aux utilisateurs de visualiser rapidement les produits disponibles ainsi que les niveaux de stock associés. Grâce à la barre de recherche et au bouton de filtrage, ils peuvent effectuer des recherches spécifiques et affiner les résultats en fonction de leurs besoins.

La page "Stock" offre ainsi une vue d'ensemble complète des produits et de leur disponibilité, ce qui facilite la gestion efficace des stocks et des approvisionnements.

#### **IV.6.9 L'interface 'Historique'**

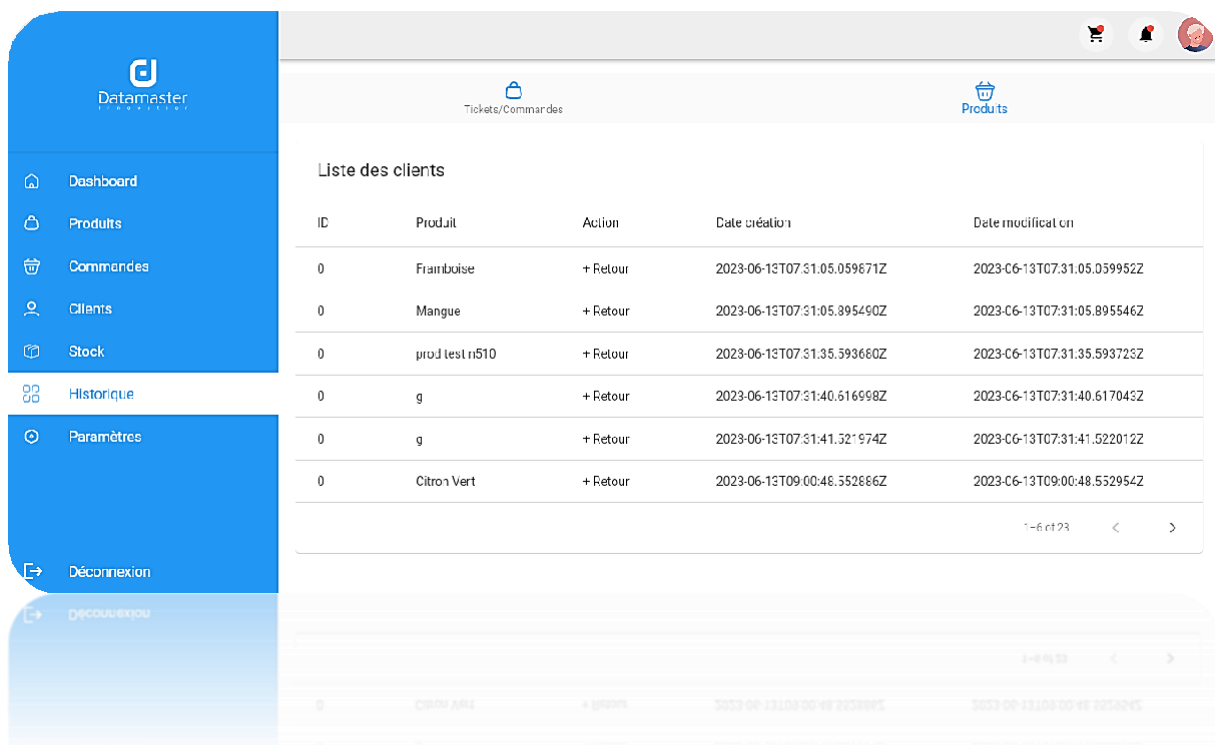
La figure ci-dessous illustre La page "Historique" qui présente aux utilisateurs un aperçu détaillé des activités réalisées sur un produit, ainsi que les commandes ou précommandes associées à ce dernier. Voici une présentation de cette page :

La liste des activités permet aux utilisateurs de retracer toutes les actions effectuées sur un produit spécifique. Ces activités peuvent inclure des mises à jour de stock, des modifications d'informations, des ajouts ou des suppressions de produits, et bien plus encore. Chaque activité est accompagnée d'une date, ce qui permet aux utilisateurs de suivre l'historique chronologique des événements.

En plus des activités, la page affiche également les commandes ou précommandes liées à ce produit. Les commandes et précommandes sont clairement identifiées et associées à leurs dates respectives. Cela permet aux utilisateurs de comprendre rapidement quand une commande ou une précommande a été passée pour ce produit.

L'objectif de la page "Historique" est de fournir aux utilisateurs une visibilité complète sur l'évolution d'un produit, les modifications apportées, ainsi que les commandes ou précommandes associées. Cette fonctionnalité permet une traçabilité précise et facilite la prise de décisions informées en matière de gestion des produits.

En résumé, la page "Historique" offre aux utilisateurs une vue d'ensemble des activités et des événements liés à un produit, permettant ainsi une gestion efficace de l'historique et des commandes ou précommandes associées.



*Figure XXVI : Interface 'Historique'*

#### IV.6.10 L'interface 'Paramètres'

Cette figure montre La page "Paramètres" qui offre aux utilisateurs un moyen de personnaliser leur expérience et de gérer les différentes options de l'application. Voici une présentation de cette page :

La page "Paramètres" comprend un Side Menu contenant cinq options distinctes. Chaque option mène à une page spécifique où les utilisateurs peuvent effectuer des actions spécifiques en fonction de leurs besoins.

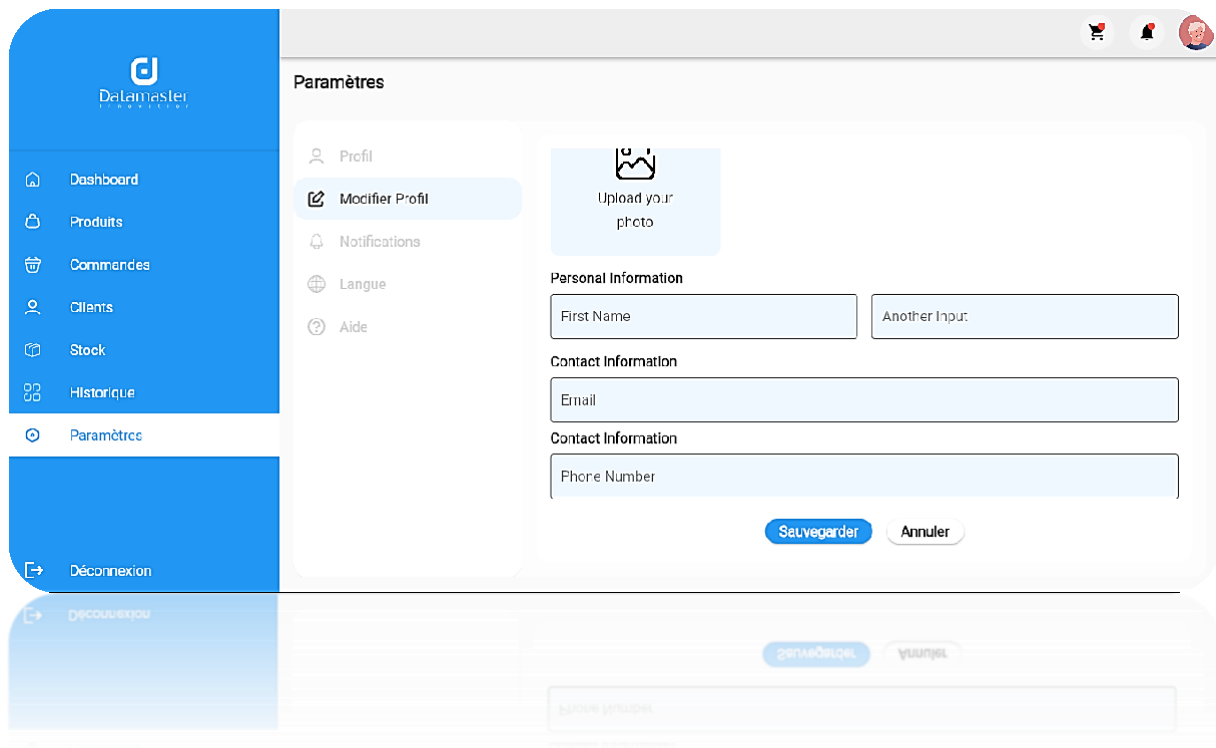
L'option "Profil" permet aux utilisateurs de visualiser leur profil personnel. Une image circulaire représentant l'utilisateur est affichée à gauche de la page, accompagnée de son nom et de son rôle. À droite de la page, les autres informations personnelles de l'utilisateur sont présentées sous forme de tableau, offrant ainsi un aperçu rapide des détails importants.

L'option "Modifier Profil" permet aux utilisateurs de mettre à jour leurs informations personnelles. Des champs d'entrée sont fournis pour saisir les données nécessaires, et un champ spécifique est disponible pour modifier la photo de profil.



L'option "Notifications" propose une fonctionnalité de gestion des notifications. Un bouton de commutation animé permet aux utilisateurs de masquer ou de révéler le cercle de notification situé en haut de la page. La page affiche également les notifications reçues, offrant ainsi une vue d'ensemble des informations importantes.

Les autres options langues et à propos du Side Menu offrent également des fonctionnalités spécifiques, mais elles ne sont pas détaillées dans cette présentation.



**Figure XXVII : Interface 'Paramètres'**

## **IV.7 Conclusion**

Dans ce dernier chapitre de ce mémoire, nous avons abordé toutes les technologies utilisées au cours de l'implémentation et la réalisation de l'application, ainsi que l'architecture du système illustré par un organigramme.

Nous avons également cité les différents obstacles rencontrés au cours de la mise en œuvre de l'application et au final nous avons cité les différentes méthodes d'une création d'un APK afin de pouvoir la déployer.

En résumé, ce chapitre permet d'avoir une idée assez claire sur la version finale du produit et leurs différentes fonctionnalités montrées à travers les interfaces.

# **Conclusion générale**

## V.1 Conclusion Générale

Nous avons abordé dans ce mémoire la problématique de la vente assistée et de la commercialisation des produits en Algérie, en mettant l'accent sur l'importance d'intégrer des solutions technologiques avancées pour améliorer l'expérience client, optimiser les processus de vente et renforcer la compétitivité des entreprises. À travers une analyse approfondie du contexte global de l'industrie des marchandises et produits, nous avons souligné les défis auxquels sont confrontées les entreprises, notamment la nécessité de s'adapter aux changements des comportements des consommateurs, des avancées technologiques et des pressions environnementales.

Le développement de notre application de vente assistée représente une réponse à ces défis. En offrant une assistance personnalisée, des recommandations ciblées et une gestion efficace des produits, notre application vise à améliorer l'expérience client et à renforcer la compétitivité des entreprises locales sur le marché national et international. Nous avons également présenté les différentes étapes de conception et de réalisation de ce projet, en mettant l'accent sur les fonctionnalités clés et les modélisations nécessaires pour son développement.

Les perspectives pour ce projet sont prometteuses. L'application de vente assistée offre un potentiel important pour transformer les pratiques commerciales en Algérie, en permettant aux entreprises de mieux répondre aux besoins et aux attentes des clients. De plus, l'évolution rapide des technologies offre de nouvelles opportunités d'innovation et d'amélioration continue de l'application, notamment en explorant des fonctionnalités telles que l'intelligence artificielle, la réalité augmentée ou l'intégration avec d'autres plateformes et services.

Cependant, le travail entrepris dans ce projet n'est pas terminé et offre encore un grand potentiel d'amélioration et de développement. En envisageant l'avenir de l'application, certaines perspectives se dessinent :

- Mise en place d'un mode déconnecté : L'implémentation d'un mode déconnecté permettrait aux utilisateurs de continuer à utiliser l'application même en l'absence de connexion internet, améliorant ainsi leur expérience d'achat.
- Recommandations personnalisées pour chaque client : En intégrant des algorithmes d'apprentissage automatique, il serait possible d'améliorer les recommandations personnalisées en fonction des préférences et des historiques d'achat de chaque client.
- Paiement sécurisé et mobile : Intégrer des solutions de paiement sécurisées et conviviales dans l'application, telles que des porte-monnaie électroniques, des services

de paiement mobiles et des méthodes de paiement sans contact. Cela offrira aux clients une expérience fluide et pratique lors du processus de paiement.

Il est également important de souligner que ce projet peut servir de base pour d'autres initiatives dans le domaine de la vente assistée en Algérie. Les résultats obtenus et les bonnes pratiques identifiées peuvent être partagés avec d'autres entreprises et institutions, favorisant ainsi l'adoption de solutions technologiques avancées et le développement d'un écosystème propice à l'innovation commerciale.

En résumé, même si notre application a déjà contribué à améliorer l'expérience client et à optimiser les processus de vente, il existe encore de nombreuses possibilités pour la faire évoluer et répondre encore mieux aux besoins des utilisateurs. Ces perspectives futures permettront de renforcer davantage l'impact positif de l'application dans le domaine de la vente assistée et d'assurer son adaptation continue aux exigences changeantes du marché.

# Bibliographie

- [1] Dan Haller. Comment le Cross-selling et l'Upselling augmentent vos ventes. TrustedShops : <https://business.trustedshops.fr/blog/cross-selling-upselling> (Consulté le 20 Mars 2023).
- [2] Tuleap : <https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes> (Consulté le 20 Mars 2023).
- [3] Alain Fernandez. Scrum, la méthode agile. PILOTER.ORG : <https://www.piloter.org/projet/methode/scrum.htm> (Consulté le 20 Mars 2023).
- [4] Google Forms, Wikipédia : [https://en.wikipedia.org/wiki/Google\\_Forms](https://en.wikipedia.org/wiki/Google_Forms) (Consulté le 20 Mars 2023).
- [5] UML (informatique) : [https://fr.wikipedia.org/wiki/UML\\_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique)) (Consulté le 20 Mars 2023).
- [6] Gantt : <https://www.gantt.com/fr/> (Consulté le 20 Mars 2023).
- [7] Xyoss. Office Powerpoint : <https://cours-informatique-gratuit.fr/dictionnaire/microsoft-office-powerpoint/> (Consulté le 22 Mars 2023).
- [8] StarUML : <https://www.wikiwand.com/fr/StarUML> (Consulté le 22 Mars 2023).
- [9] GetApp : <https://www.getapp.fr/software/103873/teamgantt> (Consulté le 22 Mars 2023).
- [10] Tibco : <https://www.tibco.com/fr/reference-center/what-is-a-logical-data-model> (Consulté le 22 Mars 2023).
- [11] BDM Tools : <https://www.blogdumoderateur.com/tools/visual-studio-code/> (Consulté le 22 Mars 2023).
- [12] <https://fr.theastrologypage.com/intellij-idea> (Consulté le 22 Mars 2023).
- [13] AndroidAide : <https://androidayuda.com/fr/android-studio/> (Consulté le 22 Mars 2023).
- [14] nextdecisions : <https://www.next-decision.fr/wiki/qu-est-ce-que-git> (Consulté le 22 Mars 2023).
- [15] DataScientest : <https://datascientest.com/github-tout-savoir> (Consulté le 22 Mars 2023).
- [16] Swagger (logiciel), Wikipédia : [https://fr.wikipedia.org/wiki/Swagger\\_\(logiciel\)](https://fr.wikipedia.org/wiki/Swagger_(logiciel)) (Consulté le 22 Mars 2023).
- [17] Damian A. Uibunlog : <https://ubunlog.com/mysql-workbench-bases-datos/> (Consulté le 22 Mars 2023).

- [18] SQL.sh : <https://sql.sh/logiciel/mysql-workbench> (Consulté le 22 Mars 2023).
- [19] java T point : <https://www.javatpoint.com/django-mvt> (Consulté le 22 Mars 2023).
- [20] Django : <https://python.doctor/page-django-introduction-python> (Consulté le 22 Mars 2023).
- [21] La rédaction, JDN. MySQL : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203595-mysql-my-structured-query-language-definition/> (Consulté le 22 Mars 2023).
- [22] Flutter (logiciel), Wikipédia : [https://fr.wikipedia.org/wiki/Flutter\\_\(logiciel\)](https://fr.wikipedia.org/wiki/Flutter_(logiciel)) (Consulté le 22 Mars 2023).
- [23] Frandroid : <https://www.frandroid.com/android/535194-quest-ce-que-flutter-loutil-permettant-de-cree-des-applications-android-et-ios> (Consulté le 22 Mars 2023).
- [24] RedHat : <https://www.redhat.com/fr/topics/api/what-is-a-rest-api> (Consulté le 22 Mars 2023).
- [25] Xyoss : <https://cours-informatique-gratuit.fr/dictionnaire/playstore/> (Consulté le 22 Mars 2023).

## *Résumé*

De nos jours la vente des produits ou de n'importe quel type de marchandise se fait encore de manière traditionnelle (paiement dans une caisse ou un guichet, carnets...), ce qui engendre des pertes atroces pour quelques commerçants en termes de temps et d'argent et causera aussi des malentendus et non-satisfactions pour un certain nombre de clients ainsi que les employés.

Dans ce contexte, notre projet vient pour pallier ses problèmes et à mettre en place une technologie et un processus de vente assistée qui représente une solution pour améliorer l'expérience client et renforcer la compétitivité des entreprises de vente en Algérie.

En collaboration avec l'équipe talentueuse de DATAMASTER, nous avons travaillé sur la conception et la réalisation de cette application, en tenant compte des besoins spécifiques du marché algérien. Nous avons ainsi pu intégrer des fonctionnalités fiables et adaptées à l'environnement commercial local qui pourront être améliorées dans le futur en intégrant l'assistance personnalisée, les recommandations ciblées et une gestion encore plus efficace des produits. Ces améliorations viseront à renforcer davantage l'expérience client et la compétitivité des entreprises de vente en Algérie, en répondant toujours mieux aux besoins et aux attentes des clients tout en s'adaptant aux évolutions du marché.

**Mots-clés :** vente, commande ou précommande, gestion multi-commande, application mobile de vente assistée, Flutter, Django.



# *Abstract*

Nowadays, the sale of products or any type of merchandise is still done in the traditional way (payment at a cash desk or counter, booklets, etc.), which generates atrocious losses for some traders in terms of time and money. money and will also cause misunderstandings and dissatisfactions for not customers as well as employees.

In this context, our project comes to solve its problems and to set up a technology and an assisted sales process which represents an innovative solution to improve the customer experience and strengthen the competitiveness of sales companies in Algeria.

By integrating features such as personalized assistance, targeted recommendations and effective product management, our application aims to meet customer needs and expectations, while adapting business practices to market changes. It offers significant potential to transform the industry of goods and products, promoting the adoption of advanced technological solutions and contributing to the economic development of the country.

**Key-words:** sale, order or pre-order, multi-order management, assisted sales mobile application, Flutter, Django.

## ملخص

في الوقت الحاضر ، لا يزال بيع المنتجات أو أي نوع من البضائع يتم بالطريقة التقليدية (الدفع في مكتب أو كاونتر ، كتيبات ، وما إلى ذلك) ، مما يؤدي إلى خسائر فادحة لبعض المتداولين من حيث الوقت والمال والإرادة يتسبب أي ضا في سوء الفهم وعدم الرضا لدى العمال والموظفين على حد سواء. في هذا السياق ، مبتكراً يأتي مشروعنا لحل مشاكله وإنشاء تقنية وعملية مبيعات مدعومة تمثل حالاً لتحسين تجربة العمال وتعزيز القدرة التنافسية لشركات المبيعات في الجزائر. من خلال دمج ميزات مثل المساعدة الشخصية والتوصيات المستهدفة وإدارة المنتجات الفعالة ، يهدف تطبيقنا إلى تلبية احتياجات العمال وتوقعاتهم ، مع تكييف ممارسات الأعمال مع تغيرات السوق. إنه يوفر إمكانات كبيرة لتحويل صناعة السلع والمنتجات ، وتعزيز اعتماد الحلول التكنولوجية المتقدمة والمساهمة في التنمية الاقتصادية للبلاد. الكلمات الدالة: البيع ، الطلب أو الطلب المسبق ، إدارة الطلبات المتعددة ، تطبيقات الهاتف المحمول للمبيعات المساعدة, Flutter,Django