

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAÏA  
FACULTÉ DES SCIENCES EXACTES  
DÉPARTEMENT D'INFORMATIQUE



MÉMOIRE DE FIN DE CYCLE  
EN VUE DE L'OBTENTION DU DIPLÔME DE MASTER PROFESSIONNEL EN  
INFORMATIQUE  
OPTION : GÉNIE LOGICIEL

**Thème**

Conception et réalisation d'une application web et mobile pour la gestion des commandes et le suivi des livraisons dans le secteur de la restauration

*Présenté par :*

KHETTARI Yassine HAMMICHE Axl

*Soutenu devant le jury composé de :*

<i>Présidente</i>	Dr HAMZA Lamia	M.C.A	U. A/Mira Béjaïa
<i>Examineur</i>	Dr SIDER Abderrahmane	M.C.A	U. A/Mira Béjaïa
<i>Encadrant</i>	Dr ALLEM Khaled	M.A.A	U. A/Mira Béjaïa

# Remerciements

Nous tenons à exprimer notre sincère reconnaissance et notre profonde gratitude aux personnes suivantes qui ont aidé à la production de cet ouvrage :

À Monsieur **Khaled ALLEM**, notre encadrant pour avoir accepté de diriger notre travail, et de nous avoir guidé tout au long de notre projet.

À Monsieur **Hassan ARIOUAT**, le responsable de l'entreprise D-soft pour nous avoir donné cette opportunité de découvrir le monde professionnel .

À tout le cadre enseignant de l'**Université Abderrahmane-Mira** et surtout nos professeurs qui par leur engagement scientifique et éducatif, durant ces cinq années d'études, ont été pour nous une source d'inspiration.

Nos vifs remerciements s'adressent aux membres du jury, pour l'honneur qu'ils nous ont fait en examinant ce mémoire de fin de cycle, soyez assurés de notre respectueuse considération.

Enfin, nous tenons à remercier tous nos proches pour leur soutien ainsi que tous nos collègues pour leur camaraderie tout au long de notre cursus.

# Dédicaces

À nos familles,

Nous dédions ce modeste travail à vous tous avec une profonde gratitude pour votre soutien inconditionnel tout au long de nos études. Votre amour, votre encouragement et votre soutien ont été les piliers essentiels de notre réussite académique. Nous sommes reconnaissant de la patience dont vous avez fait preuve face aux défis que nous avons rencontrés, créant ainsi un environnement propice à notre épanouissement.

À notre encadrant, ALLEM Khaled,

Nous tenons à exprimer une mention spéciale pour votre motivation constante et vos précieux conseils qui ont guidé chaque étape de ce travail. Votre encadrement attentif et vos orientations éclairées ont été d'une importance capitale.

À nos amis,

À Amazigh, Ghani, Nadjim, Lyes et Yacine en particulier, vos conversations enrichissantes ont contribué à élargir l'horizon de cette recherche. À tous mes amis et à toute la promotion Génie Logiciel, nous vous adressons notre reconnaissance pour votre soutien continu.

Enfin, à tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce projet, nous vous exprimons notre profonde gratitude.

Cette réussite est le fruit de notre collaboration et de votre soutien précieux.  
Merci à tous.

# Table des Matières

<b>Table des Matières</b>	<b>i</b>
<b>Table des Figures</b>	<b>iii</b>
<b>Liste des Tableaux</b>	<b>v</b>
<b>Liste des Abréviations</b>	<b>vi</b>
<b>Introduction Générale</b>	<b>1</b>
<b>1 Contexte du projet et Méthodologie de conception</b>	<b>2</b>
1.1 Introduction . . . . .	3
1.2 Contexte et objectif . . . . .	3
1.3 Présentation de l'entreprise d'accueil . . . . .	3
1.4 Étude de l'Existant . . . . .	3
1.4.1 Description des solutions existantes . . . . .	3
1.4.2 Synthèse et discussion des solutions existantes . . . . .	7
1.5 Problématique . . . . .	9
1.6 Solution proposée . . . . .	9
1.7 Applications Web et Mobile . . . . .	9
1.7.1 L'intérêt des applications web et mobiles . . . . .	9
1.8 Méthode de développement . . . . .	10
1.8.1 Processus unifié (UP) . . . . .	10
1.8.2 Langage de modélisation UML . . . . .	11
1.9 Conclusion . . . . .	12
<b>2 Spécification et Analyse des besoins</b>	<b>13</b>
2.1 Introduction . . . . .	14
2.2 Spécification des besoins . . . . .	14
2.2.1 Besoins fonctionnels . . . . .	14
2.2.2 Besoins non fonctionnels . . . . .	14
2.2.3 Identification des acteurs . . . . .	15
2.2.4 Modélisation du contexte . . . . .	15
2.2.5 Identification des cas d'utilisation . . . . .	15
2.2.6 Diagramme de cas d'utilisation global . . . . .	16
2.2.7 Description des cas d'utilisation . . . . .	18
2.3 Analyse des besoins . . . . .	24
2.3.1 Diagramme de séquence système . . . . .	24
2.4 Conclusion . . . . .	31
<b>3 Conception</b>	<b>32</b>
3.1 Introduction . . . . .	33
3.2 Diagrammes de séquence détaillé . . . . .	33
3.2.1 Diagramme de séquence détaillé du cas d'utilisation "S'inscrire" . . . . .	34
3.2.2 Diagramme de séquence détaillé du cas d'utilisation "S'authentifier" . . . . .	35

3.2.3	Diagramme de séquence détaillé du cas d'utilisation "Créer commande" . . . . .	36
3.2.4	Diagramme de séquence détaillé du cas d'utilisation "Consulter commande" . . . . .	37
3.2.5	Diagramme de séquence détaillé du cas d'utilisation "Livrer commande" . . . . .	38
3.2.6	Diagramme de séquence détaillé du cas d'utilisation "Gérer les produits" . . . . .	39
3.2.7	Diagramme de séquence détaillé du cas d'utilisation "Gérer les utilisateurs" . . . . .	40
3.3	Diagramme de classes de conception . . . . .	41
3.4	Dictionnaire de données . . . . .	42
3.5	Base de données NoSql . . . . .	42
3.5.1	Types de base de données NoSql . . . . .	43
3.5.2	Structure de la base de données . . . . .	44
3.6	Conclusion . . . . .	44
<b>4</b>	<b>Réalisation</b> . . . . .	<b>45</b>
4.1	Introduction . . . . .	46
4.2	Langages et Framework de développement . . . . .	46
4.2.1	Visual studio code . . . . .	46
4.2.2	Javascript . . . . .	46
4.2.3	NodeJs . . . . .	46
4.2.4	React . . . . .	47
4.2.5	Tailwind CSS . . . . .	47
4.2.6	ExpressJs . . . . .	47
4.2.7	Capacitor . . . . .	48
4.3	Plateformes de services web . . . . .	48
4.3.1	Services de déploiement et hébergement . . . . .	48
4.3.2	Services de paiement . . . . .	49
4.3.3	Outil de performance et optimisation . . . . .	50
4.4	Implémentation de la base de données . . . . .	50
4.4.1	Intégration et services utilisés . . . . .	50
4.5	Présentation des interfaces de l'application . . . . .	52
4.5.1	Interfaces d'inscription et de connexions pour tous les utilisateurs. . . . .	53
4.5.2	Interfaces principales Client . . . . .	53
4.5.3	Interfaces principales Administrateur . . . . .	57
4.5.4	Interfaces principales Livreur . . . . .	61
4.5.5	Interfaces de l'application mobile . . . . .	62
4.5.6	Résultats du test . . . . .	65
4.6	Conclusion . . . . .	65
	<b>Conclusion Générale et Perspectives</b> . . . . .	<b>66</b>
	<b>Bibliographie</b> . . . . .	<b>67</b>

# Table des figures

1.1	Logo de D-soft . . . . .	3
1.2	Interfaces Wajeez . . . . .	4
1.3	Interfaces de Yassir express . . . . .	5
1.4	Interface de Fast Delivery . . . . .	5
1.5	Interfaces de Deliveroo . . . . .	6
1.6	Interface de UberEats . . . . .	7
1.7	Interfaces de Just Eat takeaway . . . . .	7
1.8	Logo de QuickServe Delivery . . . . .	9
1.9	Enchaînements d'activités au cours du cycle de vie [1] . . . . .	11
1.10	Types de diagrammes UML [2]. . . . .	12
2.1	Diagramme de contexte dynamique . . . . .	16
2.2	Diagramme de cas d'utilisation global . . . . .	17
2.3	Diagramme de séquence système du cas d'utilisation "S'inscrire" . . . . .	25
2.4	Diagramme de séquence système du cas d'utilisation "S'authentifier" . . . . .	26
2.5	Diagramme de séquence système du cas d'utilisation "Créer commande" . . . . .	27
2.6	Diagramme de séquence système du cas d'utilisation "Consulter commande" . . . . .	28
2.7	Diagramme de séquence système du cas d'utilisation "Livrer commande" . . . . .	29
2.8	Diagramme de séquence système du cas d'utilisation "Gérer les produits" . . . . .	30
2.9	Diagramme de séquence système du cas d'utilisation "Gérer les utilisateurs" . . . . .	31
3.1	Diagramme de séquence détaillé "S'inscrire" . . . . .	34
3.2	Diagramme de séquence détaillé "S'authentifier" . . . . .	35
3.3	Diagramme de séquence détaillé du cas d'utilisation "Créer commande" . . . . .	36
3.4	Diagramme de séquence détaillé du cas d'utilisation "Consulter commande" . . . . .	37
3.5	Diagramme de séquence détaillé "Livrer commande" . . . . .	38
3.6	Diagramme de séquence détaillé "Gérer les produits" . . . . .	39
3.7	Diagramme de séquence détaillé "Gérer les utilisateurs" . . . . .	40
3.8	"Diagramme de classes de conception" . . . . .	41
3.9	Structure de la base de données . . . . .	44
4.1	Logo de Vs code . . . . .	46
4.2	Logo de javascript . . . . .	46
4.3	Logo de NodeJs . . . . .	47
4.4	Logo de ReactJS . . . . .	47
4.5	Logo de Tailwind CSS . . . . .	47
4.6	Logo de Express JS . . . . .	48
4.7	Logo de Capacitor . . . . .	48
4.8	Logo de Firebase . . . . .	49
4.9	Logo de Vercel . . . . .	49
4.10	Logo de Stripe . . . . .	49

---

4.11	Logo de chargily pay . . . . .	50
4.12	Logo de Lighthouse . . . . .	50
4.13	Vue d'ensemble du projet Firebase . . . . .	51
4.14	Service Authentification de Firebase . . . . .	51
4.15	Service storage de firebase . . . . .	52
4.16	Service database de firebase . . . . .	52
4.17	Interfaces d'inscription et de connexion . . . . .	53
4.18	Interface d'accueil . . . . .	54
4.19	Produits visibles accueil . . . . .	54
4.20	Interface Commande . . . . .	55
4.21	Interface paiement commande avec stripe . . . . .	55
4.22	Interface paiement de la commande chargily pay . . . . .	56
4.23	Interface commande passée avec succès . . . . .	56
4.24	Interface du tableau de bord des statistiques des produits . . . . .	57
4.25	Liste de toutes les commandes . . . . .	57
4.26	Journal des Paiements effectués avec Stripe . . . . .	58
4.27	Journal des commandes payées avec chargily pay . . . . .	58
4.28	Interface pour la gestion des produits . . . . .	59
4.29	Interface pour l'ajout d'un produit . . . . .	59
4.30	Gestion des utilisateurs . . . . .	60
4.31	Paramètres de gestion de compte . . . . .	60
4.32	Commande prête en attente d'acceptation . . . . .	61
4.33	Commande choisie par un seul livreur . . . . .	61
4.34	Icône de l'application mobile . . . . .	62
4.35	Accueil application mobile Client . . . . .	62
4.36	Panier . . . . .	63
4.37	Accueil admin mobile . . . . .	63
4.38	Interface mobile administrateur de toutes les commandes . . . . .	64
4.39	Panel administrateur . . . . .	64
4.40	Résultats des tests . . . . .	65

# Liste des tableaux

1.1	Comparatif des applications Deliveroo et Yassir Express . . . . .	8
2.1	Messages échangés entre acteurs et système . . . . .	15
2.2	Cas d'utilisations associés . . . . .	16
2.3	Description du cas d'utilisation s'inscrire . . . . .	18
2.4	Description du cas d'utilisation s'authentifier . . . . .	19
2.5	Description du cas d'utilisation Créer commande . . . . .	20
2.6	Description du cas d'utilisation Consulter commande . . . . .	21
2.7	Description du cas d'utilisation Livrer commande . . . . .	22
2.8	Description du cas d'utilisation Gérer les produits . . . . .	23
2.9	Description du cas d'utilisation Gérer les utilisateurs . . . . .	24
3.1	Dictionnaire des données du diagramme de classes . . . . .	42
3.2	Comparaison entre BDD relationnelle et BDD NoSQL . . . . .	43

# Liste des Abréviations

API	Application Programming Interface
CIP	Customer Information Platform
CSS	Cascading Style Sheets
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
iOS	iPhone Operating System
ISO	International Organization for Standardization
JS	JavaScript
JSON	JavaScript Object Notation
JWT	JSON Web Token
NoSQL	Not Only SQL
OS	Operating System
PDF	Portable Document Format
QSD	QuickServe Delivery
SEO	Search Engine Optimization
SQL	Structured Query Language
UI	User Interface
UML	Unified Modeling Language
UP	Unified Process
URL	Uniform Resource Locator

# INTRODUCTION GÉNÉRALE

Depuis quelques années, nous assistons à une transformation profonde des méthodes de travail et des processus, principalement en raison des nouvelles technologies. Aujourd'hui, que ce soit à titre personnel ou professionnel, nous sommes tous concernés par leur utilisation. Elles sont devenues indispensables dans nos vies et dans le fonctionnement des entreprises.

Avant, dans le secteur de la restauration les entreprises géraient leurs livraisons par le biais de documents papier ou de feuilles Excel. Pour assurer le suivi de ces livraisons, elles disposaient d'un service de communication dédié à la relation avec les clients, ce qui pouvait rapidement devenir ingérable lorsqu'il y avait de nombreux clients à servir.

Ce projet vise à tirer parti des nouvelles technologies en développant un système qui facilite la gestion des commandes, la réalisation des livraisons et leur suivi, afin d'atteindre un niveau de satisfaction maximale pour toutes les parties impliquées.

Ce mémoire est structuré en quatre chapitres :

Le premier chapitre, intitulé contexte du projet et méthodologie de conception, nous permettra de décrire le contexte du projet et de faire une étude de l'existant afin de dégager la problématique et proposer une solution. Ensuite, nous présenterons les applications Web, mobile ainsi que le processus de développement de logiciels qui nous facilitera l'élaboration du projet.

Le deuxième chapitre sera consacré à la spécification et à l'analyse des besoins, qui consiste à identifier les différents acteurs et décrire les besoins fonctionnels et non fonctionnels.

Le troisième chapitre sera dédié à la conception de notre application, qui aura pour objectif de détailler la description du système d'un point de vue technique.

Le dernier chapitre sera dédié à la réalisation de l'application, où nous détaillerons son développement.

Enfin, nous conclurons ce travail en résumant les connaissances acquises durant ce projet et nous exposerons quelques perspectives.

# 1

## CONTEXTE DU PROJET ET MÉTHODOLOGIE DE CONCEPTION

## 1.1 Introduction

Dans le marché grandissant que représente le marché algérien pour la livraison des repas des restaurants, plusieurs acteurs se disputent afin de récupérer la plus grande part de marché, la solution que nous proposons se démarque du fait de sa maniabilité de sa facilite d'utilisation. Pour ce faire, nous avons besoin de travailler de manière efficace et concise tout en mettant l'accent sur le produit final. Afin de réaliser notre objectif, nous avons décidé d'employer la méthode UP du fait de son fonctionnement et de la composition de notre équipe.

## 1.2 Contexte et objectif

Notre projet, intitulé 'Conception et réalisation d'une application web et Android pour la gestion des commandes et le suivi des livraisons dans le secteur de la restauration', a été initié par l'entreprise D-soft. L'objectif principal est de faciliter la gestion des commandes et accélérer le processus de livraison pour les établissements de la restauration. Cette application permettra aux gérants de suivre en temps réel toutes les commandes passées par les clients et de gérer efficacement les livraisons.

## 1.3 Présentation de l'entreprise d'accueil

D-soft est une entreprise de développement de solutions logicielles basée à Béjaïa au 22 route des Aurès, elle offre une large palette de solutions de services professionnels et de développement de technique sur mesure. Ses activités se concentrent sur le secteur de la restauration en accompagnant ses différents acteurs a améliorer leurs affaire en proposons des solutions logicielles.



FIGURE 1.1 – Logo de D-soft

## 1.4 Étude de l'Existant

Nous allons réaliser notre étude en énonçons certains des acteurs présents sur le marché national et international et faire une discussion sur les différentes fonctionnalités offertes.

### 1.4.1 Description des solutions existantes

Dans ce qui suit nous allons décrire certains des acteurs présents dans le domaine de la livraison dans le secteur de la restauration, nous commencerons par

les entreprises présentes au niveau national puis nous passerons à celles présentes au niveau international.

#### 1.4.1.1 Au niveau national

Les trois acteurs qui se démarquent sur le marché national sont les suivants :

##### — Wajeez

C'est une startup Algérienne de livraison à la demande. L'application mobile conviviale est disponible en Algérie ainsi que dans la capitale sénégalaise Dakar. Elle met en relation les utilisateurs avec un vaste réseau de restaurants, d'épiceries et d'autres services, leur offrant un accès facile à tout ce dont ils ont besoin dans les plus brefs délais, redéfinissant ainsi la commodité dans l'industrie des services.

Wajeez offre aux utilisateurs la possibilité de rechercher des restaurants en filtrant selon la localisation, le type de cuisine et autres. L'application mobile a été développée en React Native et Node js, la plateforme repose sur une base de donnée NoSQL afin de gérer les grandes quantités de données. Afin d'assurer la sécurité l'application utilise les JWT pour sécuriser les sessions utilisateurs et le chiffrement SSL pour les données sensibles [3].

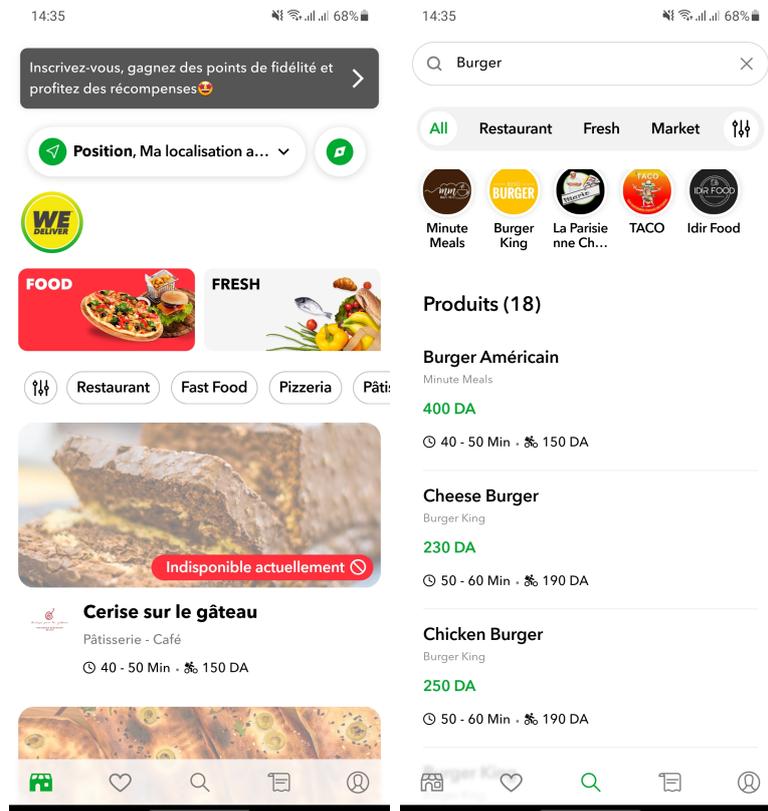


FIGURE 1.2 – Interfaces Wajeez

##### — Yassir express

C'est une entreprise de transport et de livraison qui opère dans plusieurs pays dont la France, le Canada ,la Cote d'Ivoire et surtout en Algérie. Elle propose des services de livraison de repas, de courses, de colis et de transport de personnes. Yassir utilise des applications mobiles pour faciliter la réservation et le suivi des livraisons, ce qui lui permet d'être assez populaire dans la région pour ses services de livraison de repas de restaurants [4].

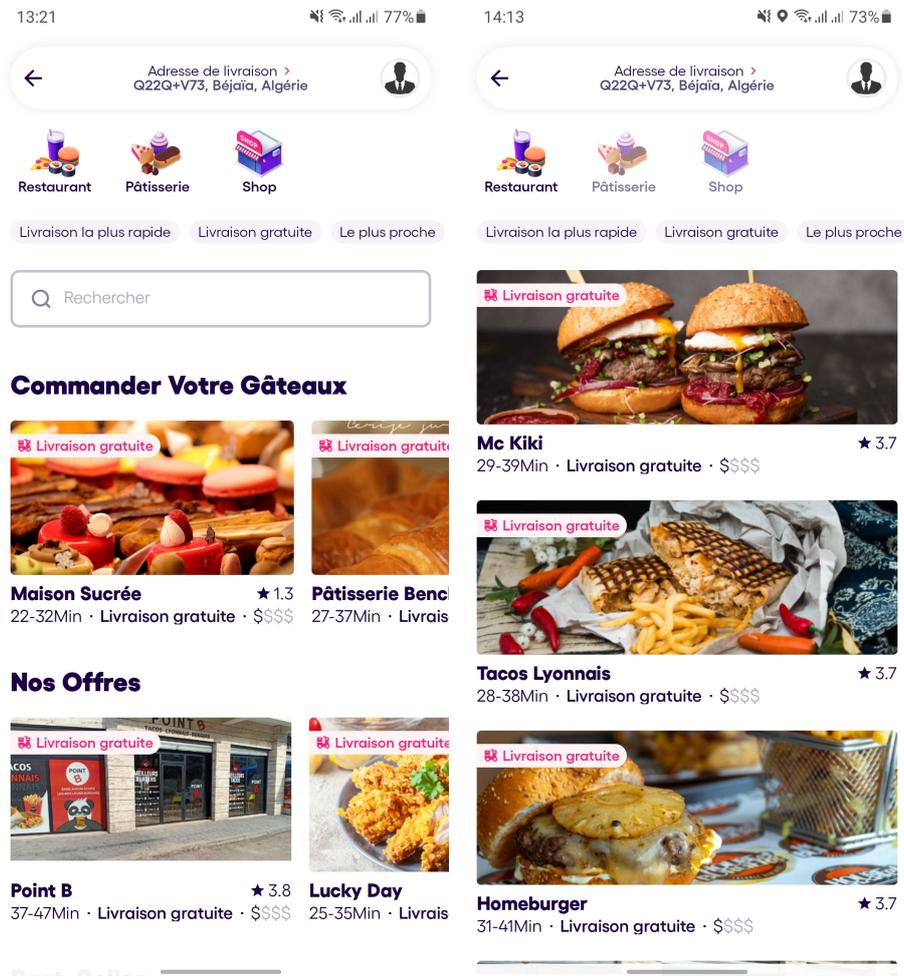


FIGURE 1.3 – Interfaces de Yassir express

### — Fast Delivery

C'est une entreprise de livraison de repas basée à Alger, en Algérie. Ils offre une plateforme de commande et de livraison rapide et pratique pour les clients, tout en offrant des opportunités pour les restaurants locaux de développer leur activité. Ils représentent une entreprise en pleine croissance, avec une équipe passionnée et déterminée à offrir les meilleurs services à leur clients et partenaires [5].

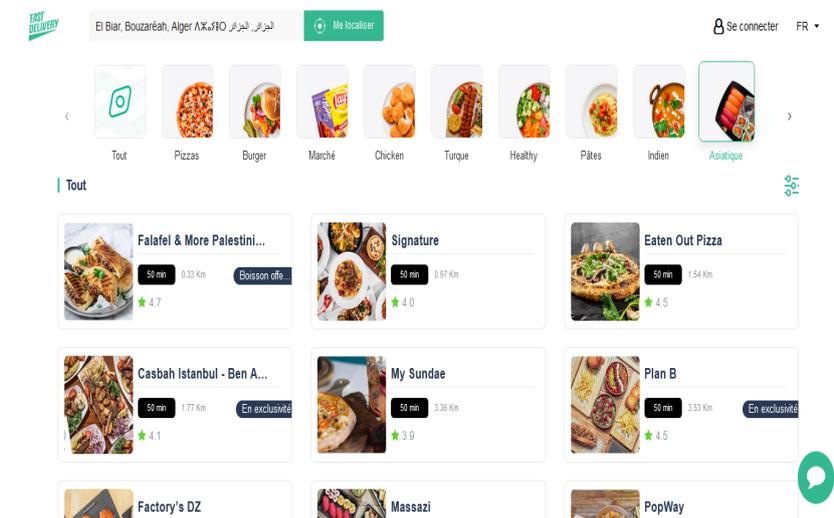


FIGURE 1.4 – Interface de Fast Delivery



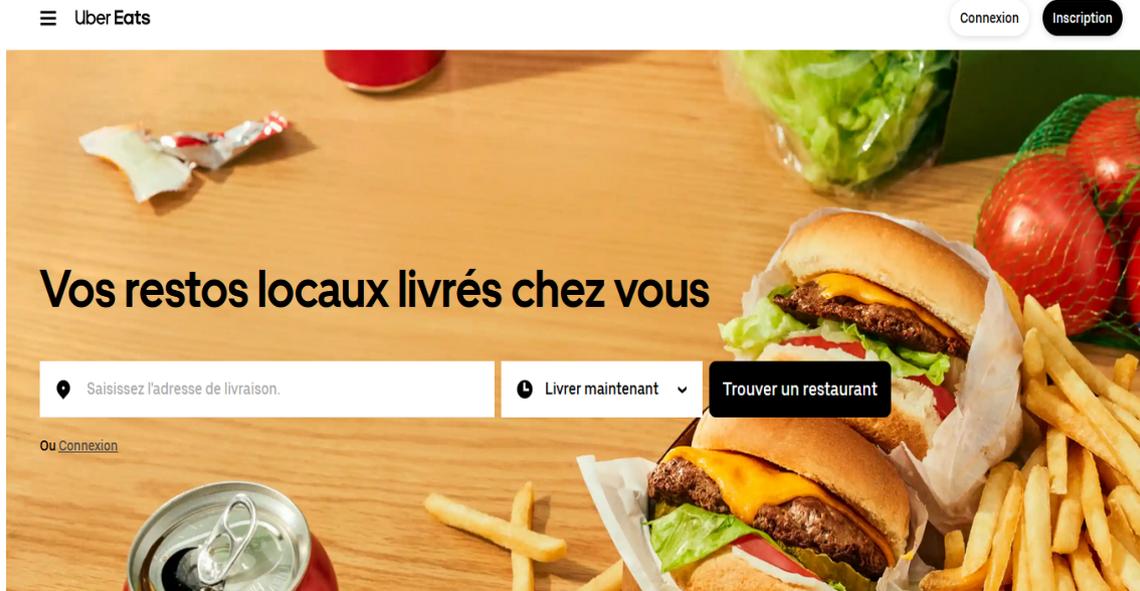


FIGURE 1.6 – Interface de UberEats

### — Just Eat takeaway

Just Eat Takeaway, anciennement Takeaway.com, est une entreprise franco-néerlandaise spécialisée dans la restauration à domicile avec commande en ligne. Le service proposé par Takeaway.com prend la forme d'un site internet répertoriant des restaurants et leurs menus, et permettant aux internautes de commander leur repas en ligne et d'être livré à domicile. La société est présente dans plusieurs pays européens [8].

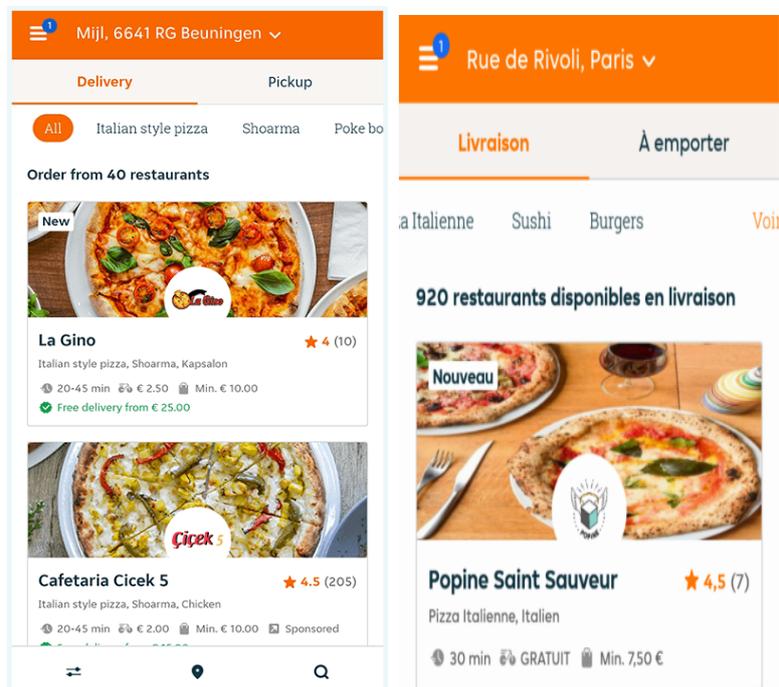


FIGURE 1.7 – Interfaces de Just Eat takeaway

## 1.4.2 Synthèse et discussion des solutions existantes

Après étude de certains des acteurs présents sur le marché national et international, donnant les descriptions techniques et processus de fonctionnement des divers applications, voici un tableau comparatif de deux acteurs, l'un présent sur le marché national et l'autre présent sur le marché international :

Critère	Deliveroo	Yassir Express
<b>Fonctionnalités</b>	<ul style="list-style-type: none"> <li>— Large choix de restaurants</li> <li>— Suivi en temps réel</li> <li>— Promotions et offres spéciales</li> <li>— Paiement en ligne sécurisé</li> </ul>	<ul style="list-style-type: none"> <li>— Couverture locale étendue</li> <li>— Livraisons rapides</li> <li>— Suivi de commande en temps réel</li> <li>— Options de paiement multiples</li> </ul>
<b>Interface Utilisateur (UI/UX)</b>	<ul style="list-style-type: none"> <li>— Interface intuitive</li> <li>— Navigation fluide</li> <li>— Design moderne</li> <li>— Application mobile disponible</li> </ul>	<ul style="list-style-type: none"> <li>— Interface conviviale</li> <li>— Navigation simple et efficace</li> <li>— Design épuré</li> <li>— Application mobile réactive</li> </ul>
<b>Technologie</b>	<ul style="list-style-type: none"> <li>— Algorithmes d'optimisation des livraisons</li> <li>— Intégration API</li> <li>— Support multi-plateforme</li> </ul>	<ul style="list-style-type: none"> <li>— Technologie de géolocalisation avancée</li> <li>— Infrastructure cloud</li> <li>— Intégration avec des systèmes de paiement locaux</li> </ul>
<b>Sécurité</b>	<ul style="list-style-type: none"> <li>— Cryptage des données grace a SSL</li> <li>— Stockage des données conforme aux normes tel que ISO27001</li> <li>— Authentification à deux facteurs</li> </ul>	<ul style="list-style-type: none"> <li>— Sécurisation des paiements</li> <li>— Chiffrement des données</li> <li>— Contrôle d'accès strict</li> </ul>
<b>Performance</b>	<ul style="list-style-type: none"> <li>— Temps de réponse rapide</li> <li>— Stabilité de l'application</li> <li>— Gestion efficace du trafic</li> </ul>	<ul style="list-style-type: none"> <li>— Haute disponibilité</li> <li>— Faible latence</li> <li>— Performance optimale même pendant les heures de pointe</li> </ul>

TABLE 1.1 – Comparatif des applications Deliveroo et Yassir Express

## 1.5 Problématique

Le secteur de la restauration fait face de plus en plus à une surcharge de travail et de nouveaux défis auxquels il doit faire face. Dans ce contexte les entreprises se doivent d'innover ou du moins assurer un service de base comparable aux concurrents pour rester compétitives.

Cette étude vise à établir l'ensemble des fonctionnalités et services minimums que se doit d'avoir toute solution voulant intégrer le marché .

C'est dans ce cadre que s'inscrit notre projet de gestion des commandes et de suivi des livraisons, qui vise à proposer une solution équivalente et évolutive pour aider les entreprises à gérer efficacement leur activité.

## 1.6 Solution proposée

Après l'étude et l'analyse des applications existantes ,en prenant en compte l'étude comparative précédente .Nous avons conçu **Quick Serve delivery** qui est une application web et mobile qui offre les services suivante :

- Une gestion simplifiée des commandes et des utilisateurs
- Offrir la possibilité du suivi GPS à nos clients.
- Diverses méthodes de paiement disponibles.

La figure ci-dessous représente le logo de notre application.

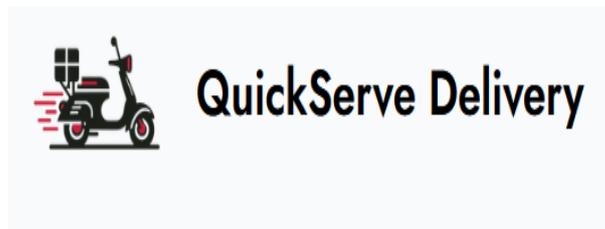


FIGURE 1.8 – Logo de QuickServe Delivery

## 1.7 Applications Web et Mobile

Dans le domaine de l'informatique, une application web, également appelée site web dynamique ou WebApp, est un programme accessible via un navigateur web, utilisant des langages comme HTML, JavaScript et CSS. Hébergée sur un serveur, elle ne nécessite pas d'installation sur l'appareil de l'utilisateur et peut être accessible directement via un signet ou un raccourci. En revanche, une application mobile est conçue pour fonctionner sur des appareils mobiles tels que les smartphones et les tablettes. Développée pour exploiter les fonctionnalités spécifiques des appareils mobiles, elle utilise des écrans tactiles, des capteurs (comme le GPS ou l'accéléromètre) et les caméras [9].

### 1.7.1 L'intérêt des applications web et mobiles

Les applications mobiles et web sont cruciales pour les entreprises, améliorant l'expérience client et augmentant les ventes. Les applications mobiles collectent des données directes, améliorent le taux de conversion et augmentent la fidélisation grâce aux notifications. Les applications web, grâce à l'évolution technologique, offrent désormais presque toutes les fonctionnalités des applications natives, avec un accès rapide via divers navigateurs, une compatibilité multi-plateforme, et la possibilité de travailler dans le cloud.

- Un accès rapide via une variété de navigateurs et la compatibilité avec tous les systèmes d'exploitation.
- Accessibilité depuis n'importe où, avec la possibilité de travailler simultanément dans le "Cloud".
- Une sécurité accrue pour les contenus.
- Collecte de données directes auprès des utilisateurs, permettant de personnaliser les offres et d'améliorer les ventes.
- Amélioration du taux de conversion en simplifiant le processus d'achat sur mobile.
- Augmentation de la fidélisation grâce aux notifications instantanées et à la communication directe avec les clients.
- Expérience utilisateur homogène et personnalisée, améliorant la satisfaction et les ventes grâce à une stratégie omnicanale.
- Notifications poussées pour informer les clients des offres spéciales, nouveautés et promotions, stimulant ainsi le trafic et les ventes.

## 1.8 Méthode de développement

Une méthode définit une séquence d'étapes, partiellement ordonnées, qui contribuent à l'obtention d'un système ou à l'évaluation d'un système existant. L'objectif d'une méthode de développement est de produire des logiciels de qualité répondant aux besoins de leurs utilisateurs dans des délais et des coûts prévisibles.

### 1.8.1 Processus unifié (UP)

Le Processus Unifié (UP) est une famille de méthodes de développement de logiciels orientés objets. Il se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML. Il définit un processus intégrant toutes les activités de conception et de réalisation au sein de cycles de développement composés d'une phase de création, d'une phase d'élaboration, d'une phase de construction et d'une phase de transition, comprenant chacune plusieurs itérations [10].

Le processus Unifié se caractérise par les propriétés suivantes :

- **Itératif et incrémental** : le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. À la fin de chaque itération, une partie exécutable du système final est produite de façon incrémentale.
- **Centré sur l'architecture** : tout système complexe doit être découpé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
- **Piloté par les risques** : les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- **Conduit par les cas d'utilisation** : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

Le processus Unifié est défini par les phases du cycle de vie comme suit :

- La phase d'initialisation : elle permet de définir la vision du projet, sa portée, sa faisabilité, son business case, afin de pouvoir décider au mieux de sa poursuite ou de son arrêt.

- La phase d'élaboration : elle poursuit trois objectifs principaux en parallèle : identifier et décrire la majeure partie des besoins des utilisateurs, construire l'architecture de base du système, et lever les risques majeurs du projet.
- La phase de construction : elle consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en effort.
- La phase de transition : elle permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux.

Le processus Unifié propose les enchaînements d'activités suivants représentés dans la figure ci-dessous :

- Exigences : présente le système du point de vue de l'utilisateur et recense les besoins fonctionnels et non fonctionnels.
- Analyse : définit une spécification complète des besoins issus des cas d'utilisation et une compréhension des exigences du client.
- Conception : décrit les différentes vues (fonctionnelles, dynamique et statique) d'une architecture, à travers un diagramme de classe et d'interaction, etc.
- Implémentation : implémentation des résultats de conception, intégration du système et des classes.
- Test : planification et mise en œuvre des tests.

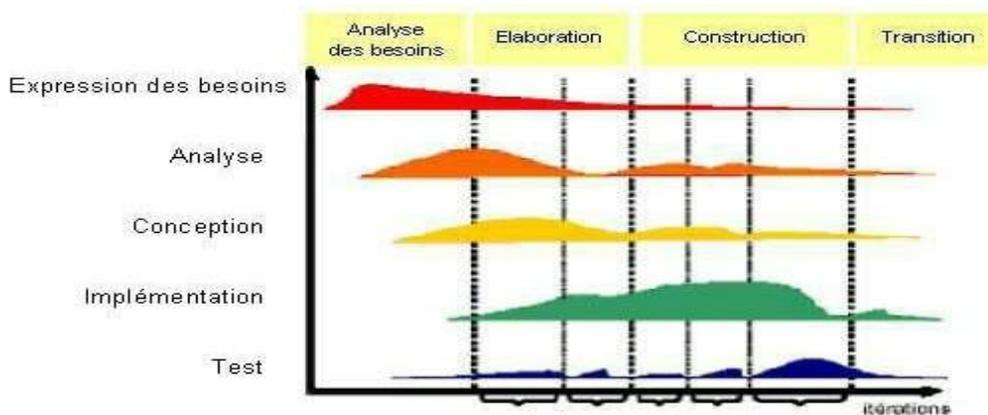


FIGURE 1.9 – Enchaînements d'activités au cours du cycle de vie [1]

## 1.8.2 Langage de modélisation UML

l'UML (Unified Modeling Language) est un langage de modélisation standardisé utilisé principalement pour spécifier, visualiser, construire et documenter les artefacts d'un système logiciel. Pour modéliser une application web spécifiquement, l'UML permet de représenter graphiquement les différents aspects fonctionnels et structurels de l'application, y compris les interactions entre ses composants, ses utilisateurs et ses différents modules [11].

UML 2.5 définit 14 types de diagrammes divisés en deux catégories, les diagrammes structurels représentant une vue statique du système, et les diagrammes comportementaux concernant le comportement dynamique du système.

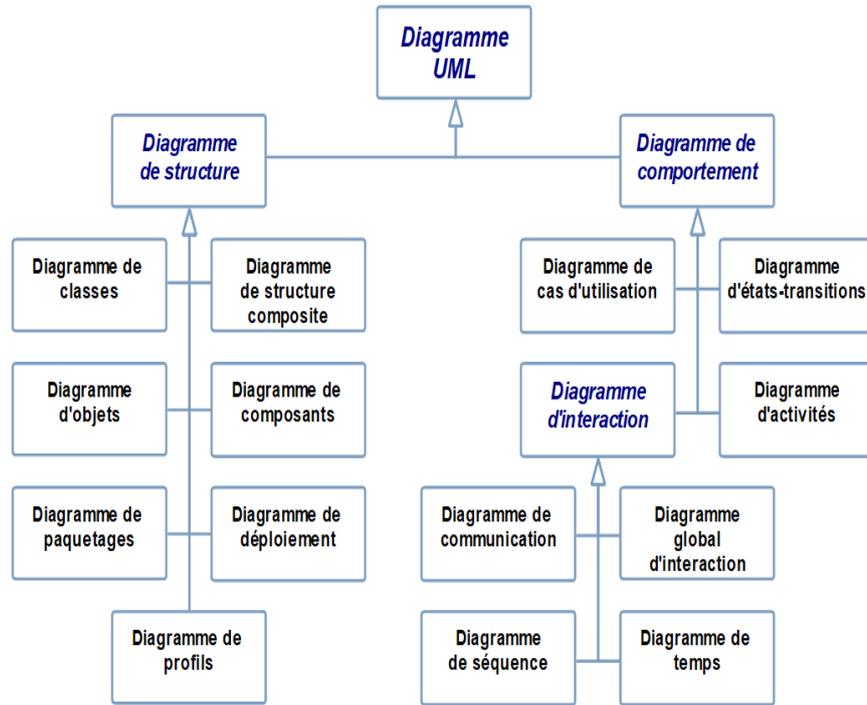


FIGURE 1.10 – Types de diagrammes UML [2].

Dans notre projet nous allons utiliser seulement les diagrammes suivants dont la définition est définie comme tel :

- **Diagramme de cas d'utilisation** : représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est utilisé dans les deux étapes de capture des besoins fonctionnels et techniques.
- **Diagramme de séquence** : est un diagramme d'interaction, il représente les échanges de messages entre objets, dans le cadre d'un fonctionnement particulier du système. Ils servent ensuite à développer en analyse les scénarios d'utilisation du système.
- **Diagramme de classe** : a toujours été le plus important dans toutes les méthodes orientées objet. C'est également celui qui contient la plus grande gamme de notations.

## 1.9 Conclusion

Dans ce chapitre, nous avons décrit le contexte général du projet et mené une étude de l'existant, en examinant les solutions disponibles sur le marché, tant au niveau national qu'international. Cette analyse nous a permis de définir le cadre du projet et de mettre en évidence les besoins non satisfaits, justifiant ainsi le développement de notre solution. Nous allons maintenant passer à la spécification et à l'analyse des besoins, où nous détaillerons les fonctionnalités et exigences nécessaires pour atteindre nos objectifs.

# 2

## SPÉCIFICATION ET ANALYSE DES BESOINS

## 2.1 Introduction

Ce chapitre marque le début de la phase d'analyse des besoins. Durant cette étape, notre objectif est d'abord d'identifier les différents acteurs du système et de comprendre les besoins fonctionnels qu'ils expriment. Ensuite, nous nous penchons sur la spécification des besoins non fonctionnels ainsi que sur leurs contraintes, ce qui nous permettra de créer le diagramme de cas d'utilisation global.

## 2.2 Spécification des besoins

La spécification des besoins nous donne une vision globale sur le projet, à travers l'identification des fonctionnalités principales à implémenter.

### 2.2.1 Besoins fonctionnels

Les besoins fonctionnels correspondent aux fonctionnalités que notre application doit offrir. Elles se résument en les fonctionnalités suivantes :

- L'inscription à l'application : Utiliser l'application nécessite une inscription où les informations de l'utilisateur sont enregistrées.
- Gestion des comptes : Les informations personnelles d'un compte utilisateur peuvent être modifiées.
- Création de la commande : le client peut créer une commande en ajoutant des produits.
- Suivi de la livraison de la commande : L'application doit permettre de suivre son état d'avancement par un livreur.
- Gestion des comptes utilisateurs : l'administrateur peut supprimer un compte utilisateur, transformer un compte client en compte livreur.
- Gestion des commandes : L'administrateur peut gérer l'état de la commande et son déroulement.
- Gestion des produits : L'administrateur peut ajouter, supprimer et modifier le prix d'un produit

### 2.2.2 Besoins non fonctionnels

Les besoins non fonctionnels n'ont pas de rapport direct avec le système mais ont un impact visuel important pour l'utilisateur. Ils représentent des exigences qu'on doit satisfaire quant aux contraintes techniques et aux besoins exprimés.

- Performance : l'application doit être rapide et réactive pour assurer une expérience utilisateur fluide, même lors de pics de demande. grâce à react et express js qui permettent des mises à jour rapides assurant une forte performance .
- Sécurité : les données des utilisateurs, y compris les informations de paiement, doivent être sécurisées et protégées contre les accès non autorisés. assurer grâce à l'utilisation de firebase qui est un service sécurisé de Google, ainsi qu'une authentification sécurisée grâce à JWT [12] .
- Disponibilité : l'application doit être disponible 24h/24 et 7j/7 pour permettre aux utilisateurs de commander à tout moment, grâce à Vercel qui est une plateforme de déploiement à travers ses différentes fonctionnalités.
- Compatibilité : l'application doit fonctionner sur différentes plateformes et appareils mobiles (iOS, Android, tablettes, etc.).
- Convivialité : L'interface utilisateur doit être intuitive et facile à utiliser, même pour les utilisateurs novices.

- Internationalisation : l'application doit prendre en charge plusieurs devises pour s'adapter aux besoins des utilisateurs internationaux, grâce notamment à stripe.
- Documentation : une documentation complète et claire doit être fournie pour les utilisateurs et les développeurs afin de faciliter l'utilisation et la maintenance de l'application.

### 2.2.3 Identification des acteurs

Un acteur représente une entité externe qui interagit avec le système ou l'application informatique en question. Les acteurs sont généralement des utilisateurs, des systèmes externes ou d'autres entités qui interagissent avec le système pour accomplir des tâches spécifiques.

Notre application contient 4 acteurs principaux qui sont les suivants :

- Visiteur : représente un acteur pouvant naviguer sur le site sans compte et effectuer une inscription.
- Client : représente l'utilisateur après son inscription et de facto pouvant créer une ou plusieurs commandes et les suivre.
- Livreur partenaire : personne physique en possession du permis effectuant les livraisons pour le compte du restaurateur après s'être authentifié.
- Administrateur : personne qui valide l'enregistrement des utilisateurs ayant accès à tous les comptes utilisateurs et pouvant de facto sanctionner certains et qui peut modifier le menu.

### 2.2.4 Modélisation du contexte

Le tableau ci-dessous illustre les différents messages échangés entre l'utilisateur et le système :

Messages Acteurs -> Système	Messages Système -> Acteurs
M1 Demande d'inscription	M1' Interface d'inscription
M2 Demande d'authentification	M2' Interface d'authentification
M3 Gérer compte utilisateur	M3' Interface gestion compte utilisateur
M4 Créer commande	M4' Affichage de la commande avec prix
M5 Sélectionner commande	M5' Interface des commandes sélectionnés
M6 Suivre les livraisons	M6' Interface de suivi des livraisons
M7 Supprimer les comptes utilisateurs	M7' Interface de suppression des comptes utilisateurs
M8 Gérer les produits	M8' Interface de gestion des produits
M9 Gérer l'état des commandes	M9' Interface de gestion de l'état des commandes

TABLE 2.1 – Messages échangés entre acteurs et système

Le diagramme suivant est un diagramme de contexte représentant les différents acteurs du système et les messages échangés entre ces derniers mentionnés précédemment.

### 2.2.5 Identification des cas d'utilisation

Un cas d'utilisation (use case) représente une séquence d'actions réalisées par le système est observable pour un acteur particulier.

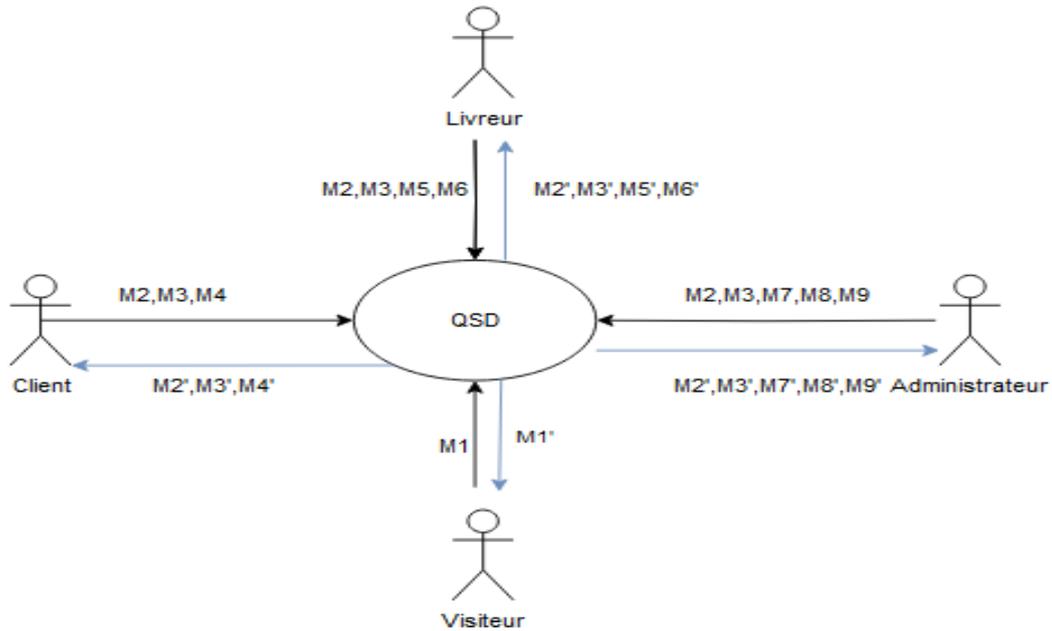


FIGURE 2.1 – Diagramme de contexte dynamique

N <sup>o</sup>	Cas d'utilisation	Acteur
1	S'inscrire	Visiteur
2	Consulter les produits	Visiteur, Client, Livreur, Administrateur
3	S'authentifier	Client, Livreur, Administrateur
4	Gérer son compte	Client, Livreur, Administrateur
5	Gestion des utilisateurs	Administrateur
6	Créer une commande (choisir son menu)	Client
7	Consulter Commande	Client
8	Livrer une commande client	Livreur
9	Gérer l'état de la commande	Administrateur
10	Gérer les produits	Administrateur

TABLE 2.2 – Cas d'utilisations associés

### 2.2.6 Diagramme de cas d'utilisation global

Dans cette section, il s'agit de présenter les différents cas d'utilisation propres à chaque acteur dans un diagramme global tel qu'illustré dans la figure suivante .

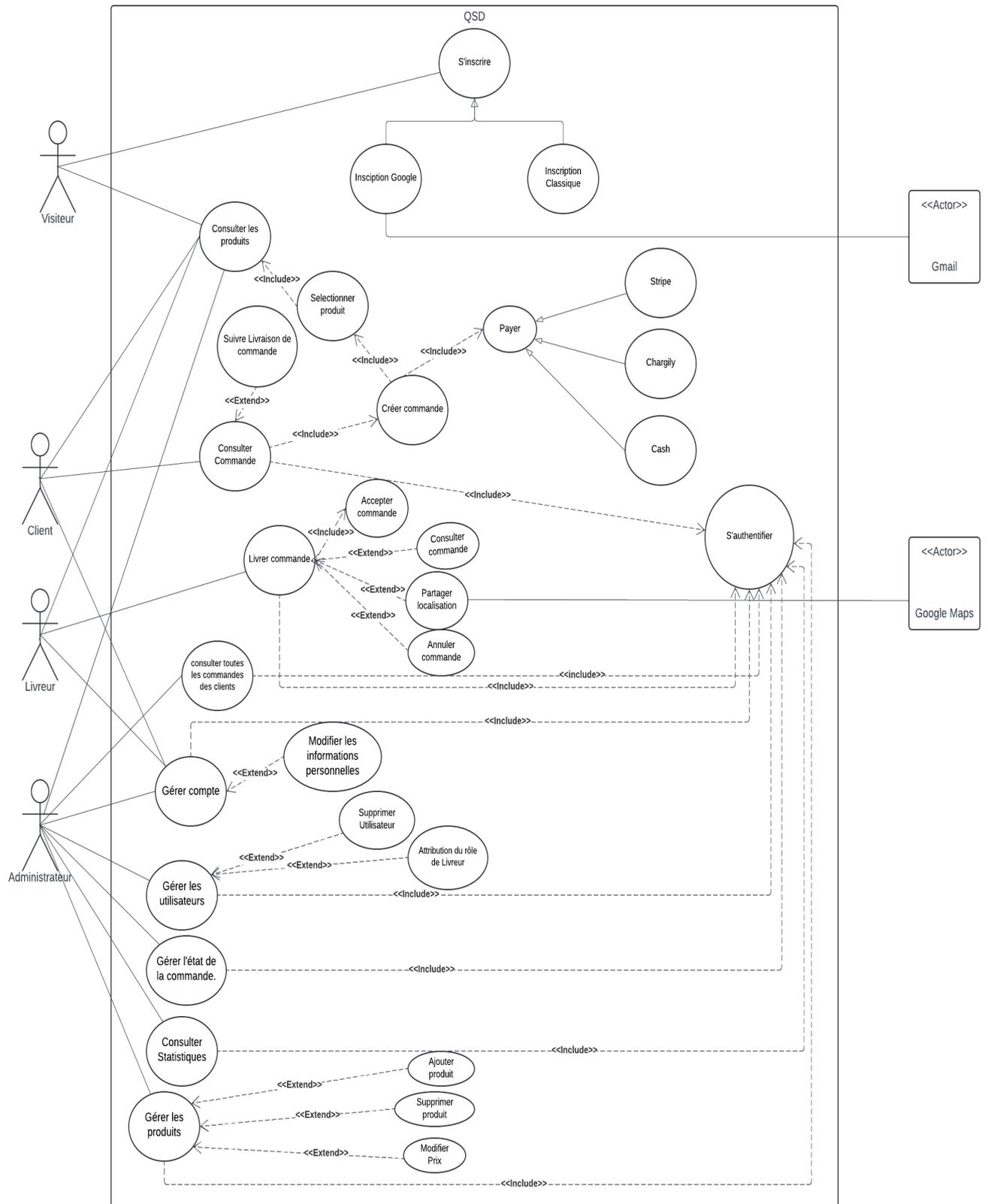


FIGURE 2.2 – Diagramme de cas d'utilisation global

## 2.2.7 Description des cas d'utilisation

Nous allons présenter une description textuelle des cas d'utilisation les plus importants.

### Description du cas d'utilisation S'inscrire :

Sommaire	
Titre	S'inscrire.
Résumé	Inscription d'un visiteur.
Acteur	Visiteur.
Description des scénarios	
Pré condition	Application accessible.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le visiteur accède a l'interface d'inscription</li> <li>2.le visiteur saisit le nom,le prénom ,l'email et le mot de passe puis valide.</li> <li>3. Le système vérifie la saisie .</li> <li>4. Le système vérifie la validité des données .</li> <li>5. Le système affiche compte créé avec succès .</li> <li>6. Le système incite le visiteur a se connecter.</li> </ol>
Scénario alternatif	<p><b>A1 : Erreur de saisie (champs vide )</b>. Le scénario démarre après le point 3 du scénario nominal :</p> <ol style="list-style-type: none"> <li>4. Le système affiche que un ou plusieurs champs sont vide.</li> <li>5.Le système invite le visiteur a remplir les champs vide.</li> </ol> <p>le scénario nominal reprend au point 2.</p> <p><b>A2 : Erreur de saisie (e-mail et mot de passe non valide )</b>. Le scénario démarre après le point 4 du scénario nominal :</p> <ol style="list-style-type: none"> <li>5. Le système affiche que l'email et le mot de passe ne sont pas valide.</li> <li>6.Le système invite le visiteur a vérifier son e-mail et mot de passe.</li> </ol> <p>le scénario nominal reprend au point 2.</p>
Scénario d'exception	<p><b>E1 : Données erronées (e mail inexistant)</b>. Le scénario démarre après le point 4 du scénario nominal :</p> <ol style="list-style-type: none"> <li>5. Le système affiche inscription impossible,veuillez vérifier votre e-mail</li> </ol>
Post conditions	Compte utilisateur créé.

TABLE 2.3 – Description du cas d'utilisation s'inscrire

## Description du cas d'utilisation S'authentifier :

<b>Sommaire</b>	
Titre	S'authentifier
Résumé	Authentifier les utilisateurs connectés.
Acteur	Client, Livreur, Administrateur.
<b>Description des scénarios</b>	
Pré condition	L'utilisateur doit posséder un compte . application accessible.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur accède a l'interface d'authentification.</li> <li>2. L'utilisateur saisit l'e-mail et le mot de passe puis valide.</li> <li>3. Le système vérifie la saisie .</li> <li>4. Le système vérifie la validité des données .</li> <li>5. Le système Affiche authentification réussie.</li> <li>6. Le compte client s'affiche en haut a droite de la page d'accueil</li> </ol>
Scénario alternatif	<p><b>A1 : Erreur de saisie (champs vide )</b>. Le scénario démarre après le point 3 du scénario nominal : 4. Le système indique qu'il y a des champs non remplis. Le scénario nominal reprend au point 2.</p> <p><b>A2 : Erreur de saisie ( e-mail non valide)</b>. Le scénario démarre après le point 4 du scénario nominal : 5. Le système indique que l'email n'est pas valable. Le scénario nominal reprend au point 2.</p>
Scénario d'exception	<p><b>E1 : Données erronées (mot de passe incorrect ou compte utilisateur inexistant)</b>. Le scénario démarre après le point 4 du scénario nominal : 5. Le système affiche authentification impossible.</p>
Post conditions	Authentification réussie.

TABLE 2.4 – Description du cas d'utilisation s'authentifier

Description du cas d'utilisation Créer commande :

<b>Sommaire</b>	
Titre	Créer commande.
Résumé	Création d'une commande client.
Acteur	Client.
<b>Description des scénarios</b>	
Pré condition	Authentification.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le client cherche les produits désirés.</li> <li>2. Le client les ajoute a son panier et valide.</li> <li>3. Le système vérifie les données.</li> <li>4. Le système affiche la commande en entier .</li> <li>5. Le système affiche le montant de la commande avec l'option de paiement.</li> <li>6. Le client choisit l'option de paiement et valide.</li> <li>7. Le système vérifie les données.</li> <li>8. Le système affiche commande passée.</li> <li>9. Le système renvoie le client a l'accueil</li> </ol>
Scénario alternatif	<p><b>A1 : Produit indisponible .</b>                      Le scénario démarre après le point 3 du scénario nominal.                      4.Le système affiche que un produit sélectionné est indisponible.                      5.Le système invite le client a refaire sa commande.                      Le scénario nominal reprend au point 2.</p> <p><b>A2 : Quantité du produit désirée insuffisante.</b>                      Le scénario démarre après le point 3 du scénario nominal :                      4.Le système affiche que la quantité du produit désirée dépasse la quantité disponible.                      5. Le système invite le client a refaire sa commande.                      Le scénario nominal reprend au point 2.</p>
Scénario d'exception	<p><b>E1 : Données erronées (option de paiement indisponible ).</b>                      Le scénario démarre après le point 7 du scénario nominal :                      8. Le système affiche la commande ne peut être validée.</p>
Post conditions	Commande créée avec succès et Paiement possible.

TABLE 2.5 – Description du cas d'utilisation Créer commande

## Description du cas d'utilisation Consulter commande

Sommaire	
Titre	Consulter commande.
Résumé	Consultation d'une commande client créée.
Acteur	Client.
Description des scénarios	
Pré condition	Authentification
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le client clique sur l'interface de consultation.</li> <li>2. Le système affiche la ou les commandes déjà effectuées.</li> <li>3. Le client sélectionne une commande spécifique.</li> <li>4. Le système vérifie les données.</li> <li>5. Le système affiche les détails de la commande spécifique .</li> <li>6. Le système affiche l'interface de consultation .</li> </ol>
Scénario alternatif	<p><b>A1 : la commande appartient au client,les données ont été altérées.</b></p> <p>Le scénario démarre après le point 4 du scénario nominal :</p> <ol style="list-style-type: none"> <li>5. certaines informations sur la commande ont été altérés.</li> <li>6. Le système invite le client a envoyer un message pour notifier l'administrateur</li> </ol> <p>Le scénario nominal reprend au point 6.</p>
Scénario d'exception	<p><b>E1 : Commandes n'appartenant pas au client .</b></p> <p>Le scénario démarre après le point 4 du scénario nominal :</p> <ol style="list-style-type: none"> <li>5. Le système affiche le message que la commande est erronée.</li> </ol>
Post conditions	Commande consultée.

TABLE 2.6 – Description du cas d'utilisation Consulter commande

## Description du cas d'utilisation Livrer commande :

Sommaire	
Titre	Livrer commande.
Résumé	Acceptation et livraison d'une commande client.
Acteur	Livreur.
Description des scénarios	
Pré condition	Authentification.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le livreur accède a l'interface de consultation des commandes disponibles.</li> <li>2. Le livreur choisit une commande spécifique.</li> <li>3. Le système sélectionne la commande choisie .</li> <li>4. Le système vérifie la validité des données .</li> <li>5. Le système affiche la commande choisie dans l'interface des commandes actives.</li> <li>6. Une fois la commande livrée le livreur clique sur commande effectuée</li> <li>7. Le système affiche l'interface des commandes disponibles</li> </ol>
Scénario alternatif	<p><b>A1 :Le système enregistre la mauvaise commande.</b></p> <p>Le scénario démarre après le point 4 du scénario nominal :</p> <ol style="list-style-type: none"> <li>5. Le système enregistre la mauvaise commande</li> <li>6. Le système affiche un message d'erreur</li> </ol> <p>Le scénario nominal reprend au point 3.</p>
Scénario d'exception	<p><b>E1 : La commande est incorrecte.</b></p> <p>Le scénario démarre après le point 4 du scénario nominal :</p> <ol style="list-style-type: none"> <li>5. Le système affiche le message la commande ne peut être livrée</li> </ol>
Post conditions	Commande en cours de livraison.

TABLE 2.7 – Description du cas d'utilisation Livrer commande

Description du cas d'utilisation Gérer les produits :

Sommaire	
Titre	Gérer les utilisateurs
Résumé	Permettre à l'administrateur d'influer sur les comptes utilisateurs
Acteur	Administrateur
Description des scénarios	
Pré-conditions	Authentification
Scénario nominal	<p>1. L'administrateur accède à l'interface de gestion des produits.</p> <p>2. L'administrateur choisit l'action souhaitée :</p> <p>Option 1 : Ajouter un produit.</p> <p>2.1. Le système affiche le formulaire pour l'ajout du produit.</p> <p>2.2. L'administrateur remplit le formulaire d'ajout d'un produit.</p> <p>2.3. L'administrateur soumet le formulaire.</p> <p>2.4. Le système vérifie les informations fournies.</p> <p>2.5. Le système ajoute le produit à la base de données.</p> <p>2.6 Le système affiche un message de confirmation de l'ajout du produit.</p> <p>Option 2 : Supprimer un produit.</p> <p>2.7. L'administrateur sélectionne le produit à supprimer.</p> <p>2.8. L'administrateur clique sur l'option "Supprimer".</p> <p>2.9. Le système demande une confirmation de suppression.</p> <p>2.10. L'administrateur confirme la suppression.</p> <p>2.11. Le système supprime le produit de la base de données.</p> <p>Option 3 : Modifier le prix du produit.</p> <p>2.12. L'administrateur sélectionne le produit a modifier le prix.</p> <p>2.13. L'administrateur saisit le nouveau prix.</p> <p>2.14. L'administrateur confirme la saisie.</p> <p>2.15. Le système mets a jour le prix du produit.</p> <p>3. Le système affiche l'interface de gestion des produits.</p>
Scénario alternatif	<p><b>A1 : L'ajout du produit ne passe pas</b></p> <p>Le scénario démarre après le point 2.4 du scénario nominal :</p> <p>2.5. Le système informe l'administrateur des informations sont incorrectes.</p> <p>2.6. Le système affiche ajout impossible.</p> <p>Le scénario nominal reprend au point 3.</p>
Scénario d'exception	<p><b>E1 : L'administrateur supprime le produit sans confirmation.</b></p> <p>Le scénario démarre après le point 2.8 du scénario nominal :</p> <p>2.9. Le système supprime le produit sans confirmation.</p> <p>2.10. Le système affiche que le produit est supprimé.</p>
Post conditions	Informations de certains comptes utilisateurs changées.

TABLE 2.8 – Description du cas d'utilisation Gérer les produits

## Description du cas d'utilisation Gérer les utilisateurs :

Sommaire	
Titre	Gérer les utilisateurs
Résumé	Permettre à l'administrateur d'influer sur les comptes utilisateurs
Acteur	Administrateur
Description des scénarios	
Pré-conditions	Authentification
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur accède à l'interface de gestion des utilisateurs.</li> <li>2. L'administrateur choisit un compte client.</li> <li>3. Le système affiche les options de gestion pour les comptes clients.</li> <li>4. L'administrateur choisit l'action souhaitée : Option 1 : Modifier un compte client en compte livreur.                             <ol style="list-style-type: none"> <li>4.1. Le système vérifie le compte.</li> <li>4.2. L'administrateur confirme le changement.</li> <li>4.3. Le système change le rôle du compte client en livreur.</li> </ol>                             Option 2 : Supprimer un compte utilisateur.                             <ol style="list-style-type: none"> <li>4.4. L'administrateur confirme la suppression.</li> <li>4.5. Le système supprime le compte utilisateur de la base de données.</li> </ol> </li> <li>5. Le système informe l'administrateur des modifications effectuées.</li> <li>6. Le système renvoie l'interface de gestion des utilisateurs.</li> </ol>
Scénario alternatif	<p><b>A1 : Le changement ne s'effectue pas</b></p> <p>Le scénario démarre après le point 4.1 du scénario nominal :</p> <ol style="list-style-type: none"> <li>4.2. Le système informe l'administrateur que le compte ne peut être modifié.</li> <li>4.3. Le système demande à l'administrateur de vérifier les propriétés du compte.</li> </ol> <p>Le scénario nominal reprend au point 6.</p>
Scénario d'exception	<p><b>E1 : L'administrateur supprime le mauvais utilisateur.</b></p> <p>Le scénario démarre après le point 4.4 du scénario nominal :</p> <ol style="list-style-type: none"> <li>4.5. Le système supprime le mauvais compte.</li> </ol>
Post conditions	Informations de certains comptes utilisateurs changées.

TABLE 2.9 – Description du cas d'utilisation Gérer les utilisateurs

## 2.3 Analyse des besoins

L'analyse des besoins permet de définir les exigences du système avant de passer à la conception. C'est ce qui a permis entre autre de définir les diagrammes de séquence système suivants.

### 2.3.1 Diagramme de séquence système

La description écrite ne peut pas illustrer la séquence des événements de manière efficace. Il est donc nécessaire d'utiliser un diagramme de séquence pour compléter la description des cas d'utilisation précédents et montrer clairement les enchaînements des événements.

2.3.1.1 Diagramme de séquence système du cas d'utilisation "S'inscrire"

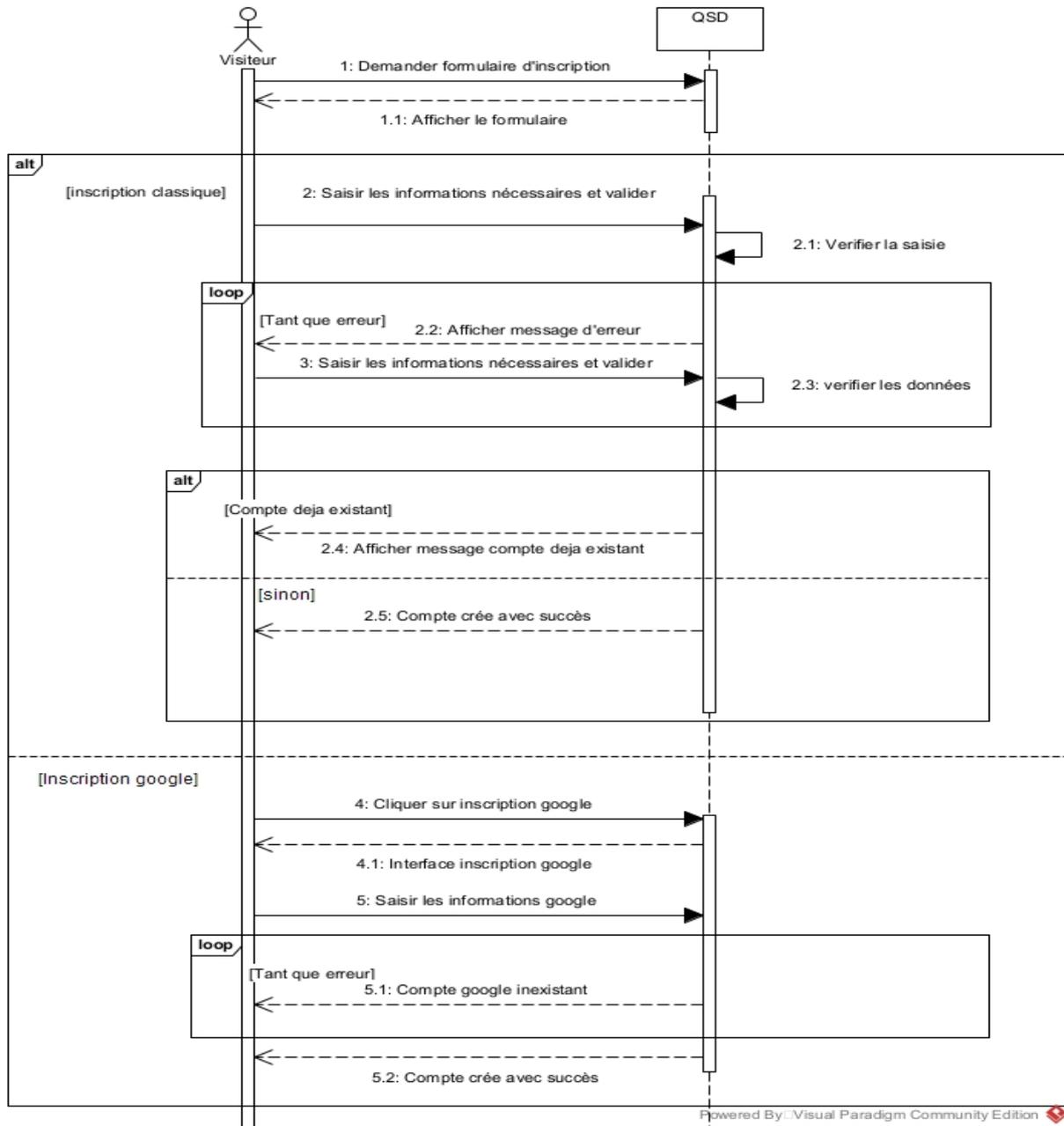


FIGURE 2.3 – Diagramme de séquence système du cas d'utilisation "S'inscrire"

2.3.1.2 Diagramme de séquence système du cas d'utilisation 'S'authentifier'

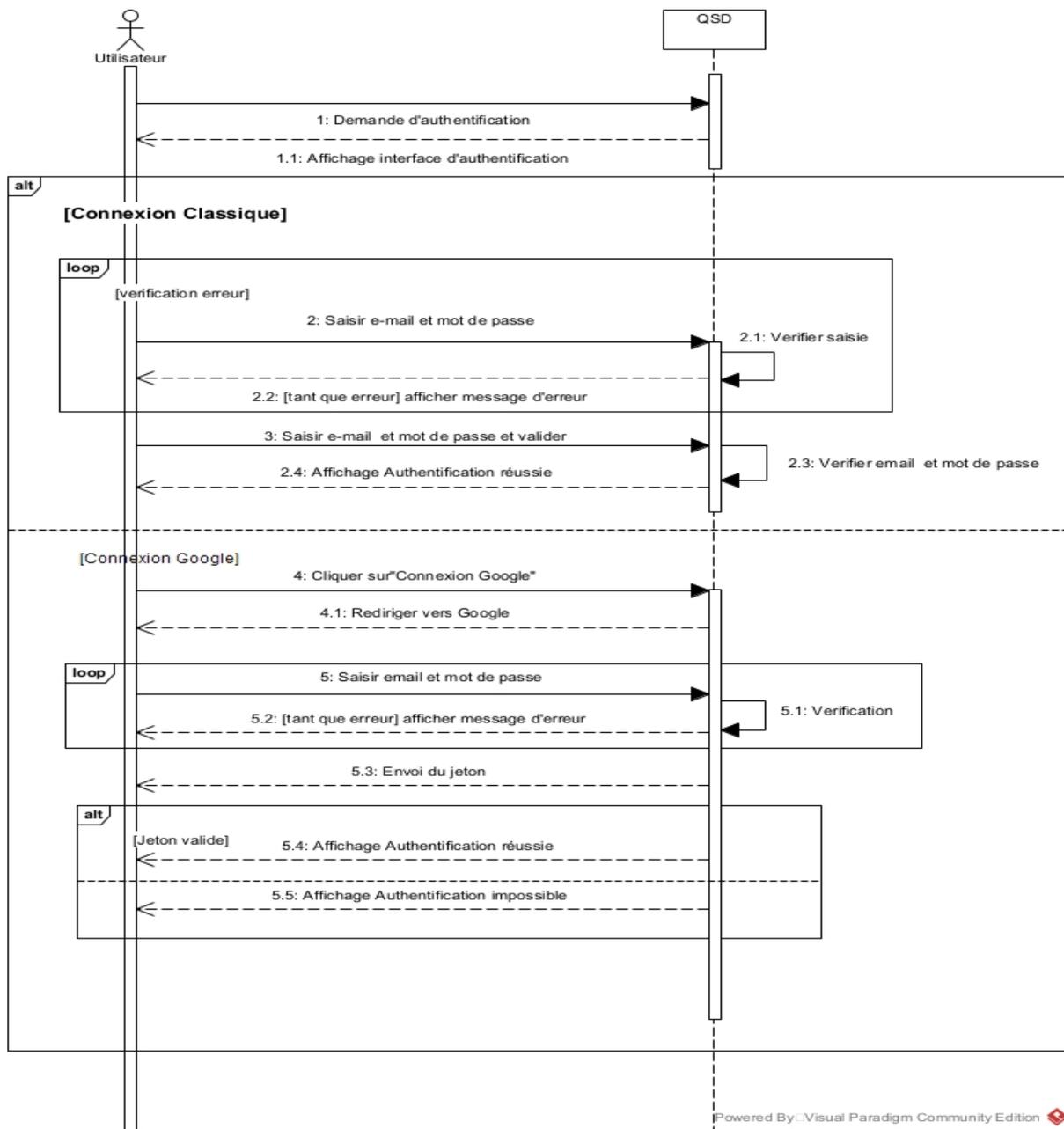


FIGURE 2.4 – Diagramme de séquence système du cas d'utilisation 'S'authentifier'

2.3.1.3 Diagramme de séquence système du cas d'utilisation 'Créer commande'

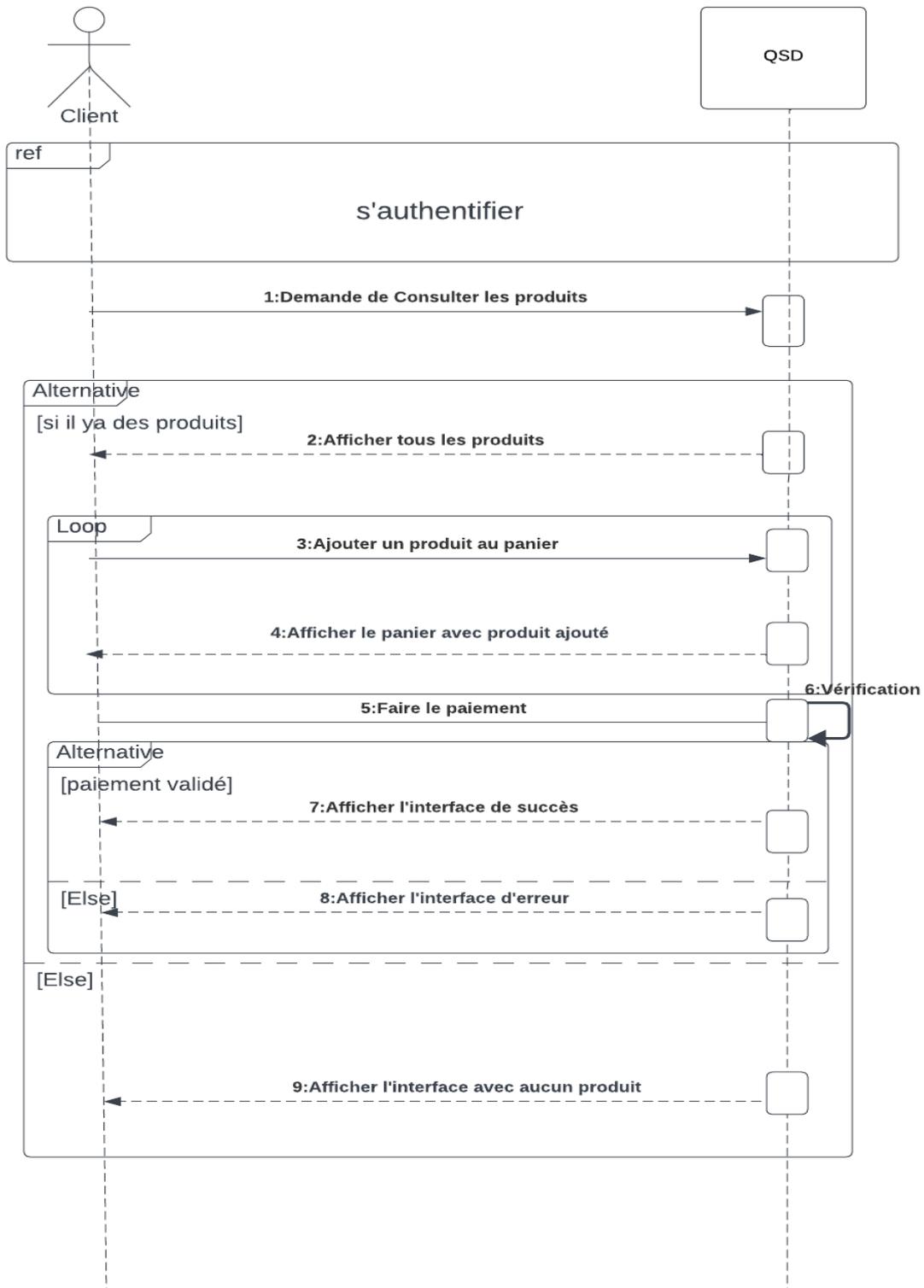


FIGURE 2.5 – Diagramme de séquence système du cas d'utilisation 'Créer commande'

2.3.1.4 Diagramme de séquence système du cas d'utilisation ' Consulter commande'

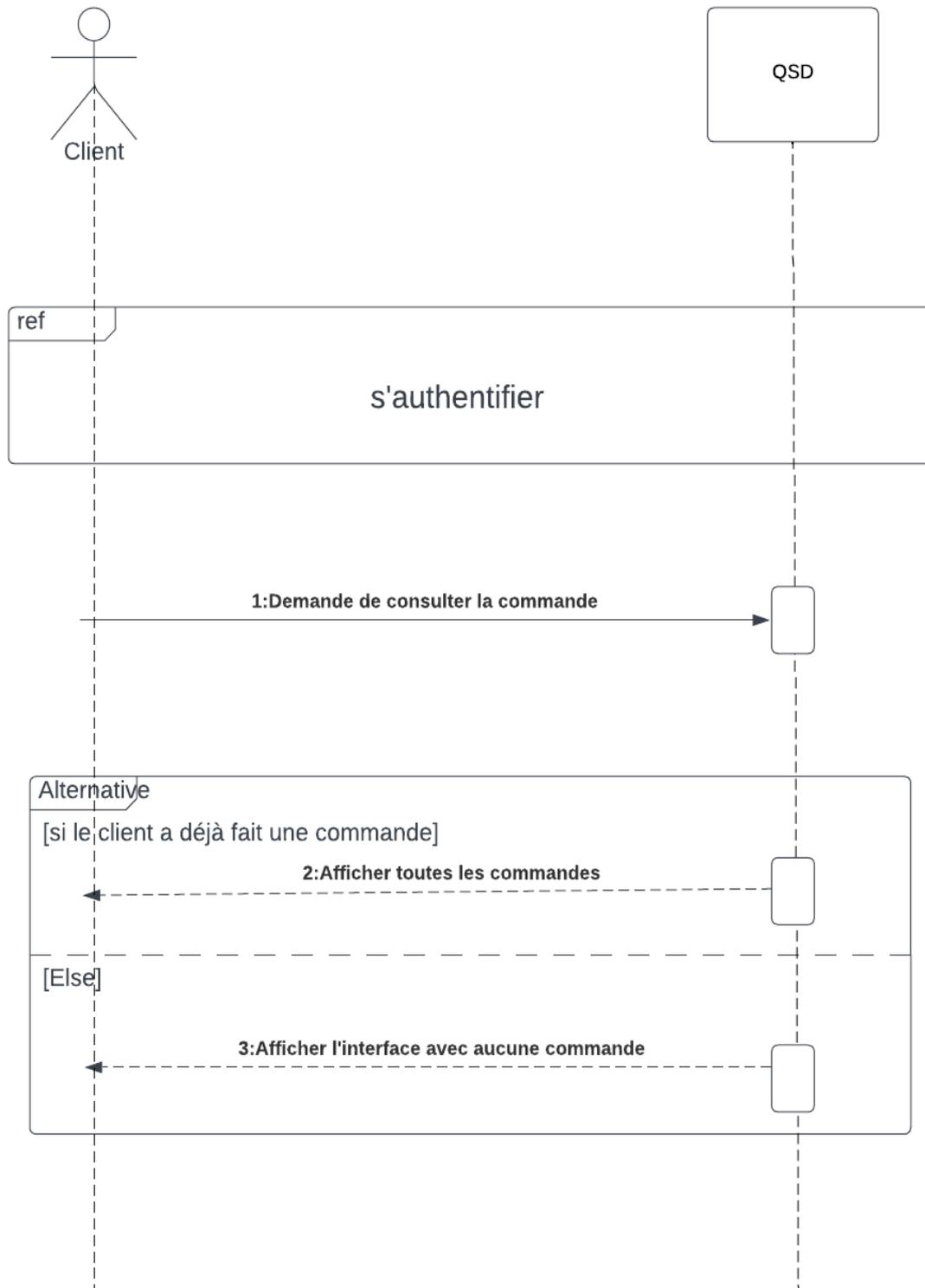


FIGURE 2.6 – Diagramme de séquence système du cas d'utilisation ' Consulter commande'

2.3.1.5 Diagramme de séquence système du cas d'utilisation 'Livrer commande'

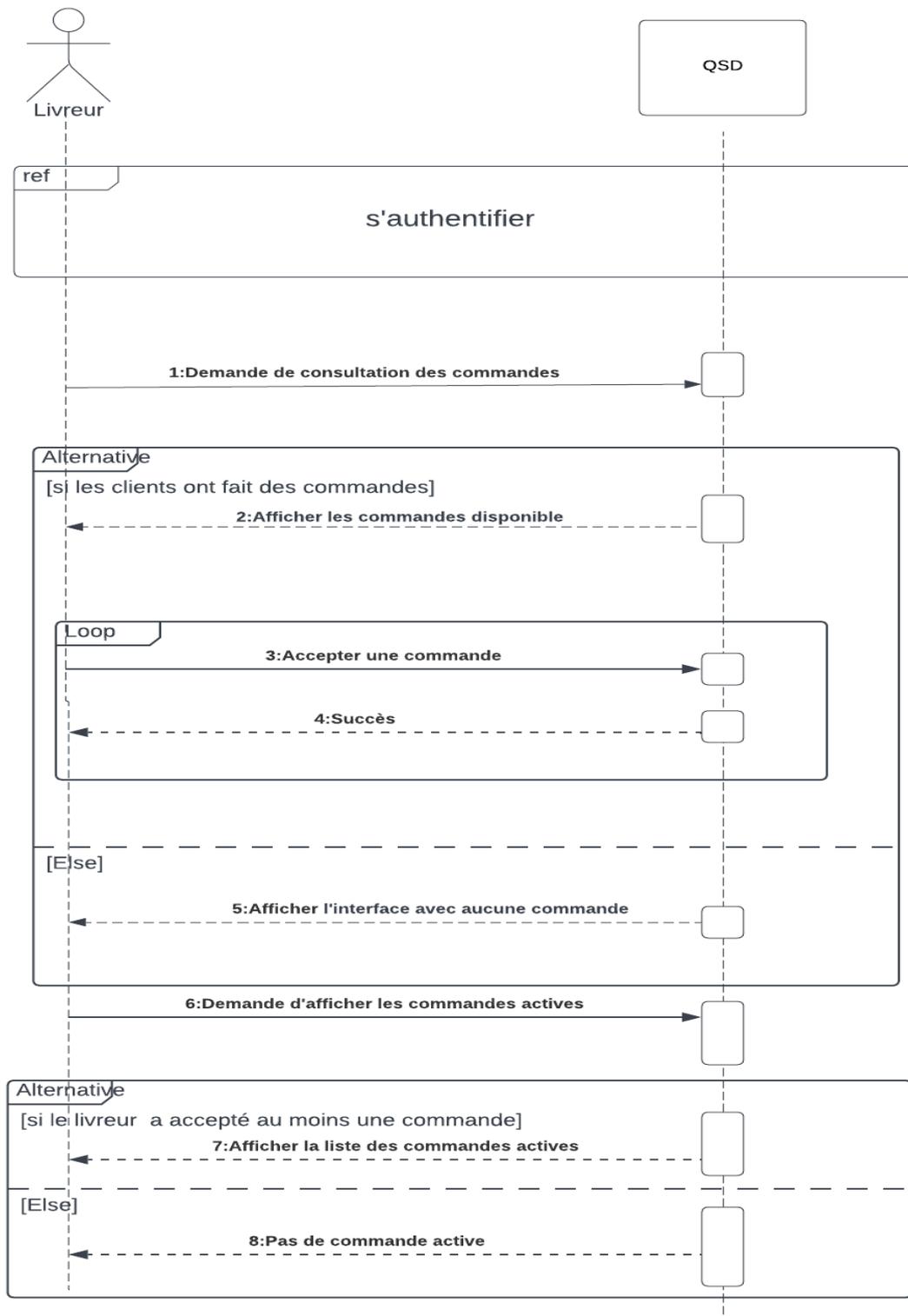


FIGURE 2.7 – Diagramme de séquence système du cas d'utilisation 'Livrer commande'

2.3.1.6 Diagramme de séquence système du cas d'utilisation 'Gérer les produits'

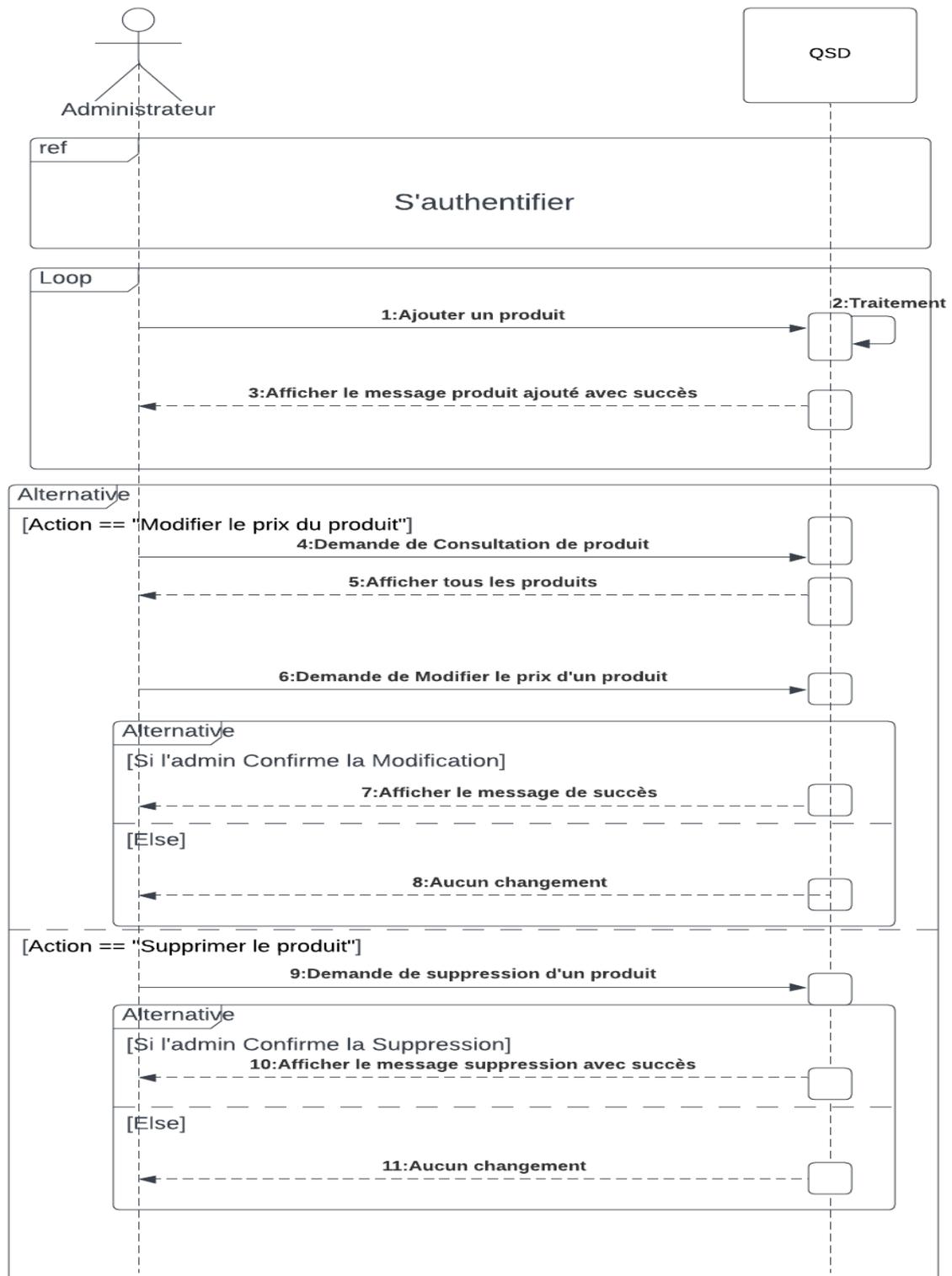


FIGURE 2.8 – Diagramme de séquence système du cas d'utilisation 'Gérer les produits'

### 2.3.1.7 Diagramme de séquence système du cas d'utilisation 'Gérer les utilisateurs'

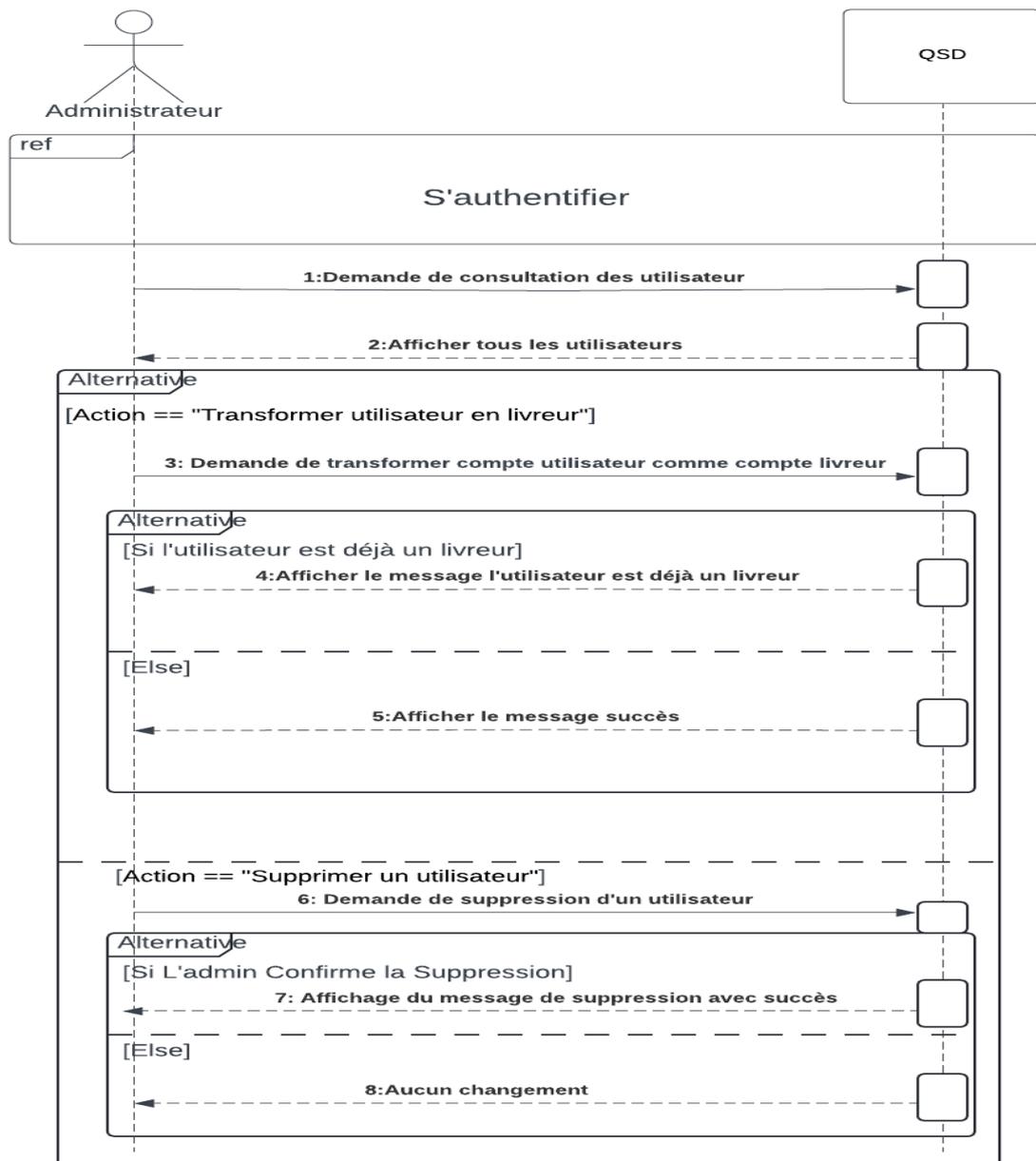


FIGURE 2.9 – Diagramme de séquence système du cas d'utilisation 'Gérer les utilisateurs'

## 2.4 Conclusion

Après cette étape, nous avons pu identifier et décrire les objectifs et les besoins de notre système, les représentant également à travers des diagrammes de cas d'utilisation et de séquence. Dans le chapitre suivant, nous aborderons la phase de conception où nous détaillerons ces besoins de manière approfondie.

# 3

## CONCEPTION

## 3.1 Introduction

Dans ce chapitre, nous allons créer les diagrammes de séquence d'interaction en utilisant les diagrammes de séquence système décrits précédemment. Ces nouveaux diagrammes mettront en scène les objets intervenant dans les différents cas d'utilisation. Ensuite, nous utiliserons ces objets pour concevoir le diagramme de classe. Enfin, nous expliquerons le choix des bases de données non relationnelles et les raisons qui nous ont poussés à opter pour cette solution dans notre projet.

## 3.2 Diagrammes de séquence détaillé

Pour chaque diagramme de séquence système, nous établirons le diagramme de séquence détaillé correspondant.

Du fait de notre choix pour une architecture MVC les objets composant le système sont de 3 types :

- La Vue : représente l'interface entre l'utilisateur et le système.
- Le Contrôleur : représente la logique de traitement du système déclenchée par une action de l'utilisateur.
- Le Modèle : représente les données et la logique métier du système, telles que décrites dans les cas d'utilisation.

### 3.2.1 Diagramme de séquence détaillé du cas d'utilisation "S'inscrire"

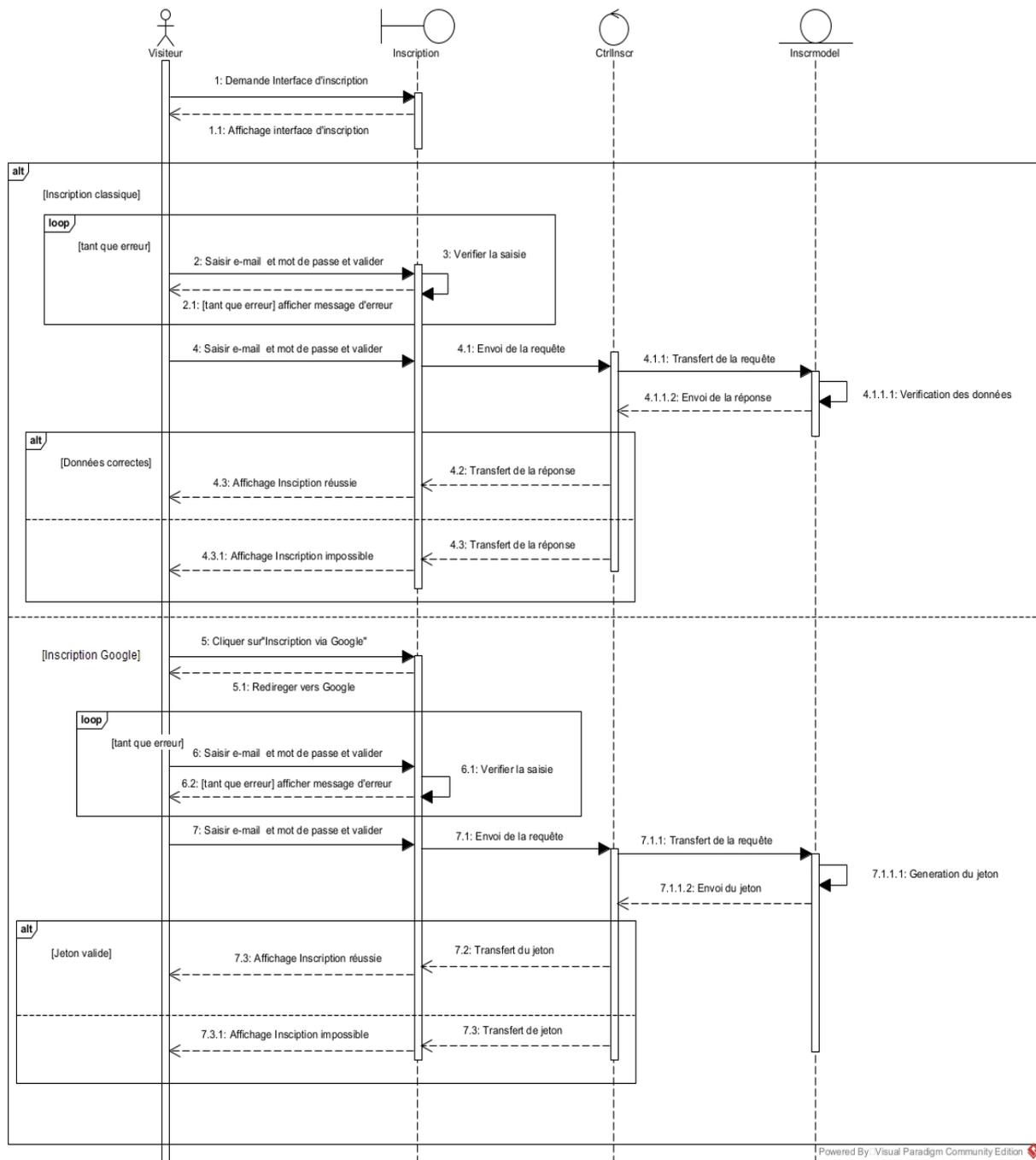


FIGURE 3.1 – Diagramme de séquence détaillé "S'inscrire"

### 3.2.2 Diagramme de séquence détaillé du cas d'utilisation "S'authentifier"

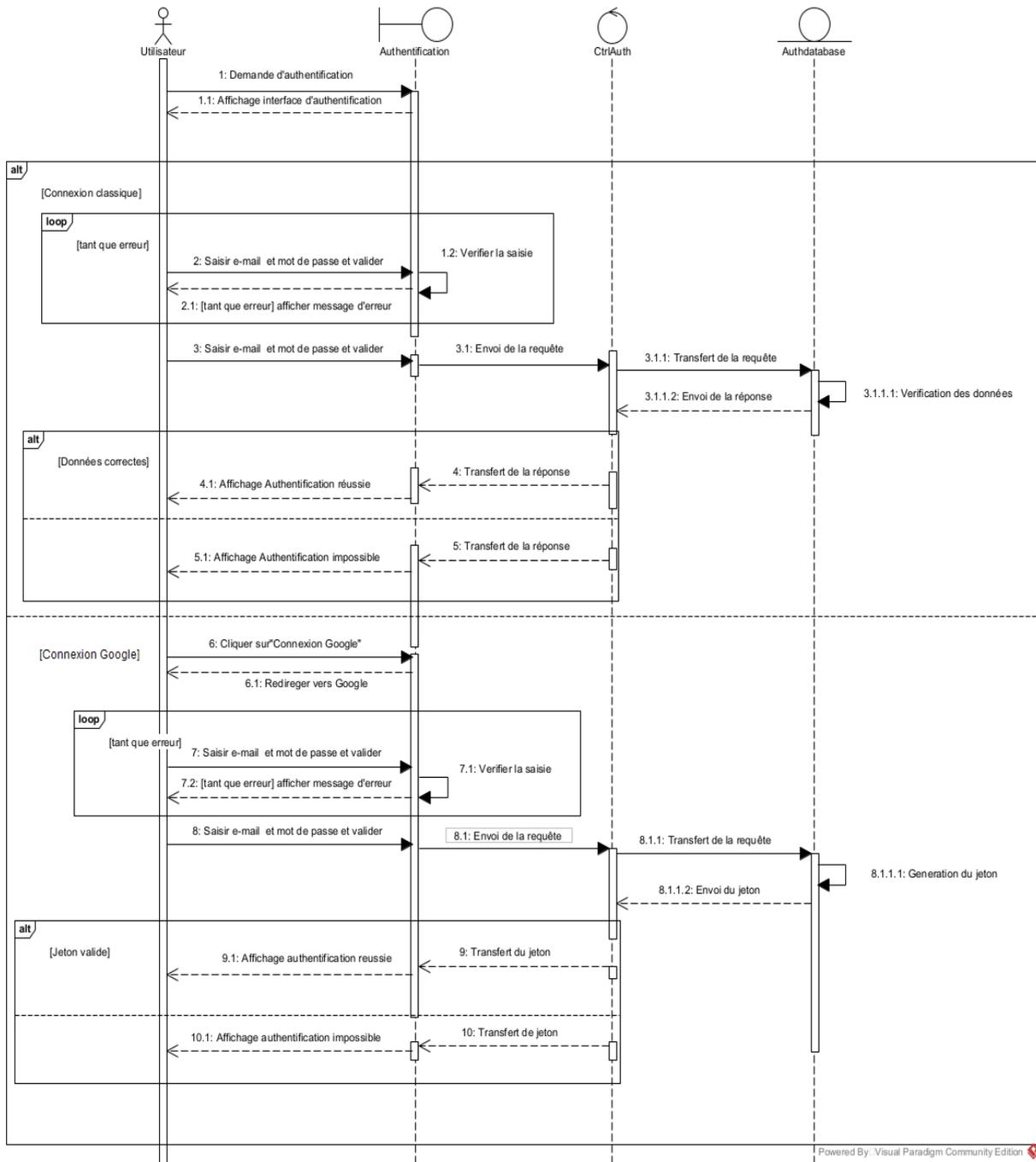


FIGURE 3.2 – Diagramme de séquence détaillé "S'authentifier"

### 3.2.3 Diagramme de séquence détaillé du cas d'utilisation “Créer commande”

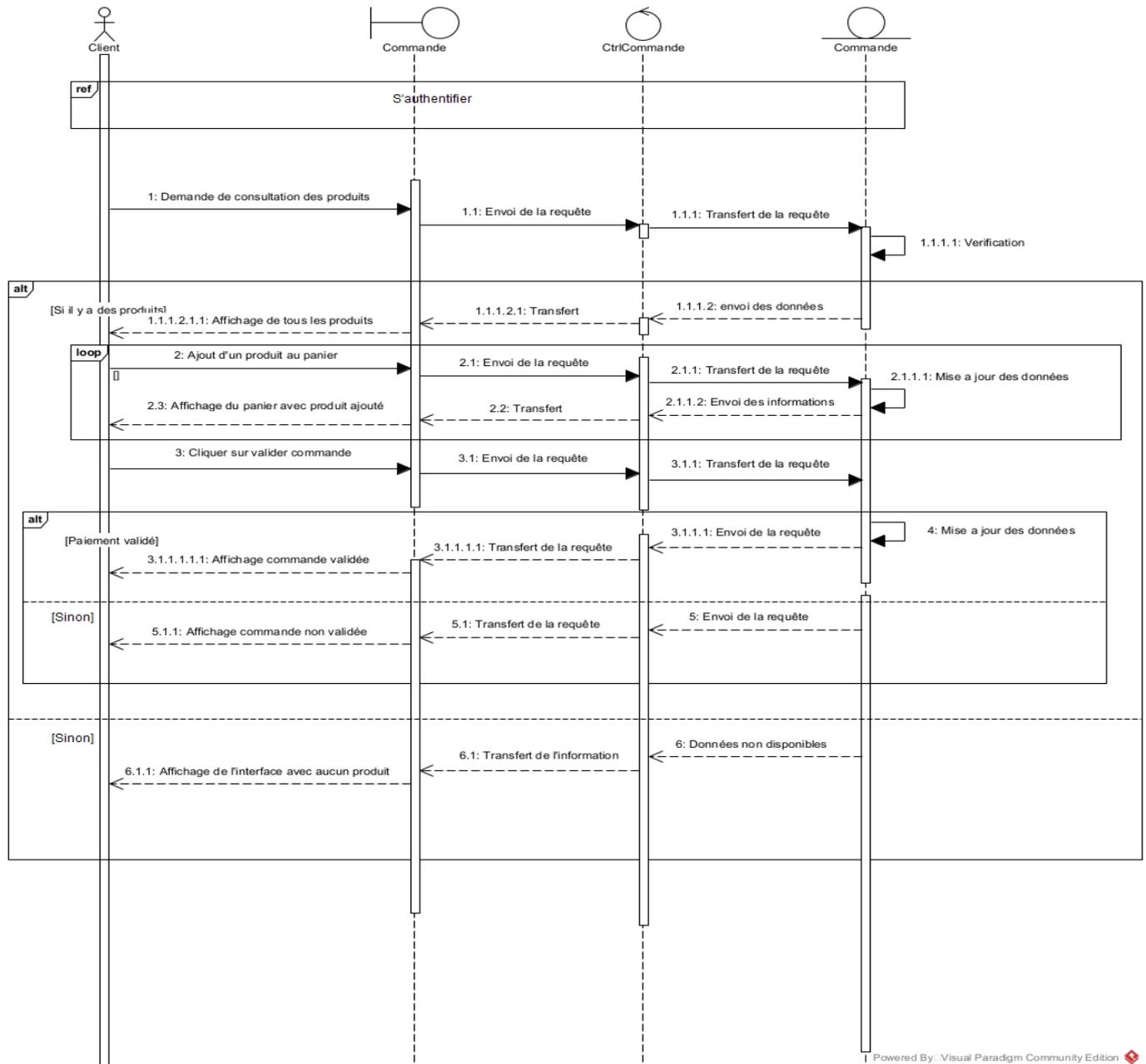


FIGURE 3.3 – Diagramme de séquence détaillé du cas d'utilisation “Créer commande”

### 3.2.4 Diagramme de séquence détaillé du cas d'utilisation “Consulter commande”

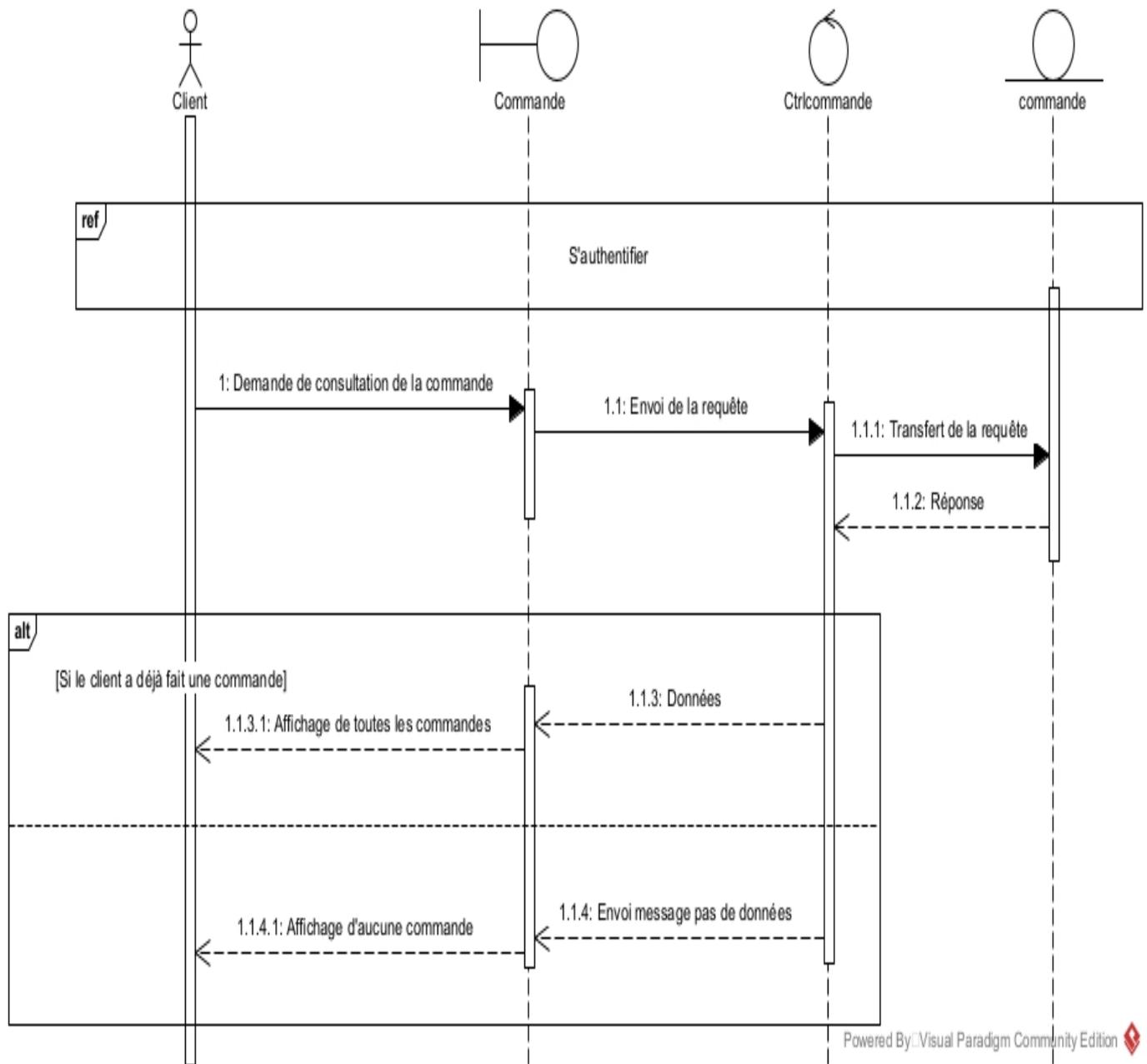
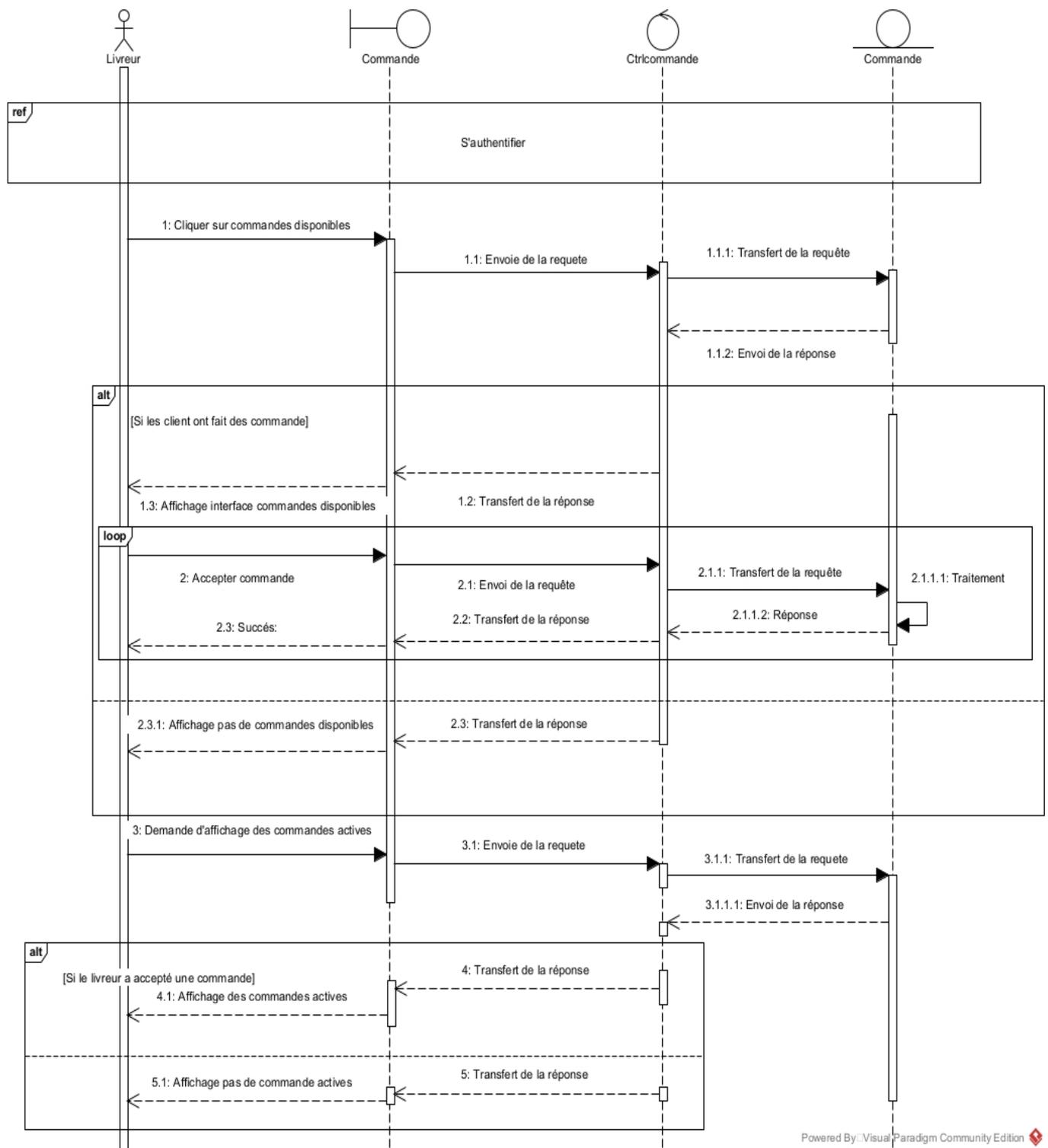


FIGURE 3.4 – Diagramme de séquence détaillé du cas d’utilisation “Consulter commande”

### 3.2.5 Diagramme de séquence détaillé du cas d'utilisation “Livrer commande”



### 3.2.6 Diagramme de séquence détaillé du cas d'utilisation “Gérer les produits”

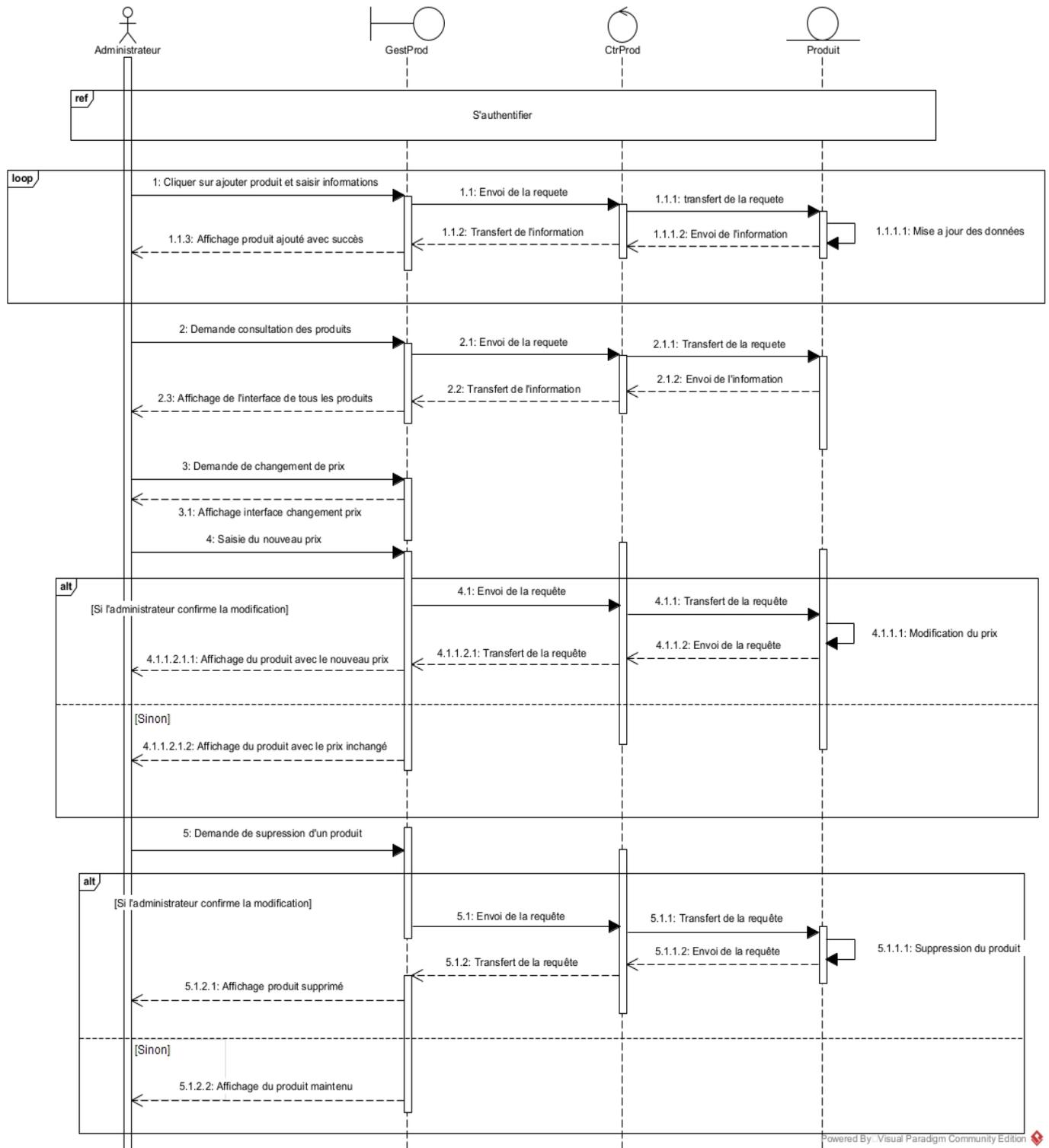


FIGURE 3.6 – Diagramme de séquence détaillé “Gérer les produits”

### 3.2.7 Diagramme de séquence détaillé du cas d'utilisation “Gérer les utilisateurs”

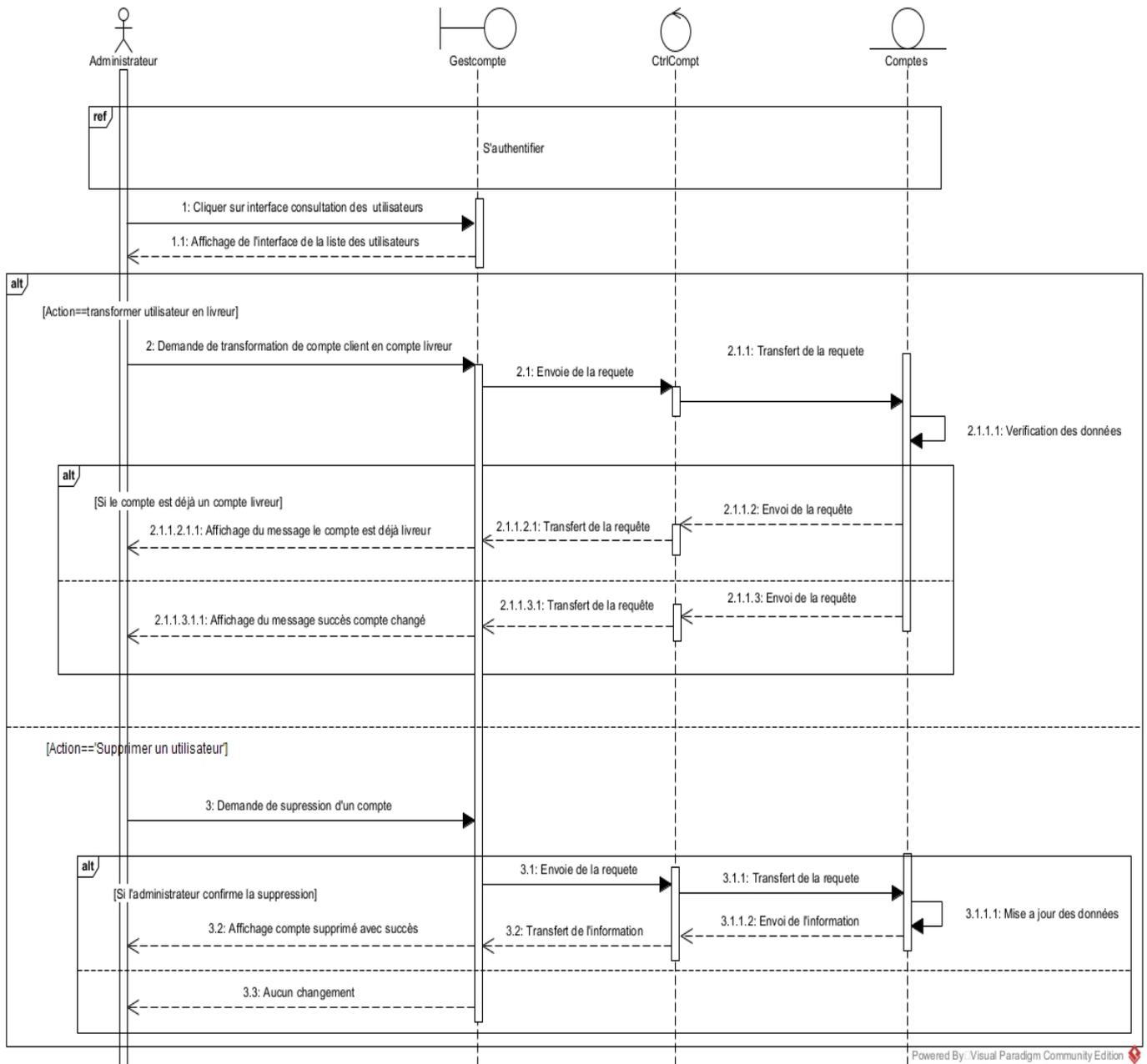


FIGURE 3.7 – Diagramme de séquence détaillé “Gérer les utilisateurs”

### 3.3 Diagramme de classes de conception

Le diagramme de classes offre une vision statique et abstraite de la structure interne d'un système. Il représente les différentes classes d'objets, leurs attributs, méthodes et relations sans considérer le déroulement temporel des interactions entre ces objets. C'est un outil essentiel pour comprendre la composition et les liens entre les entités du système, mais il ne montre pas comment ces entités interagissent dynamiquement dans des scénarios spécifiques.

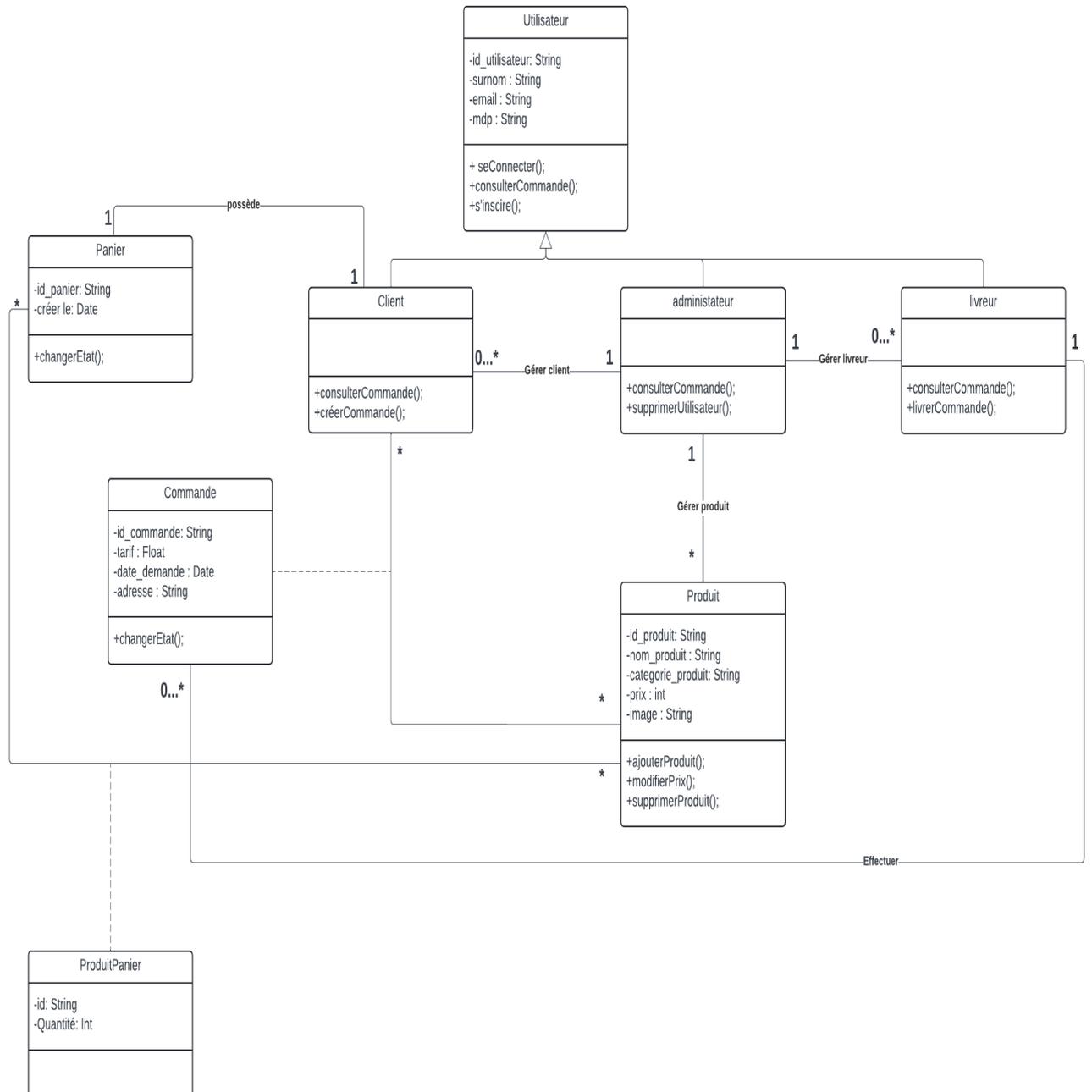


FIGURE 3.8 – "Diagramme de classes de conception"

### 3.4 Dictionnaire de données

Le tableau suivant décrit les types de données des attributs de chaque classe collectée dans le diagramme de classe.

Classe	Attribut	Désignation	Type
Utilisateur	id-utilisateur	identifiant	String
	surnom	combinaison du nom et prénom de l'utilisateur	String
	email	email de l'utilisateur	String
	mdp	mot de passe de l'utilisateur	String
Commande	id-commande	identifiant de la commande	String
	tarif	tarif de livraison	int
	date-demande	date à laquelle est effectuée la commande	Date
	adresse	adresse de livraison	String
Produit	id-produit	identifiant du produit	String
	nom-produit	nom du produit	String
	categorie-produit	catégorie du produit	String
	prix	prix du produit	int
	image	aspect visuel du produit	String
Panier	id-panier	identifiant du panier	String
	créer-le	date de création	Date
ProduitPanier	id	identifiant	String
	Quantité	quantité du produit dans le panier	Date

TABLE 3.1 – Dictionnaire des données du diagramme de classes

### 3.5 Base de données NoSql

Dans le contexte actuel où émergent différents types de systèmes de bases de données (relationnelles et non-relationnelles), le choix du type et du système de base de données pour stocker de grandes quantités de données dans les applications de big data est devenu un défi important. Les bases de données NoSQL ont émergé pour répondre à ces limitations. Elles sont conçues pour offrir des performances élevées et une grande évolutivité en relâchant certaines des contraintes des bases de données relationnelles traditionnelles, telles que les propriétés ACID (Atomicité, Cohérence, Isolation, Durabilité) [13].

Fonctionnalités	BDD relationnelle	BDD NoSQL
<b>Complexité de requêtage</b>	Dans le schéma	Au niveau de la couche applicative (Contraintes, join)
<b>Temps de réponse sur de gros volume de données</b>	Performance limitée pour le croisement de données sur de gros volumes	Performant
<b>Haute disponibilité</b>	Clustering : on a une machine maitre, on copie la base sur des serveurs esclaves (avec SQL server 2012 et Always On il y aura une base de données copiée dédiée à la lecture)	base de données distribuée : stockage des données sur différents serveurs (partitionnement automatique)
<b>Montée en charge</b>	Scalabilité verticale : on augmente les performances hardware du serveur utilisé	Scalabilité horizontale : ajout d'autant de serveurs dont on a besoin pour augmenter les performances
<b>Schéma d'interrogation</b>	SELECT GROUP BY	MAP/REDUCE

TABLE 3.2 – Comparaison entre BDD relationnelle et BDD NoSQL

### 3.5.1 Types de base de données NoSql

Les bases de données NoSQL, conçues pour gérer des volumes massifs de données non structurées ou semi-structurées, se distinguent des bases de données relationnelles traditionnelles par leur flexibilité et leur capacité à évoluer horizontalement. Elles se divisent principalement en quatre types [14] :

1. **Bases de données clé-valeur** : Ces bases de données stockent des paires clé-valeur, où chaque clé est unique et permet d'accéder à une valeur associée. Elles sont particulièrement efficaces pour les opérations de recherche rapide et les applications nécessitant un accès rapide aux données, comme Redis et Amazon DynamoDB.
2. **Bases de données orientées colonnes** : Ces bases de données stockent les données dans des colonnes plutôt que dans des lignes. Cela permet de lire et écrire efficacement de grandes quantités de données. Les bases de données orientées colonnes sont idéales pour les applications analytiques et les systèmes de gestion de grandes quantités de données, tels que Apache Cassandra et HBase.
3. **Bases de données orientées documents** : Ces bases de données stockent les données sous forme de documents, généralement en JSON, BSON, ou XML. Chaque document peut avoir une structure différente, ce qui permet une grande flexibilité dans la modélisation des données. Les bases de données orientées documents sont bien adaptées aux applications nécessitant une gestion flexible des données, comme MongoDB et CouchDB.
4. **Bases de données orientées graphes** : Ces bases de données sont conçues pour stocker et gérer des relations complexes entre les données. Elles représentent les données sous forme de graphes, avec des nœuds, des arêtes, et des propriétés. Les bases de données orientées graphes sont parfaites pour les applications nécessitant une analyse approfondie des relations entre les données, telles que les réseaux sociaux, les recommandations, et la détection de fraudes. Des exemples incluent Neo4j et Amazon Neptune.

### 3.5.2 Structure de la base de données

Nous avons décidé de travailler avec une base de données NoSQL orientée document au format JSON, car elles sont optimisées pour des requêtes rapides et des écritures fréquentes, assurant ainsi une haute performance. La représentation du schéma de la base de données en arborescence json est comme suit :

```
{
  "User": {
    "userId": "",
    "email": "",
    "displayName": "",
    "password": "",
    "createdAt": ""
  },
  "Product": {
    "productId": "",
    "product_name": "",
    "product_category": "",
    "product_price": 0.0,
    "imageURL": ""
  },
  "CartItem": {
    "cartItemId": "",
    "userId": "",
    "productId": "",
    "quantity": 0,
    "createdAt": ""
  },
  "Order": {
    "orderId": "",
    "userId": "",
    "amount": 0.0,
    "created": "",
    "status": "",
    "items": [],
    "sts": "",
    "total": 0.0
  }
}
```

FIGURE 3.9 – Structure de la base de données

## 3.6 Conclusion

Dans ce chapitre, nous avons présenté les diagrammes de séquence d'interaction correspondant à certains cas d'utilisation de notre système. Ensuite, nous avons élaboré le diagramme de classes de notre système, illustrant de manière globale la structure des éléments constituant la base de données associée à notre application. Dans le chapitre suivant nous allons énoncer les différents outils qui ont permis de développer notre application ainsi que quelques interfaces illustrant les différentes fonctionnalités de notre application.

# 4

## RÉALISATION

## 4.1 Introduction

Dans ce chapitre, la réalisation du projet sera d'abord présentée par les outils de développement ainsi que les choix d'implémentation de la base de données et des services invoqués. Ensuite, l'architecture logicielle, qui sert à organiser le code source du projet, sera décrite.

Par la suite, une présentation des scénarios d'exécution des cas d'utilisation sera illustrée par l'enchaînement des interfaces graphiques de l'application.

## 4.2 Langages et Framework de développement

La mise en œuvre du projet a nécessité l'utilisation de plusieurs outils, langages et framework de développement dont les principaux sont les suivants :

### 4.2.1 Visual studio code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré [15].



FIGURE 4.1 – Logo de Vs code

### 4.2.2 Javascript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web<sup>4</sup>. Une grande majorité des sites web l'utilisent<sup>5</sup>, et la majorité des navigateurs web disposent d'un moteur JavaScript<sup>6</sup> pour l'interpréter [16].



FIGURE 4.2 – Logo de javascript

### 4.2.3 NodeJs

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

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

Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Ce qui autorise, lors du déploiement de sites internet et d'applications web développés avec Node.js, de ne pas installer et utiliser des serveurs webs tels que Nginx ou Apache [17].



FIGURE 4.3 – Logo de NodeJs

#### 4.2.4 React

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre. Elle est maintenue par Meta (anciennement Facebook) ainsi que par une communauté de développeurs individuels et d'entreprises depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état [18].

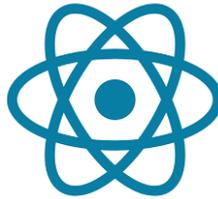


FIGURE 4.4 – Logo de ReactJS

#### 4.2.5 Tailwind CSS

Tailwind CSS est un framework CSS open source. La fonctionnalité principale de cette bibliothèque est, contrairement à d'autres frameworks CSS comme Bootstrap, qu'elle ne procure pas une série de classes prédéfinies pour des éléments tels que des boutons ou des tables. À la place, Tailwind crée une liste de classes CSS « utilitaires » pouvant être utilisés pour ajouter un style à chaque élément en les mélangeant et en les agençant [19].



FIGURE 4.5 – Logo de Tailwind CSS

#### 4.2.6 ExpressJs

Express est un framework web minimaliste et flexible pour Node.js, qui offre un ensemble robuste de fonctionnalités pour les applications web et mobiles. Grâce à

une multitude de méthodes utilitaires HTTP et de middleware à votre disposition, permettant la création d'une API robuste est rapide et facile [20] .



FIGURE 4.6 – Logo de Express JS

### 4.2.7 Capacitor

Capacitor est une plateforme open-source conçue pour le développement d'applications hybrides, permettant aux applications web d'accéder aux fonctionnalités natives des appareils mobiles. Elle nous permet de créer des applications multi-plateformes pour iOS, Android et des applications Web progressives (PWAs) en utilisant JavaScript, HTML et CSS. Avec Capacitor, nous pouvons créer des applications performantes qui exploitent les fonctionnalités natives des appareils tout en maintenant un seul code source pour plusieurs plateformes [21] .



FIGURE 4.7 – Logo de Capacitor

## 4.3 Plateformes de services web

Afin de réaliser au mieux notre application, nous avons utilisé les services web suivants :

### 4.3.1 Services de déploiement et hébergement

Les services de déploiement et d'hébergement sont des solutions fournies par des plateformes spécialisées pour permettre aux développeurs de rendre leurs applications et sites web accessibles en ligne

#### 4.3.1.1 Firebase

Firebase est un outil d'aide au développement d'applications web, Android, iOS, et Unity. Il propose un hébergement en nuage et utilise NoSQL pour héberger des bases de données. Il propose des logiciels utilisés pour le développement d'applications mobiles pour enregistrer des données, envoyer des notifications et des publicités, remonter les erreurs et les clics effectués sur l'application. Il fonctionne dans une application et est invisible de l'utilisateur de l'application [22].



FIGURE 4.8 – Logo de Firebase

#### 4.3.1.2 Vercel

Vercel est une plateforme de déploiement qui permet aux développeurs de publier rapidement des sites web et des applications frontales. Elle offre une infrastructure serverless optimisée pour le déploiement automatique, la mise à l'échelle transparente et des performances élevées [23].



FIGURE 4.9 – Logo de Vercel

### 4.3.2 Services de paiement

Un service de paiement est une plateforme ou un système qui facilite le transfert d'argent entre différentes entités.

#### 4.3.2.1 Stripe

Stripe est une plateforme de services financiers et de logiciels pour les entreprises qui leur permet d'accepter des paiements en ligne et de gérer leurs activités financières. Fondée en 2010, Stripe est largement utilisée par des entreprises de toutes tailles pour faciliter les transactions en ligne de manière sécurisée et efficace [24].



FIGURE 4.10 – Logo de Stripe

#### 4.3.2.2 Chargily pay

Chargily Pay est une passerelle de paiement en ligne algérienne qui facilite l'acceptation des paiements via plusieurs méthodes de paiement locales, telles que les cartes EDAHABIA et CIB. Elle utilise des protocoles de sécurité avancés pour garantir la protection des informations sensibles et des transactions financières de vos clients [25].



FIGURE 4.11 – Logo de chargily pay

### 4.3.3 Outil de performance et optimisation

Afin de tester notre application nous avons utiliser le service suivant :

#### 4.3.3.1 Lighthouse

Lighthouse est un outil automatisé de qualité web et de performance fourni par Google. Il évalue les performances d'une page web en matière d'accessibilité, de bonnes pratiques, de SEO (optimisation pour les moteurs de recherche) et de performances. C'est un outil précieux pour l'analyse et l'amélioration des performances des sites web [26].



FIGURE 4.12 – Logo de Lighthouse

## 4.4 Implémentation de la base de données

Dans cette partie nous allons décrire comment configurer l'environnement Firebase afin de gérer notre base de données NoSQL et les services complémentaires fournis :

### 4.4.1 Intégration et services utilisés

Firestore regroupe un ensemble de services que nous exploitons en suivant les étapes suivantes :

- Créer un projet dans la plateforme Firebase : <https://console.firebase.google.com/u/0/>
- Ajouter une application au projet créer précédemment

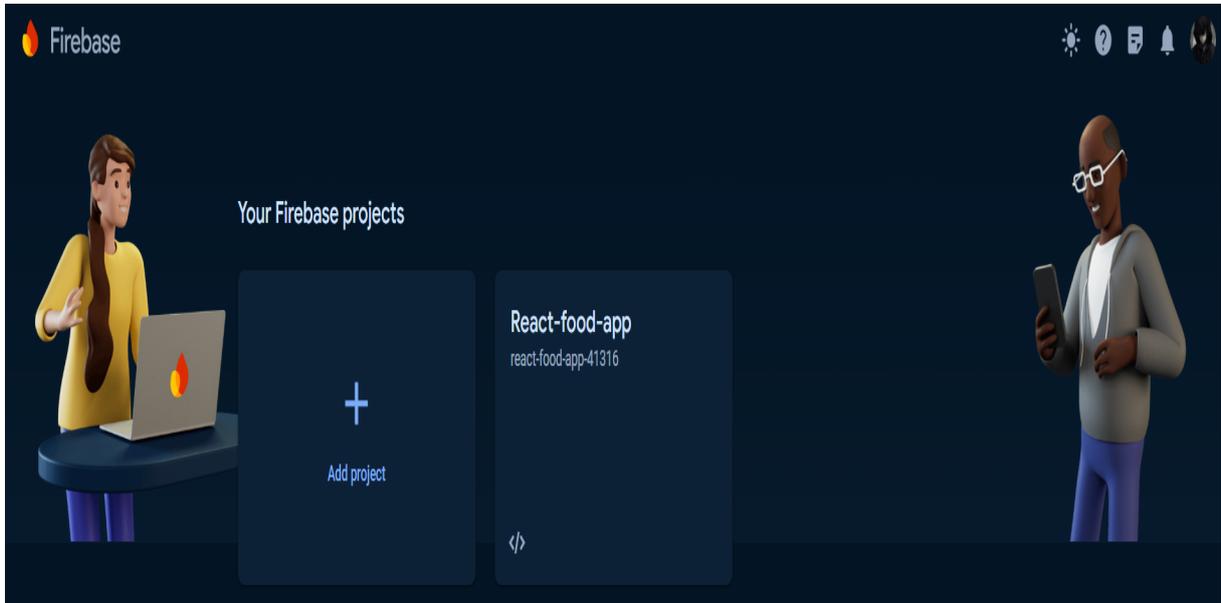


FIGURE 4.13 – Vue d'ensemble du projet Firebase

Nous pouvons à présent utiliser les services d'authentification, base de données, notification, etc.

On a utilisé les services Firebase suivants :

- a) **Authentification** : Grâce à Firebase Authentication, nous avons pu mettre en place des systèmes d'authentification robustes, incluant des méthodes de connexion via email/mot de passe, Google, et bien d'autres méthodes son réalisables.

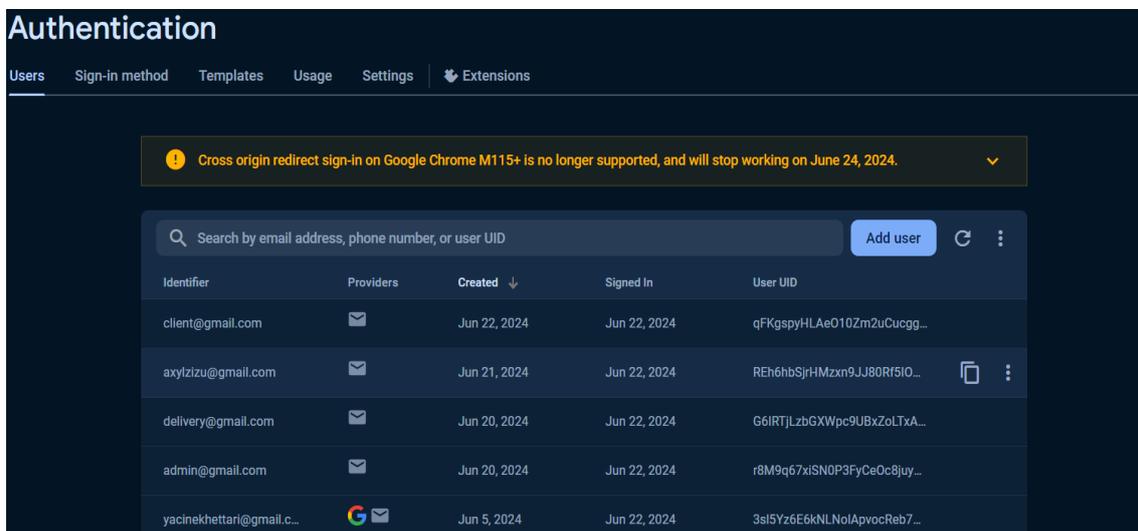


FIGURE 4.14 – Service Authentification de Firebase

- b) **Cloud Storage** : C'est une fonctionnalité puissante qui permet de stocker et de gérer des fichiers dans le cloud de manière sécurisée et évolutive. Avec Firebase Storage, nous pouvons stocker des images, des vidéos, des documents et d'autres types de fichiers directement depuis notre application web et mobile ,de même pour les fichiers des utilisateurs.

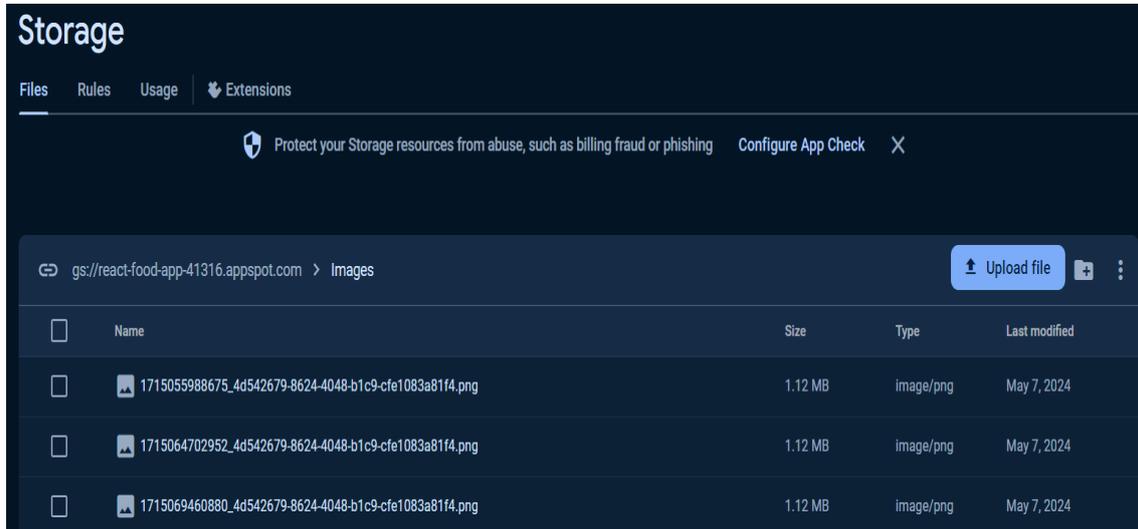


FIGURE 4.15 – Service storage de firebase

- c) **Firestore database :** Firestore est une base de données flexible et évolutive pour le développement d'applications mobiles, web et serveur, proposée par Firebase et Google Cloud Platform. C'est une base de données NoSQL orientée documents qui permet de stocker, synchroniser et interroger des données pour vos applications

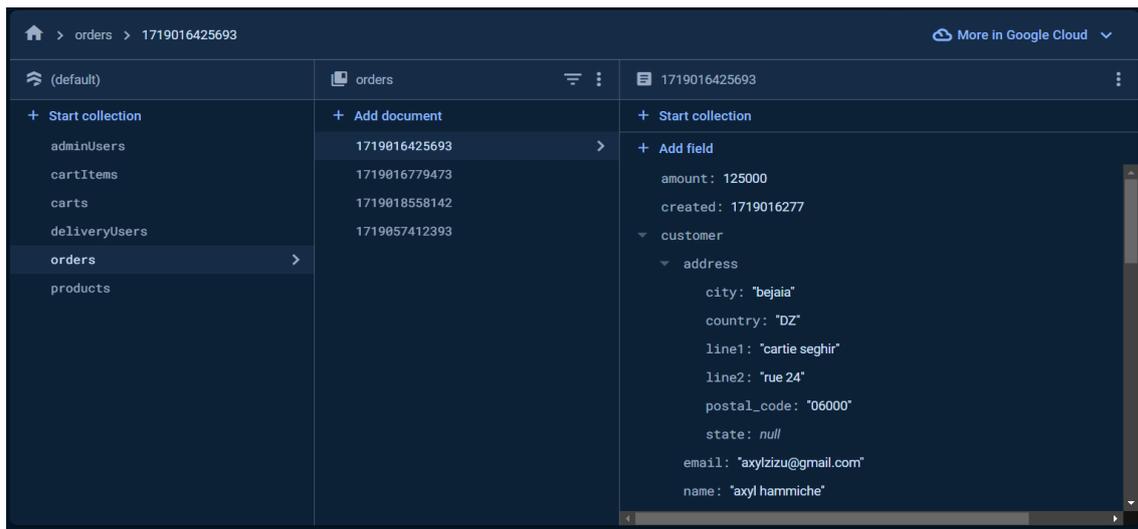


FIGURE 4.16 – Service database de firebase

## 4.5 Présentation des interfaces de l'application

Cette section représente les interfaces homme machine des différents utilisateurs, elle vise à donner une représentation visuelle des différents aspects de l'application aussi bien mobile que web.

### 4.5.1 Interfaces d'inscription et de connexions pour tous les utilisateurs.

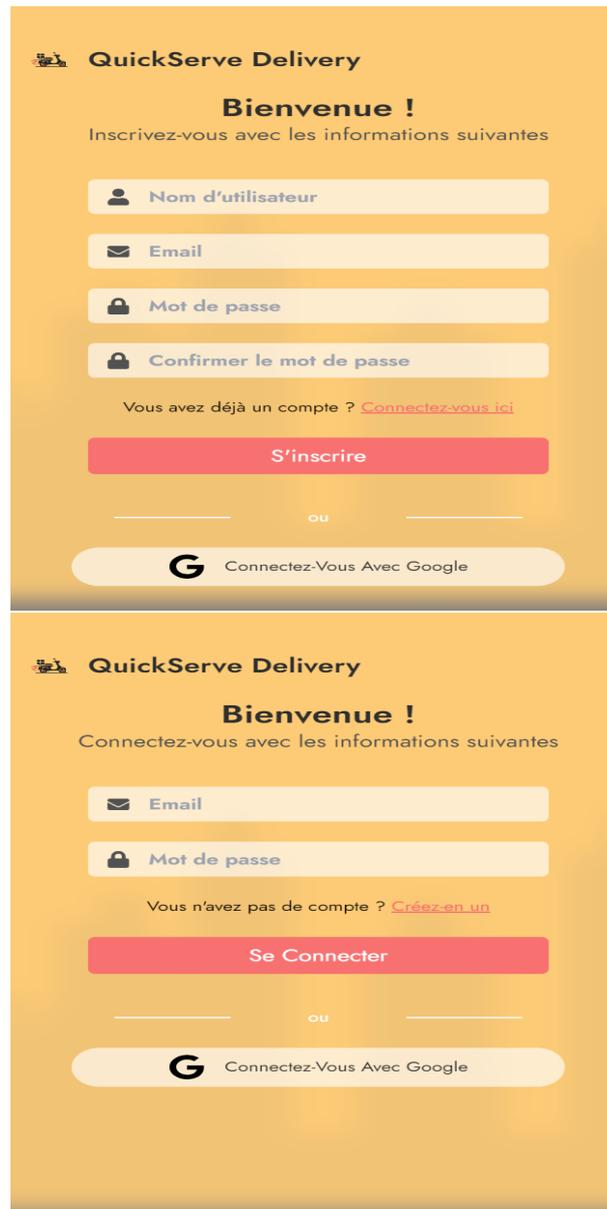


FIGURE 4.17 – Interfaces d'inscription et de connexion

### 4.5.2 Interfaces principales Client

Dans ce qui suit nous allons monter les interfaces qui apparaîtront pour le visiteur et le client.

#### 4.5.2.1 Interfaces d'accueil

Lorsque l'application est lancée, elle affiche une interface d'accueil qui présente les diverses actions disponibles pour les visiteurs.

a) première partie de la page d'accueil ou on peut le bouton d'inscription, connexion, notre logo ainsi que le slogan.



FIGURE 4.18 – Interface d'accueil

b) Suite de la page d'accueil ou on peut voir le menu avec possibilité de choisir la catégorie.

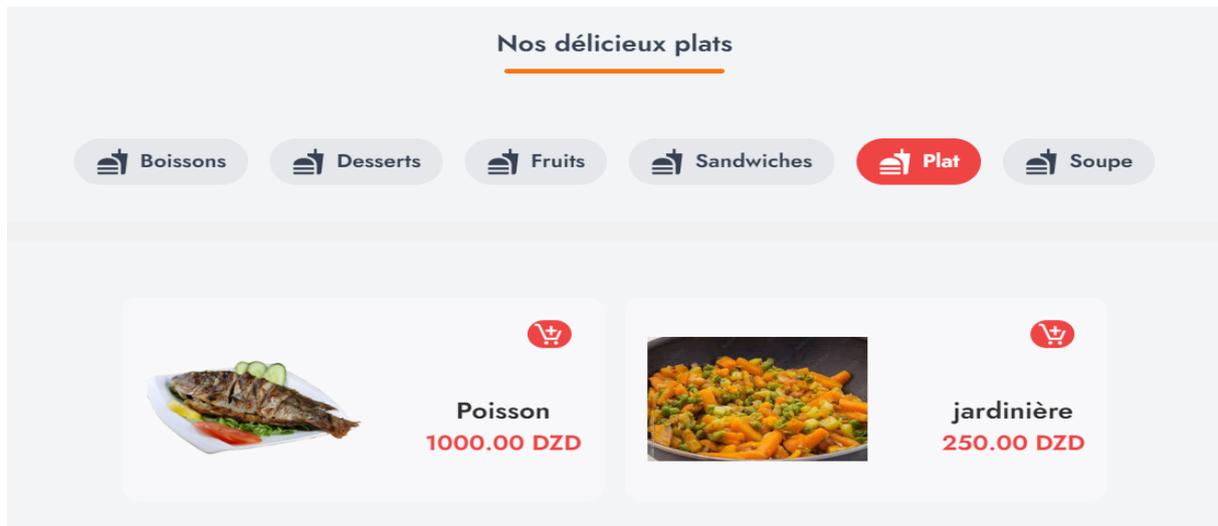


FIGURE 4.19 – Produits visibles accueil

### 4.5.2.2 Interfaces des commandes

-Dans ce qui suit nous allons montrer dans les figures 4.18, 4.19, 4.20, 4.21 qui représentent les interfaces des commandes ainsi que les différents moyens de paiement possibles :

a)Interface de la création de la commande et du choix du type de paiement.

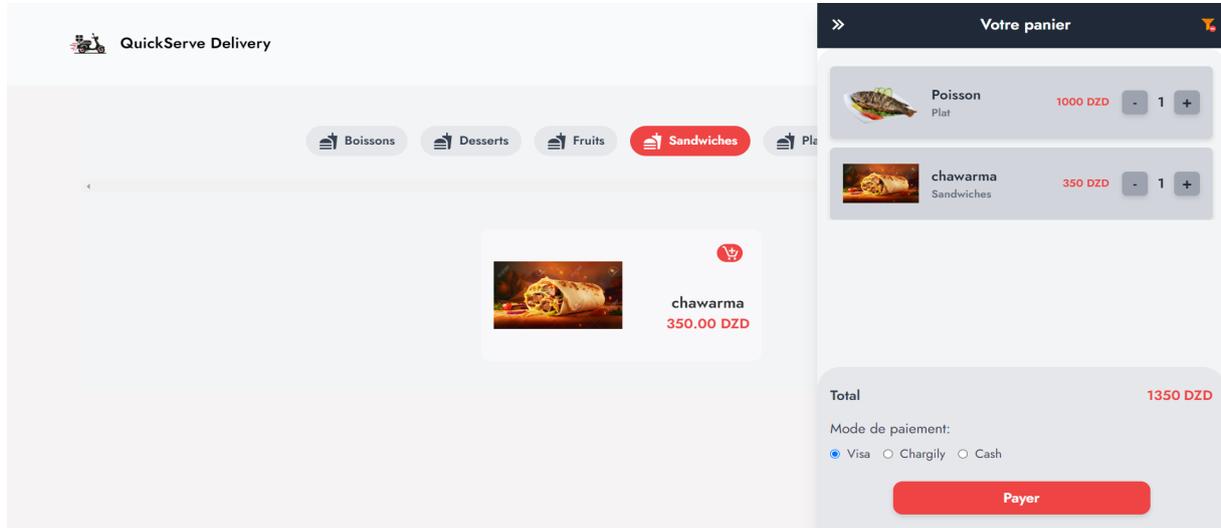


FIGURE 4.20 – Interface Commande

b)Interface montrant le paiement d'une commande avec Stripe

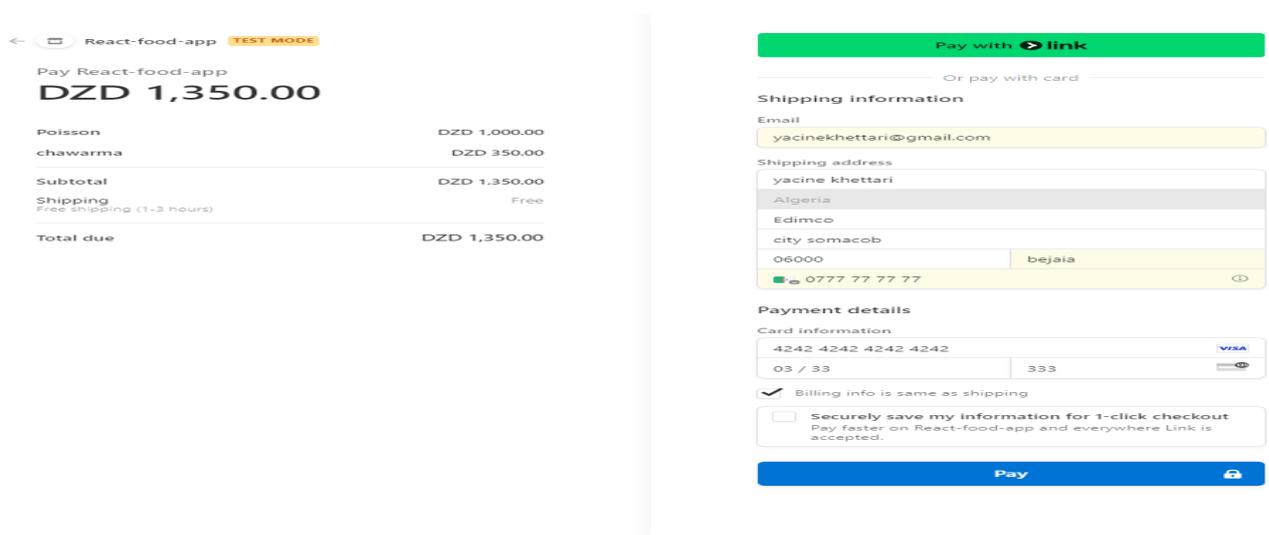


FIGURE 4.21 – Interface paiement commande avec stripe

c) Interface renvoyant au paiement de la commande via chargily, et donc pouvant utiliser au choix une carte cib ou edahabia.

The screenshot shows a payment interface powered by Chargily Pay. It is in 'Test mode' and displays the following details:

- Payee:** react-food-app
- Billing Details:**
  - Email: client@gmail.com
  - Full name: client
  - Phone number: 0777777777
  - Country: Algeria
  - State: bouira
  - Address: bouira
- Payment method:** Edahabia card (selected) and CIB card.
- Agreement:** I agree to the Terms Of Use.
- Amount:** Pay 230.00 DZD
- Buttons:** Cancel, Fail, and Expire.
- Summary:** Subtotal 230.00 DZD, Fees + 0.00 DZD, Total Due 230.00 DZD.

FIGURE 4.22 – Interface paiement de la commande chargily pay

d) Affichage de la commande créée avec succès peu importe le type de paiement utilisé.

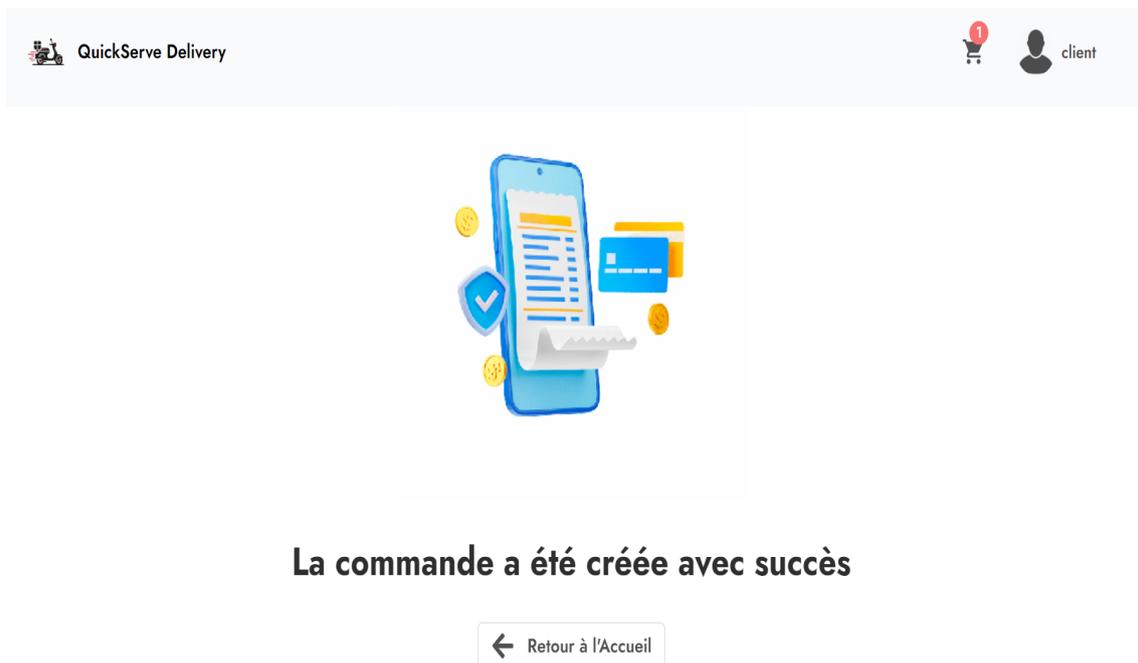


FIGURE 4.23 – Interface commande passée avec succès

### 4.5.3 Interfaces principales Administrateur

a) Interface d'accueil d'administrateur ou on peut voir les statistiques des commandes, et peut consulter les détails des nombres des produits.



FIGURE 4.24 – Interface du tableau de bord des statistiques des produits

b) Interface représentant la liste de toutes les commandes effectuées par les clients ou l'administrateur peut changer l'état des commandes (en préparation, prêt...).

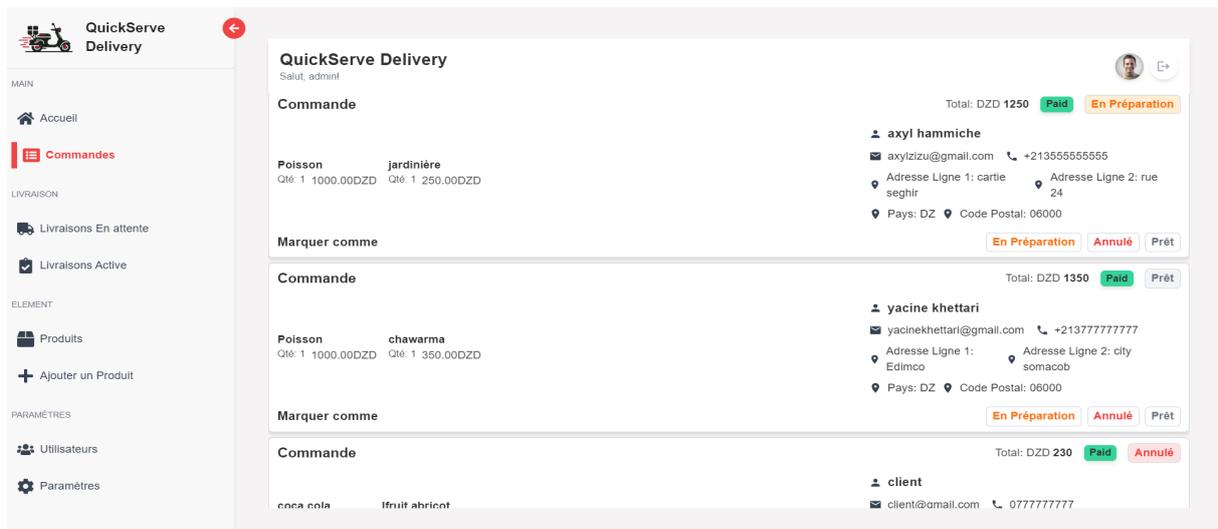


FIGURE 4.25 – Liste de toutes les commandes

c) Journal des commandes payées avec stripe .

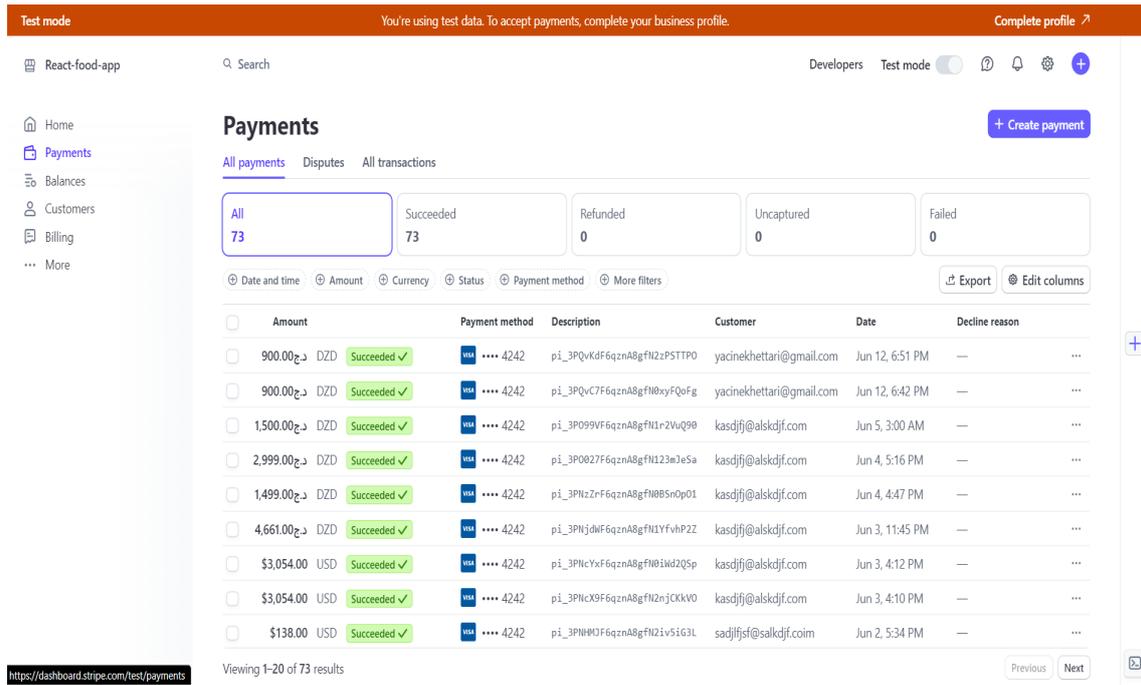


FIGURE 4.26 – Journal des Paiements effectués avec Stripe

d) Journal des commandes payées avec chargily pay.

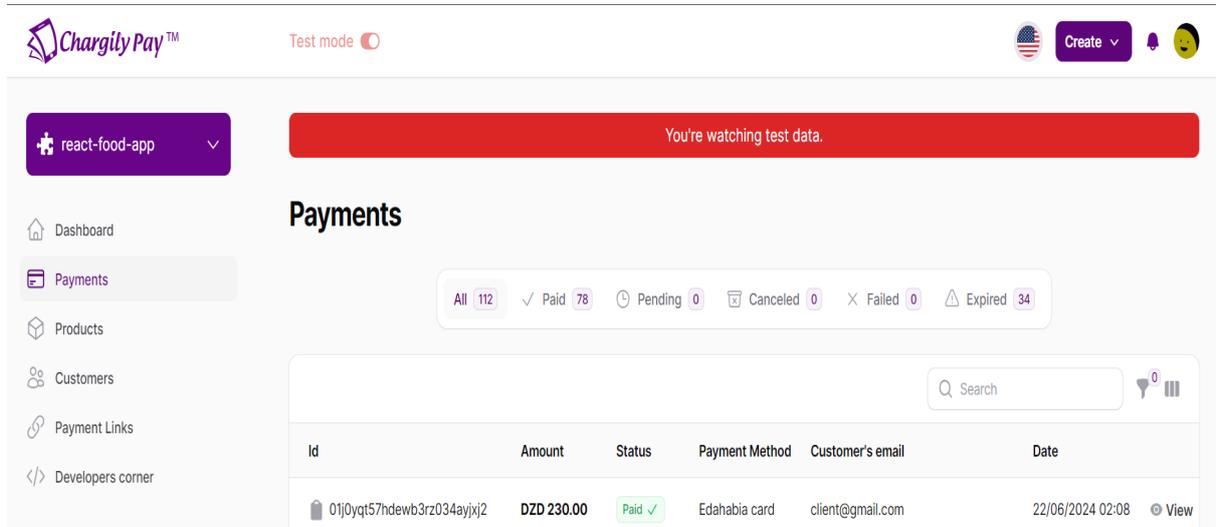


FIGURE 4.27 – Journal des commandes payées avec chargily pay

e) Interface de gestion des produits, où l'administrateur peut apporter des modifications à tous les produits.

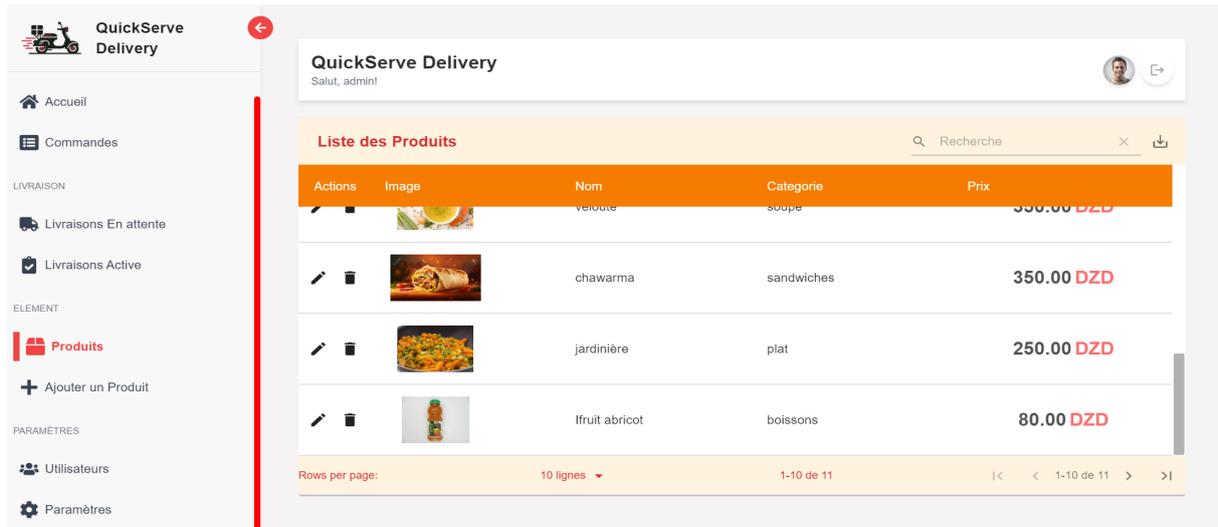


FIGURE 4.28 – Interface pour la gestion des produits

f) Interface représentant comment l'ajout d'un produit s'effectue par l'administrateur.

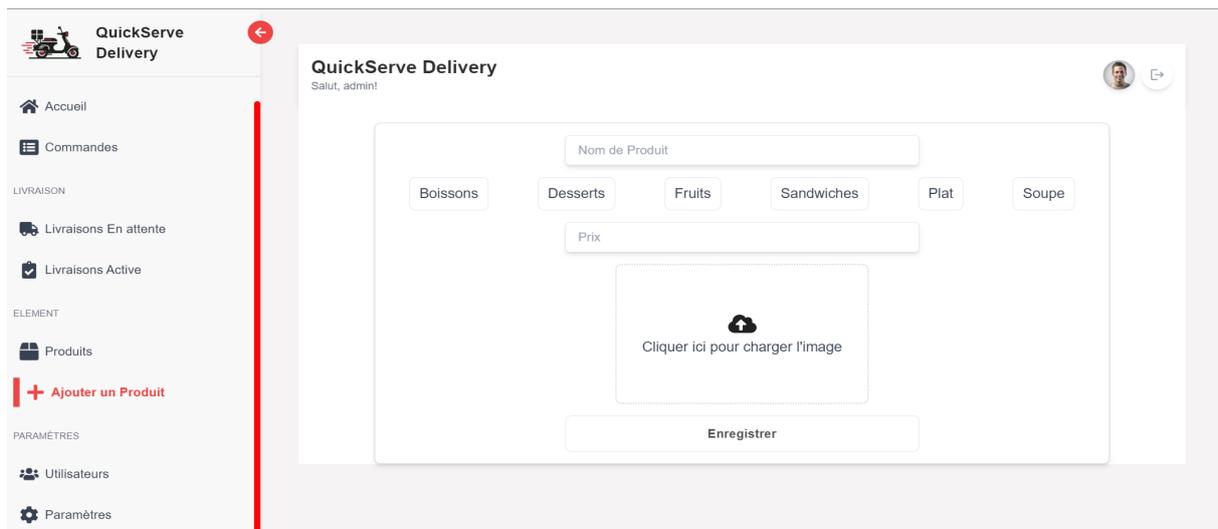


FIGURE 4.29 – Interface pour l'ajout d'un produit

g) Dans l'interface suivante l'administrateur a accès aux comptes utilisateurs pouvant ainsi transformer un compte client en un compte livreur ou en supprimer un.

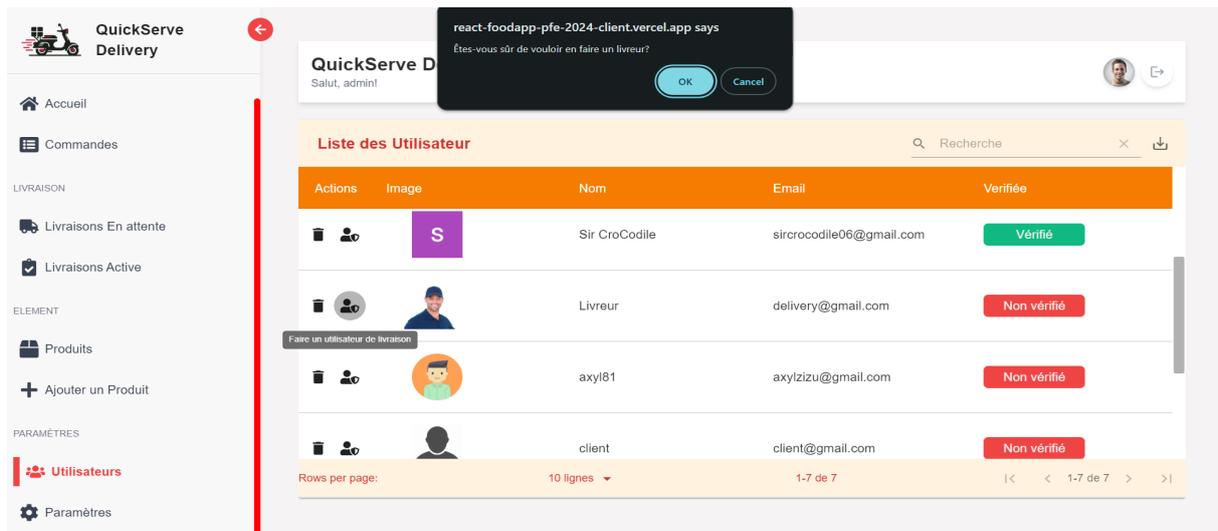


FIGURE 4.30 – Gestion des utilisateurs

h) Interface montrant comment la modification des informations personnelles du compte peut .

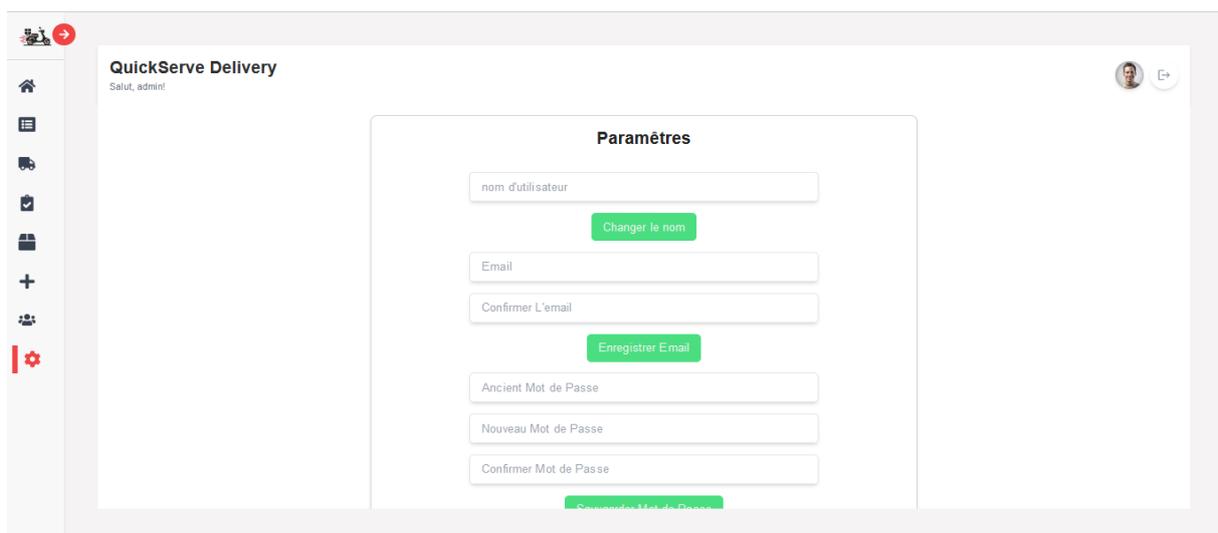


FIGURE 4.31 – Paramètres de gestion de compte

#### 4.5.4 Interfaces principales Livreur

Dans cette partie nous allons démontrer les 2 interfaces propres au livreur.

a) La figure 4.30 ci-dessous représente l'ensemble des commandes disponibles visibles par tous les livreurs.

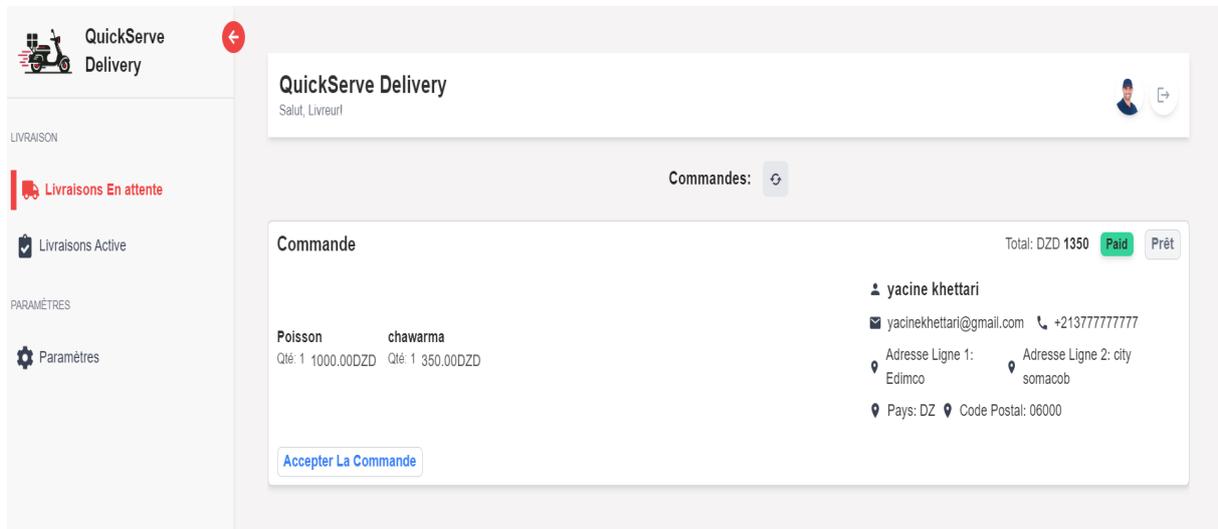


FIGURE 4.32 – Commande prête en attente d'acceptation

b) Dans ce qui suit l'interface propre à un livreur après avoir sélectionné une commande dans la section commandes actives.

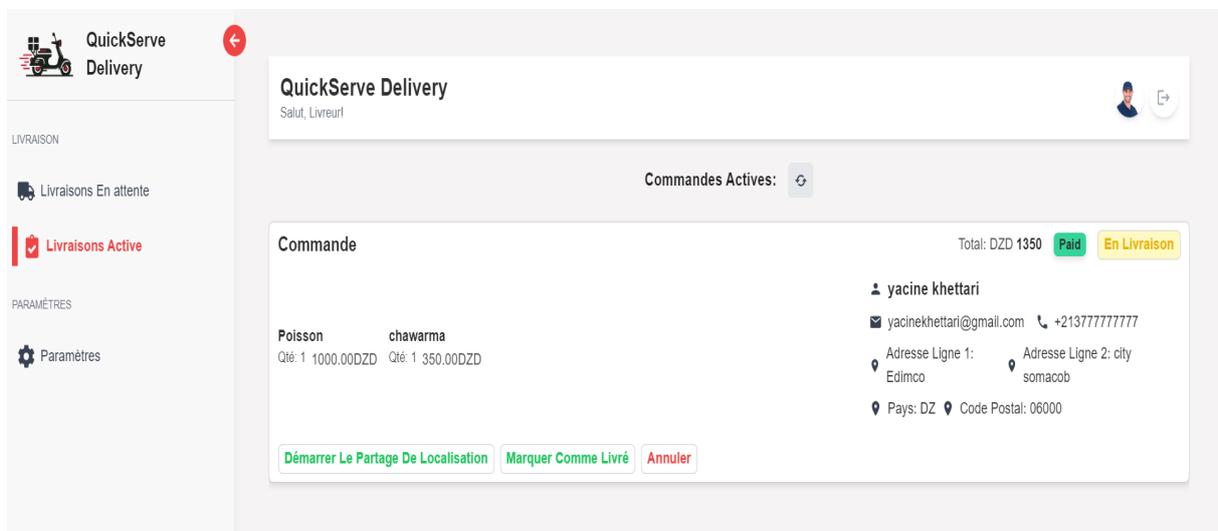


FIGURE 4.33 – Commande choisie par un seul livreur

### 4.5.5 Interfaces de l'application mobile

Dans ce qui suit nous allons montrer quelques interfaces de notre application mobile pour les différents utilisateurs, après avoir montré l'icône de notre application.

#### 4.5.5.1 Icône de l'application mobile

Voici ci-dessous l'icône de notre application mobile qui



FIGURE 4.34 – Icône de l'application mobile

#### 4.5.5.2 Interfaces Client

Dans cette partie nous allons illustrer les interfaces pour les différentes fonctionnalités propres aux clients.

a) Les interfaces ci-dessous représente l'accueil de l'application mobile.

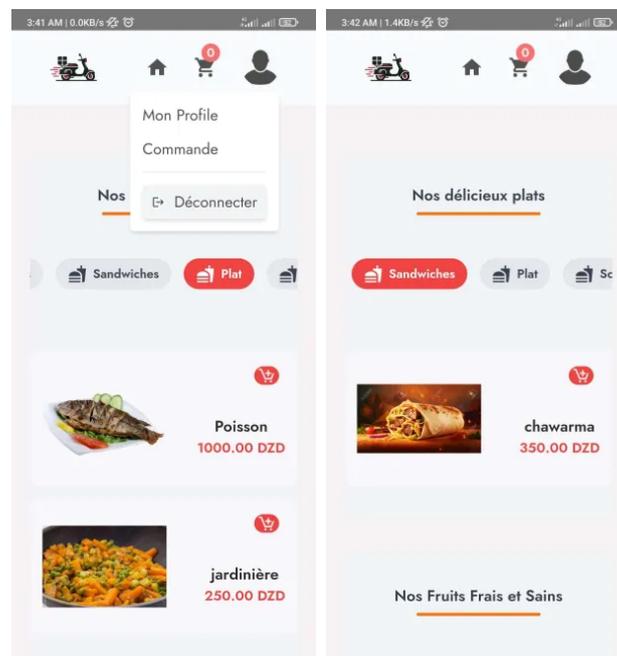


FIGURE 4.35 – Accueil application mobile Client

b) Voici ci-dessous le rendu lors de la création d'une commande via l'application mobile.

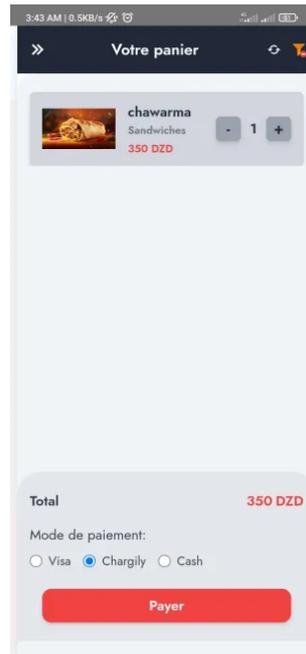


FIGURE 4.36 – Panier

#### 4.5.5.3 Interfaces Administrateur

Dans cette partie nous nous intéresserons aux fonctionnalités propres à l'administrateur.

a) Dans la figure 4.35 suivante on retrouve le rendu de l'accueil pour l'Administrateur lors de la connexion sur l'application mobile.

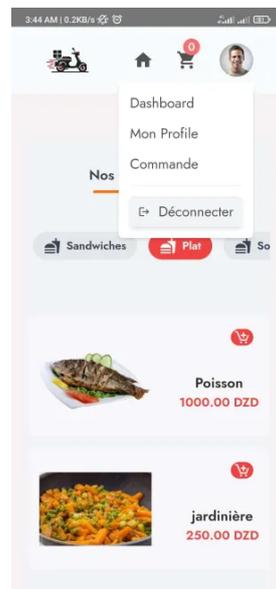


FIGURE 4.37 – Accueil admin mobile

b) L'interface représentée par la figure suivante montre l'historique de toutes les commandes effectuées par les clients ainsi que la possibilité de les trier selon l'état d'avancement.

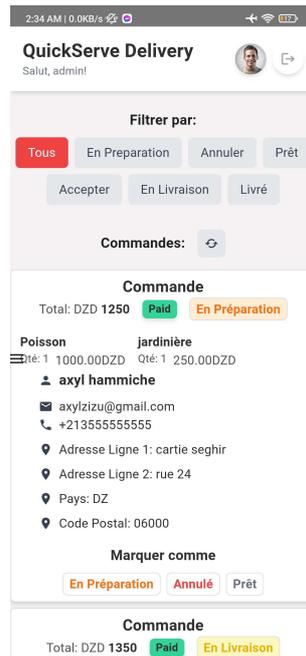


FIGURE 4.38 – Interface mobile administrateur de toutes les commandes

c) Dans ce qui suit nous avons l'interface du panel administrateur par lequel sont démontrées la plupart des actions possibles pour l'administrateur.

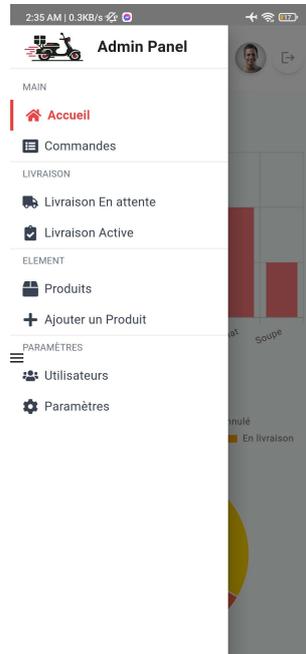


FIGURE 4.39 – Panel administrateur

### 4.5.6 Résultats du test

La figure suivante représente les résultats des différents tests de performance, d'accessibilité, de bonnes pratiques et de SEO effectués grâce au service lighthouse.



FIGURE 4.40 – Résultats des tests

## 4.6 Conclusion

Dans cet ultime chapitre nous avons présenter la réalisation de notre projet, décrivant ainsi tous les outils, langages et services utilisés afin de réaliser notre projet. Pour parfaire notre description nous avons présenter les différentes interfaces des principales fonctionnalités.

# CONCLUSION GÉNÉRALE ET PERSPECTIVES

Dans ce projet, nous avons conçu et réalisé une application permettant aux utilisateurs de passer une commande dans le secteur de la restauration et d'être livré. L'élaboration de ce projet a suivi plusieurs étapes. Dans un premier temps, nous avons réalisé une étude de l'existant, ce qui nous a permis d'examiner les solutions disponibles, et de poser la problématique et déterminer les objectifs auxquels notre solution doit répondre. En adoptant le processus de développement UP et le langage de modélisation UML, nous avons spécifié et analysé les besoins et conçu notre application web et android. La réalisation de notre application web QuickServe Delivery a été faite en utilisant principalement les outils : react js, firebase, express js. Quant à l'application mobile, nous avons opté pour l'outil capacitor afin de créer une application native et permettre sa distribution.

La réalisation de ce projet nous a permis d'enrichir nos connaissances dans les différentes technologies apprises au cours de notre dernière année tel que react tailwind ou express, sans oublier notre découverte de l'environnement firebase. Ce travail s'est avéré bénéfique, car il nous a donné un aperçu du monde professionnel, nous aidant à mieux nous organiser pour accomplir les tâches qui nous sont confiées dans les meilleures conditions et des délais les plus courts.

Cependant, des améliorations restent envisageables pour notre application. Nous comptons ajouter de nouvelles fonctionnalités, telles que :

- La prise en charge de plusieurs restaurateurs augmentant ainsi le choix sur notre application offrant une scalabilité accrue.
- intégrer un assistant vocal au lancement de l'application afin de permettre aux personnes en situation handicaps de passer leur commandes sans aide extérieure.
- mettre en place des modules permettant au restaurateur d'annoncer des promotions ou autres.
- inclure le système de code promo aussi bien pour les livraisons que pour le menu.

# Bibliographie

- [1] “cycle de vie processus up.” <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/>, Consulté le : 23 mars 2024.
- [2] “types de diagrammes uml.” <https://cours.khalilmamouny.com/uml-survol-des-14-diagrammes/>, Consulté le : 23 mars 2024.
- [3] “Wajeez.” <https://wajeez.dz/fr/>, Consulté le : 10 mars 2024.
- [4] “Yassir express.” <https://express.yassir.io/>, Consulté le : 12 mars 2024.
- [5] H. D. eurl, “Fast delivery.” <https://www.fastdelivery.dz/qui-sommes-nous/>, Consulté le : 17 mars 2024.
- [6] “Deliveroo.” <https://fr.wikipedia.org/wiki/Deliveroo>, Consulté le : 17 mars 2024.
- [7] “Ubereats.” <https://www.ubereats.com/fr>, Consulté le : 19 mars 2024.
- [8] “Justeat takeaway.” <https://careers.justeattakeaway.com/fr/fr/about-take-away>, Consulté le : 20 mars 2024.
- [9] “Interet des applications web et mobile.” <https://designli.co/blog/the-pros-and-cons-of-building-a-mobile-app-vs-a-web-app/>, Consulté le : 22 mars 2024.
- [10] I. Jacobson, G. Booch, and J. Rumbaugh, *Le processus unifié de développement logiciel*. Eyrolles, 2000.
- [11] P. Roques, *UML 2 : Modéliser une application web*. Eyrolles, 2007.
- [12] “Json web token.” <https://jwt.io/introduction>, Consulté le : 23 mai 2024.
- [13] C. A. Györödi, D. V. Dumșe-Burescu, D. R. Zmaranda, R. Györödi, G. A. Gabor, and G. D. Pecherle, “Performance analysis of nosql and relational databases with couchdb and mysql for application’s data storage,” *Applied Sciences*, vol. 10, no. 23, 2020. <https://www.mdpi.com/2076-3417/10/23/8524>, Consulté le : 3 mai 2024.
- [14] M. Fowler and P. J. Sadalage, *NoSQL Distilled : A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley Professional, 2012.
- [15] “Visual studio code.” <https://code.visualstudio.com/docs>, Consulté le : 3 mai 2024.
- [16] “Javascript.” <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, Consulté le : 3 mai 2024.

- 
- [17] “Nodejs.” <https://nodejs.org/docs/latest/api/documentation.html>, Consulté le : 3 mai 2024.
- [18] “Reactjs.” <https://fr.react.dev/>, Consulté le : 3 mai 2024.
- [19] “tailwind.” <https://v2.tailwindcss.com/docs>, Consulté le : 3 mai 2024.
- [20] “Express.” <https://expressjs.com/>, Consulté le : 3 mai 2024.
- [21] “Capacitor.” <https://capacitorjs.com/docs>, Consulté le : 3 mai 2024.
- [22] “Firebase.” <https://firebase.google.com/docs>, Consulté le : 3 mai 2024.
- [23] “Vercel.” <https://vercel.com/docs>, Consulté le : 01 juin 2024.
- [24] “Stripe.” <https://docs.stripe.com/>, Consulté le : 15 mai 2024.
- [25] “chargily pay.” <https://dev.chargily.com/pay-v2/introduction>, Consulté le : 17 mai 2024.
- [26] “Lighthouse.” <https://developer.chrome.com/docs/lighthouse/overview>, Consulté le : 01 juin 2024.

## Résumé

La gestion de commande et le suivi de livraison jouent un rôle crucial dans le domaine de la restauration, où la rapidité et l'efficacité sont essentielles pour satisfaire les clients et maintenir une réputation positive. Dans le cadre de ce mémoire de fin de cycle pour l'obtention d'un master en génie logiciel, nous avons conçu et réalisé QuickServe Delivery, une application web et mobile de gestion de commandes et de suivi de livraison dans le secteur de la restauration. QSD est une application performante ayant une interface conviviale et intuitive, elle permet aux clients de parcourir une sélection variée de plats, de passer leurs commandes en quelques clics, et de suivre leur progression en temps réel jusqu'à leur porte. Que ce soit pour une commande rapide à domicile, au bureau ou lors d'événements spéciaux, notre application garantit une satisfaction client optimale à chaque étape du processus de livraison.

**Mots clés :** Restauration, Service de livraison, Gestion de commande, Application mobile, Application web, QuickServe Delivery.

## Abstract

Order Management and delivery tracking play a crucial role in the restaurant industry, where speed and efficiency are essential to satisfy customers and maintain a positive reputation. As part of this final year project for obtaining a master's degree in software engineering, we have designed and developed QuickServe Delivery, a web and mobile application for order management and delivery tracking in the restaurant sector. QSD is a high-performance application with a user-friendly and intuitive interface. It allows customers to browse a wide selection of dishes, place their orders with just a few clicks, and track their progress in real time up to their doorstep. Whether for a quick order at home, at the office, or during special events, our application guarantees optimal customer satisfaction at every stage of the delivery process.

**Keywords :** Restaurant industry, Delivery service, Order management, mobile application, Web application, QuickServe Delivery.