

REPUBLIQUE ALGÉRIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE A-MIRA DE BEJAIA
FACULTÉ DES SCIENCES EXACTES DEPARTEMENT INFORMATIQUE



Thème

Conception et développement d'une application e-commerce

ENCADRÉ PAR

Dr **Farah** Zoubeyr

PRÉSENTÉ PAR

Mr **BRAHIMI** Rayan

Mr **MECHEKOUR** Billal

Mr **MESSAOUDENE** Said

Mr **MERSEL** Lyes

Remerciements

Tout d'abord, nous exprimons notre gratitude à Dieu Tout-Puissant pour nous avoir accordé le courage, la force, la patience et la santé nécessaires pour mener à bien ce modeste travail.

Nous tenons à exprimer notre plus profonde et sincère gratitude envers Dr **Farah Zoubeyr** pour leur dévouement exceptionnel et leur contribution inestimable.

Votre engagement indéfectible, votre professionnalisme remarquable et votre bienveillance constante ont été des piliers essentiels à la réussite de notre projet. Chaque étape a été marquée par votre expertise et votre disponibilité sans faille.

Cette collaboration a été pour nous une expérience enrichissante, et nous espérons vivement avoir l'opportunité de travailler à nouveau ensemble dans le futur. Nous tenons également à souligner l'apport intellectuel et méthodologique dans la réalisation de ce travail.

L'équipe tient à exprimer sa reconnaissance envers tous ceux qui ont soutenu et participé à cette réalisation.

Cordialement, L'équipe

TABLE DES MATIÈRES

CHAPITRE 1 : Introduction	1
1.1 Contexte général	1
1.2 Objectifs généraux du projet	1
1.3 Méthodologie suivie	1
1.3.1 Principes fondamentaux	1,2
CHAPITRE 2 : Présentation générale du projet	3
2.1 Description de l'application MEGA SHOP	3
2.2 Acteurs du système	3
2.3 Fonctionnalités principales	3
2.4 Présentation du projet	4
2.5 Originalité du projet	4
CHAPITRE 3 : Étude de l'existant	5
3.1 Analyse de quelques plateformes similaires	5
3.1.1 Points communs	5
3.1.2 Différences et avantages de MEGA SHOP	8,9
CHAPITRE 4 : Cahier des charges	6
4.1 Objectifs fonctionnels	6
4.1.1 Pour les visiteurs	6
4.1.2 Pour les clients	6
4.1.3 Pour les vendeurs	6
4.1.4 Pour les administrateurs	6
4.2 Objectifs non fonctionnels	6
4.2.1 Sécurité	6,7
4.2.2 Performance	7
4.2.3 Ergonomie et Accessibilité	7
4.3 Contraintes techniques	7
4.3.1 Environnement de développement	7
4.3.2 Infrastructure	7
4.3.3 Sécurité	7
4.3.4 Performance	7
4.3.5 Compatibilité	8
4.4 Cas d'utilisation principaux	8
4.4.1 Processus d'achat complet	8
4.4.2 Gestion de produit par un vendeur	8

TABLE DES MATIÈRES

4.4.3 Modération par l'administrateur	8
4.4.4 Évolution du statut utilisateur	8
CHAPITRE 5 : Analyse et conception	
5.1 Introduction	9
5.2 Diagramme de cas d'utilisation	9
5.3 Diagrammes de séquence	10
5.3.1 Diagramme de séquence pour le cas d'utilisation "S'authentifier"	10
5.3.2 Diagramme de séquence pour le cas d'utilisation "Commenter un produit"	11
5.3.3 Diagramme de séquence pour le cas d'utilisation "Créer un compte client"	12
5.3.4 Diagramme de séquence pour le cas d'utilisation "Gérer les produits"	13
5.4 Diagramme de déploiement	14
5.5 Diagramme d'états-transitions pour le processus de paiement	15
5.6 Diagramme de contexte	
5.7 Diagramme de classe	15
5.8 Modèle relationnel de données	16
5.9 Architecture de l'application	17,18
5.9.1 Couche Présentation (Frontend)	18
5.9.2 Couche API (Backend)	18
5.9.3 Couche Service	18
5.9.4 Couche Accès aux données	18
5.9.5 Base de données	18
5.10 Conclusion	18
CHAPITRE 6 : Réalisation de l'application	18,19
6.1 Livrables	19
6.2 Interfaces principales	20
6.2.1 Interfaces principales de l'application	20
6.2.2 Interfaces principales de client	20,21
6.2.3 Interfaces principales de l'administrateur	21
6.2.4 Interfaces principales de vendeur	22,23
CHAPITRE 7 : Conclusion et perspectives	
7.1 Bilan du projet	24
7.2 Améliorations futures possibles	25

TABLE DES FIGURES

FIG.1 - Diagramme de cas d'utilisation	9
FIG.2 - Diagramme de séquence pour le cas d'utilisation "S'authentifier"	10
FIG.3 - Diagramme de séquence pour le cas d'utilisation "Commenter un produit"	11
Figure 4 – Diagramme de séquence pour le cas d'utilisation "Créer un compte client"	12
Figure 5 – Diagramme de séquence pour le cas d'utilisation "Gérer les produits"	13
Figure 6 – Diagramme de déploiement	14
Figure 7 – Diagramme états-transitions pour le processus de paiement	14
Figure 8 – Diagramme de contexte	15
Figure 9 – Diagramme de classe	15
Figure 10 – Interface principale de l'application version pc et mobile	16
Figure 11 – Interface de produits version pc et mobile	20
Figure 12 – Interface de panier version pc et mobile	20
Figure 13 – Interface paramètres version pc et mobile	21
Figure 14 – Interface commande version pc et mobile	21
Figure 15 – Interface dashboard version pc	21
Figure 16 – Interface gestion d'utilisateurs version pc et mobile	22
Figure 17 – Interface ma boutique version pc et mobile	22
Figure 18 – Interface mes ventes version pc et mobile	22
Figure 19 – Interface notifications version pc et mobile	23
Figure 20 – Interface paramètres version pc et mobile	23
Figure 21 – Interface dashboard version pc et mobile	23
Figure 22 – Interface notifications version pc et mobile	24
Figure 23 – Interface mes commandes version pc et mobile	24

CHAPITRE 1 : INTRODUCTION

1.1 Contexte général

Avec l'évolution rapide des technologies web et la généralisation de l'accès à Internet, le commerce en ligne s'est imposé comme un mode d'achat incontournable. Il permet aux consommateurs de faire leurs achats à tout moment, sans contrainte géographique, et aux vendeurs de proposer leurs produits à une large clientèle. Dans ce contexte, les plateformes e-commerce jouent un rôle clé dans la digitalisation des échanges commerciaux.

Dans le cadre de notre formation universitaire, il nous a été demandé de développer une application web de gestion e-commerce répondant à un cahier des charges . Ce projet nous a permis de mobiliser nos compétences en développement logiciel, en conception de base de données, en modélisation UML et en gestion d'une application web full-stack permettant aux utilisateurs de vendre, acheter et gérer des produits dans un environnement numérique sécurisé. Conçue dans un cadre pédagogique, cette application intègre l'ensemble des fonctionnalités essentielles d'une plateforme e-commerce moderne, incluant une simulation du processus de paiement.

1.2 Objectifs généraux du projet

1. Créer une plateforme intuitive facilitant les transactions commerciales en ligne.
2. Mettre en place un système de gestion efficace pour les produits, les commandes et les comptes utilisateurs.
3. Intégrer une simulation de paiement réaliste conforme aux exigences pédagogiques.
4. Offrir une expérience utilisateur différenciée selon le type d'acteur (visiteur, client, vendeur, administrateur).
5. Garantir la sécurité des données et des transactions

1.3 Méthodologie suivie

1.3.1 Principes fondamentaux

Pour ce projet, nous avons adopté une approche de développement agile, plus précisément la méthodologie DSDM (Dynamic Systems Development Method), qui nous a permis de :

- Suivre un développement itératif et incrémental adapté à notre contexte académique
- Prioriser nos efforts selon le principe **MoSCoW** (Must have, Should have, Could have, Won't have).
- Impliquer activement les utilisateurs finaux dans le processus de développement.
- Assurer la livraison dans les délais grâce à des timelines bien définies.
- Maintenir l'accent sur la qualité et l'adéquation aux besoins métier.

Cette méthodologie a été choisie pour sa rigueur et son pragmatisme, particulièrement adaptés aux projets avec des contraintes de temps fixes comme le nôtre. Le cycle de développement a ainsi suivi les phases principales de DSDM :

1. Pré-projet et étude de faisabilité
2. Étude fonctionnelle et conception
3. Développement itératif et incrémental en timeboxes
4. Mise en œuvre et feedback continu
5. Post-projet et évaluation

Nous avons également mis en place des outils de versionnage (Git/GitHub) et de communication (Discord) pour faciliter la collaboration entre les membres de l'équipe, tout en respectant les principes fondamentaux de DSDM comme la communication active et la livraison dans les délais.

CHAPITRE 2 : PRÉSENTATION GÉNÉRALE DU PROJET

2.1 Description de l'application MEGA SHOP

MEGA SHOP est une plateforme e-commerce complète qui permet à différents types d'utilisateurs d'interagir dans un écosystème commercial unifié. L'application offre une interface intuitive pour parcourir, rechercher et acheter des produits, ainsi que des fonctionnalités avancées pour les vendeurs et les administrateurs.

La plateforme est construite avec des technologies modernes (Next.js, TypeScript, Prisma) qui garantissent performance et évolutivité. L'architecture adoptée permet une séparation claire entre la présentation, la logique métier et l'accès aux données.

2.2 Acteurs du système

L'application MEGA SHOP distingue quatre types d'acteurs principaux:

- 1. Visiteur:** Utilisateur non authentifié pouvant consulter le catalogue, rechercher des produits et gérer un panier temporaire.
- 2. Client:** Utilisateur authentifié pouvant effectuer des achats, évaluer des produits et gérer son compte personnel.
- 3. Vendeur:** Utilisateur disposant de priviléges pour mettre en vente des produits, gérer son inventaire et interagir avec les évaluations clients.
- 4. Administrateur :** utilisateur disposant de droits étendus pour gérer l'ensemble de la plateforme, superviser les utilisateurs et modérer le contenu.

2.3 Fonctionnalités principales

- Gestion complète du catalogue de produits
- Système de recherche avancée
- Panier d'achat et processus de commande
- Simulation de paiement sécurisé
- Système d'évaluation et d'avis
- Gestion des comptes utilisateurs
- Système de notifications
- Tableau de bord administrateur et vendeur
gestion des produits
- Gestion des produits de la boutique

2.4 Présentation du projet

Dans un monde où le commerce en ligne ne cesse de croître, notre projet, intitulé MEGA SHOP, s'inscrit dans une démarche d'innovation et de praticité. Il s'agit d'une application web de gestion e-commerce qui vise à simplifier les achats en ligne tout en offrant aux utilisateurs la possibilité de vendre leurs propres produits sur la plateforme.

MEGA SHOP se distingue des plateformes classiques par sa double fonctionnalité : elle permet aux visiteurs d'explorer un large catalogue de produits, de rechercher, commenter, ajouter des articles à leur panier, et de finaliser leurs commandes. Une fois inscrits, les utilisateurs peuvent également gérer leurs comptes, suivre leurs commandes, publier leurs propres produits à vendre, et recevoir des notifications.

L'application prévoit également une interface d'administration complète, accessible après authentification, permettant la gestion des produits (du site et des vendeurs), des comptes utilisateurs, des commandes, ainsi que la simulation des paiements et le suivi des gains liés aux ventes.

La conception de MEGA SHOP a été menée en suivant une démarche méthodologique rigoureuse, depuis l'analyse des besoins jusqu'au développement technique, en utilisant des outils modernes adaptés au développement web. Ce projet nous a permis de mettre en pratique nos connaissances en programmation full-stack, en gestion de base de données et en modélisation logicielle, tout en répondant à un besoin concret du commerce numérique actuel.

2.5 Originalité du projet

Notre application MEGA SHOP se distingue par plusieurs caractéristiques innovantes:

- 1. Architecture hybride de rendu:** Combinaison optimale de Server Side Rendering (SSR) et Client Side Rendering (CSR) pour maximiser les performances tout en garantissant une expérience utilisateur fluide.
- 2. Système de transition de rôle:** Un parcours utilisateur permettant d'évoluer de simple client à vendeur via un processus de validation administré, créant ainsi un écosystème dynamique et auto-alimenté.
- 3. Chatbot d'assistance intégré:** Solution d'aide contextuelle intelligente permettant de guider les utilisateurs dans leurs interactions avec la plateforme.
- 4. Système de signalement communautaire:** Mécanisme permettant aux utilisateurs de participer à la modération de la plateforme en signalant les contenus inappropriés.
- 5. Simulation pédagogique de paiement:** Reproduction fidèle du processus de paiement en ligne sans les risques associés aux transactions réelles, tout en préparant l'infrastructure pour une éventuelle intégration avec des services comme Stripe ou Slick Pay.

CHAPITRE 3 : ÉTUDE DE L'EXISTANT

3.1 Analyse de quelques plateformes similaires

Amazon

- Forces: Vaste catalogue, système de recommandation avancé, logistique intégrée, programme de fidélité (Prime)
- Faiblesses: Interface parfois surchargée, problèmes de contrefaçon, complexité pour les nouveaux vendeurs

Jumia

- Forces: Adapté aux marchés africains, options de paiement locales, forte présence mobile
- Faiblesses: Délais de livraison variables, support client parfois critiqué

Alibaba/AliExpress

- Forces: Prix compétitifs, grande variété de produits, plateforme B2B et B2C
- Faiblesses: Qualité variable des produits, longs délais de livraison, barrière linguistique

Points communs et différences avec MEGA SHOP

3.1.1 Points communs

- **Structure multi vendeur:** comme Amazon et AliExpress, MEGA SHOP permet à plusieurs vendeurs de proposer leurs produits.
- **Système d'évaluation:** Tous les sites majeurs incluent un système d'avis et de notation, également présent dans MEGA SHOP.
- **Recherche et filtrage:** Fonctionnalités essentielles présentes sur toutes les plateformes.
- **Gestion de panier:** Processus d'achat en plusieurs étapes similaires

3.1.2 Différences et avantages de MEGA SHOP

Orientation pédagogique: Contrairement aux plateformes commerciales, MEGA SHOP intègre une simulation de paiement pour des fins éducatives.

Simplicité d'utilisation: Interface épurée et intuitive comparée à l'interface parfois surchargée d'Amazon.

Transition client-vendeur facilitée: Processus simplifié pour devenir vendeur, contrairement aux plateformes comme Amazon qui séparent distinctement ces rôles.

Modération communautaire: Système de signalement plus accessible que sur la plupart des plateformes existantes.

Spécialisation vestimentaire: Contrairement aux marketplaces généralistes, MEGA SHOP se concentre exclusivement sur le secteur de l'habillement, permettant une organisation optimisée des catégories et filtres (tailles, matières, styles), ainsi qu'une expérience d'achat plus cohérente et adaptée aux spécificités de ce marché.

CHAPITRE 4 : CAHIER DES CHARGES

4.1 Objectifs fonctionnels

4.1.1 Pour les visiteurs

- Consulter le catalogue de produits
- Rechercher des produits par divers critères
- Consulter les évaluations des produits
- Gérer un panier d'achat temporaire
- Créer un compte client

4.1.2 Pour les clients

- Toutes les fonctionnalités des visiteurs
- Passer et suivre des commandes
- Évaluer les produits achetés
- Signaler des produits inappropriés
- Gérer les notifications et le compte personnel

4.1.3 Pour les vendeurs

- Toutes les fonctionnalités des clients
- Gérer un catalogue de produits (ajout, modification, suppression)
- Consulter les statistiques de vente et les gains (tableau de bord)

4.1.4 Pour les administrateurs

- Gérer l'ensemble du catalogue
- Valider, suivre et annuler les commandes
- gérer les comptes utilisateurs
- Modérer les avis et traiter les signalements
- Consulter les statistiques de vente et les gains (tableau de bord)

4.2 Objectifs non fonctionnels

4.2.1 Sécurité

- Chiffrement des mots de passe (BCrypt)
- Gestion des sessions par JWT avec rafraîchissement automatique
- Sécurisation des routes API
- Protection contre les attaques courantes

- Limitation des tentatives de connexion
- Sécurité des images uploadées

4.2.2 Performance

- Optimisation des images
- Pagination des listes de produits
- Stratégie de rendu hybride (SSR/CSR)

4.2.3 Ergonomie et Accessibilité

- Interface responsive et compatible multi-navigateurs
- Accessibilité pour utilisateurs avec besoins spécifiques

4.3 Contraintes techniques

4.3.1 Environnement de développement:

- Utilisation obligatoire de TypeScript comme langage de programmation
- Adoption d'un framework full stack (Next.js) pour une cohérence technique
- Respect des bonnes pratiques de développement (typage strict, tests unitaires)

4.3.2 Infrastructure:

- Déploiement sur Vercel pour le frontend et l'API
- Base de données Postgres héberger chez Supabase,
- Limitations de stockage imposées par les formules gratuites.

4.3.3 Sécurité:

- Conformité avec les standards OWASP pour la sécurité web
- Implémentation du JWT avec des contraintes de temps d'expiration
- Stockage sécurisé des informations sensibles

4.3.4 Performance:

- Temps de chargement initial sous 3 secondes
- Temps de réponse des API sous 500ms
- Optimisation des requêtes vers la base de données

4.3.5 Compatibilité:

- Support des navigateurs modernes uniquement (Chrome, Firefox, Safari, Edge)
- Adaptation aux écrans de taille minimale 320px

3.4 Cas d'utilisation principaux

3.4.1 Processus d'achat complet:

- Un visiteur navigue dans le catalogue
- Il ajoute des produits à son panier
- Il crée un compte ou se connecter
- Il complète le processus de commande
- Il effectue la simulation de paiement
- Il reçoit une confirmation de commande

3.4.2 Gestion de produit par un vendeur:

- Un client demande à devenir vendeur
- L'administrateur approuve la demande
- Le vendeur ajoute un nouveau produit
- Il configure les détails et le prix
- Il publie le produit sur la plateforme
- Il suit les ventes et répond aux avis

3.4.3 Modération par l'administrateur:

- Un client signale un produit inapproprié
- L'administrateur reçoit le signalement
- Il examine le produit concerné
- Il prend une décision (suppression, avertissement)
- Il notifie le vendeur

3.4.4 Évolution du statut utilisateur:

- Un visiteur crée un compte client
- Le client effectue plusieurs achats
- Il demande le statut de vendeur
- Le client obtient les priviléges vendeur
- Il peut désormais vendre tout en restant client

CHAPITRE 5 : ANALYSE ET CONCEPTION

Diagrammes UML

5.1 Introduction

La phase d'analyse et de conception de l'application MEGA SHOP a été menée en respectant rigoureusement les principes de la méthode DSDM (Dynamic Systems Development Method) adoptée pour ce projet. Conformément à cette approche Agile, nous avons privilégié une conception itérative où les modèles et diagrammes ont évolué progressivement au fil des temps.

5.2 Diagramme de cas d'utilisation

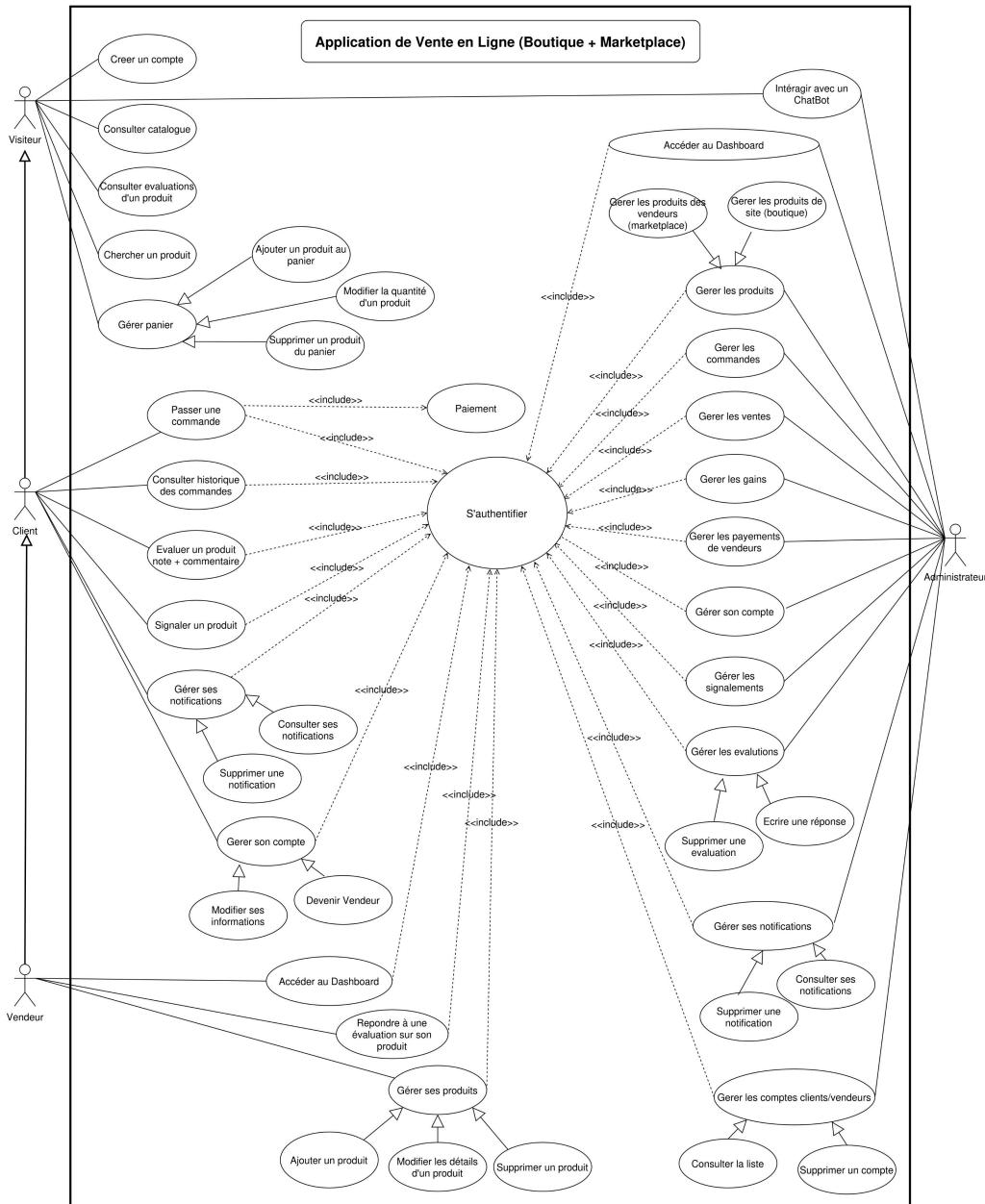


FIG.1 - Diagramme de cas d'utilisation

5.3 Diagramme de séquence

5.3.1 Diagramme de séquence pour le cas d'utilisation "S'authentifier"

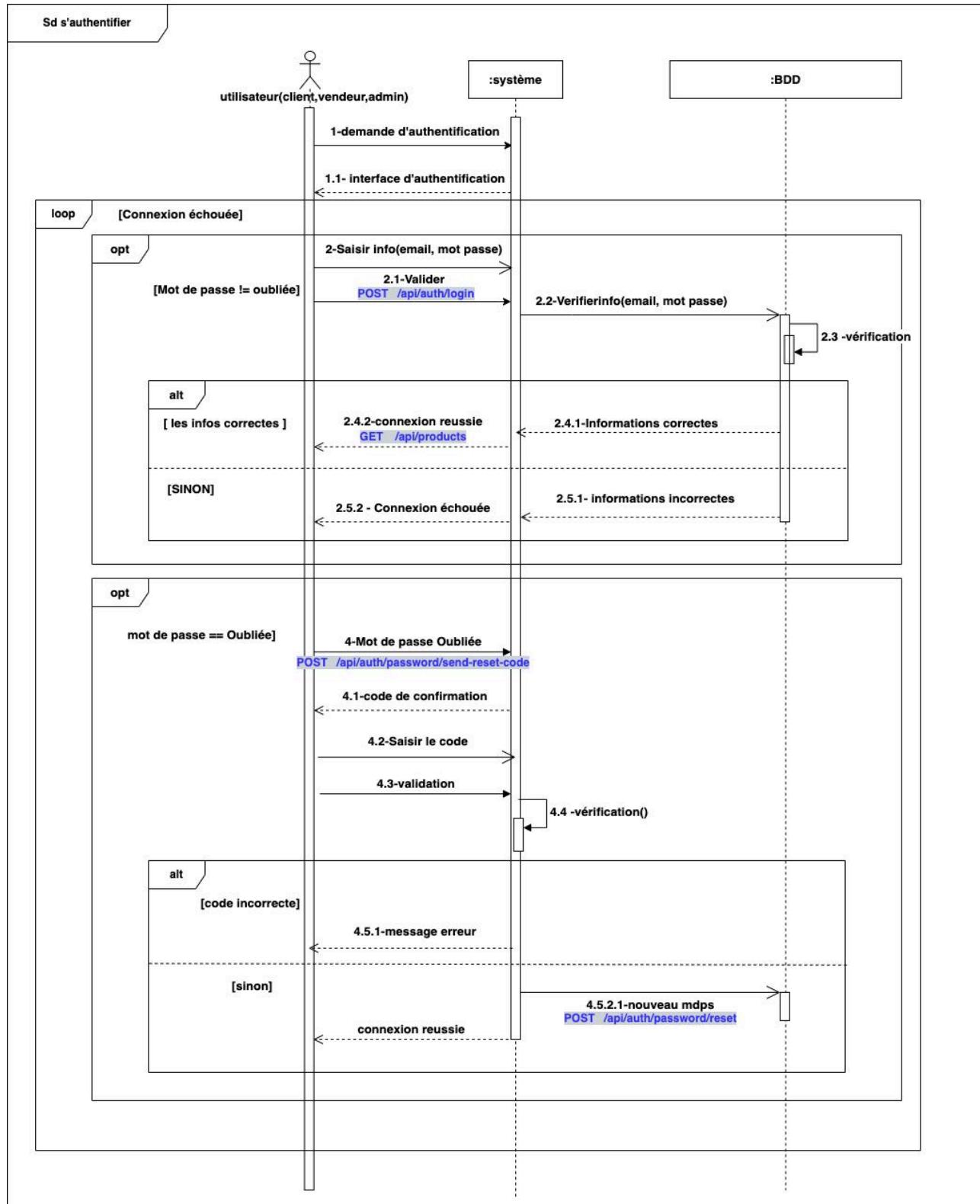


FIG.2 - Diagramme de séquence pour le cas d'utilisation "S'authentifier"

5.3.2 Diagramme de séquence pour le cas d'utilisation "commenter un produit"

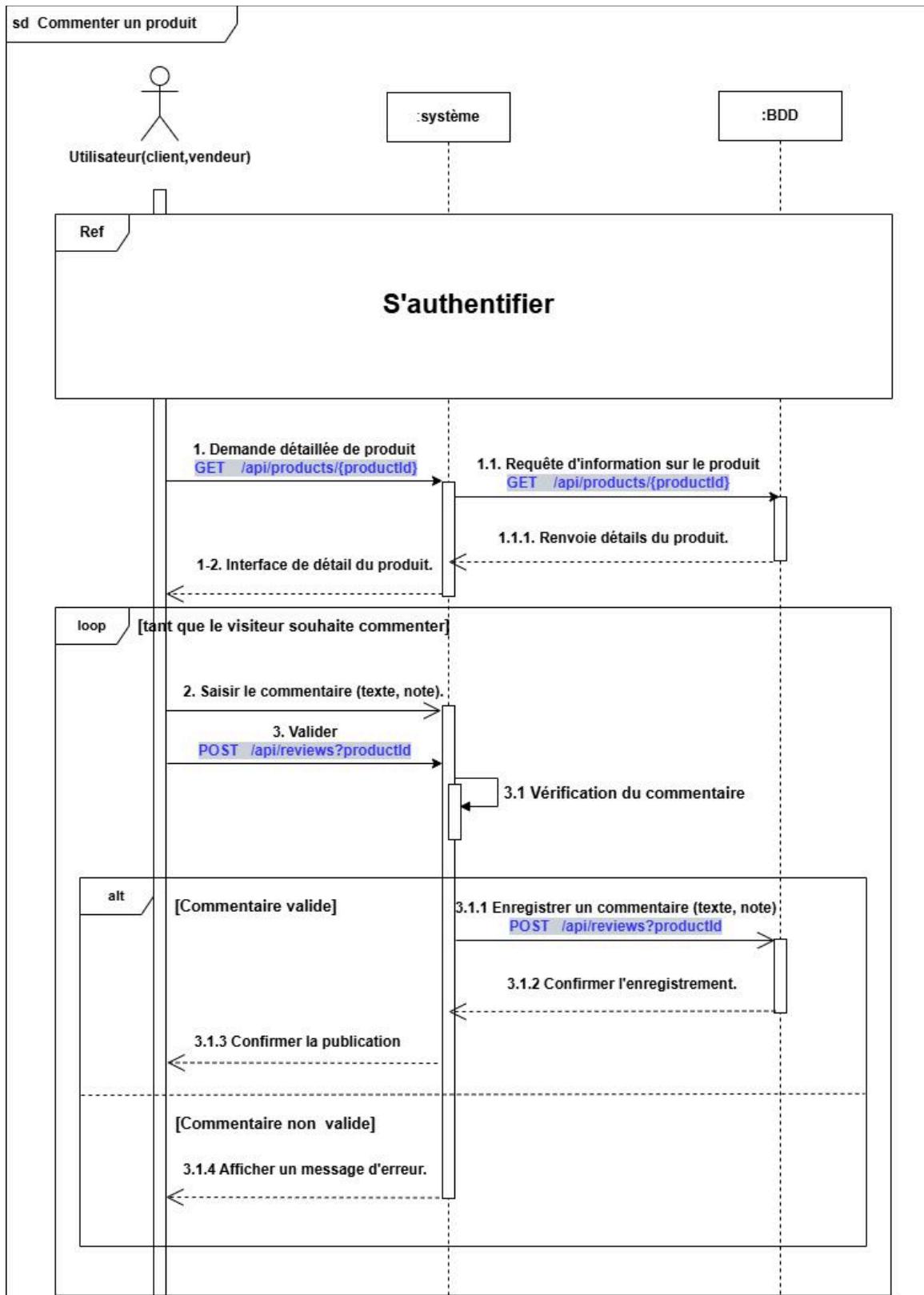


FIG.3 - Diagramme de séquence pour le cas d'utilisation "Commenter un produit"

5.3.3 Diagramme de séquence pour le cas d'utilisation “Créer un compte client”

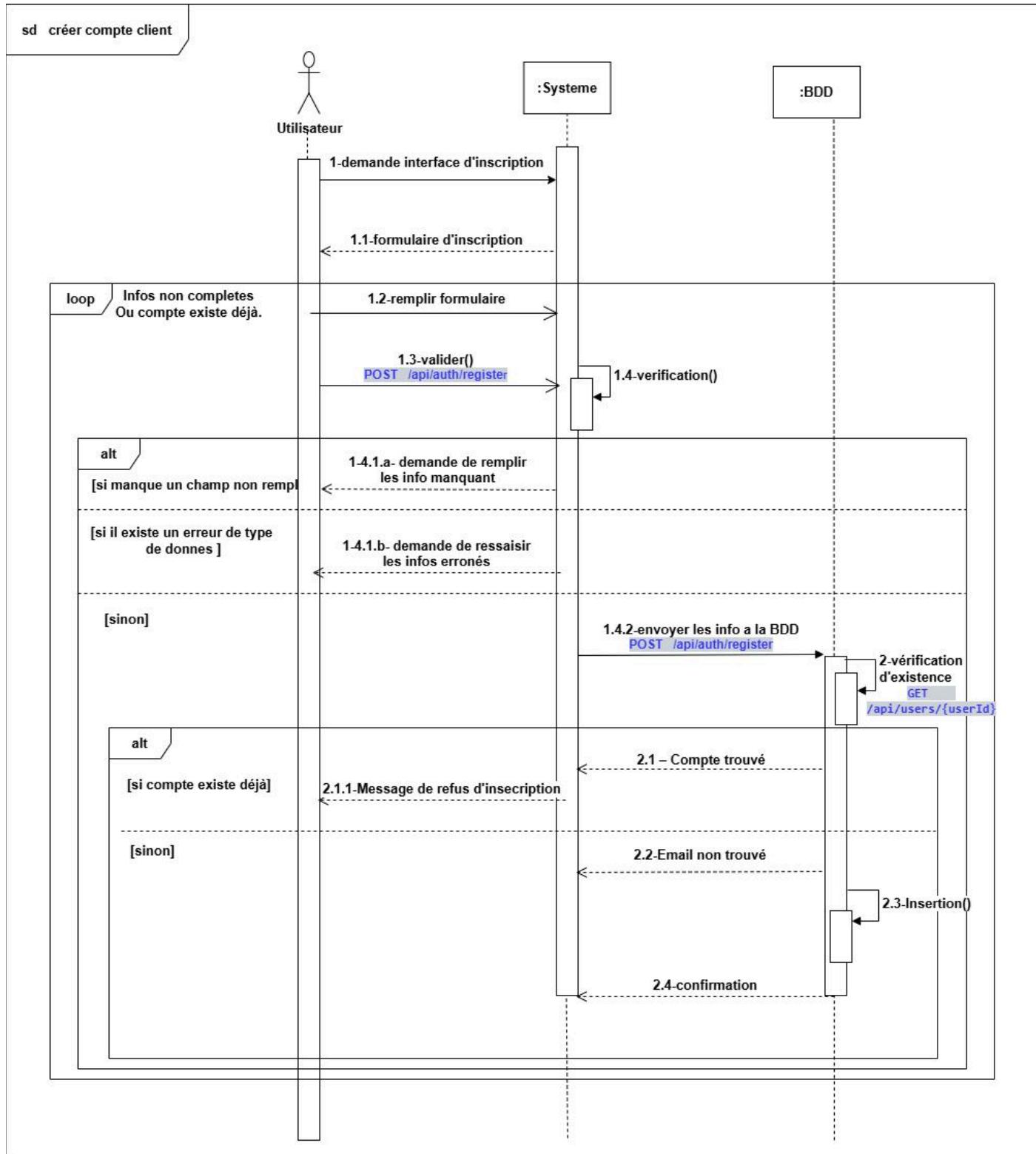
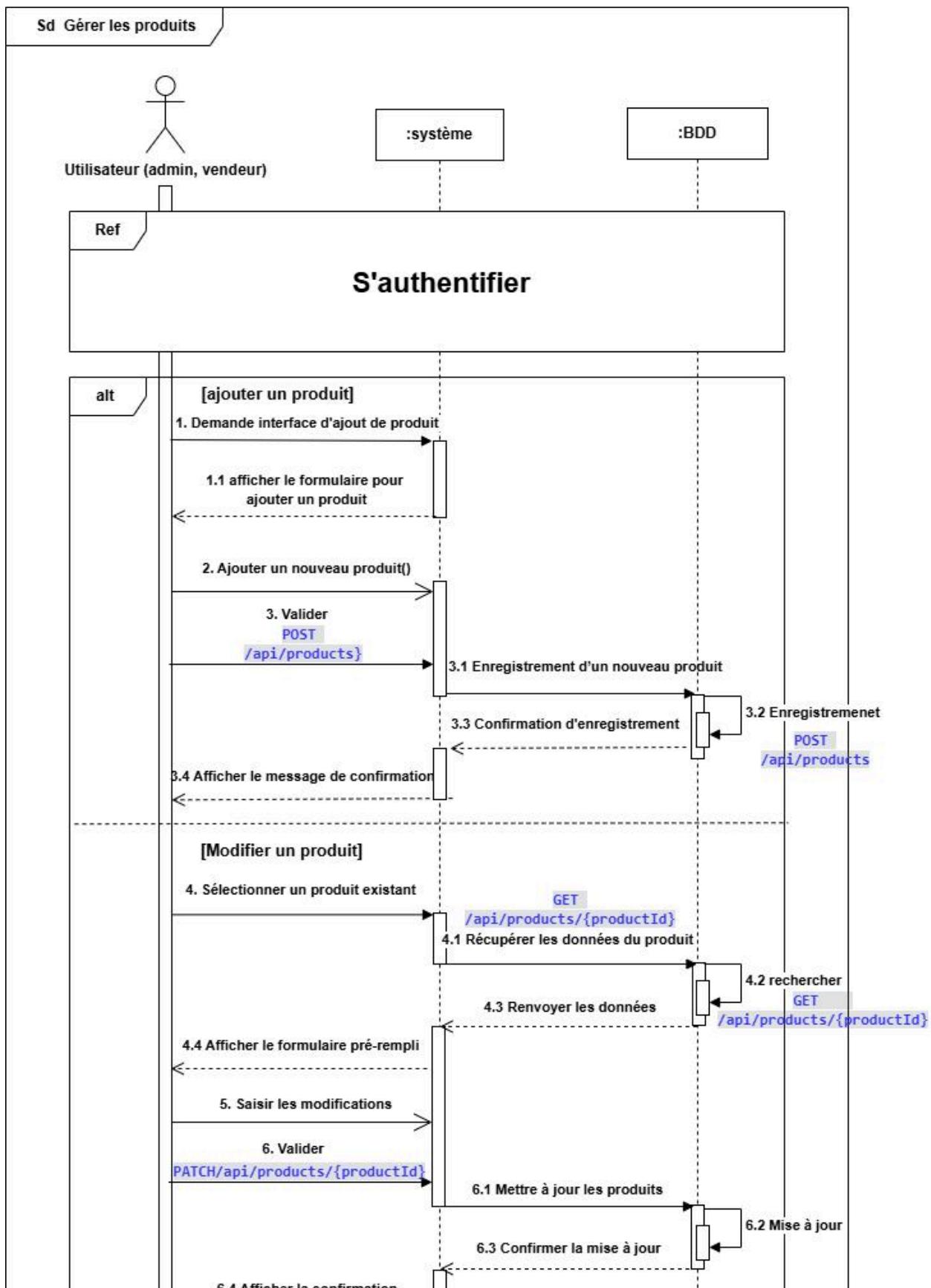


FIG.4 - Diagramme de séquence pour le cas d'utilisation "Créer un compte client"

5.3.4 Diagramme de séquence pour le cas d'utilisation “Gérer les produits”



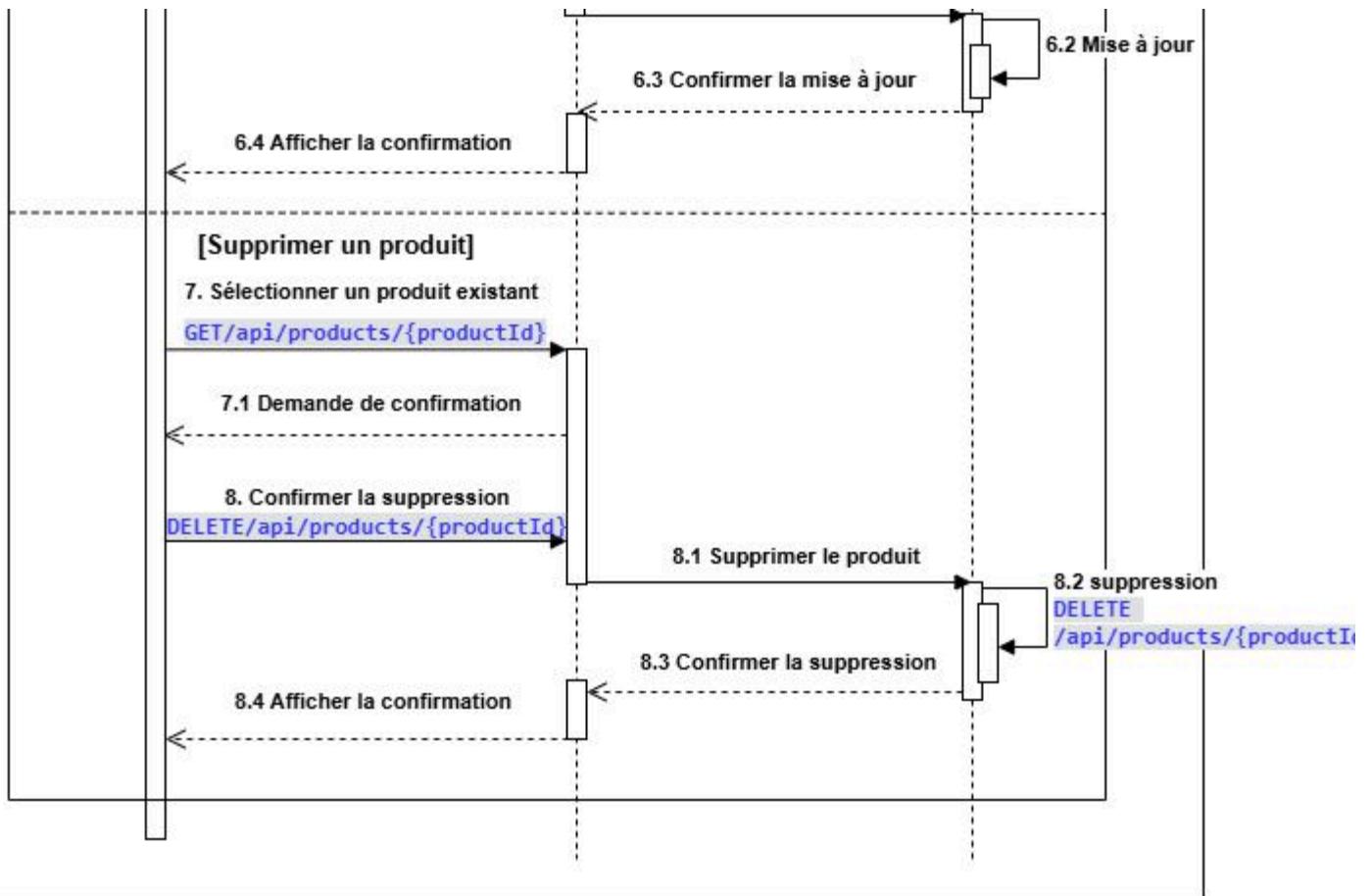


FIG.5 - Diagramme de séquence pour le cas d'utilisation "Gérer les produits"

5.4 Diagramme de déploiement

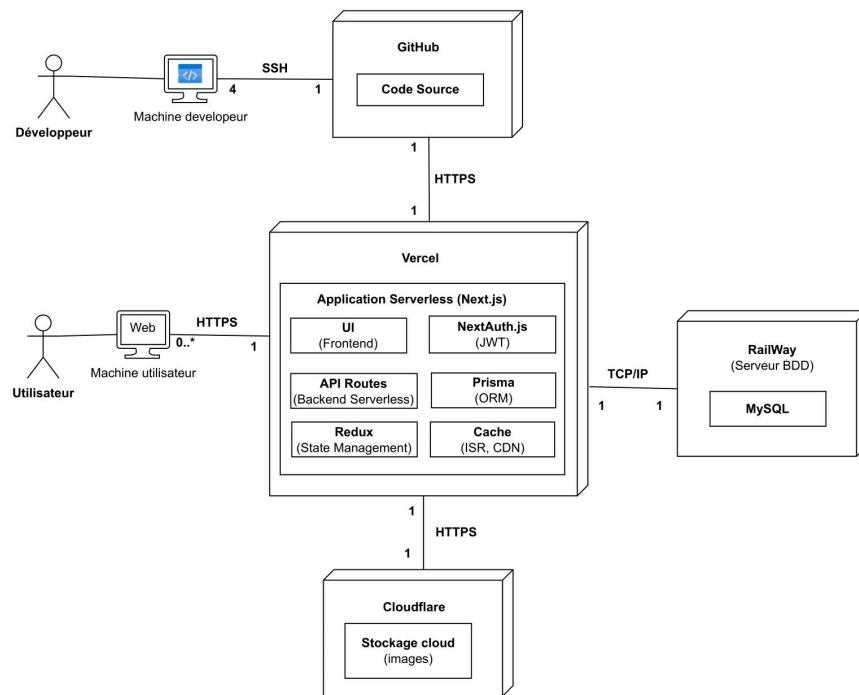


FIG.6 - Diagramme de déploiement

5.5 Diagramme d'états-transitions pour le processus de paiement

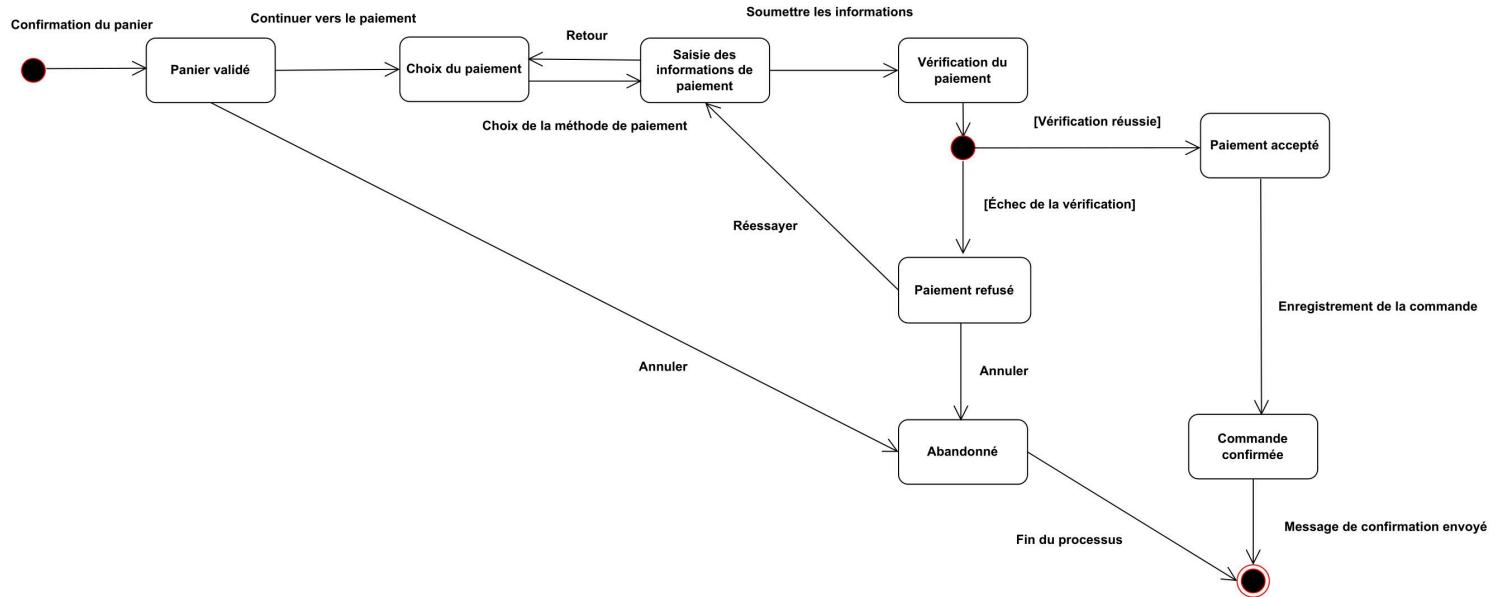


FIG.7 - Diagramme d'états-transitions pour le processus de paiement

5.6 Diagaramme de contexte

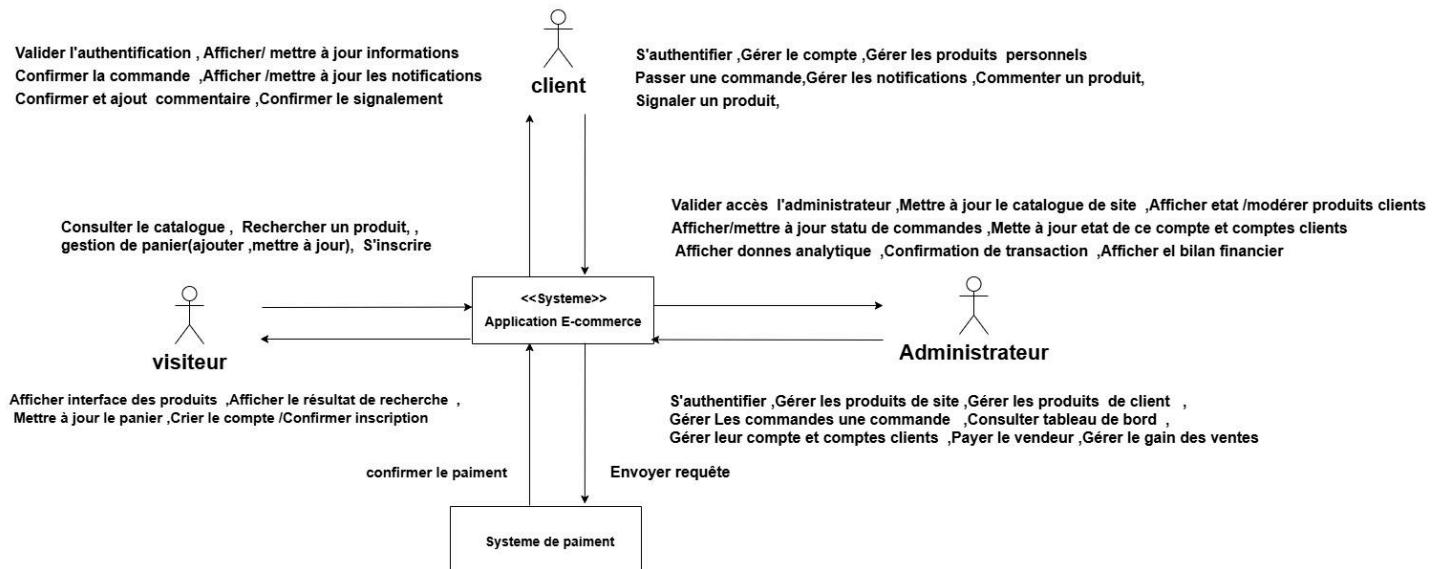


FIG.8 - Diagaramme de contexte

5.7 Diagramme de classe

Un diagramme de classe est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques¹. Il permet de modéliser la structure et le comportement d'un système orienté objet en décrivant ses classes, ses attributs, ses opérations et les interactions entre ses objets. Dans cette sous-section, nous allons détailler les éléments d'un diagramme de classe, tels que la représentation graphique des classes, des associations, de l'héritage, de la visibilité et des multiplicités. Nous allons également illustrer ces concepts par des exemples concrets tirés du domaine de l'application

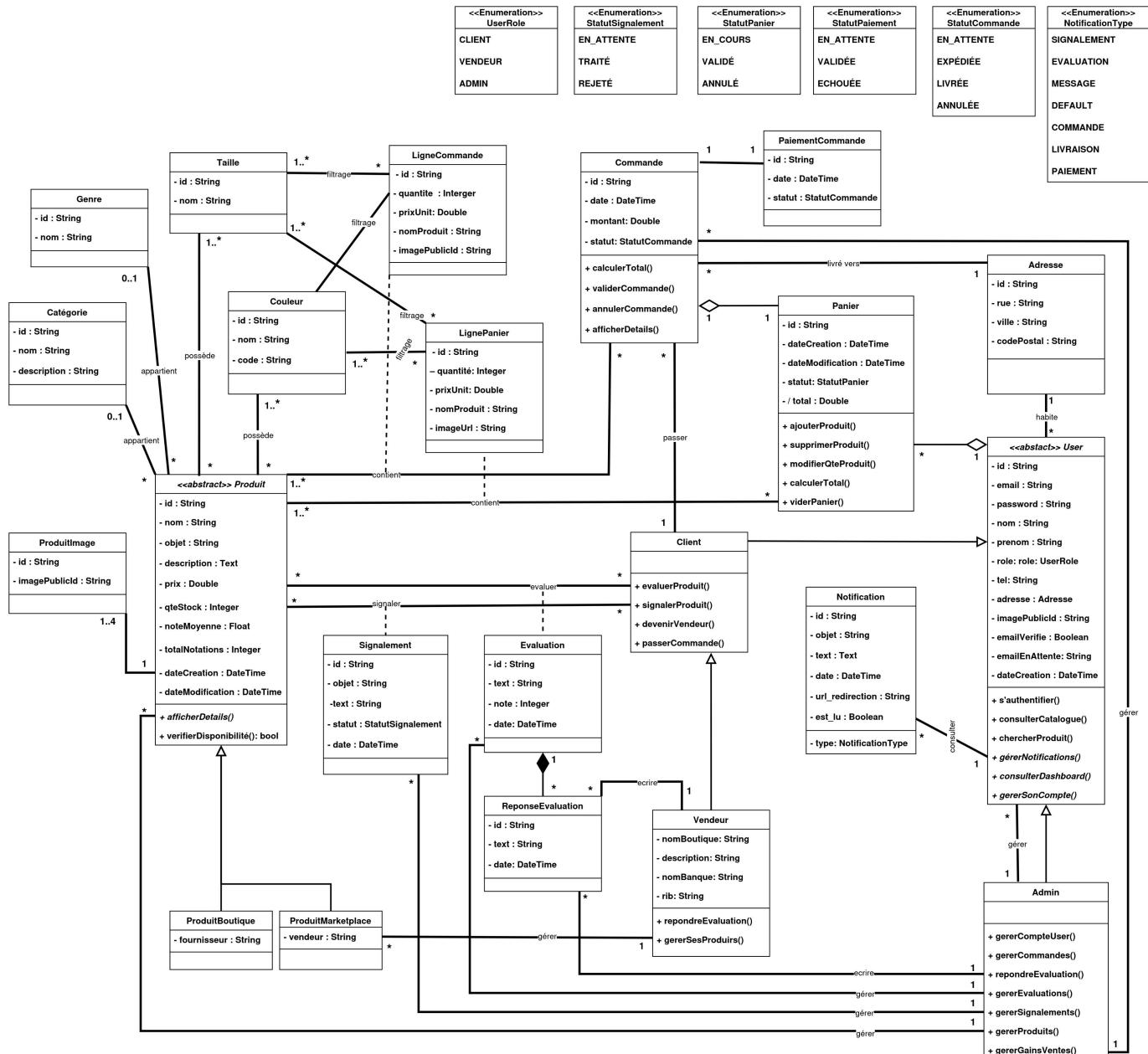


FIG.9 - Diagramme de classe

5.8 Model relationnelle de données

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles. Ce modèle repose sur des principes mathématiques mis en avant par E.F. Codd et est souvent implémenté dans une base de données. Dans le modèle relationnel, les données sont organisées sous formes de tables, qu'on nomme des relations, qui sont associées entre elles. Le modèle relationnel doit respecter plusieurs règles, telles que l'unicité des enregistrements, la définition des clés primaires et étrangères, la normalisation des tables, etc. Dans cette sous-section, nous allons voir comment passer du modèle conceptuel au modèle relationnel, en transformant les classes, les associations, les compositions et les héritages du diagramme UML en tables et en attributs du modèle relationnel.

Pour ce qui concerne notre modèle relationnel :

- **adresse** (id, rue, ville, code_postal, pays)
- **user**(id, email, password, nom, prenom, tel, url_image, date_creation, email_verifie, role, #adresse_id)
- **client** (id : #Fk to user.id)
- **vendeur** (id : #FK to client.id, nom_affichage, rib)
- **client** (id : #Fk to user.id)
- **categorie**(id, nom, description)
- **filtre**(id, nom, valeur)
- **filtre_produit**(#id_filtre, #id_produit)
- **produit**(id, titre, objet, description, prix, qte_stock, note_moyenne, total_notations, image_url, date_creation, date_modification, #categorie_id)
- **produit_boutique** (#produit_id, fournisseur)
- **produit_marketplace** (#produit_id, #vendeur_id)
- **evaluation** (id, #user_id, #produit_id, note, text, date)
- **reponse_evaluation**(id, text, date, #user_id, #evaluatotion_id)
- **signalement**(id, objet, text, statut, date, #client_id, #produit_id)
- **notification**(id,objet, text, date, url_redirection, est_lu, #user_id)
- **panier**(id, statut, date_creation, date_modification, #user_id)
- **ligne_panier**(id, #panier_id, #produit_id, nom_produit, quantite, prix_unit, image_url)
- **commande**(id, date, montant, statut, #adresse_id, #panier_id, #client_id)
- **ligne_commande**(id, #commande_id, #produit_id, nom_produit, quantité, prix_unit, image_url)
- **filtre_ligne_commande**(#filtre_id, #ligne_commande_id)

- **filtre_ligne_panier**(#filtre_id, #ligne_panier_id)
- **paiement_commande**(id, date, statut, #commande_id)
- **paiement_vendeur**(id, montant, date, statut, #vendeur_id)

5.9 Architecture de l'application

Notre application MEGA SHOP est construite selon une architecture moderne en couches qui sépare clairement les responsabilités:

5.9.1 Couche Présentation (Frontend)

- **Composants React:** Interface utilisateur interactive
- **Pages Next.js:** Structures de pages avec routes prédéfinies
- **État global:** Gestion de l'état avec Context API et hooks personnalisés
- **Styles:** Utilisation de Tailwind CSS pour un design responsive

5.9.2 Couche API (Backend)

- **Routes API Next.js:** Points d'entrée pour les opérations CRUD
- **Middleware d'authentification:** Vérification des tokens JWT
- **Validation des données:** Vérification des entrées utilisateur
- **Gestion des erreurs:** Traitement unifié des exceptions

5.9.3 Couche Service

- **Logique métier:** Encapsulation des règles de l'application
- **Services transversaux:** Authentification, notifications, paiements

5.9.4 Couche Accès aux données Prisma ORM:

Interface typée avec la base de données

- **Modèles de données:** Représentation des entités du système
- **Migrations:** Gestion des changements de schéma

5.9.5 Base de données

- **MySQL:** Stockage relationnel des données
- **Schéma:** Structure normalisée pour optimiser les performances

Cette architecture en couches offre plusieurs avantages:

- **Séparation des préoccupations:** Chaque couche a une responsabilité bien définie
- **Testabilité:** Possibilité de tester chaque couche indépendamment
- **Maintenabilité:** Facilité d'évolution et de correction
- **Scalabilité:** Possibilité d'optimiser ou de remplacer des composants spécifiques

5.10 Conclusion

L'analyse et la conception de MEGASHOP ont été menées avec une attention particulière à la modularité et à l'évolutivité du système. Les diagrammes UML ont permis de modéliser précisément les interactions entre les différents acteurs et composants du système, tandis que le diagamme de classe a établi une base de données robuste et normalisée.

L'architecture en couches adoptée représente un compromis idéal entre complexité et maintenabilité, permettant d'isoler les changements et de limiter leur impact sur l'ensemble du système. Cette approche facilite également la répartition du travail entre les membres de l'équipe, chacun pouvant se concentrer sur une couche spécifique.

La décomposition fonctionnelle et la séparation des responsabilités ont été au cœur de notre démarche de conception, assurant ainsi que chaque module puisse évoluer indépendamment tout en conservant une cohérence globale. Cette approche architecturale nous permet d'anticiper les futures extensions du système, notamment l'intégration de nouvelles méthodes de paiement ou le développement d'applications mobiles.

En définitive, les choix techniques et architecturaux réalisés durant cette phase de conception constituent le fondement solide sur lequel repose l'ensemble de notre application e-commerce.

CHAPITRE 6 : RÉALISATION DE L'APPLICATION

Le chapitre 6 est consacré à la réalisation de l'application.

Il détaillera les étapes de développement et les choix techniques effectués.

Le chapitre 6 sera divisé en plusieurs sections :

- Section 6.1 : Architecture et design.

- Section 6.2 : Implémentation et codage.

- Section 6.3 : Tests et validation.

- Section 6.4 : Conclusion et perspectives.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

Le chapitre 6 sera terminé par une conclusion et des perspectives pour l'avenir.

6.1 Livrables:

- [🌐 Lien de l'application en ligne](https://project-megashop.vercel.app)
<https://project-megashop.vercel.app>
- [🔗 Code source sur GitHub](https://github.com/lyes-mersel/megashop)
<https://github.com/lyes-mersel/megashop>
- [📁 Google Drive \(Script MySQL + code généré par l'AGL\)](https://drive.google.com/drive/folders/1oer1IjKjUbgMOYDR0GvrETKF_K6EykvD?usp=sharing)
https://drive.google.com/drive/folders/1oer1IjKjUbgMOYDR0GvrETKF_K6EykvD?usp=sharing

6.2 Interfaces principales

6.2.1 Interfaces principales de application

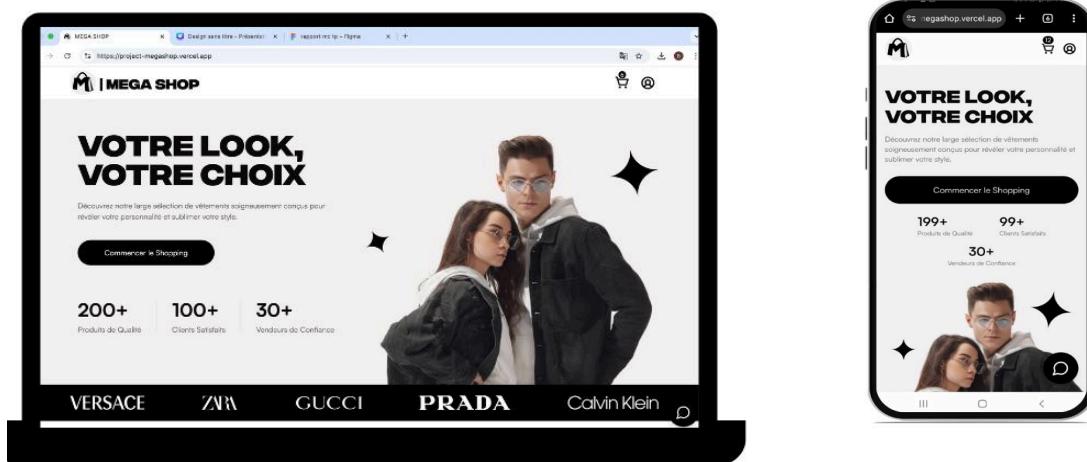


FIG.8-10 - Interface principale de l'application version pc et mobile

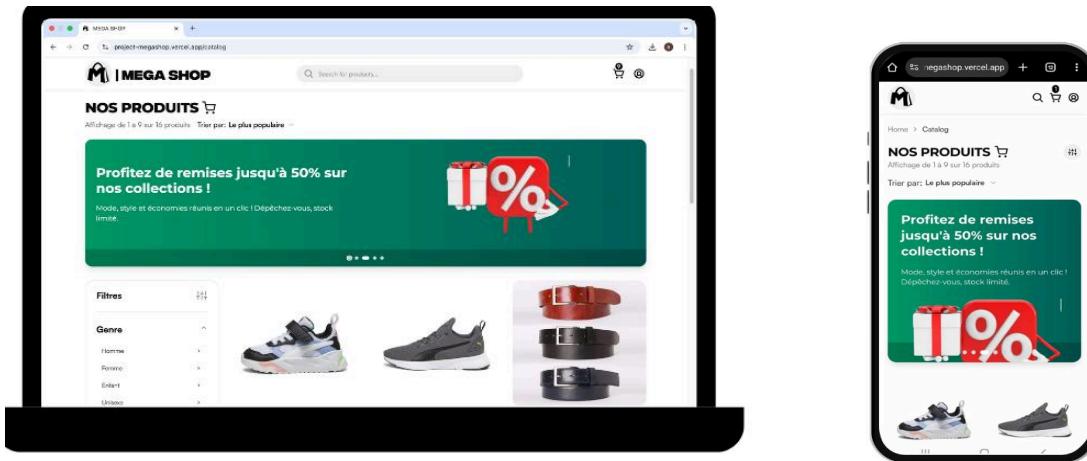


FIG.11 - Interface de produits version pc et mobile

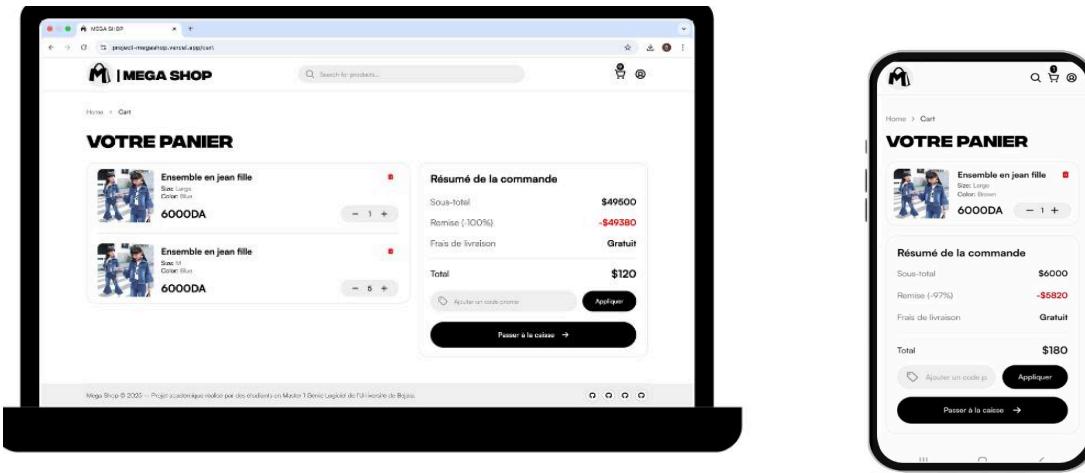


FIG.12 - Interface de panier version pc et mobile

6.2.2 Interfaces principales de client

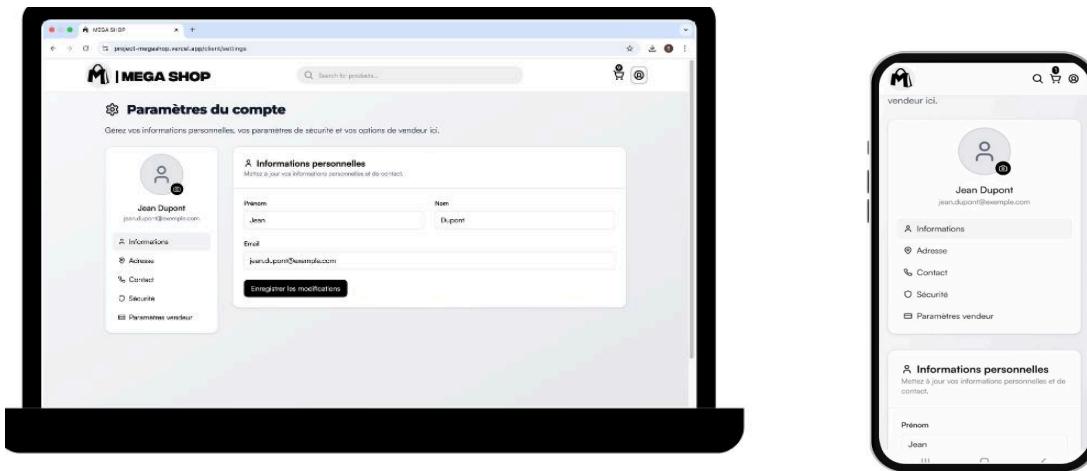


FIG.13 - Interface paramètres version pc et mobile

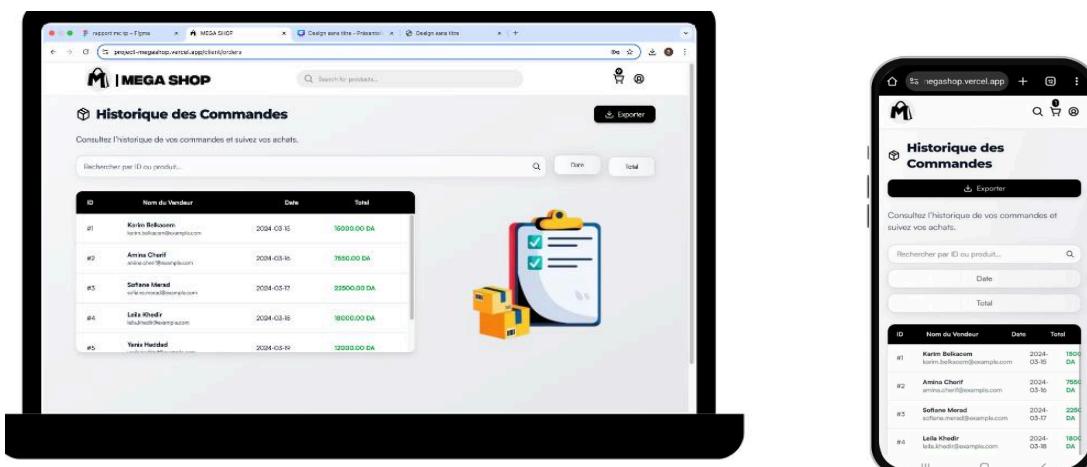


FIG.14 - Interface commande version pc et mobile

6.2.3 Interfaces principales de l'administrateur

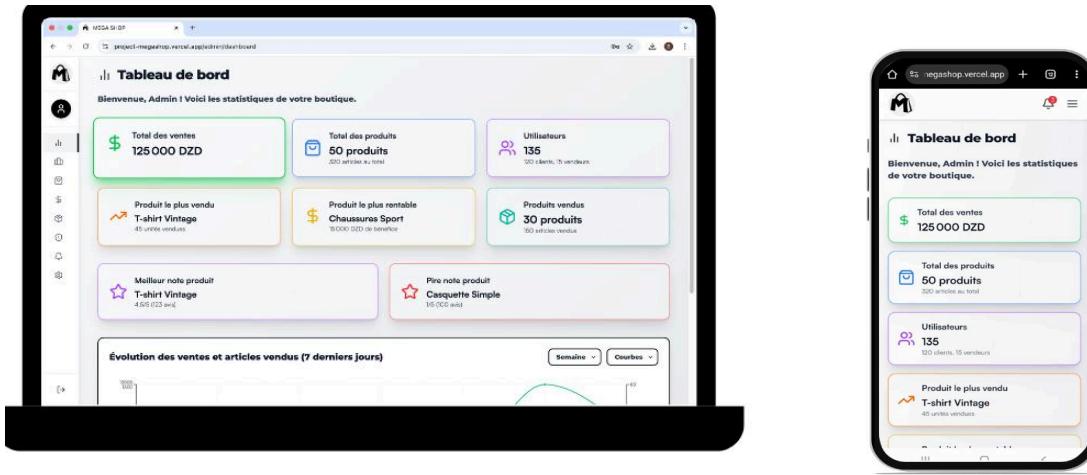


FIG.15 - Interface dashboard version pc et pc

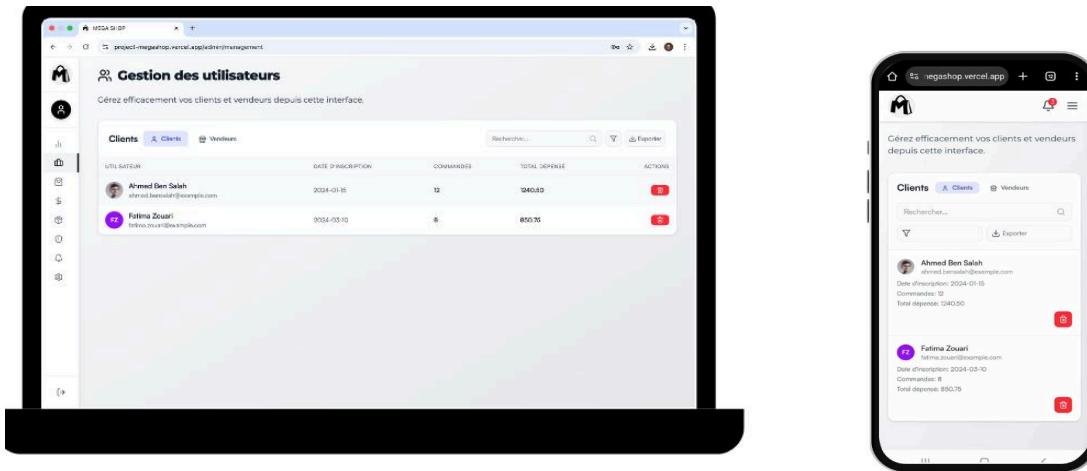


FIG.16 - Interface gestion d'utilisateurs version pc et mobile

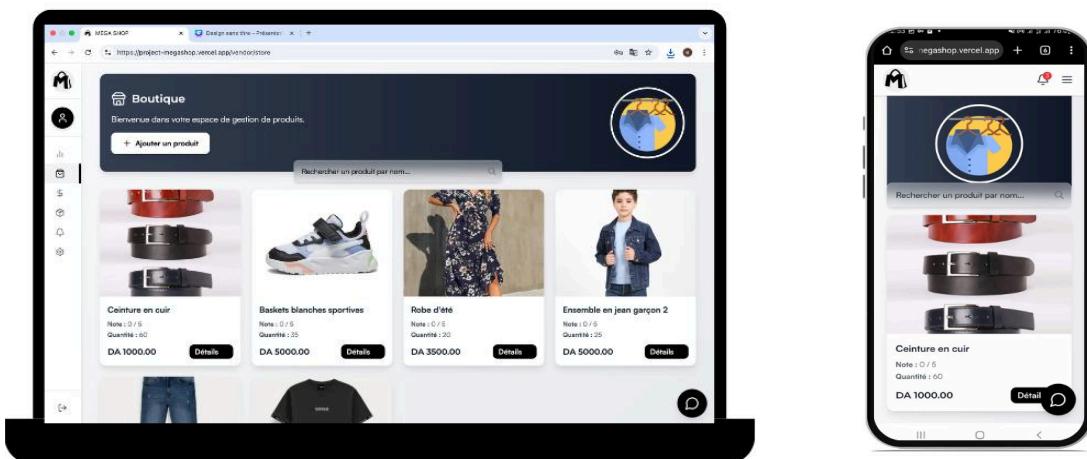


FIG.17 - Interface ma boutique version pc et mobile

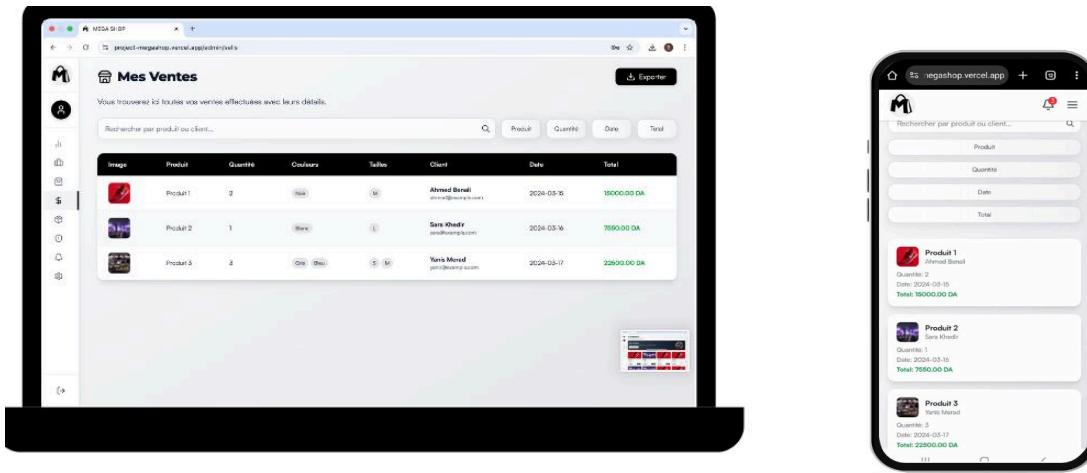


FIG.18 - Interface mes ventes version pc et mobile

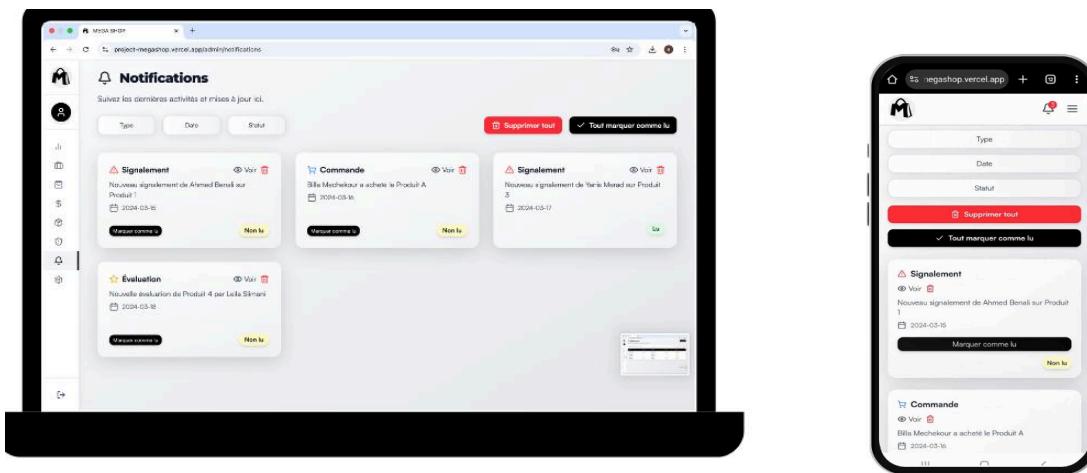


FIG.19 - Interface notifications version pc et mobile

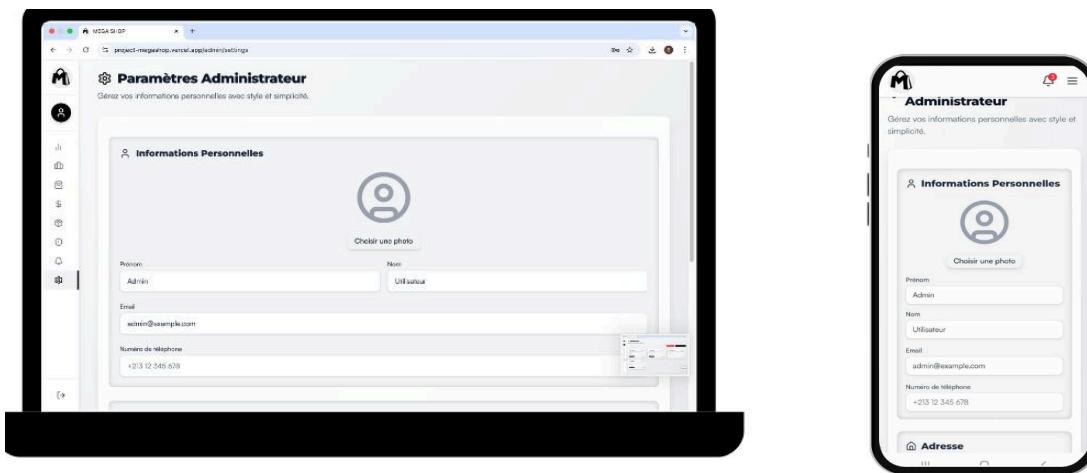


FIG.20 - Interface paramètres version pc et mobile

6.2.4 Interfaces principales de vendeur

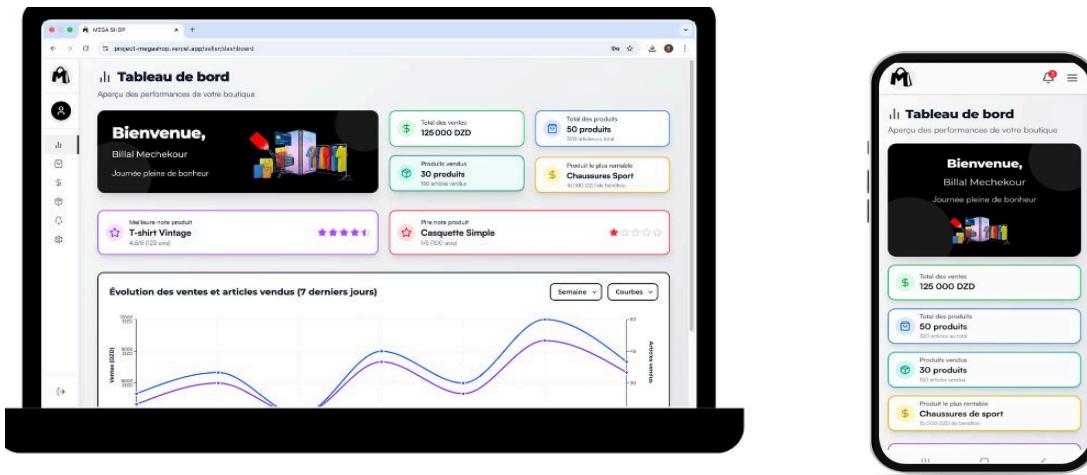


FIG.21 - Interface dashbord version pc et mobile

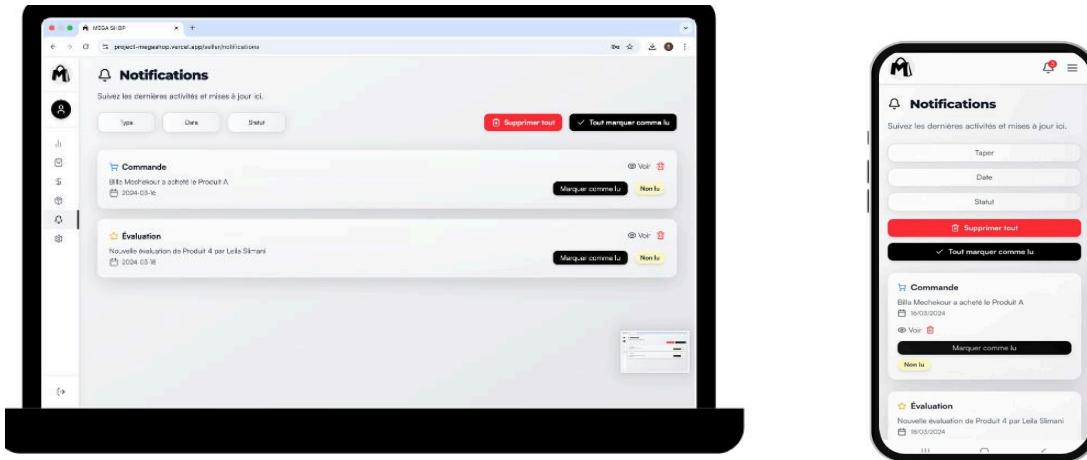


FIG.22 - Interface notifications version pc et mobile

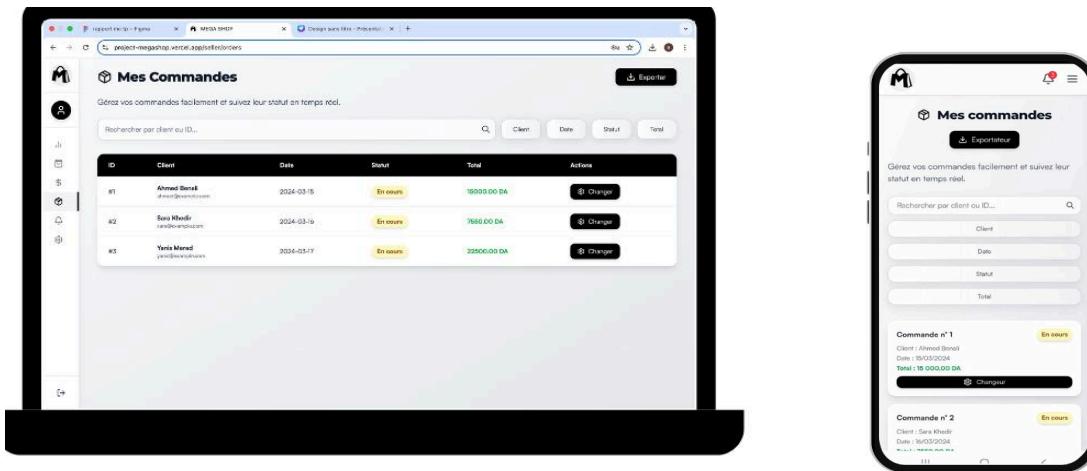


FIG.23 - Interface mes commandes version pc et mobile

CHAPITRE 7 : CONCLUSION ET PERSPECTIVES

Le présent chapitre a pour objectif de synthétiser les résultats obtenus et d'identifier les perspectives futures.

En premier lieu, il est important de souligner que les résultats obtenus sont en grande partie conformes aux prévisions initiales.

Cependant, il existe quelques différences significatives entre les résultats obtenus et les prévisions initiales.

Ensuite, il est nécessaire de identifier les perspectives futures pour améliorer les résultats obtenus.

Enfin, il est important de prendre en compte les résultats obtenus pour améliorer les méthodes et les procédures utilisées.

7.1 Bilan du projet

Le développement de l'application e-commerce MEGA SHOP a été une expérience enrichissante qui nous a permis de mettre en pratique nos connaissances théoriques dans un contexte réel. Nous avons réussi à créer une plateforme fonctionnelle qui répond aux exigences définies dans le cahier des charges.

Ce projet nous a confronté à des défis techniques et organisationnels, nous poussant à rechercher des solutions innovantes et à approfondir notre compréhension des technologies modernes de développement web.

La méthodologie Agile adoptée s'est révélée particulièrement adaptée à notre contexte, permettant une progression constante et des ajustements réguliers en fonction des retours et des difficultés rencontrées.

7.2 Améliorations futures possibles

Bien que notre application MEGA SHOP réponde aux exigences initiales, plusieurs axes d'amélioration ont été identifiés pour les futures versions:

1. Intégration de paiements réels:

- Implémentation de Stripe pour les transactions internationales
- Intégration de Slick Pay pour les paiements locaux en Algérie
- Mise en place d'un système de portefeuille électronique interne

2. Extension mobile:

- Développement d'une application mobile native (React Native)
- Adaptation de l'interface pour une expérience mobile optimisée
- Implémentation de fonctionnalités spécifiques (notifications push, scan de produits)

3. Amélioration de l'intelligence artificielle:

- Système de recommandation basé sur les préférences utilisateur
- Détection automatique de contenus inappropriés

4. Enrichissement logistique:

- Module de suivi des livraisons en temps réel
- Intégration avec des services logistiques locaux
- Optimisation des coûts de livraison par regroupement intelligent

5. Expansion sociale:

- Intégration de fonctionnalités de partage sur réseaux sociaux
- Système d'affiliation pour encourager le partage
- Fonctionnalités communautaires (forums, groupes d'achat)