

REPUBLIQUE DU CAMEROUN

Paix-Travail-Patrie

MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR

Université de Yaoundé I

Institut Saint Jean



REPUBLIC OF CAMEROON

Peace-Work-Fatherland

MINISTRY OF HIGHER
EDUCATION

University of Yaounde I

Institut Saint Jean



RAPPORT DE STAGE ACADEMIQUE

CONCEPTION ET RÉALISATION D'UNE APPLICATION WEB ET MOBILE DE GESTION DES COMMANDES : CAS DE ASIEL

Stage effectué du **7 avril 2025** au **7 juillet 2025** à **Mickmaq**

En vue de l'obtention d'une Licence Professionnelle en Informatique.

Option : Conception et développement d'applications pour l'économie numérique

Rédigé et présenté par

PAYONG KOUAWOUO BRICE VALERY

Etudiant en **3ème année Licence Professionnelle**

Sous l'encadrement

Académique de :

M. TAPIGUE Delphin

Enseignant à l'institut Universitaire Saint Jean

Professionnel de :

M. FOUAPON Hassan

Directeur Mickmaq Cameroun

Année Académique

2024-2025

DEDICACE

À ma famille

REMERCIEMENTS

L'aboutissement de ce travail n'est pas uniquement le fruit de notre effort personnel, mais également celui de nombreuses personnes qui nous ont soutenus et accompagnés tout au long de cette expérience.

Nous tenons tout d'abord à exprimer notre profonde gratitude à Monsieur **FOUAPON Hassan**, Directeur Général de l'entreprise **Mickmaq**, pour avoir accepté notre demande de stage et nous avoir accueillis au sein de sa structure.

Nos remerciements s'adressent également à notre encadreur académique, **Monsieur TAPIGUE Delphin**, pour ses conseils avisés, sa disponibilité et son accompagnement tout au long de cette année.

Nous remercions sincèrement tous les **membres de notre famille** ainsi que nos **enseignants de l'Université Saint Jean**, dont le soutien constant a été précieux à la réalisation de ce travail.

Enfin, nous adressons nos remerciements les plus chaleureux à **l'ensemble du personnel de Mickmaq**, pour leur disponibilité, leur bienveillance et leur collaboration tout au long de notre stage.

SOMMAIRE

LISTE DES TABLEAUX	4
LISTE DES FIGURES.....	5
LISTE DES ABREVIATIONS	6
AVANT-PROPOS	7
INTRODUCTION GÉNÉRALE.....	8
CHAPITRE I : PRÉSENTATION DE L'ENTREPRISE ET SON ENVIRONNEMENT	9
INTRODUCTION	10
I PRÉSENTATION DE MICKMAQ	10
II DÉROULEMENT DU STAGE	14
CONCLUSION	16
CHAPITRE II : IDENTIFICATION DU PROBLÈME	17
INTRODUCTION	18
PRÉSENTATION DU SERVICE ASIEL.....	18
I IDENTIFICATION ET ORIGINE DU PROBLÈME	19
II SOLUTIONS EXISTANTES.....	20
CONCLUSION	21
CHAPITRE III : SOLUTION ET IMPLÉMENTATION	22
INTRODUCTION	23
I DÉFINITION DU BESOIN ET SPÉCIFICATIONS DU PROJET ASIEL	23
II ARCHITECTURE DE LA SOLUTION	27
III MÉTHODE ET LANGAGE DE MODÉLISATION	29
IV TECHNOLOGIES UTILISÉES	40
V RÉSULTAT	42
VI DIFFICULTÉS RENCONTRÉES ET ÉVALUATION PERSONNELLE	45
CONCLUSION	46
CONCLUSION GÉNÉRALE.....	47
RÉFÉRENCES	IX
A) BIBLIOGRAPHIE	IX
B) WEBOGRAPHIE	IX
ANNEXES	X
TABLES DES MATIÈRES.....	XIII

LISTE DES TABLEAUX

Tableau 1: Emploi de temps	14
Tableau 2 Taches effectuées.....	16
Tableau 3: description textuelle use case Gestion des utilisateurs	31
Tableau 4: description textuelle use case Gestion des commandes	32
Tableau 5: use case gestion des produits	33
Tableau 6: description textuelle use case Gestion des produits	34
Tableau 7: Technologies utilisées	41
Tableau 8:Tableau des différentes tâches effectuées durant le stage	xi
Tableau 9: Fiche signalétique.....	xi

LISTE DES FIGURES

Figure 1: Organigramme de Mickmaq	13
Figure 2: Diagramme de gantt.....	15
Figure 3 : Architecture de la solution	28
Figure 4: Use case Gestion des utilisateurs	30
Figure 5: Use case Gestions des commandes	31
Figure 6: Diagramme de classe	34
Figure 7: diagramme de séquence d'authentification.....	36
Figure 8: Diagramme de séquence passer commande.....	37
Figure 9: Diagramme de déploiement	38
Figure 10: Déploiement application spring Boot sur Render	39
Figure 11: déploiement base de données MYSQL sur Railway.....	39
Figure 12: Interface connexion application web	42
Figure 13: interface d'administration des produits	42
Figure 14: Interface d'administration des commandes	43
Figure 15: mot de passe oublier.....	43
Figure 16: page login.....	43
Figure 17: connexion.....	43
Figure 18: Panier	44
Figure 19: Liste des commandes	44
Figure 20: Détails d'un produit	44
Figure 21: liste des produits	44
Figure 22: page d'accueil 2.....	44
Figure 23: page d'accueil	44
Figure 24: Localisation Mickmaq	xii

LISTE DES ABREVIATIONS

API	: Application Programming Interface
BD	: Base de Données
CMS	: Content Management System
CRM	: Customer Relationship Management
CRM	: Customer Relationship Management
CSV	: Comma-Separated Values
ERP	: Enterprise Resource Planning
HTTPS	: HyperText Transfer Protocol Secure
ISI	: Informatique et Systèmes d'Information
IUSJ	: Institut Universitaire Saint Jean
JDBC	: Java Database Connectivity
JWT	: JSON Web Token
PDF	: Portable Document Format
PERT	: Program Evaluation and Review Technique
RH	: Ressources Humaines
SARL	: Société À Responsabilité Limitée
SGBDR	: Système de Gestion de Base de Données Relationnelle
UML	: Unified Modeling Language
UTT	: Université des Technologies de Troyes
VS code	: Visual Studio Code
CESI	: Centre des Études Supérieures Industrielles
EFREI	: École Française d'Électronique et d'Informatique
EPF	: École Polytechnique Féminine
ESIEA	: École supérieure d'Informatique Électronique Automatique
CTI	: Commission des Titres d'Ingénieur
IPES	: Institution Privée d'Enseignement Supérieur

AVANT-PROPOS

Nouvellement accrédité CTI et Label EUR-ACE, Saint Jean ingénieur est la 1ère IPES (Institution Privée d'Enseignement Supérieur) camerounaise et une des toutes premières Ecoles privées en Afrique subsaharienne à bénéficier de cette prestigieuse reconnaissance. Cette accréditation obtenue pour 5 ans permettra aux détenteurs du Diplôme d'Ingénieur de se prévaloir du titre d'Ingénieur diplômé de l'état Français (Graduate Engineer), (en plus de l'état Camerounais) qui est une appellation largement reconnue à l'international, et garantit la qualité de la formation concernée. ISJ/SJI est sous la tutelle de l'Ecole Nationale Supérieure Polytechnique de Yaoundé (POLYTECH Yaoundé) Ses formations pour les Cycles Ingénieur et licence professionnelle se déclinent en Français et en Anglais, tandis que celle pour le cycle Master professionnel se fait en Français et en cours du soir. Voici nos écoles partenaires : UTT, 3IL, ISEP, EFREI, EPF, UMONS, ESTP, CESI, ESTP, INP GRENOBLE, INP Clermont, ESME, ESIEA, BUILDERS, UTBM, Ontario Tech University, Northern Alberta Institute of Technology.

INTRODUCTION GÉNÉRALE

Dans un contexte où le numérique transforme profondément les modes de fonctionnement des entreprises, les structures camerounaises s'adaptent progressivement à ces mutations technologiques pour renforcer leur compétitivité. C'est dans cette dynamique que s'inscrit l'Institut Universitaire Saint-Jean (IUSJ), qui, afin de favoriser l'insertion professionnelle de ses étudiants et de valoriser l'apprentissage par la pratique, impose un stage de fin de cycle de trois mois à tous les étudiants de licence. Ce stage a pour but de confronter les étudiants à des problématiques réelles d'entreprise et de leur permettre de mobiliser les compétences acquises en formation. C'est ainsi que nous avons effectué notre stage du 7 avril au 7 juillet 2025 au sein de **Mickmaq Cameroun**, une entreprise spécialisée dans le développement de solutions numériques et le e-commerce.

Durant cette période, nous avons participé à un projet initié par l'entreprise **ALEN SARL**, visant à développer une plateforme numérique dénommée **ASIEL**, dont l'objectif est de promouvoir les produits locaux camerounais via un système digital intégré. Dans le cadre de ce projet, notre mission principale a consisté à concevoir et à développer un **module de gestion des commandes**, accessible à la fois via une application mobile pour les clients, et une interface Web pour les administrateurs. Ce module devait permettre une gestion fluide, centralisée et en temps réel de toutes les commandes effectuées sur la plateforme.

Le problème posé peut se formuler ainsi : **comment concevoir un système numérique unifié permettant la gestion simultanée et efficace des commandes client sur mobile et leur administration via une interface web ?** En effet, de nombreuses plateformes locales souffrent encore d'un manque d'intégration entre les outils de vente et les systèmes de gestion, ce qui nuit à la performance globale et à l'expérience utilisateur. L'intérêt de cette étude réside dans la mise en œuvre d'une solution technique concrète, apportant une réelle valeur ajoutée à la structuration des processus commerciaux d'ASIEL. Elle contribue non seulement à améliorer le suivi des ventes et des livraisons, mais aussi à renforcer la transparence et la réactivité du système. Ce travail nous a permis d'évoluer dans un environnement professionnel exigeant, de développer notre autonomie et d'apporter une contribution tangible à un produit destiné à un usage à grande échelle. Ce rapport est structuré en trois chapitres, le premier chapitre présente l'environnement de travail et l'entreprise d'accueil, le deuxième chapitre expose la problématique, les objectifs, ainsi que les exigences du système et en fin le troisième chapitre détaille la conception, la mise en œuvre technique de la solution, et les résultats obtenus.

CHAPITRE I : PRÉSENTATION DE L'ENTREPRISE ET SON ENVIRONNEMENT

INTRODUCTION

Ce premier chapitre a pour objectif de présenter l'entreprise **Mickmaq**, son environnement organisationnel ainsi que le cadre dans lequel s'est déroulé le stage. Il est structuré en deux grandes sections.

Dans un premier temps, une présentation générale de Mickmaq sera faite. Cette section permettra de comprendre son historique, ses missions, sa vision ainsi que les principales activités qu'elle exerce dans le secteur des technologies de l'information. Une attention particulière sera portée sur l'organisation administrative de l'entreprise, en mettant en lumière la direction ou le service d'accueil dans lequel le stage a été réalisé. Il s'agira notamment de bien situer cette entité par rapport aux autres départements de l'entreprise, de démontrer son rôle stratégique dans le fonctionnement global de Mickmaq et de souligner l'intérêt de l'analyse qui en découlera.

Dans un second temps, le déroulement du stage sera présenté de manière détaillée. Cette partie décrira les différentes activités réalisées, les tâches accomplies, ainsi que les problèmes identifiés au cours de cette période d'apprentissage. Les causes et manifestations de ces problèmes seront analysées afin d'introduire les axes de réflexion abordés dans les chapitres suivants.

I PRÉSENTATION DE MICKMAQ

a) Contexte

Mickmaq est une plateforme africaine de marketing-commerce qui propose un marché numérique et une plateforme de livraison. Créée en 2019, Mickmaq propose des solutions informatiques et encourage l'utilisation des nouvelles technologies au Cameroun.

b) Missions

La mission de Mickmaq est principalement axée sur l'autonomisation et la pérennisation de l'usage des technologies. Cela inclut :

- ✓ Concevoir et réaliser des logiciels mobiles pour les entreprises et les particuliers ;
- ✓ Accompagner les jeunes entreprises dans leur croissance rapide grâce aux technologies informatiques ;
- ✓ Offrir des formations et des certifications pour renforcer les ressources humaines qualifiées dans divers domaines ;
- ✓ Contribuer activement au développement durable du monde à travers des solutions innovantes et la réalité virtuelle.

c) Vision

La vision de Mickmaq est de proposer une plateforme pour les grossistes, distributeurs, chauffeurs et petites entreprises, prête à exploiter des stratégies de commerce omnicanal pour accroître leurs ventes.

L'objectif est d'améliorer l'expérience d'achat en gros en ligne et de livraison, favorisant ainsi la croissance des petites entreprises en Afrique.

d) Activités

Les activités de Mickmaq couvrent plusieurs domaines, notamment l'**informatique**, l'**ingénierie** et la **formation**. On peut citer :

- ✓ Conception, réalisation et hébergement de sites web ;
- ✓ Développement et maintenance de logiciels ;
- ✓ Formations dans les domaines liés aux logiciels ;
- ✓ Conseil en informatique et innovation ;
- ✓ Support technique informatique.

e) Organisation administrative

Mickmaq est organisée administrativement comme suit :

La Direction Générale

Il s'agit du plus haut niveau hiérarchique de l'entreprise, chargé de :

- ✓ Veiller au bon fonctionnement de chaque département ;
- ✓ Définir les stratégies de projet ;
- ✓ Fournir un leadership et une orientation aux employés ;
- ✓ Prendre des décisions critiques pouvant affecter les opérations ou la réputation de l'entreprise ;
- ✓ Servir de point de contact pour les parties prenantes clés (investisseurs, partenaires, etc.).

Département des Ressources Humaines

Ce département est responsable de :

- ✓ Recruter et embaucher le personnel ;
- ✓ Servir d'intermédiaire entre les employés et la direction ;
- ✓ Gérer les programmes de rémunération et les avantages sociaux ;
- ✓ Développer et mettre en œuvre les politiques et procédures de l'entreprise, en conformité avec les lois du travail ;
- ✓ Assurer la gestion administrative des dossiers du personnel, des données et des systèmes RH.

Département de la Communication

Ce département est chargé de :

- ✓ Gérer les relations publiques (image et réputation de l'entreprise) ;

- ✓ Élaborer et diffuser des messages à destination des parties externes (clients, partenaires, etc.) ;
- ✓ Favoriser la communication interne efficace ;
- ✓ Créer et gérer des contenus sur différentes plateformes et canaux.

Département des Affaires Financières

Ce département est responsable de :

- ✓ Élaborer et gérer les plans financiers et budgets de l'entreprise ;
- ✓ Préparer et présenter des rapports financiers fiables et ponctuels à la direction, aux parties prenantes et aux autorités de régulation ;
- ✓ Gérer les coûts et dépenses de l'organisation ;
- ✓ Assurer la conformité aux réglementations fiscales, etc.

Département Technique

Ce département est chargé de :

- ✓ Gérer l'infrastructure technologique de l'entreprise (réseaux, matériel, serveurs, etc.) ;
- ✓ Participer à la gestion et la maintenance des données de l'entreprise ;
- ✓ Mettre en œuvre et maintenir des mesures de cybersécurité pour protéger les actifs numériques ;
- ✓ Gérer les relations avec les fournisseurs de technologies et les prestataires externes.

Département de Génie Logiciel

Ce département est responsable de :

- ✓ Concevoir et développer des applications logicielles et des systèmes ;
- ✓ Assurer la maintenance et le support des applications tout au long de leur cycle de vie ;
- ✓ Évaluer, planifier et réaliser des projets logiciels, etc.

f) Organigramme de Mickmaq

L'organigramme de Mickmaq, qui illustre la structure organisationnelle de l'entreprise ainsi que les différents postes et services disponibles se présente ainsi qu'il suit.

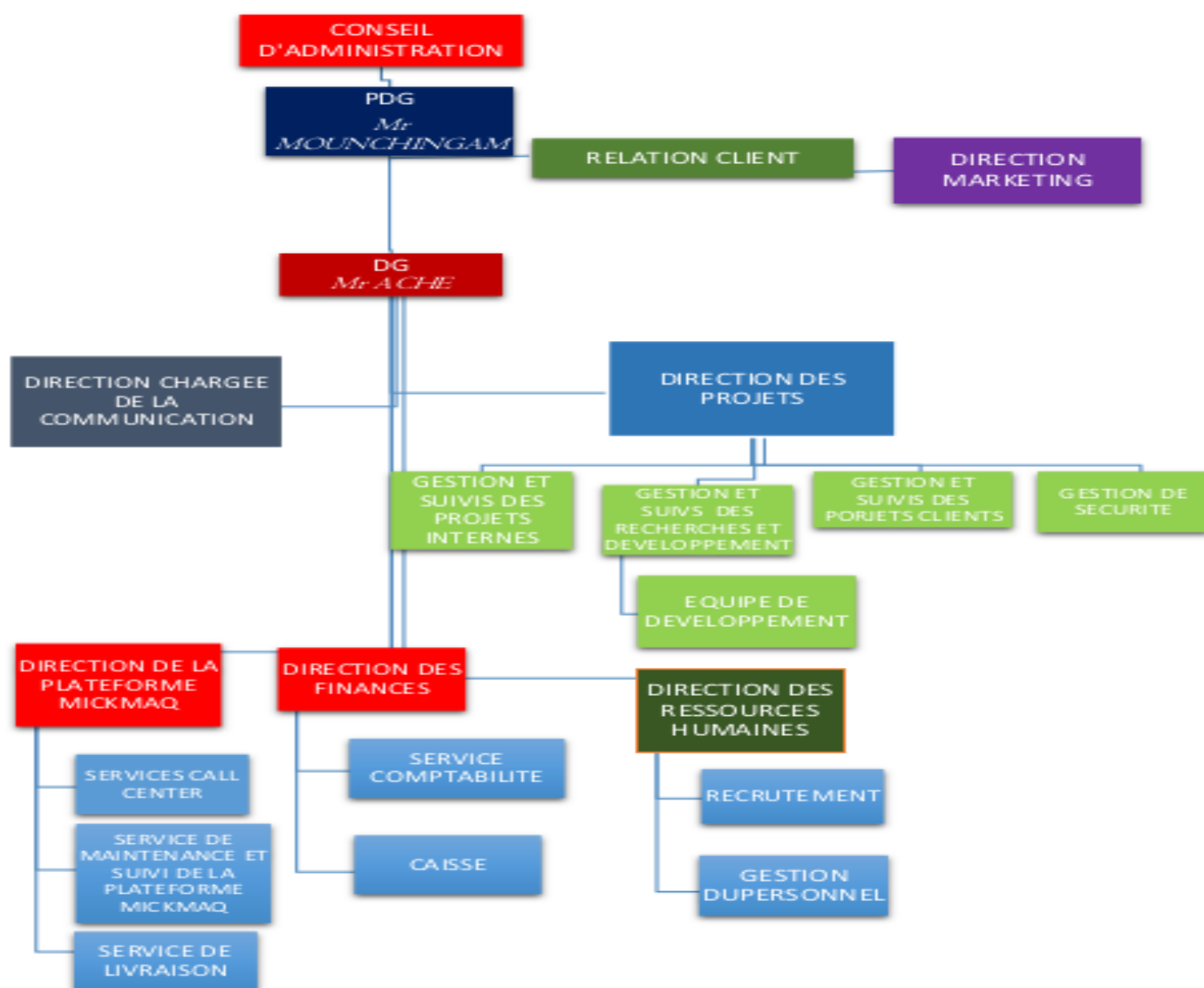


Figure 1: Organigramme de Mickmaq

II DÉROULEMENT DU STAGE

a) Accueil et intégration

À notre arrivée, nous avons été chaleureusement accueillis par l'équipe technique, sous la supervision de **Monsieur Hassan**, notre encadreur professionnel. Après une présentation générale de l'entreprise, nous avons rapidement été intégrés dans l'environnement de travail et mis en condition pour débiter le projet qui nous avait été assigné.

Les premiers jours ont été consacrés à la prise en main de l'environnement de développement, à l'installation des outils nécessaires (*Spring Boot, Angular, Ionic, MySQL, GitHub*, etc.), ainsi qu'à l'étude du cahier des charges fonctionnel. Nous avons ensuite été briefés sur les exigences techniques du projet, les normes de codage en vigueur dans l'entreprise, et les différents livrables attendus.

b) Emploi de temps

Jour/ heure	8h-00 12h-00	12h-00 13h-00	13h-00 17h-00
Lundi	Temps de travail	Pause	Temps de travail
Mardi	Temps de travail	Pause	Temps de travail
Mercredi	Temps de travail	Pause	Temps de travail
Jeudi	Temps de travail	Pause	Temps de travail
Vendredi	Temps de travail	Pause	Temps de travail
Samedi	Temps de travail	Temps de travail	

Tableau 1: Emploi de temps

c) Planification du travail

La **planification du travail** constitue une étape essentielle dans la réussite de tout projet. Elle permet de définir clairement les tâches à accomplir, de mobiliser efficacement les ressources disponibles, et d'anticiper les délais de réalisation. Une bonne planification facilite également le suivi de l'avancement du projet et la gestion des imprévus.

Dans le cadre de notre projet de stage, nous avons mis en place une planification rigoureuse afin d'assurer une organisation structurée du travail. Pour cela, nous avons utilisé un outil de gestion de projet bien adapté aux environnements informatiques : **le diagramme de Gantt**.

✓ Diagramme de Gantt

Le diagramme de Gantt est un outil visuel qui permet de représenter graphiquement le calendrier d'un projet. Chaque tâche est représentée par une barre horizontale dont la longueur correspond à la durée prévue de la tâche. Les tâches sont disposées sur une échelle de temps, permettant de visualiser leur séquence et leur chevauchement éventuel.

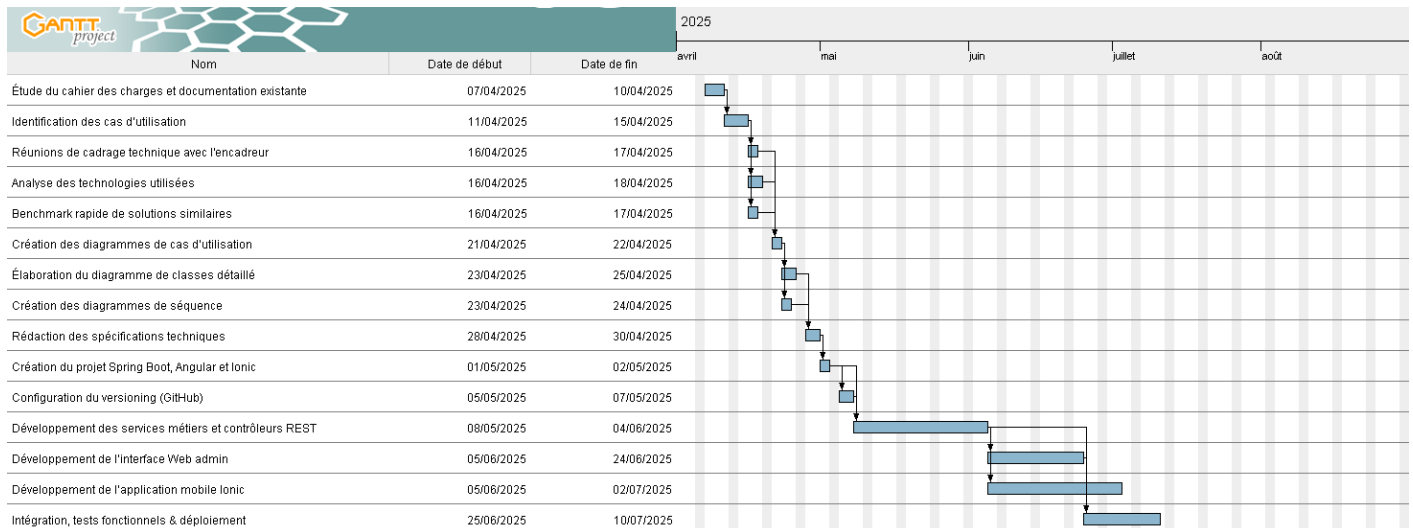


Figure 2: Diagramme de gantt

d) Taches effectuées

Le stage étant orienté projet, il s'est articulé autour de la conception et du développement d'un module de gestion des commandes pour la plateforme numérique **ASIEL**, un écosystème porté par l'entreprise cliente **ALEN SARL**. Le projet intègre des interfaces web et mobiles, et vise à centraliser la gestion des produits, des commandes et des utilisateurs.

Au cours des semaines suivantes, nous avons été chargés de plusieurs tâches spécifiques : modélisation UML, conception du modèle de base de données, développement du backend avec Spring Boot, implémentation des interfaces à l'aide d'Angular et Ionic, mise en place des APIs REST, intégration des WebSockets pour la gestion en temps réel des commandes, déploiement avec Docker, et enfin, rédaction de la documentation technique.

La suite de ce rapport présente, de manière chronologique, les différentes étapes, livrables et tâches que nous avons réalisées tout au long du stage.

Principales tâches effectuées durant le stage	Description des tâches effectuées	Commentaires
Analyse et prise en main du projet	Redaction du cahier des charges, identification des cas d'utilisation, réunions techniques avec le tuteur, benchmark de solutions similaires.	Une étape essentielle pour comprendre les objectifs du projet et ses contraintes techniques.

Modélisation et conception UML	Création des diagrammes de cas d'utilisation, diagramme de classes, de séquence, spécifications techniques et conception de la base de données.	Cette phase nous a permis de bien structurer l'application avant d'entrer dans la phase de développement.
Mise en place de l'environnement de travail	Installation des outils nécessaires (IDE, Git ...), création des projets Angular, Ionic et Spring Boot, mise en place du système de versioning.	Nous avons découvert plusieurs outils professionnels que nous n'avions jamais utilisés auparavant.
Développement du backend	Développement des entités, services, contrôleurs REST, sécurité JWT, tests unitaires et implémentation des WebSockets.	Une phase technique complexe, qui nous a beaucoup appris sur les architectures REST et la sécurité des API.
Développement des interfaces utilisateur (Web & Mobile)	Développement de l'interface d'administration avec Angular, application mobile Ionic avec authentification, panier, notifications, suivi des commandes.	Nous avons apprécié cette partie car elle nous a permis de voir l'impact direct de notre travail sur l'expérience utilisateur.
Déploiement, test et documentation finale	Intégration Web/Mobile/Backend, déploiement, tests finaux, rédaction de la documentation technique et du rapport de stage.	Une étape finale complète et satisfaisante qui a marqué l'aboutissement du projet.

Tableau 2 Taches effectuées

CONCLUSION

On retient que l'entreprise Mickmaq vise à devenir une plateforme de e-commerce innovante, facilitant la connexion entre consommateurs et commerçants, tout en offrant des services de livraison efficaces. Nous avons également observé que Mickmaq se distingue par son engagement envers des produits naturels et respectueux de l'environnement, visant à créer une communauté de parents et d'entreprises soucieux de durabilité.

CHAPITRE II : IDENTIFICATION DU PROBLÈME

INTRODUCTION

Ce chapitre vise à mettre en lumière la problématique principale rencontrée au cours du stage effectué au sein de l'entreprise MICKmaq, dans le cadre du projet ASIEL développé pour le compte de l'entreprise ALEN SARL. Il s'agira, dans un premier temps, d'identifier avec précision le problème observé, en analysant son origine, ses manifestations ainsi que ses conséquences sur le bon fonctionnement du projet. Dans un second temps, nous présenterons les solutions existantes susceptibles de répondre à cette problématique, tout en soulignant leurs limites par rapport au besoin spécifique du client. Cette analyse permettra de situer clairement la valeur ajoutée de l'intervention réalisée durant le stage.

I PRÉSENTATION DU SERVICE ASIEL

✓ Vision d'ASIEL

ASIEL est une initiative de l'entreprise **ALEN SARL**, une structure camerounaise dynamique évoluant dans les domaines de la **gestion**, de la **négociation** et du **commerce**. Consciente des défis de modernisation du commerce local et des enjeux de visibilité pour les producteurs nationaux, ALEN SARL a lancé le projet ASIEL' avec pour ambition de **révéler les richesses commerciales du Cameroun** à travers une plateforme numérique innovante.

La vision d'ASIEL repose sur la **valorisation des entreprises locales** dont les activités s'inscrivent dans une démarche communautaire, socialement responsable, et orientée vers un rayonnement sur les marchés national et international. En mettant en avant des commerces aux valeurs vérifiées, ASIEL' se positionne comme un **pont entre les entrepreneurs locaux et le consommateur moderne**, à travers une stratégie digitale structurée.

✓ Raisons de la création d'ASIEL

La création d'ASIEL s'appuie sur un ensemble d'observations concrètes liées aux réalités économiques et commerciales du Cameroun. Parmi les constats ayant motivé ce projet, on peut citer :

- **La difficulté d'accès aux commerces physiques** et le **non-respect fréquent des délais** de livraison ou de service ;
- **Le manque de canaux de distribution efficaces** pour les entrepreneurs locaux, combiné à **un déficit de visibilité** et à une préférence pour les produits étrangers ;
- **La disparition précoce de nombreuses entreprises locales**, faute de stratégie commerciale adaptée au contexte camerounais et de soutien de la part des consommateurs ;

- La **digitalisation croissante** des modes de consommation, qui oblige désormais les entreprises à s'aligner sur les usages numériques, notamment via Internet et les terminaux mobiles.

ASIEL' naît donc de la volonté d'**apporter des solutions concrètes et adaptées** à ces problématiques, en proposant une **plateforme numérique accessible, professionnelle et flexible**.

✓ Les services proposés par ASIEL

Dans sa configuration, ASIEL' propose une **expérience complète de e-commerce et de livraison** structurée autour de trois principaux services complémentaires :

- **ASIEL e-market** : Une plateforme de commerce électronique conçue pour répondre aux besoins des consommateurs tout en mettant en avant les producteurs et entrepreneurs locaux. Elle permet une navigation fluide, la constitution facile de paniers, la passation de commandes sécurisée, ainsi que le suivi transparent de leur traitement.
- **ASIEL Delivery (Livraison standard)** : Un service de livraison indépendant du e-market. Il permet d'acheminer des colis ou commandes passés sur la plateforme ou en dehors. Ce service garantit une **livraison standard** dans un **délai minimal de 4 heures**.
- **SUPER ASIEL (Livraison express)** : Un service de livraison ultra-rapide, destiné à répondre aux **besoins urgents des clients**, indépendamment de l'achat via l'application. Il propose des délais de livraison **beaucoup plus courts** que ceux du service standard, tout en assurant fiabilité et efficacité.

II IDENTIFICATION ET ORIGINE DU PROBLÈME

Durant notre stage, nous avons été impliqués dans le développement d'un système numérique appelé ASIEL. Ce système devait intégrer plusieurs modules, dont un **module de gestion des commandes** permettant aux clients de passer des commandes via une application mobile, et aux gestionnaires de suivre, traiter et administrer ces commandes via une interface Web.

Cependant, un problème majeur a été identifié : **l'absence d'un système intégré permettant la gestion centralisée et synchronisée des commandes entre les différents canaux d'utilisation (mobile et web)**. En effet, dans les premières versions de la plateforme, les commandes passées sur l'application mobile n'étaient ni correctement enregistrées, ni facilement administrables depuis le back-office, ce qui créait :

- ✓ Des pertes ou duplications de commandes ;
- ✓ Des retards dans le traitement et la livraison ;
- ✓ Une mauvaise expérience utilisateur (clients non informés, délais non respectés) ;

- ✓ Une surcharge de travail pour les gestionnaires (suivi manuel, erreurs humaines).

Ce dysfonctionnement impactait directement l'image de la plateforme ASIEL et compromettait la réussite de son lancement commercial, rendant urgente la mise en place d'une solution fiable.

III SOLUTIONS EXISTANTES

Pour répondre aux problématiques de gestion des commandes multicanaux, plusieurs solutions logicielles largement utilisées sur le marché peuvent être envisagées. Parmi elles, on retrouve :

1. Les CMS e-commerce (WooCommerce, Shopify, Prestashop)

Ces plateformes sont conçues pour la création rapide de boutiques en ligne. Elles intègrent nativement des modules de gestion de produits, de paiement, de livraison et de suivi des commandes.

Ce qu'elles apportent :

- ✓ Interface prête à l'emploi pour les utilisateurs et les administrateurs.
- ✓ Suivi des commandes, notifications client, gestion des stocks.
- ✓ Large écosystème de plugins pour ajouter des fonctionnalités.

Limites dans le contexte d'ASIEL :

- ✓ **Complexité d'adaptation** : ces outils ne sont pas conçus pour s'intégrer facilement à une application mobile externe existante. Adapter leur architecture aux besoins spécifiques d'ASIEL nécessiterait un développement lourd et coûteux.
- ✓ **Manque de personnalisation métier** : difficile de refléter des processus internes spécifiques (ex. types de livraisons différenciés comme SUPER ASIEL).

2. Les ERP/CRM personnalisables (Odoo, Zoho)

Ces outils permettent une gestion centralisée des ressources de l'entreprise, y compris les ventes, les clients et les commandes. Ils offrent des modules configurables, adaptés à diverses industries.

Ce qu'ils apportent :

- ✓ Gestion complète de l'activité (commandes, clients, livraisons, stocks).
- ✓ Tableau de bord décisionnel et reporting intégré.
- ✓ Automatisation des flux métiers.

Limites dans le contexte d'ASIEL :

- ✓ **Coût élevé** : les versions avancées ou personnalisées sont payantes, avec des frais de licence ou d'abonnement qui dépassent les moyens d'une jeune entreprise comme ASIEL.
- ✓ **Surcharge fonctionnelle** : ces outils sont souvent trop complexes ou trop larges pour un projet ciblé ; de nombreuses fonctionnalités resteraient inutilisées.
- ✓ **Intégration difficile** : relier ces systèmes à une application mobile Ionic et à une interface Angular personnalisée aurait entraîné des surcoûts techniques et logistiques importants.

En résumé, bien que ces solutions présentent des atouts indéniables, elles ne répondent pas de manière satisfaisante aux **exigences spécifiques du projet ASIEL**, notamment en ce qui concerne :

- ✓ L'intégration fluide entre l'application mobile et le back-office web,
- ✓ La personnalisation de la logique métier locale (types de produits, modes de livraison, rôles des utilisateurs),
- ✓ Les contraintes budgétaires et techniques du projet,
- ✓ La nécessité d'un système léger, autonome, fonctionnel même dans des environnements à faible connectivité.

Ainsi, il était impératif de concevoir une **solution sur mesure, légère, adaptée au contexte local**, et capable d'assurer la **gestion fluide des commandes en temps réel**, tout en étant compatible à la fois avec l'application mobile client et l'interface web d'administration.

CONCLUSION

Ce chapitre nous a permis d'identifier le **problème central** observé durant le stage, à savoir l'**absence d'un module intégré de gestion des commandes**, limitant ainsi la fluidité des opérations sur la plateforme ASIEL. Après avoir évalué les **solutions existantes** et leurs limites, nous avons mis en évidence la nécessité d'une solution **sur mesure**, conçue spécifiquement pour répondre aux besoins et contraintes du projet. L'élaboration d'un **cahier des charges rigoureux** a permis d'encadrer le développement du module de manière structurée et conforme aux attentes du client. Dans le chapitre suivant, nous décrirons la **solution technique mise en œuvre**, en présentant les différentes étapes de conception, de développement, de test et d'intégration du module de gestion des commandes.

CHAPITRE III : SOLUTION ET IMPLÉMENTATION

INTRODUCTION

Dans ce troisième chapitre, nous présentons de manière détaillée la solution mise en œuvre pour répondre efficacement au problème identifié au chapitre précédent, à savoir le manque d'intégration entre la gestion des commandes sur l'application mobile et l'interface Web d'administration. Cette solution a été développée dans le cadre du projet ASIEL, pour le compte de l'entreprise ALEN SARL, en partenariat avec Mickmaq. Nous exposerons dans un premier temps la démarche de modélisation adoptée, en précisant le langage utilisé et les outils mobilisés. Ensuite, nous décrirons le processus de conception et de réalisation de la solution, en mettant en avant les fonctionnalités clés du module de gestion des commandes. Enfin, une évaluation de la solution implémentée permettra de mesurer son impact et son efficacité.

I DÉFINITION DU BESOIN ET SPÉCIFICATIONS DU PROJET ASIEL

a) Mission

Développer une **application administrative centralisée** pour **ASIEL**, permettant de gérer les commandes issues de multiples canaux (e-market, ASIEL Delivery, SUPER ASIEL) et d'administrer l'écosystème mobile. Cette solution interconnectera une interface web (Angular) pour les administrateurs et une application mobile (Ionic) pour les clients via un backend unique (Spring Boot).

b) Objectifs

- ✓ **Centraliser la gestion** des commandes et des données opérationnelles.
- ✓ **Automatiser les processus** de suivi, notification et reporting.
- ✓ **Optimiser l'expérience administrateur** via une interface web intuitive.
- ✓ **Garantir la cohérence des données** entre l'app mobile et le backoffice.

c) Portée

L'application couvrira :

- ✓ Passation et suivi des commandes (mobile et web).
- ✓ Gestion des produits, vendeurs et utilisateurs.
- ✓ Notifications temps réel et reporting statistique.
- ✓ Administration technique de l'application mobile.

Hors périmètre :

- ✓ Gestion des paiements (intégration supposée existante).
- ✓ Modules externes (ex : logistique livraison).

d) Besoins Fonctionnels

Pour la solution proposée, on distingue les besoins suivants :

1. Authentification (Web et Mobile)

Le système doit assurer la gestion des accès utilisateurs, en tenant compte des rôles et niveaux d'autorisation. Les fonctionnalités attendues sont :

- ✓ Génération et gestion de tokens JWT pour les sessions sécurisées ;
- ✓ Restriction d'accès selon les rôles : administrateur, livreur, client, commerçant ;
- ✓ Création, modification et suppression de comptes utilisateurs par les super-administrateurs ;
- ✓ Gestion des sessions : expiration automatique, déconnexion manuelle, révocation de token ;

2. Gestion des Utilisateurs (Interface Web Admin)

Une interface centralisée permet la gestion globale des profils utilisateurs :

- ✓ Affichage de la liste des utilisateurs avec filtres (rôle, statut, activité récente) ;
- ✓ Création, modification, désactivation ou suppression de comptes ;
- ✓ Consultation de l'historique d'activités d'un utilisateur donné ;
- ✓ Gestion des rôles : client, commerçant, livreur, administrateur.

3. Passation de Commandes (Application Mobile – Client)

L'application mobile doit permettre aux clients de passer des commandes de manière fluide et intuitive :

- ✓ Consultation du catalogue de produits disponibles ;
- ✓ Ajout de plusieurs articles au panier ;
- ✓ Validation de commande à partir du panier ;
- ✓ Accès à l'historique des commandes passées ;
- ✓ Réception de notifications sur les mises à jour des commandes.

4. Suivi & Traitement des Commandes (Interface Web Admin)

Les administrateurs doivent pouvoir gérer l'ensemble du cycle de vie des commandes via une interface centralisée :

- ✓ Visualisation de toutes les commandes triées par statut (en attente, en cours, livrée, annulée) ;
- ✓ Accès aux détails complets de chaque commande (client, panier, adresse, livraison) ;
- ✓ Modification de l'état d'une commande ;
- ✓ Réactivation d'une commande échouée ou annulée ;

- ✓ Historique complet du traitement des commandes ;
- ✓ Filtres de recherche : par statut, client, commerçant, ou période.

5. Notifications & Alertes (WebSockets)

Le système doit être capable de transmettre des informations en temps réel aux utilisateurs concernés :

- ✓ Envoi de notifications push (mise à jour de commande, confirmation, livraison, etc.) ;
- ✓ Ciblage des notifications en fonction de critères spécifiques (zone géographique, rôle, activité, commande) ;
- ✓ Lancement de campagnes promotionnelles ou d'annonces événementielles ;
- ✓ Suivi de l'historique des notifications envoyées.

6. Statistiques & Reporting (Interface Web Admin)

L'interface d'administration doit proposer des outils de suivi et d'analyse des performances du système :

- ✓ Affichage d'indicateurs clés : nombre de commandes, livraisons effectuées, utilisateurs actifs ;
- ✓ Génération de graphiques dynamiques : produits les plus commandés, taux de livraison, etc. ;
- ✓ Export des données au format CSV ou PDF pour reporting externe ;
- ✓ Filtres de statistiques : par commerçant, produit, ville, catégorie ou période.

7. Gestion des Produits & Vendeurs (Interface Web Admin)

Les administrateurs doivent pouvoir gérer le contenu du catalogue et les profils des vendeurs :

- ✓ Création, modification et suppression de produits ;
- ✓ Attribution d'un produit à un vendeur et à une catégorie ;
- ✓ Gestion des statuts des produits : actif, en attente, rejeté, ou en rupture de stock ;
- ✓ Création, édition ou désactivation de comptes vendeurs.

e) Besoins Non Fonctionnels

Outre les fonctionnalités attendues, la solution doit également répondre à un certain nombre d'exigences non fonctionnelles. Celles-ci concernent la qualité globale du système en termes de sécurité, de performance, d'accessibilité, de disponibilité et d'évolutivité.

- ✓ **Sécurité** : Authentification sécurisée pour tous les utilisateurs (chiffrement des données sensibles).

- ✓ **Accessibilité multiplateforme** : Application mobile Android (pouvant être portée sous iOS), interface Web responsive.
- ✓ **Performance** : Le système doit répondre aux requêtes utilisateur en moins de 3 secondes.
- ✓ **Disponibilité** : Le système doit être disponible au minimum 99% du temps.
- ✓ **Extensibilité** : Le système doit permettre d'ajouter d'autres fonctionnalités comme un module de paiement ou de gestion de stock.
- ✓ **Sauvegarde et récupération** : Sauvegarde automatique des données critiques (commandes, produits, utilisateurs).
- ✓ **Ergonomie** : Interfaces intuitives adaptées aux utilisateurs non technophiles.

f) Caractéristiques des utilisateurs

❖ Profils d'utilisateurs

1. Clients (utilisateurs de l'application mobile)

- ✓ **Rôle** : Ce sont les personnes qui utilisent l'application pour passer une commande, payer et suivre la livraison.
- ✓ **Niveau d'instruction** : Très varié. Certains ont fait des études, d'autres non.
- ✓ **Expérience** : Certains sont habitués à utiliser des applications, d'autres non.
- ✓ **Connaissances techniques** : En général limitées. L'application doit donc être simple et intuitive.

2. Commerçants (vendeurs partenaires)

- ✓ **Rôle** : Ce sont les restaurateurs ou commerçants qui préparent les commandes passées par les clients via l'application.
- ✓ **Niveau d'instruction** : Formation liée à leur métier (cuisine, vente, etc.).
- ✓ **Expérience** : Habitués à gérer leur commerce, parfois aussi en ligne.
- ✓ **Connaissances techniques** : Moyennes. Ils utilisent le portail pour voir les commandes, mettre à jour leurs produits ou modifier leur disponibilité.

3. Administrateurs de la plateforme

- ✓ **Rôle** : Ils gèrent l'ensemble du système, assurent le bon fonctionnement de la plateforme, supervisent les utilisateurs et règlent les problèmes techniques.
- ✓ **Niveau d'instruction** : Diplômes en informatique, gestion ou domaines similaires.
- ✓ **Expérience** : Expérience en gestion de systèmes et en support.
- ✓ **Connaissances techniques** : Bon niveau. Ils s'occupent de la sécurité, des données, de la configuration des rôles, et de la génération de rapports.

❖ **Besoins et attentes des utilisateurs**

- ✓ **Facilité d'accès** : Tous doivent pouvoir accéder facilement à l'appli, quel que soit leur profil.
- ✓ **Interface simple** : L'application doit être claire, bien organisée, sans complexité inutile.
- ✓ **Aide disponible** : Un support doit être proposé pour les utilisateurs qui rencontrent des difficultés.
- ✓ **Suivi clair** : Les clients suivent leurs commandes, les livreurs leur trajet, les commerçants leurs ventes.
- ✓ **Outils adaptés** : Chaque utilisateur doit avoir des outils simples mais efficaces pour accomplir ses tâches rapidement.

II ARCHITECTURE DE LA SOLUTION

L'application repose sur une **architecture logicielle en 3 tiers (Three-Tier Architecture)**, qui permet une séparation claire des responsabilités, une meilleure évolutivité, et une maintenance facilitée.

Présentation des 3 Tiers :

❖ **Tier 1 : Couche Présentation (Front-End)**

C'est la couche visible par les utilisateurs. Elle se décline en deux applications distinctes :

- ✓ **Application mobile (Ionic)** : utilisée par les clients pour consulter les produits, passer des commandes, suivre les livraisons, recevoir des notifications, etc.
- ✓ **Application Web Admin (Angular)** : utilisée par les administrateurs, commerçants, et livreurs pour gérer les utilisateurs, commandes, produits, statistiques, etc.

Responsabilités :

- ✓ Affichage des interfaces utilisateurs (UI)
- ✓ Collecte et validation des entrées utilisateurs
- ✓ Communication avec l'API via des appels HTTP (REST)
- ✓ Affichage des notifications (via WebSocket)

Communication :

- ✓ Utilise **HTTPS** pour interagir avec la couche métier (API REST sécurisée)
- ✓ Reçoit des notifications temps réel via **WebSockets**

❖ **Tier 2 : Couche Métier / Logique (Back-End API)**

Cette couche est le **cœur du système**. Elle est responsable de la gestion des règles métier et du traitement des données.

Technologie envisagée : Java (Spring Boot)

Responsabilités :

- ✓ Authentification & autorisation (JWT, rôles)
- ✓ Traitement des commandes et gestion des statuts
- ✓ Gestion des utilisateurs, produits, etc.
- ✓ Application des règles de validation métier (stocks, livraisons, paiements...)
- ✓ Centralisation des logs, des erreurs, et de la journalisation

Communication :

- ✓ Fournit une **API RESTful** sécurisée pour le front-end
- ✓ Communique avec la base de données via l'ORM Hibernate

❖ Tier 3 : Couche Données (Base de Données)

Cette couche contient l'ensemble des données persistées de l'application.

Technologies :

Base relationnelle (MySQL) : pour les entités structurées (utilisateurs, commandes, produits, stocks...)

Responsabilités :

- ✓ Stockage sécurisé et structuré des données
- ✓ Exécution des requêtes via l'ORM Hibernate
- ✓ Sauvegarde, réplication et restauration des données critiques
- ✓ Support de la traçabilité (via journaux d'audit, timestamping, soft deletes...)

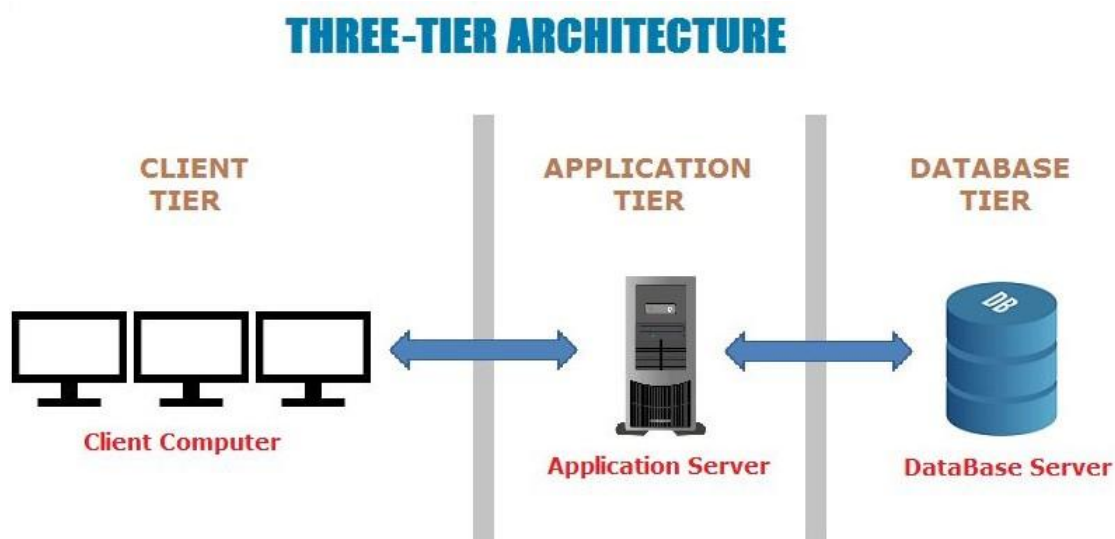


Figure 3 : Architecture de la solution

III MÉTHODE ET LANGAGE DE MODÉLISATION

a) Méthode de modélisation

Au cours de notre stage, nous avons choisi d'adopter la méthodologie agile Scrum pour la gestion de notre projet. Scrum est une méthode de gestion de projet qui met l'accent sur la collaboration, la communication et la livraison rapide de résultats. Cette approche repose sur des itérations courtes appelées sprints, permettant à l'équipe de travailler de manière efficace et itérative. Chaque semaine, nous avons défini des objectifs pour le sprint en cours et avons planifié les fonctionnalités à inclure dans ce sprint. Pendant le sprint, nous avons utilisé GitHub pour partager notre travail et fournir des livrables à la fin de chaque deux semaines.

b) Langage de modélisation

Pour modéliser le système, nous avons opté pour **UML (Unified Modeling Language)**, un langage de modélisation graphique largement utilisé dans le domaine de l'ingénierie logicielle. UML permet de représenter de manière standardisée les différentes composantes d'un système logiciel, tout en facilitant la communication entre les parties prenantes : analystes, développeurs, concepteurs, clients et testeurs.

L'objectif principal de l'utilisation d'UML est d'assurer une **compréhension claire, partagée et cohérente** du fonctionnement attendu du système, de sa structure, ainsi que de ses interactions internes et externes.

UML propose 14 types de diagrammes, répartis en deux grandes catégories :

- ✓ **Diagrammes structurels** (par exemple : diagramme de classes, de composants, de déploiement) ;
- ✓ **Diagrammes comportementaux** (par exemple : diagramme de cas d'utilisation, de séquence, d'activités).

Dans le cadre de ce projet, nous avons choisi d'utiliser les **quatre diagrammes suivants**, jugés les plus pertinents pour représenter les besoins et le fonctionnement de notre solution :

1. **Diagramme de cas d'utilisation** : pour illustrer les interactions entre les utilisateurs (acteurs) et le système, en identifiant les fonctionnalités principales attendues.
2. **Diagramme de classes** : pour modéliser la structure statique du système à travers les entités, leurs attributs et leurs relations.
3. **Diagramme de séquence** : pour représenter les échanges entre les objets du système dans le temps, notamment lors du traitement d'une commande.

4. Diagramme de déploiement : pour représenter les différents composants logiciels et les environnements matériels sur lesquels ils sont déployés.

c) Les diagrammes

✓ Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est utilisé pour représenter les fonctionnalités du système du point de vue de l'utilisateur. Il permet de visualiser les différents cas d'utilisation, les acteurs impliqués dans ces cas d'utilisation et les relations entre les acteurs et les cas d'utilisation. Le diagramme de cas d'utilisation est utile pour comprendre les besoins et les exigences des utilisateurs et pour identifier les fonctionnalités qui doivent être implémentées dans le code.

Gestion des utilisateurs.

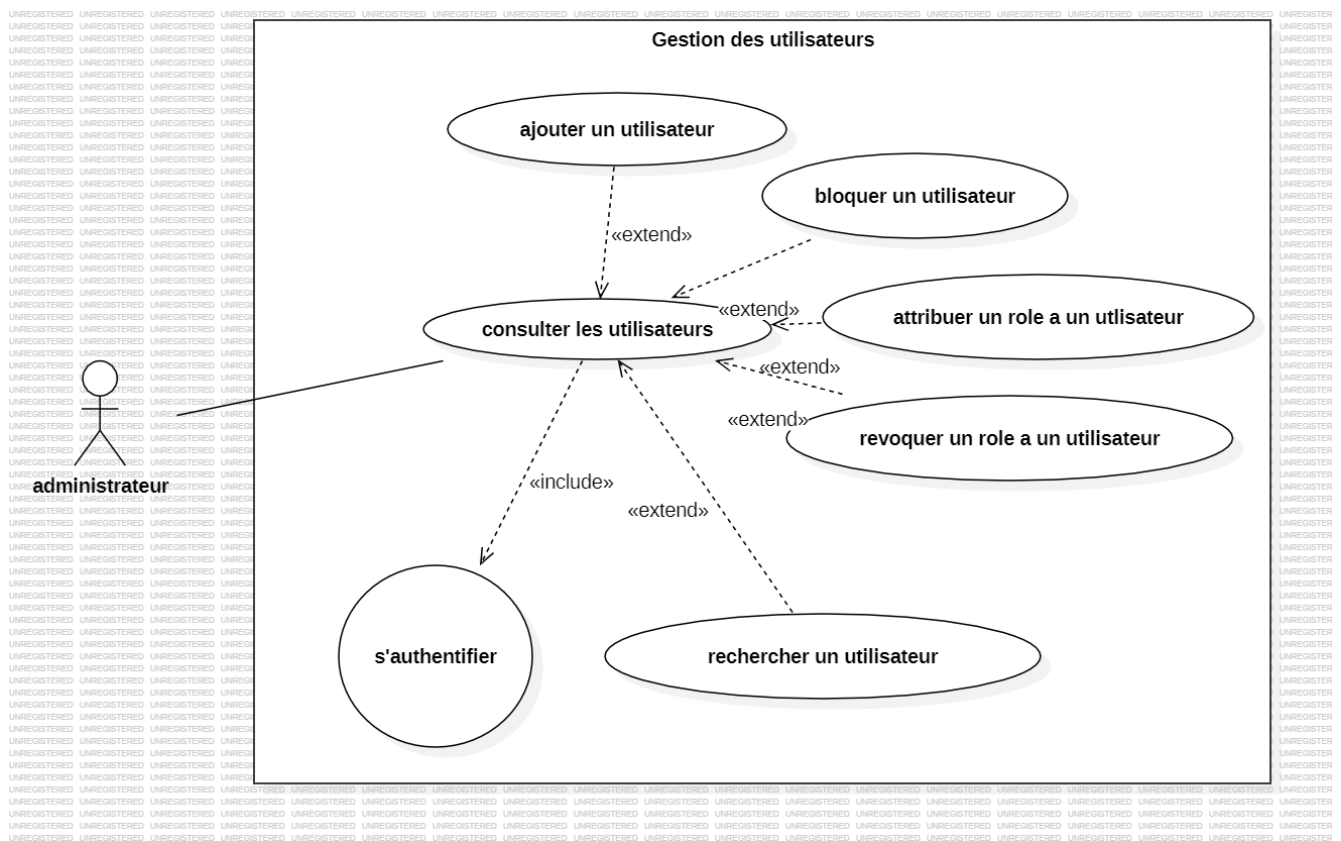


Figure 4: Use case Gestion des utilisateurs

Intitulé	Gestion des utilisateurs
Objectif	Permettre à l'administrateur de consulter, créer, modifier, désactiver des comptes utilisateurs et gérer leurs rôles.
Acteurs	Administrateur
Préconditions	- L'administrateur est authentifié avec un compte disposant des droits appropriés.

	- La plateforme et la base de données sont accessibles.
Post-conditions	<ul style="list-style-type: none"> - La liste des utilisateurs est mise à jour selon l'opération réalisée. - Les modifications sont immédiatement enregistrées en base de données.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur accède à la section « Gestion des utilisateurs ». 2. Il clique sur « Consulter la liste des utilisateurs » pour afficher tous les comptes. 3. Pour ajouter, il sélectionne « Ajouter un utilisateur », saisit les informations (nom, email, rôle, mot de passe) et valide. 4. Pour modifier, il choisit « Modifier un utilisateur », met à jour les champs et enregistre. 5. Pour désactiver, il clique sur « Désactiver un utilisateur » et confirme. 6. Pour gérer les rôles, il sélectionne « Attribuer/Révoquer rôles », ajuste les permissions et confirme. 7. Le système affiche à chaque fois un message de confirmation.
Scénarios alternatifs	<ul style="list-style-type: none"> - Erreur de saisie : Si des données sont invalides (ex. email mal formé), le système affiche une erreur et propose la correction. - Conflit de rôle : Si l'attribution d'un rôle viole une règle de sécurité, l'opération est bloquée et un avertissement s'affiche. - Désactivation impossible : Si l'utilisateur a des commandes en cours, la désactivation est refusée et le système liste les commandes à traiter d'abord.

Tableau 3: description textuelle use case Gestion des utilisateurs

Gestion des Commandes.

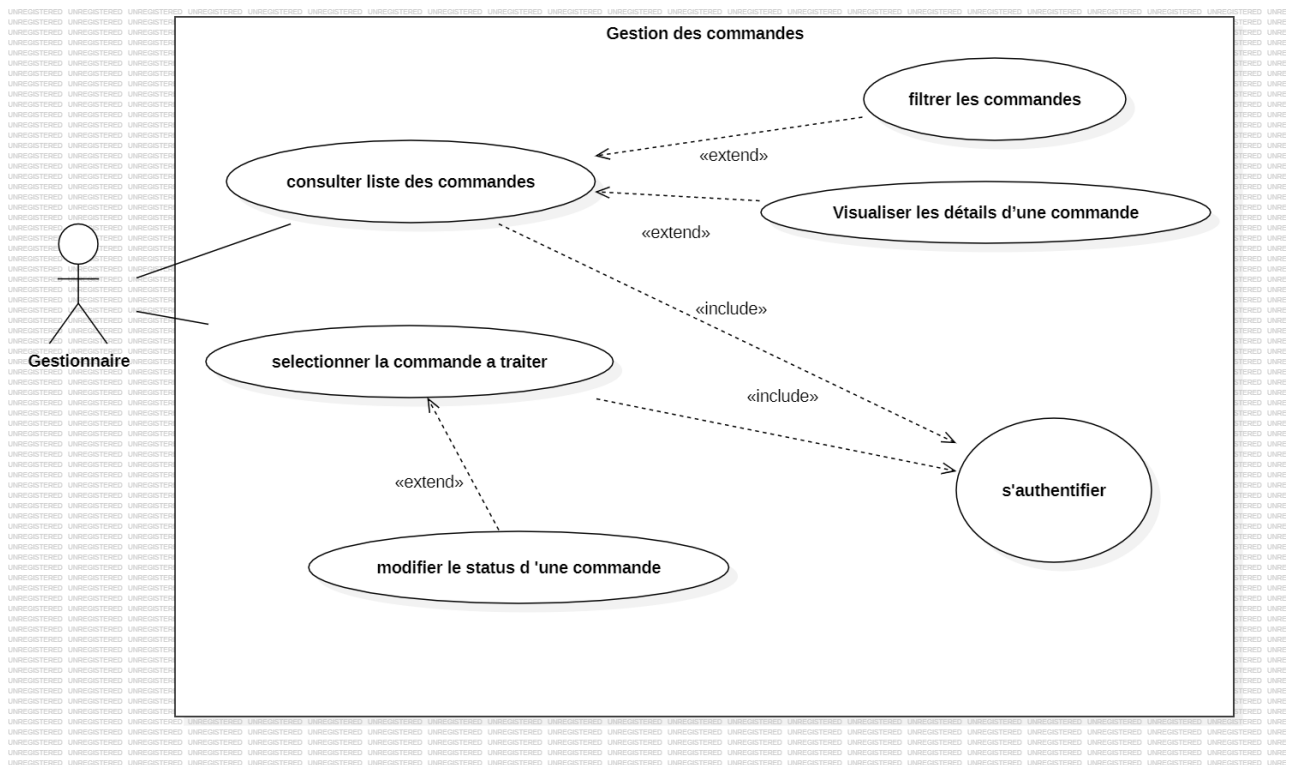


Figure 5: Use case Gestions des commandes

Intitulé	Gestion des commandes
Objectif	Permettre au gestionnaire de visualiser, filtrer et mettre à jour le statut des commandes sur la plateforme.
Acteurs	Gestionnaire
Préconditions	<ul style="list-style-type: none"> - Le gestionnaire est authentifié avec un compte disposant des droits appropriés. - Des commandes existent déjà dans le système. - La plateforme et la base de données sont accessibles.
Post-conditions	<ul style="list-style-type: none"> - Le ou les statuts des commandes sélectionnées sont mis à jour en base de données. - Les clients et livreurs concernés reçoivent une notification du changement de statut.
Scénario principal	<ol style="list-style-type: none"> 1. Le gestionnaire accède à la section « Gestion des commandes ». 2. Il consulte la liste des commandes reçues. 3. Il applique éventuellement des filtres (date, statut, client). 4. Il sélectionne une commande à traiter. 5. Il modifie le statut (par exemple : EN_COURS → LIVRÉE) et confirme. 6. Le système enregistre la mise à jour et affiche un message de succès. 7. Le module déclenche l'envoi d'une notification au client et/ou au livreur.
Scénarios alternatifs	<ul style="list-style-type: none"> - A1. Filtre sans résultat : Si aucun enregistrement ne correspond aux critères de filtre, un message indique « Aucune commande trouvée » et le gestionnaire peut modifier ses critères. - A2. Statut invalide : Si le statut choisi n'est pas autorisé selon la règle métier (ex. passage direct de EN_ATTENTE à ANNULÉE sans raison), le système bloque l'opération et affiche un avertissement. - A3. Erreur de mise à jour : En cas de problème technique (base inaccessible), un message d'erreur s'affiche et invite à réessayer plus tard.

Tableau 4: description textuelle use case Gestion des commandes

Gestion des produits

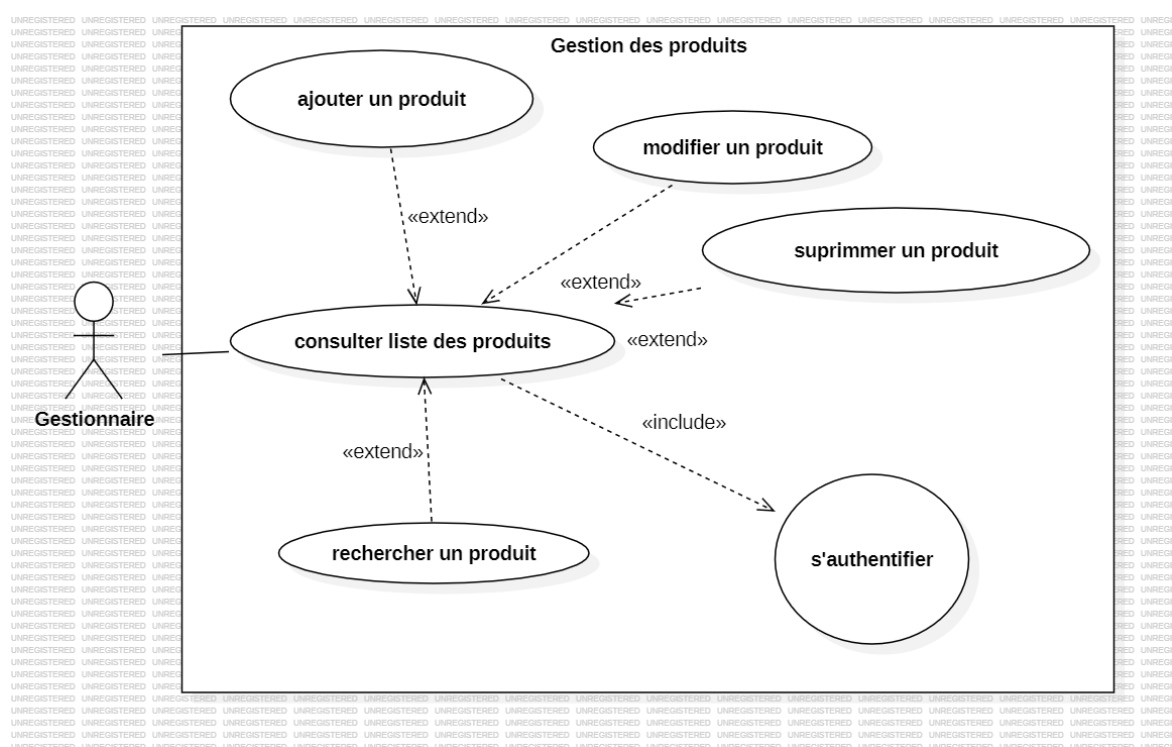


Tableau 5: use case gestion des produits

Intitulé	Gestion des produits
Objectif	Permettre au gestionnaire d'ajouter, modifier, consulter ou supprimer des produits sur la plateforme.
Acteurs	Gestionnaire
Préconditions	<ul style="list-style-type: none"> - Le gestionnaire est connecté avec les droits adéquats. - La base de données produit est opérationnelle.
Post-conditions	<ul style="list-style-type: none"> - Les informations des produits sont mises à jour dans le système. - Les modifications sont immédiatement visibles pour les clients.
Scénario principal	<ol style="list-style-type: none"> 1. Le gestionnaire accède à la section « Gestion des produits ». 2. Il consulte la liste des produits existants. 3. Il clique sur un bouton pour ajouter un nouveau produit ou modifier un produit existant. 4. Il remplit ou édite les champs requis (nom, prix, stock, description, image). 5. Il valide l'action. 6. Le système enregistre les modifications et affiche un message de confirmation.
Scénarios alternatifs	<ul style="list-style-type: none"> - A1. Produit déjà existant : Si le nom du produit est déjà utilisé, un message d'erreur s'affiche et empêche l'ajout. - A2. Champs invalides : Si des champs obligatoires sont vides ou incorrects (ex : prix négatif), le système bloque l'action.

- **A3. Erreur technique** : Si la base de données est inaccessible, le système affiche une erreur et annule l'opération.

Tableau 6: description textuelle use case Gestion des produits

✓ Diagramme de classe

Le diagramme de classes présenté ci-dessous illustre la structure logique de notre application de gestion des commandes. Il met en évidence les principales entités du système telles que les utilisateurs, les vendeurs, les produits, les commandes et les notifications. Chaque classe regroupe des attributs pertinents ainsi que les opérations qu'elle peut effectuer, ce qui reflète le comportement attendu dans le système. Les relations (comme les associations ou les cardinalités) permettent de comprendre comment ces éléments interagissent entre eux : par exemple, un utilisateur peut passer plusieurs commandes, chaque commande contient plusieurs produits, et un vendeur peut gérer plusieurs articles. Ce diagramme sert de référence pour la conception technique, en assurant une cohérence entre les besoins fonctionnels et la structure du code.

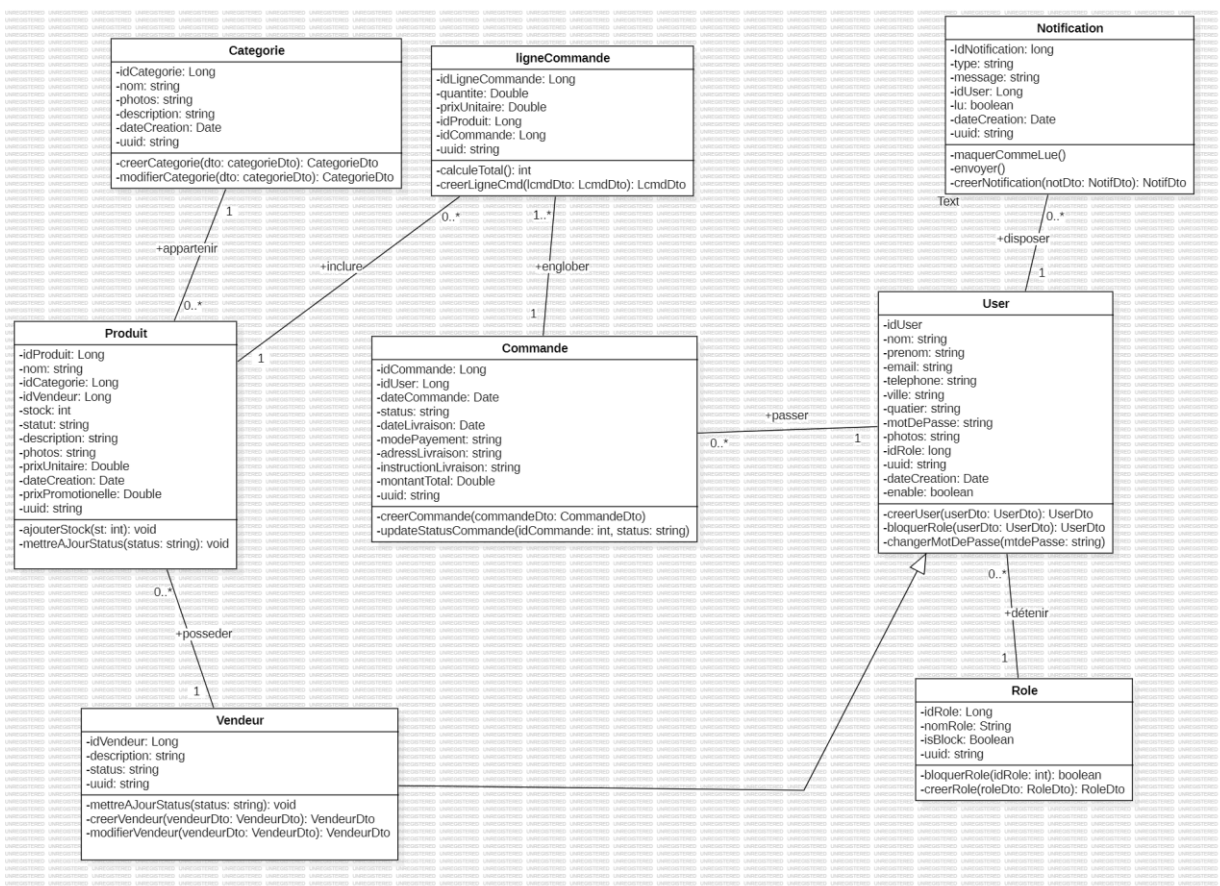


Figure 6: Diagramme de classe

Description textuelle du diagramme de classes

- ✓ **User** : Représente un utilisateur de la plateforme. Il contient des informations personnelles telles que le nom, l'adresse email, le mot de passe, ainsi que des données

de localisation. Chaque utilisateur est associé à un rôle qui définit ses permissions d'accès au système (client, vendeur, administrateur...).

- ✓ **Vendeur** : Entité représentant les fournisseurs de produits. Chaque vendeur peut publier plusieurs produits. Il possède un statut (actif, inactif) et des données descriptives telles que l'email, le téléphone et les informations d'identification.
- ✓ **Produit** : Objet mis en vente sur la plateforme. Chaque produit appartient à une catégorie et est lié à un vendeur. Il possède des attributs comme le stock, le prix, la description et des photos, et peut également avoir un prix promotionnel.
- ✓ **Catégorie** : Sert à organiser les produits en groupes logiques. Chaque produit appartient à une seule catégorie, qui permet de faciliter la navigation et la recherche sur la plateforme.
- ✓ **Commande** : Représente une transaction effectuée par un utilisateur. Elle contient la date, le mode de paiement, l'adresse de livraison, un statut, ainsi que le montant total. Une commande peut contenir plusieurs lignes de commande.
- ✓ **LigneCommande** : Détaille un article spécifique dans une commande. Elle indique la quantité, le prix unitaire, et la référence du produit. Elle permet de gérer les contenus des paniers et des factures.
- ✓ **Notification** : Permet d'alerter les utilisateurs lors d'événements clés (commande validée, livraison en cours, compte bloqué...). Chaque notification est liée à un utilisateur, possède un type, un message et un état de lecture.
- ✓ **Rôle** : Définit les droits d'accès d'un utilisateur. Les rôles (ex. : client, admin, vendeur) permettent de restreindre ou d'autoriser certaines actions dans le système.

Les relations entre les classes sont exprimées à travers des associations (ex : un utilisateur passe plusieurs commandes), des cardinalités et parfois des dépendances. Des méthodes métiers essentielles sont également précisées pour certaines classes afin d'illustrer les actions que ces entités peuvent accomplir (comme créer une commande, ajouter un produit, ou envoyer une notification).

✓ Diagramme de séquence

Le diagramme de séquence est utilisé pour représenter la dynamique d'un système, c'est-à-dire les interactions entre les objets et les messages échangés entre eux. Il permet de visualiser les séquences d'actions qui se produisent dans le système, l'ordre dans lequel ces actions se produisent et les conditions de synchronisation. Le diagramme de séquence est utile pour comprendre le comportement du système et pour identifier les méthodes et les messages qui seront implémentés dans le code.

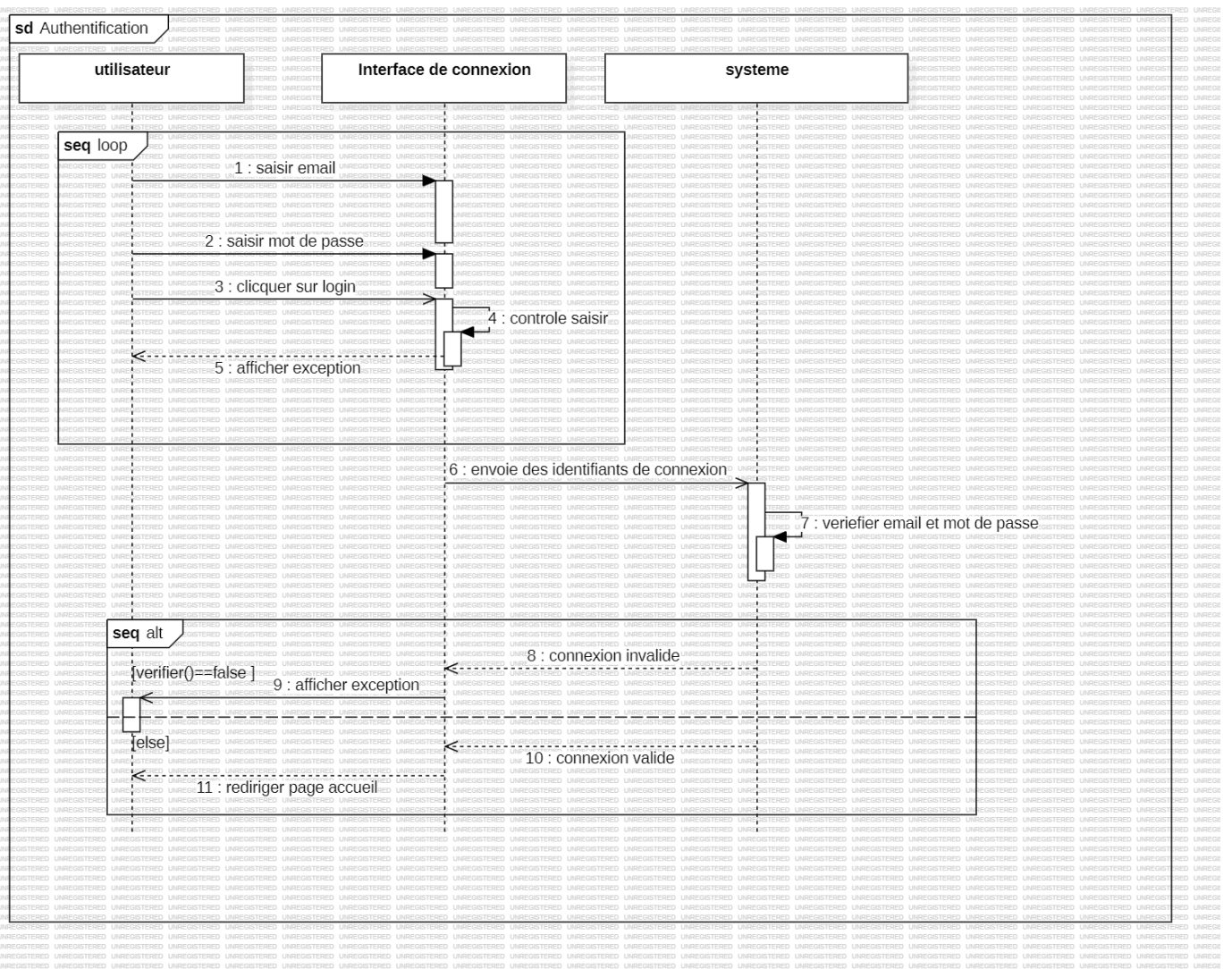


Figure 7: diagramme de séquence d'authentification

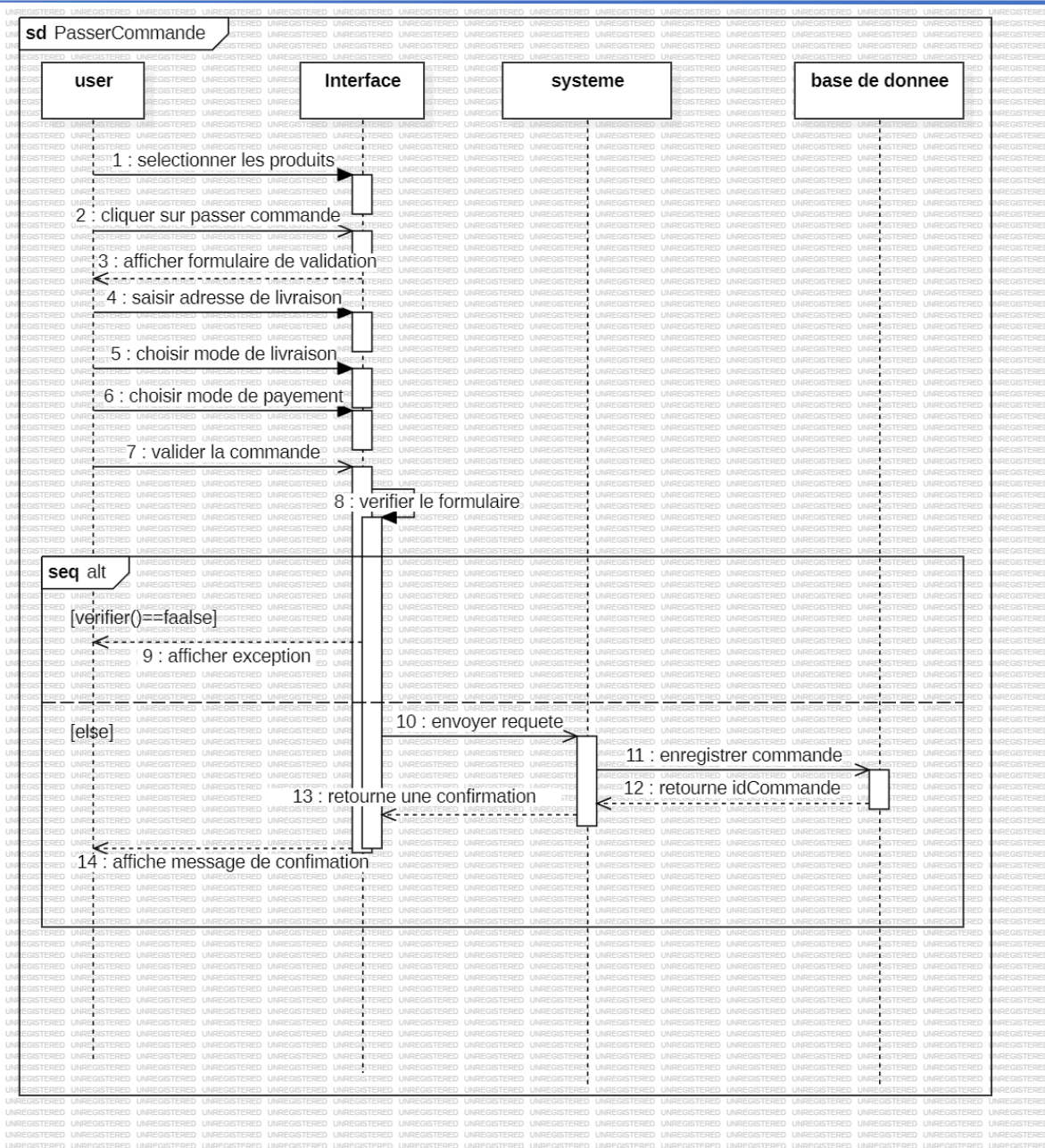


Figure 8: Diagramme de séquence passer commande

✓ Diagramme de déploiement

Afin de mieux visualiser l'architecture physique de notre application Asiel, un **diagramme de déploiement UML** a été réalisé. Celui-ci permet de représenter les différents composants logiciels et les environnements matériels sur lesquels ils sont déployés, ainsi que les interactions entre ces éléments via divers protocoles de communication. Ce diagramme nous permet d'avoir une vision claire de la structure technique du projet et des différentes couches de déploiement.

1 Client Web (Navigateur)

- ✓ Permet à l'administrateur ou aux utilisateurs d'accéder à l'interface Angular via

- ## 2 Summary: Frontend (Netflix)

- ✓ Héberge l'application **Angular** utilisée par l'équipe d'administration pour gérer

- ## 2 Server Backend (Router)

✓ Composant central de l'application. héberge AP RES développée en

- ✓ Le serveur de base de données Railway via JDBC
- ✓ L'interface web Angular
- ✓ L'application mobile Ionic via HTTPS

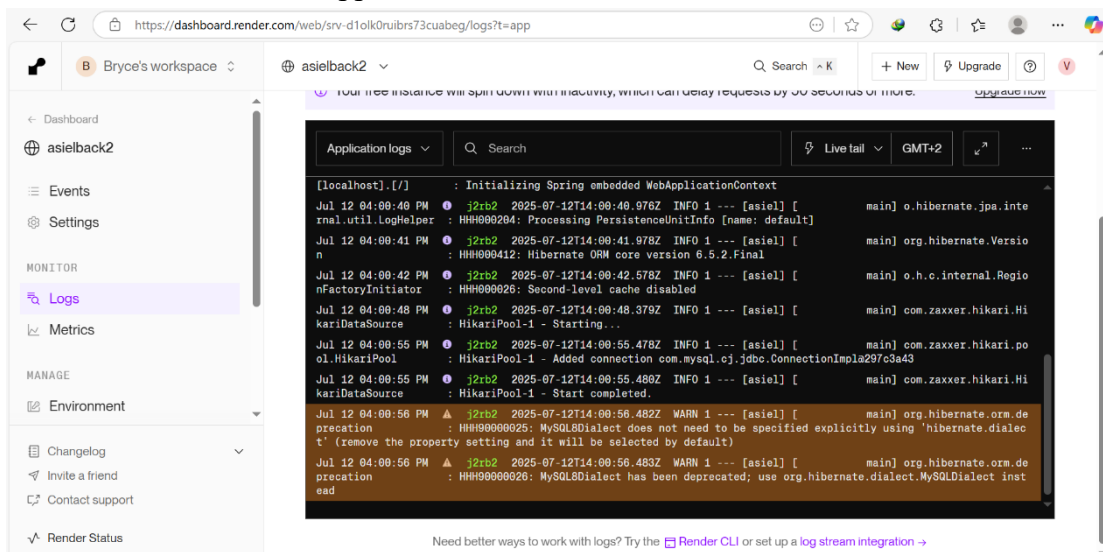


Figure 10: Déploiement application spring Boot sur Render

4. Serveur de base de données (Railway)

- Héberge une **base de données MySQL** contenant les entités principales de l'application (produits, utilisateurs, commandes...).
- Elle est directement liée au backend via une connexion **JDBC sécurisée**.

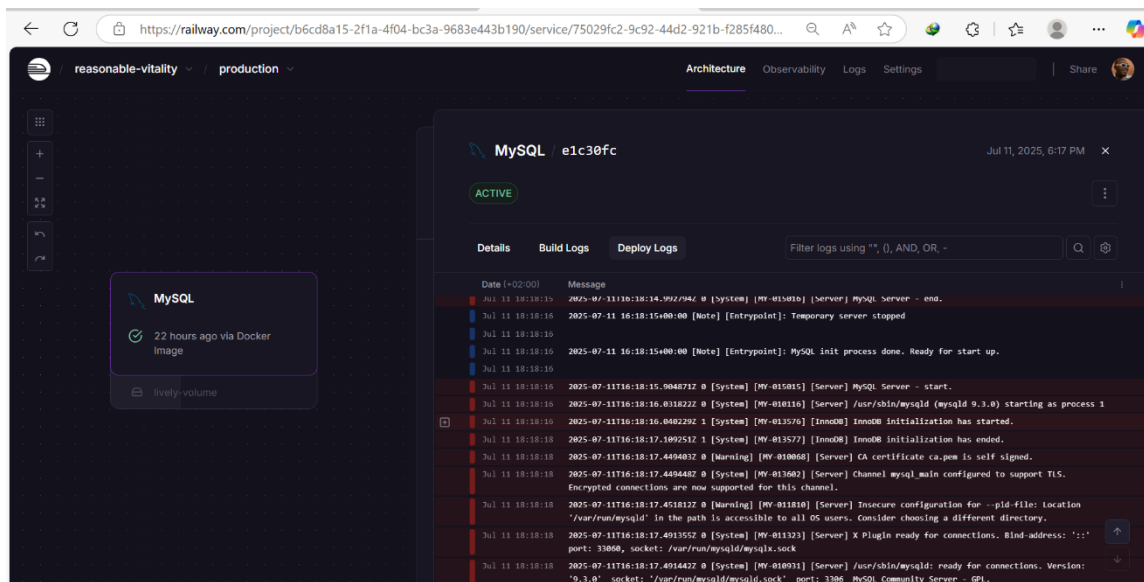


Figure 11: déploiement base de données MYSQL sur Railway

5. Smartphone (Utilisateur)




- ✓ L'application **mobile Flutter** est déployée sur les smartphones.

- ✓ Elle interagit avec le backend via **HTTPS** pour accéder aux fonctionnalités (authentification, commandes, notifications...).

6. Application mobile (Ionic)

- ✓ Déployée localement sur le terminal de l'utilisateur.

IV TECHNOLOGIES UTILISÉES

Outils	Role	Symboles
Angular	Framework d'application web qui fournit un cadre complet pour la création d'application web complexe	
MySQL	MySQL qui est un système de gestion de base de données relationnelle (SGBDR) open-source largement utilisé. Nous l'avons utilisé pour stocker et gérer des données dans un format structuré, ce qui nous a facilité la récupération et la manipulation des données.	
CSS	CSS (Cascading Style Sheets) qui est un langage de feuilles de style utilisé pour définir la présentation visuelle d'une page web. Il nous a permis de définir des styles pour les éléments HTML tels que les polices, les couleurs, les marges, les bordures et les mises en page	

Outils	Role	Symboles
Visual studio	Pour l'environnement de développement intégré	
GitHub	Pour la gestion des versions du code source et le travail en collaboration sur notre projet	
Spring Boot	Qui est un Framework open-source pour le développement d'applications Java a été utile ici dans le développement de certains API (Application Programming Interface) au cours du développement de notre application.	
IntelliJ	Un environnement de développement intégré (IDE) commercial développé par JetBrains. Nous l'avons utilisé donc comme IDE pour développer certains API en Spring Boot.	
Postman	C'est un outil de développement d'API qui facilite la création, le test, la documentation et le partage d'API. C'est lui qui nous a permis de tester certains API que nous avons développée.	

Tableau 7: Technologies utilisées

V RÉSULTAT

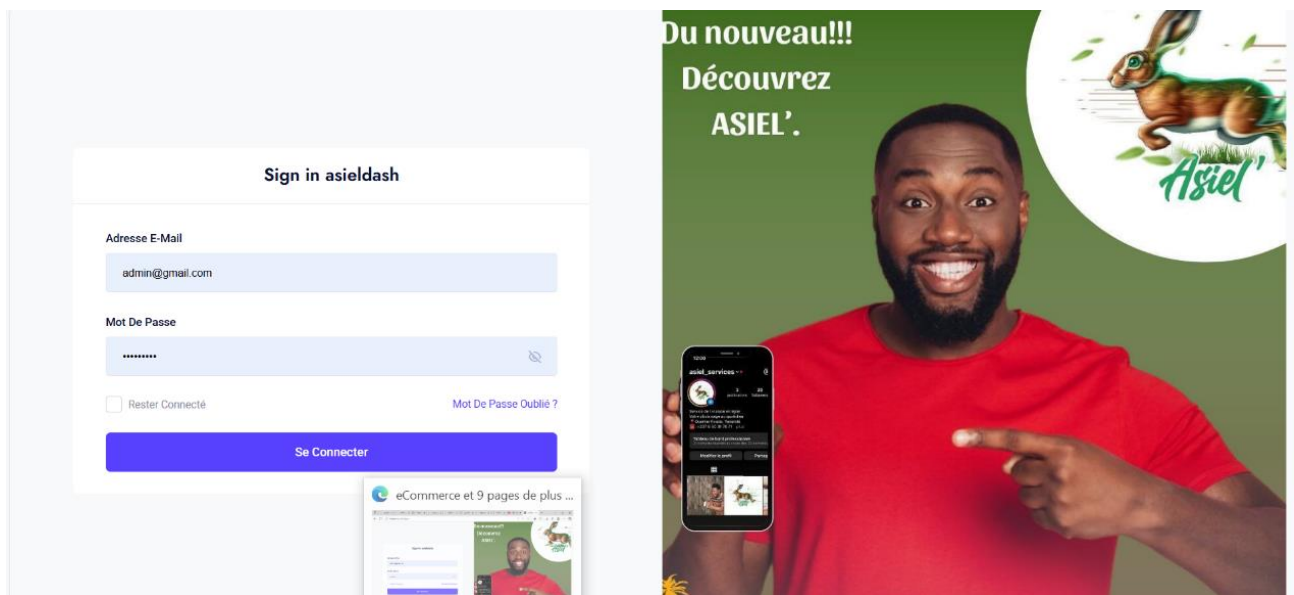


Figure 12: Interface connexion application web

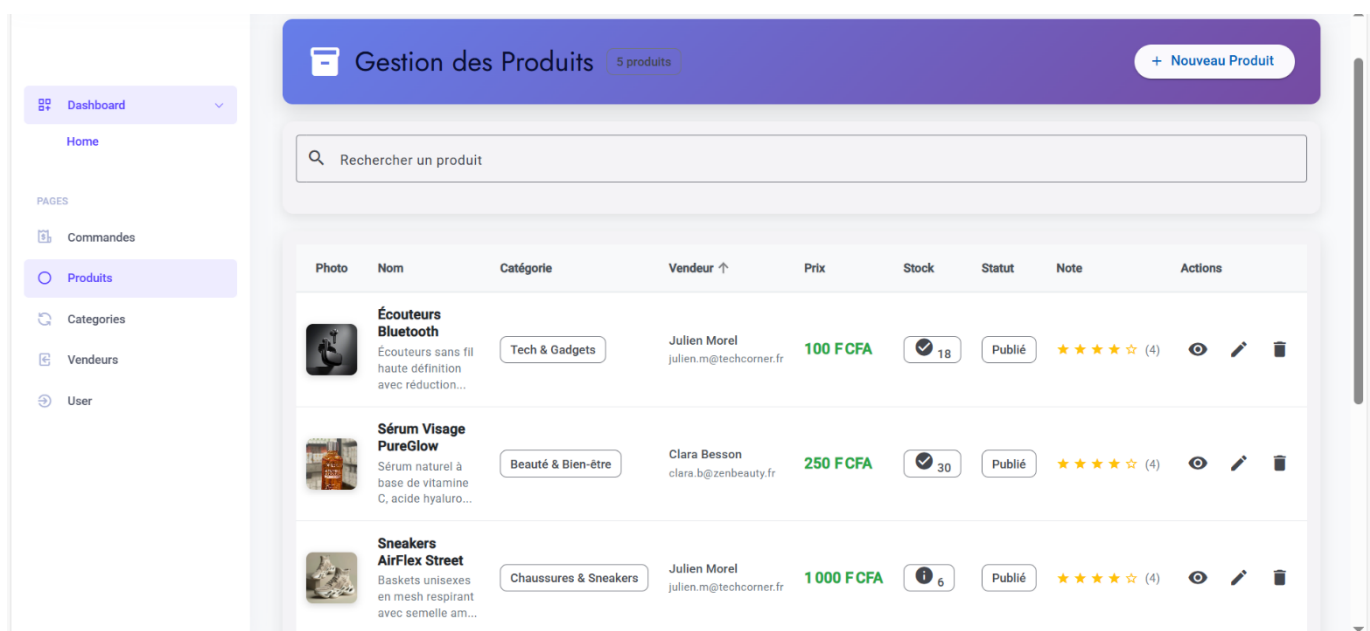


Figure 13: interface d'administration des produits

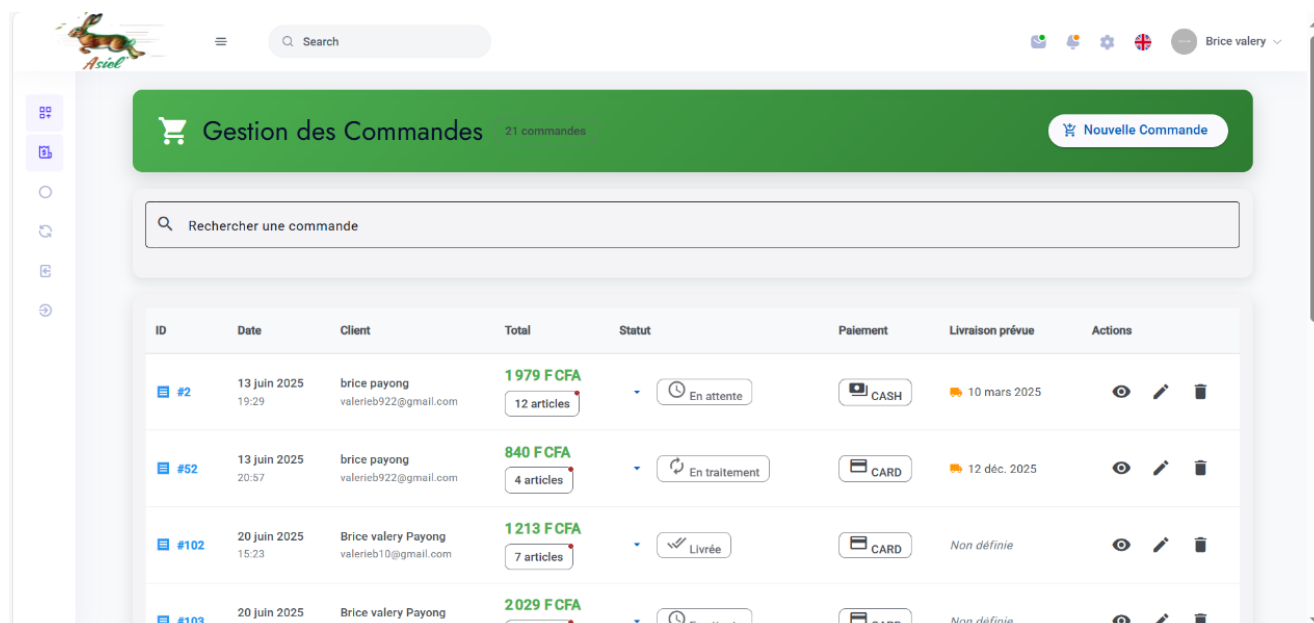


Figure 14: Interface d'administration des commandes

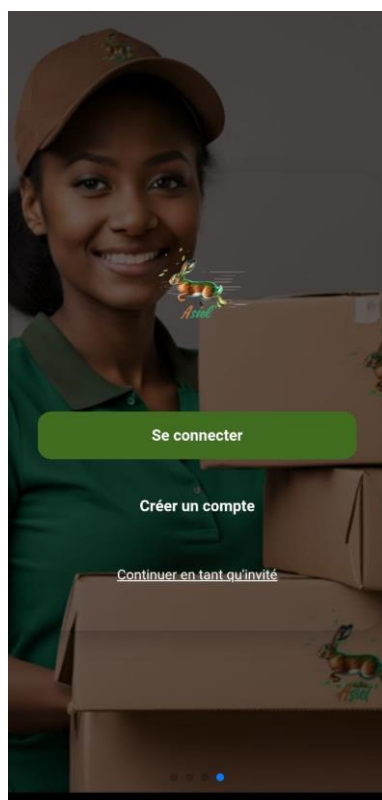


Figure 17: connexion

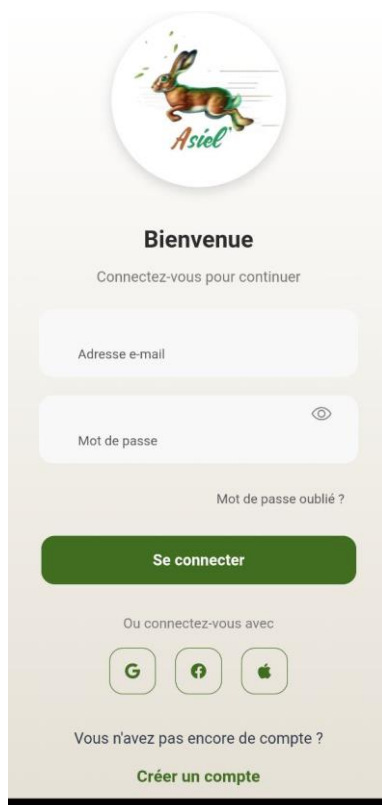


Figure 16: page login

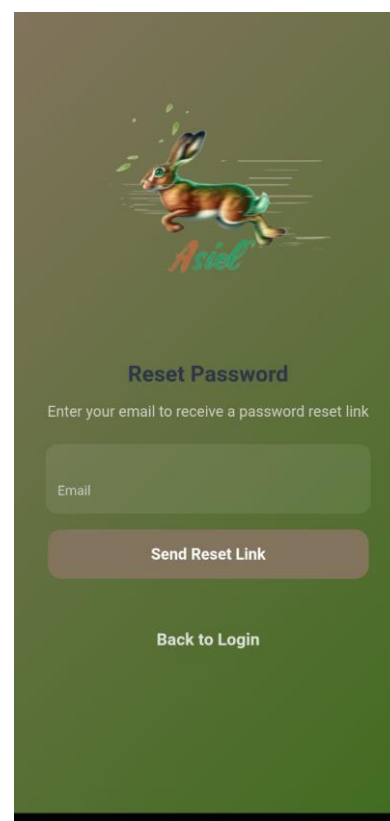


Figure 15: mot de passe oublier

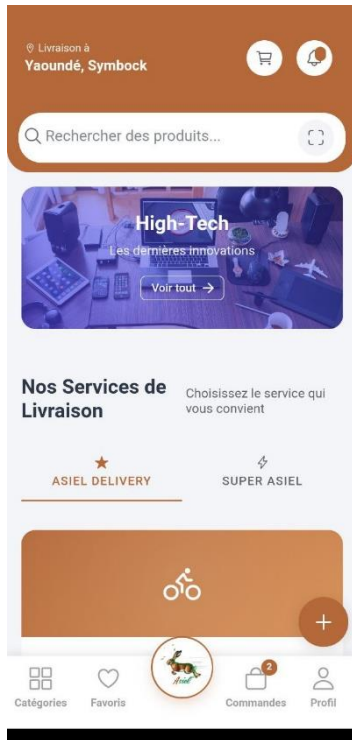


Figure 23: page d'accueil

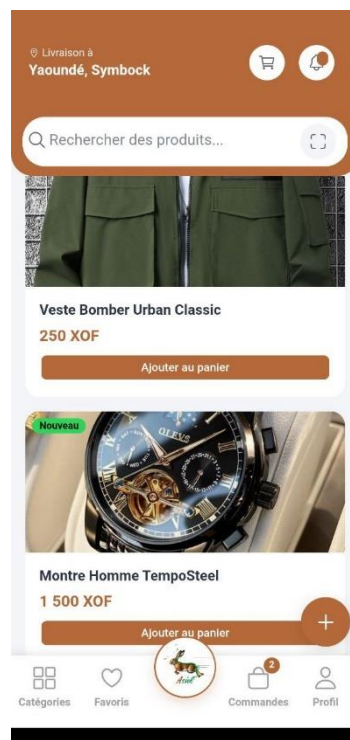


Figure 22: page d'accueil 2

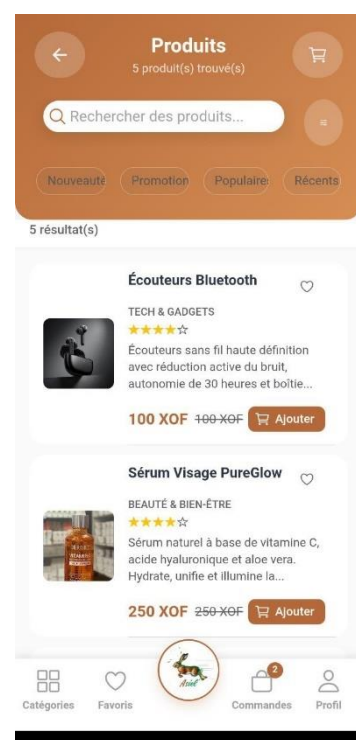


Figure 21: liste des produits



Figure 20: Détails d'un produit

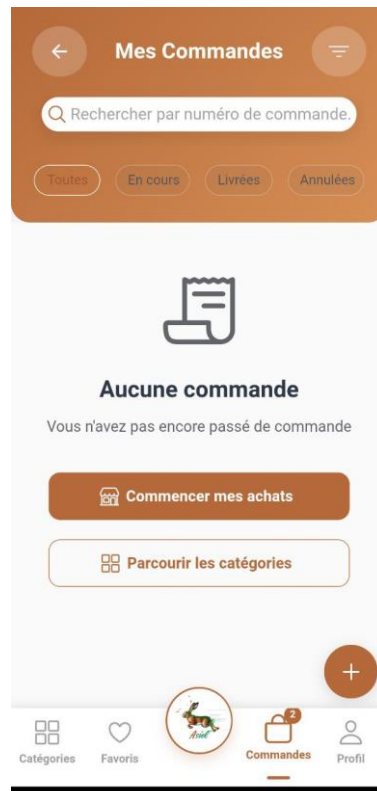


Figure 19: Liste des commandes

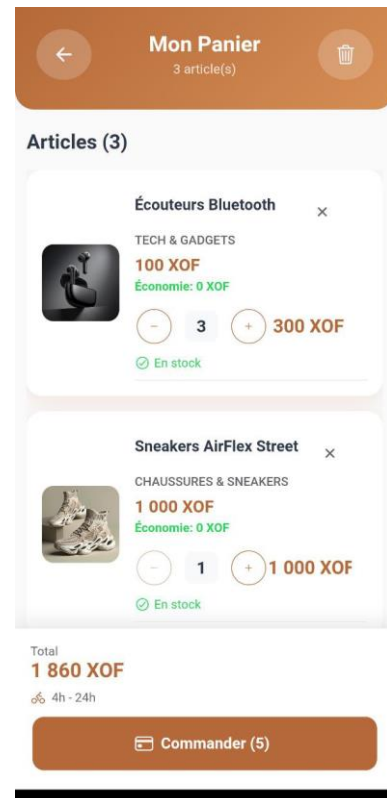


Figure 18: Panier

VI DIFFICULTÉS RENCONTRÉES ET ÉVALUATION PERSONNELLE

a) Difficultés rencontrées

Au cours de notre stage chez **MICKmaq**, dans le cadre du développement de la plateforme **ASIEL**, plusieurs difficultés ont été rencontrées, tant sur le plan technique que sur le plan organisationnel :

- ✓ **Contraintes de temps** : La durée limitée du stage (3 mois) a imposé une gestion rigoureuse du temps pour livrer une version fonctionnelle du module de gestion des commandes. Cela a parfois entraîné des périodes de travail intensif.
- ✓ **Complexité technique du projet** : La mise en place d'un système centralisé, interconnectant une application mobile (Ionic) et une interface Web (Angular) via un backend commun (Spring Boot), nécessitait une bonne maîtrise de plusieurs technologies. La gestion des échanges entre les composants, notamment via API REST et WebSockets, a été particulièrement exigeante.
- ✓ **Manque de documentation initiale** : Le projet étant en phase de démarrage, certaines spécifications fonctionnelles n'étaient pas encore formalisées. Cela a nécessité de nombreuses réunions d'éclaircissement avec les parties prenantes pour bien cerner les attentes et définir les fonctionnalités à implémenter.
- ✓ **Tests et débogage en environnement réel** : Les premiers tests sur les environnements de production simulés ont révélé des dysfonctionnements (mauvaise synchronisation des commandes, erreurs d'authentification), qu'il a fallu corriger rapidement pour garantir une expérience utilisateur fluide.

b) Évaluation personnelle

Malgré les difficultés rencontrées, ce stage a été extrêmement enrichissant sur plusieurs plans :

- ✓ **Compétences techniques renforcées** : Nous avons pu approfondir notre maîtrise de technologies modernes telles que Spring Boot, Angular, Ionic, ainsi que des outils de gestion de version (**Git**) et de base de données (MySQL). Nous avons également appris à intégrer des API, gérer des authentifications sécurisées (JWT) et implémenter des systèmes de notifications en temps réel.
- ✓ **Méthodologie de travail** : Ce projet nous a permis d'expérimenter le **travail en mode agile**, avec des livrables réguliers, des échanges avec les autres développeurs, et l'adoption de bonnes pratiques de développement (commentaires, modularité, tests unitaires).
- ✓ **Développement de l'autonomie et du sens des responsabilités** : L'implication directe dans un projet réel destiné à un usage commercial nous a obligé à faire preuve d'initiative, de rigueur et de professionnalisme. Nous avons appris à prendre des décisions techniques justifiées et à gérer les imprévus avec réactivité.

En somme, ce stage a été une expérience formatrice, tant sur le plan technique que personnel. Il a confirmé notre intérêt pour le développement d'applications web et mobiles, et renforcé notre ambition de nous investir dans des projets à fort impact. Nous en ressortons plus confiants, mieux préparés pour la vie professionnelle, et conscients des efforts à fournir pour continuer à progresser dans le domaine du développement logiciel.

CONCLUSION

En conclusion, la solution que nous avons proposée dans ce chapitre constitue une réponse concrète et efficace au problème que nous avons identifié. À cet effet, nous avons spécifié les besoins fonctionnels et non fonctionnels, défini l'architecture de notre système, décrit la méthode de modélisation utilisée, ainsi que les différents outils mis en œuvre.

CONCLUSION GÉNÉRALE

Ce rapport de stage retrace de manière structurée l'ensemble du travail effectué au sein de l'entreprise **MICKmaq**, dans le cadre du projet **ASIEL** porté par **ALEN SARL**. Durant ces trois mois, l'accent a été mis sur trois axes principaux : la présentation du cadre professionnel et du projet, l'identification d'un problème concret lié à la gestion multicanale des commandes, ainsi que la conception et le développement d'une solution numérique adaptée. Cette solution repose sur une application mobile dédiée aux clients et une interface web d'administration interconnectée.

La méthodologie adoptée s'est basée sur une analyse détaillée des besoins, suivie d'une modélisation fonctionnelle avec UML, puis d'un développement progressif à l'aide de technologies modernes telles que **Spring Boot**, **Angular** et **Ionic**. Ce stage a constitué une expérience formatrice qui nous a permis de renforcer nos compétences techniques tout en collaborant efficacement avec d'autres développeurs dans un environnement professionnel exigeant.

L'un des faits marquants de cette expérience reste la reconnaissance de notre travail par les responsables de **MICKmaq**, qui ont choisi de nous intégrer à part entière à l'équipe du projet **ASIEL** en tant que **développeur Front-End**, en charge de l'évolution de l'application web d'administration. Des perspectives d'amélioration restent ouvertes, notamment l'intégration de systèmes de paiement sécurisés et l'optimisation des performances en cas de forte affluence. Ce stage représente pour nous un véritable tremplin vers le monde professionnel et le début d'une prise de responsabilités concrètes dans le domaine du développement d'applications web et mobiles.

RÉFÉRENCES

a) Bibliographie

- ✓ Institut Saint Jean, (s.d.), *Guide de rédaction du rapport de stage projet*, Document pédagogique interne.
- ✓ Nahsang, M. T., (2024), *Paramétrage des modules de gestion des ventes et des achats dans Odoo ERP*, Rapport de stage, Institut Universitaire Saint Jean, année académique 2024-2025, Licence professionnelle 3ième année.
- ✓ Nkeutio Ndongmo, B., (2024), *Application mobile de prestation de services pour AVI Center : génération de contrats et gestion de logements*, Rapport de stage, Institut Universitaire Saint Jean, année académique 2024-2025, Licence professionnelle 3ième année.

b) Webographie

- ✓ Oracle, (2023), « Spring Boot Documentation », *Spring.io*, <https://docs.spring.io/spring-boot>.
- ✓ Angular Team, (2023), « Angular Documentation », *Angular.io*, <https://angular.io/docs>.
- ✓ Ionic Framework, (2023), « Ionic Docs », *Ionicframework.com*, <https://ionicframework.com/docs>.
- ✓ OpenAI, (2024), « Introduction to GPT Models », *OpenAI.com*, <https://platform.openai.com/docs>.
- ✓ À propos de nous – Institut Universitaire Saint Jean : <https://www.universitesaintjean.org>
- ✓ Angular HTTP guide : <https://angular.io/guide/understanding-communicating-with-http#getting-error-details>
- ✓ Documentation Spring Framework : <https://docs.spring.io/spring-framework/reference/>
- ✓ Tutoriels Spring Boot sur YouTube : https://www.youtube.com/results?search_query=spring+boot+gestion+de+stock
- ✓ Documentation Swagger : <https://swagger.io/docs/>
- ✓ Flickr (ressources médias) : <https://www.flickr.com/>

ANNEXES

Date	Tâche effectuée	Outils utilisés	Avec qui ?
Du 07/04/2025 au 12/04/2025	Étude du cahier des charges, prise en main du projet	Word, Google Docs	M. Hassan, Équipe de développement
Du 12/04/2025 au 17/04/2025	Analyse des technologies utilisées (Spring Boot, Angular, Ionic) + Benchmark	YouTube, Docs officiels	Individuel
Du 17/04/2025 au 20/04/2025	Installation de l'environnement de travail (IDE, Git, MySQL)	VS Code, IntelliJ, GitHub, DBeaver	Individuel
Du 17/04/2025 au 26/04/2025	Modélisation UML : cas d'utilisation, diagrammes de classes et de séquence	StarUML,	Individuel
Du 26/04/2025 au 30/04/2025	Conception du modèle relationnel et rédaction des spécifications techniques	MySQL, DBeaver, Word	Individuel
Du 01/05/2025 au 19/05/2025	Développement Backend : entités, services, API REST, sécurité JWT	Spring Boot, Postman, IntelliJ	Individuel
Du 19/05/2025 au 22/05/2025	Tests unitaires backend + intégration Swagger pour la documentation	Swagger UI, Postman	Individuel
Du 22/05/2025 au 3/06/2025	Développement du Frontend (Angular) – Authentification + modules	Angular, VS Code	Individuel
Du 03/06/2025 au 07/06/2025	Intégration de l'API Flickr pour gestion des images produits	Angular, API Flickr	Individuel
Du 07/06/2025 au 10/06/2025	Création de l'interface admin : tableau de bord, statistiques	Angular, Chart.js	Individuel
Du 11/06/2025 au 30/06/2025	Développement de l'app mobile (Ionic) : catalogue, panier, commandes	Ionic	Individuel
Du 01/07/2025 au 03/07/2025	Intégration des notifications push (websocket)	Ionic, websocket	Individuel
Du 03/07/2025 au 10/07/2025	Intégration Web ↔ Backend ↔ Mobile + tests fonctionnels	Postman, navigateur, VS Code	Individuel

Du 05/07/2025 au 10/07/2025	Correction des bugs + validation avec le tuteur	Git	M. Hassan
Du 07/04/2025 au 10/07/2025	Rédaction de la documentation technique et du rapport final	Word, Google Docs	Individuel
Du 10/07/2025 au 16/07/2025	Finalisation du support	PowerPoint, Word	Individuel, M. Hassan, M. Tapigue

Tableau 8: Tableau des différentes tâches effectuées durant le stage

Raison sociale	Mickmaq SARL
Année de création	Cameroun en 2019
Régime d'imposition	Simplifié
Siège social	Yaoundé, derrière le lieu-dit « Rond-point express »
Activités principales	E-Commerce
Contacts	Téléphone : 6 80 051 785
Site Internet	https://www.mickmaq.com
Forme juridique	SARL
Nom du directeur	M. Fomenky Ache

Tableau 9: Fiche signalétique



Figure 24: Localisation Mickmaq

TABLES DES MATIÈRES

DEDICACE	1
REMERCIEMENTS	2
SOMMAIRE	3
LISTE DES TABLEAUX	4
LISTE DES FIGURES	5
LISTE DES ABREVIATIONS	6
AVANT-PROPOS	7
INTRODUCTION GÉNÉRALE	8
CHAPITRE I : PRÉSENTATION DE L'ENTREPRISE ET SON ENVIRONNEMENT	9
INTRODUCTION	10
I PRÉSENTATION DE MICKMAQ	10
a) <i>Contexte</i>	10
b) <i>Missions</i>	10
c) <i>Vision</i>	11
d) <i>Activités</i>	11
e) <i>Organisation administrative</i>	11
f) <i>Organigramme de Mickmaq</i>	12
II DÉROULEMENT DU STAGE	14
a) <i>Accueil et intégration</i>	14
b) <i>Emploie de temps</i>	14
c) <i>Planification du travail</i>	14
d) <i>Taches effectuées</i>	15
CONCLUSION	16
CHAPITRE II : PRÉSENTATION DU PROBLÈME	17
INTRODUCTION	18
PRÉSENTATION DU SERVICE ASIEL	18
I IDENTIFICATION ET ORIGINE DU PROBLÈME	19
II SOLUTIONS EXISTANTES	20
CONCLUSION	21
CHAPITRE III : SOLUTION ET IMPLÉMENTATION	22

INTRODUCTION	23
I DÉFINITION DU BESOIN ET SPÉCIFICATIONS DU PROJET ASIEL.....	23
a) <i>Mission</i>	23
b) <i>Objectifs</i>	23
c) <i>Portée</i>	23
d) <i>Besoins Fonctionnels</i>	23
e) <i>Besoins Non Fonctionnels</i>	25
f) <i>Caractéristiques des utilisateurs</i>	26
II ARCHITECTURE DE LA SOLUTION	27
III MÉTHODE ET LANGAGE DE MODÉLISATION	29
a) <i>Méthode de modélisation</i>	29
b) <i>Langage de modélisation</i>	29
c) <i>Les diagrammes</i>	30
IV TECHNOLOGIES UTILISÉES.....	40
V RÉSULTAT	42
VI DIFFICULTÉS RENCONTRÉES ET ÉVALUATION PERSONNELLE	45
a) <i>Difficultés rencontrées</i>	45
b) <i>Évaluation personnelle</i>	45
CONCLUSION	46
CONCLUSION GÉNÉRALE.....	47
RÉFÉRENCES BIBLIOGRAPHIQUES	IX
ANNEXES.....	X
TABLES DES MATIÈRES	XIII