

UNIVERSITÉ CADI AYYAD

Ecole Nationale des Sciences Appliquées de Marrakech

Filière : Génie Informatique

Rapport du Projet de Fin de Semestre

Thème :

Digitalisation du commerce des matériaux de
construction et de bricolage au Maroc :
Conception et développement d'une application web

Présent par :

AHMAMO Hamza
MGHIRA Oussama
AOURIR Aïssam

Encadré par :

M.Walid Bouarifi

Remerciements

Nous tenons tout d'abord à exprimer notre profonde gratitude à notre encadrant, M. Walid Bouarifi, pour sa guidance, ses conseils avisés et son soutien constant tout au long de ce projet. Sa disponibilité, son expertise et son engagement ont été essentiels à la réussite de ce travail.

Nous souhaitons également adresser nos remerciements à l'ensemble de l'équipe pédagogique de l'ENSA de Marrakech pour la qualité de l'enseignement qui nous a été dispensé.

Nous exprimons aussi notre reconnaissance à notre ami Elkaamouss Aissam, qui nous a prodigué de précieux conseils et une aide significative dans le cadre de ce projet.

Nous tenons par ailleurs à remercier les membres du jury, notamment M. Abdelghafour Atlas, pour leur investissement et le temps qu'ils ont consacré à évaluer notre travail. Leurs retours constructifs et leur expertise nous ont permis d'enrichir et d'améliorer notre projet.

Enfin, nous adressons notre gratitude à toutes les personnes ayant contribué, de près ou de loin, à la réalisation de ce travail

Résumé

Ce projet ambitieux vise à développer une application web de gestion commerciale multi-rôles en utilisant **Angular**, **Spring Boot** et **MySQL**. L'objectif est de centraliser les interactions entre les **administrateurs**, **fournisseurs** et **clients** au sein d'une plateforme intuitive, performante et sécurisée.

L'administrateur dispose d'un tableau de bord complet lui permettant de **recruter et gérer les fournisseurs**, **valider ou refuser les commandes**, et **gérer les profils clients**. Du côté des fournisseurs, l'application leur permet de **créer et organiser des catégories**, **ajouter des produits**, et **gérer leur catalogue** en toute autonomie. Quant aux clients, ils peuvent **parcourir les produits disponibles** et **passer des commandes en quelques clics**, garantissant une expérience fluide et accessible.

Grâce à une architecture moderne et bien structurée, ce système assure une séparation claire des responsabilités et une communication fluide entre les différentes parties prenantes. Ce projet met en œuvre les meilleures pratiques du développement full-stack pour offrir une solution robuste et évolutive, répondant aux besoins d'une plateforme e-commerce B2B/B2C moderne.

ABSTRACT

Ce projet consiste en la conception et le développement d'une application web de gestion commerciale destinée à faciliter les interactions entre administrateurs, fournisseurs et clients. L'application est structurée autour de trois profils utilisateurs principaux : **l'administrateur**, chargé de la gestion des fournisseurs, des clients et des commandes ; **le fournisseur**, responsable de la création et la gestion des catégories et des produits ; et **le client**, qui peut consulter les produits et passer des commandes.

Le système repose sur une architecture **full-stack** utilisant **Angular** pour l'interface utilisateur, **Spring Boot** pour le backend, et **MySQL** pour la gestion des données. Cette application propose une interface intuitive et une logique métier claire, permettant un contrôle précis sur le cycle de vie des commandes et une gestion efficace des produits.

L'objectif principal est de fournir une solution web robuste, évolutive et sécurisée, répondant aux besoins d'une plateforme e-commerce multi-utilisateurs, en assurant une bonne séparation des rôles, une gestion fluide des données, et une expérience utilisateur optimisée.

introduction

Dans un contexte où le commerce en ligne prend une place de plus en plus importante, la mise en place de plateformes web permettant la gestion centralisée des interactions entre différents types d'utilisateurs devient une nécessité. Ce projet s'inscrit dans cette dynamique en proposant le développement d'une application web de gestion commerciale, destinée à faciliter la collaboration entre administrateurs, fournisseurs et clients.

L'objectif principal est de concevoir un système complet, intuitif et sécurisé permettant à chaque acteur d'interagir selon ses droits et responsabilités : l'administrateur gère les utilisateurs et valide les commandes, le fournisseur administre ses produits et ses catégories, et le client explore les produits disponibles pour passer ses commandes.

répondre à ces exigences, nous avons adopté une architecture full-stack moderne, combinant **Angular** pour le développement du frontend, **Spring Boot** pour le backend, et **MySQL** comme système de gestion de base de données. Cette architecture permet une séparation claire des responsabilités, une maintenance facilitée, et une évolutivité du système.

Ce rapport présente l'ensemble des étapes de réalisation du projet, depuis l'analyse des besoins jusqu'à la mise en œuvre technique, en passant par la modélisation, la conception et le développement de l'application. Il met également en évidence les choix technologiques, les défis rencontrés et les solutions apportées.

Sommaire

Remerciements	02
Résumé.....	03
Abstract.....	04
Introduction.....	04
Chapitre 1: Contexte général du projet	09
I. <u>Problématique</u>	09
II. Solution.....	10
III. <u>Diagramme de GANTT :</u>	11
Chapitre 2: Conception et Mise en place du projet	12
I. Introduction	12
II. <u>Conception UML</u>	12
1. Diagramme cas d'utilisation:	12
2. Diagramme de classes :	14
3. Diagramme de séquence :	16
4. La base de données :	17
III. Conclusion	17
Chapitre 3: Etude Technique Du Projet	18
I. Architecture technique.....	18
1. <u>Modele Rest API</u>	18
2. Spring Boot/Angular/MySQL :	18
3. Technologies et utiles utilisees :	19
II. Arborescente du projet:	24
1. <u>Back-End</u>	24
2. <u>Front-End</u>	27
Chapitre 4: Réalisation du projet	30
1. Pages commune:	30
2. Pages admin:.....	35
3. pages fournisseur	40
4. Pages client:	43
Chapitre 5 : Synthèses et Orientation Futures	42
I. conclusion :	44

II. Perspectives :	44
III. References :	45

Table des figures

Figure 1 : Diagramme de GANTT.....	11
Figure 2: Diagramme de cas d'utilisation.....	13
Figure3: Diagramme de classes.....	15
Figure4 : Diagramme de séquence.....	16
Figure5 : Architecture REST API.....	18
Figure6: <i>Architecture de combinaison</i> Spring Boot/Angular/MySQL.....	19
Figure7: Architecture du Back-End.....	25
Figure8 : Architecture du Front-End.....	27
Figure9 : Interface de d'accueil.....	30
Figure10: Interface d'inscription.....	31
Figure11: <i>Interface de connexion</i>	32
Figure12: <i>Interface de verification d'inscription</i>	33
Figure13: <i>Interface de reinitialisation de mot de passe</i>	34
<i>Figure 14: Interface tableau de bord pour admin</i>	35
<i>Figure 15: Interface de gestion des Produits</i>	35

<i>Figure 16: Interface de gestion des commandes.....</i>	<i>36</i>
<i>Figure 17: Interface de gestion des categories des produits.....</i>	<i>36</i>
<i>Figure 18: Interface de gestion des categories des produits.....</i>	<i>37</i>
<i>Figure 19: Interface de gestion des fournisseurs.....</i>	<i>38</i>
<i>Figure 20: Interface de gestion des categories des produits.....</i>	<i>39</i>
<i>Figure 21: Interface de gestion des produits par fournisseur.....</i>	<i>40</i>
<i>Figure 22: Interface de modification de produits par un fournisseur.....</i>	<i>41</i>
<i>Figure 23: Interface de creation des categories par un fournisseur.....</i>	<i>42</i>
<i>Figure 24: Interface de creation des produits par un fournisseur.....</i>	<i>42</i>
<i>Figure 25: Interface d'affichage d'un client.....</i>	<i>43</i>
<i>Figure 26: Interface d'affichage du profil d'un client.....</i>	<i>44</i>

Chapitre 1 : *Contexte général du projet*

I. Problématique :

Avec l'évolution rapide du numérique, de nombreux secteurs ont entamé leur transition vers la digitalisation afin d'optimiser leurs services et de mieux répondre aux besoins des consommateurs. Toutefois, au Maroc, le domaine du commerce des matériaux de construction et de bricolage reste encore largement dépendant des méthodes traditionnelles. Ce manque de digitalisation crée des obstacles à la visibilité des magasins, à la gestion des stocks, et à l'accessibilité des produits pour les clients, malgré le fait que ce secteur soit particulièrement répandu et essentiel à l'économie locale.

La mise en place d'une application web dédiée permettrait non seulement de moderniser l'expérience d'achat, mais aussi de faciliter la mise en relation entre vendeurs et clients, en centralisant les offres et en simplifiant les processus de commande. En intégrant des fonctionnalités comme la recherche intelligente de produits, la gestion des catalogues en ligne, ou encore un système de communication directe entre clients et commerçants, cette solution contribuerait à combler le fossé numérique existant dans ce domaine.

Une telle plateforme favoriserait également une meilleure gestion commerciale, une visibilité élargie à l'échelle nationale, et une meilleure satisfaction des clients grâce à une expérience fluide, rapide et adaptée aux nouvelles habitudes de consommation. En somme, la digitalisation du

commerce des matériaux de construction et de bricolage représente une étape clé pour moderniser un secteur à fort potentiel, tout en répondant aux exigences actuelles du marché.

II. Solution:

Pour répondre à ce problème, nous avons développé une application web moderne dédiée à la digitalisation du commerce des matériaux de construction et de bricolage. Cette solution vise à combler le manque de présence numérique des magasins marocains opérant dans ce secteur, tout en modernisant leur fonctionnement et en facilitant l'expérience client. L'application a été conçue en utilisant des technologies web robustes telles que **Java, Spring Boot, Angular**, et **MySQL**, garantissant à la fois performance, évolutivité et sécurité.

Premièrement, nous avons mis en place un espace dédié aux vendeurs, leur permettant de créer et de gérer facilement leurs catalogues de produits, d'actualiser les prix, les quantités disponibles, et de suivre les commandes en temps réel. Cette interface intuitive leur permet de mieux gérer leur activité tout en augmentant leur visibilité auprès d'une clientèle plus large.

Deuxièmement, les clients disposent d'un module de recherche avancée leur permettant de trouver rapidement des matériaux ou outils selon plusieurs critères : catégorie, prix, disponibilité, lieu de livraison, etc. L'application intègre également un système de **favoris**, une gestion simplifiée du **panier d'achat**, et un suivi de **commande en temps réel**, offrant ainsi une expérience fluide et moderne à l'utilisateur final.

En outre, afin de favoriser l'accessibilité et d'encourager l'adoption du numérique dans ce secteur, l'application propose une fonctionnalité d'**invitation** permettant aux commerçants de recommander facilement la plateforme à d'autres professionnels du domaine. Cela contribue à élargir l'écosystème de l'application et à renforcer sa communauté d'utilisateurs.

Enfin, nous avons intégré un tableau de bord dynamique pour les administrateurs, leur permettant de suivre l'évolution des ventes, de modérer les annonces, de gérer les comptes utilisateurs (vendeurs et clients), et d'assurer le bon fonctionnement général de la plateforme.

En centralisant les offres, en facilitant la gestion commerciale, et en simplifiant l'accès des clients aux produits, notre solution numérique transforme en profondeur le commerce des matériaux de construction et de bricolage au Maroc. Elle renforce la compétitivité des commerçants tout en répondant aux attentes des consommateurs en matière de praticité, de rapidité et de transparence.

III. Diagramme de GANTT:

Nous avons utilisé un diagramme de Gantt dans notre rapport de projet pour visualiser la planification du projet et les différentes tâches impliquées. Ce diagramme permet une vue d'ensemble claire du projet, en mettant en évidence les dépendances entre les tâches et en identifiant les tâches critiques. Nous mettons à jour régulièrement le diagramme pour refléter les changements et ajustements réalisés. Le diagramme de Gantt joue un rôle central dans notre rapport, offrant une représentation visuelle concise de la planification et de l'évolution du projet.

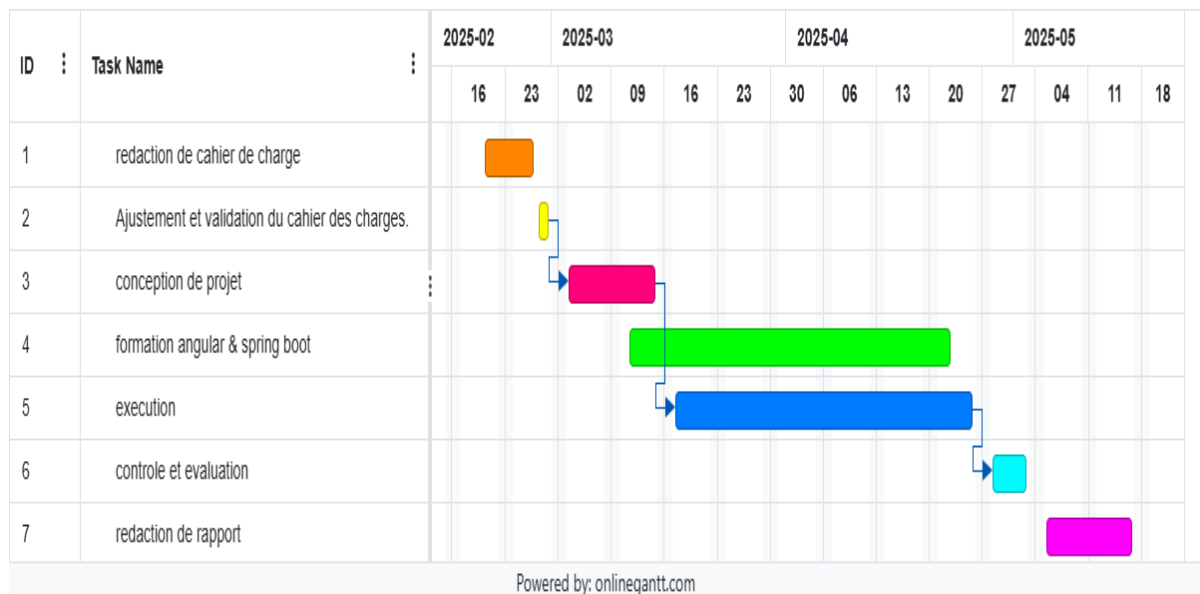


Figure 1 : Diagramme de GANTT.

Chapitre 2: Conception et Mise en place du projet:

I. Introduction :

Dans cette phase de conception, notre objectif est de modéliser en détail l'architecture de notre application web de gestion commerciale multi-utilisateurs, afin de garantir son bon fonctionnement et sa réussite. Une planification rigoureuse et une conception bien structurée sont essentielles pour établir une base technique solide, anticiper les éventuelles difficultés et assurer un développement fluide.

Dans ce chapitre, nous allons présenter les différents éléments clés de la conception, notamment les **diagrammes UML** tels que le **diagramme de cas d'utilisation**, le **diagramme de classes** et le **diagramme de séquence**, ainsi que le **modèle de la base de données**. Ces représentations graphiques nous permettent de **visualiser les rôles et les interactions des utilisateurs**, **modéliser les entités métier** et **organiser les flux de traitement** de l'application.

Chaque composant de cette phase de conception contribue à structurer notre application développée avec **Angular** en frontend, **Spring Boot** en backend, et **MySQL** comme base de données, en assurant une séparation claire des responsabilités et une cohérence dans la gestion des fonctionnalités.

II. Conception UML:

1. Diagramme de cas d'utilisation :

Le diagramme des cas d'utilisation ci-dessous illustre les différentes interactions possibles entre les utilisateurs (Administrateur, Fournisseur, Client) et le système. Il représente de manière globale les fonctionnalités principales offertes par l'application:

- ✓ L'**administrateur** peut gérer les fournisseurs, gérer les clients, consulter et approuver ou refuser les commandes.
- ✓ Le **fournisseur** peut gérer ses catégories de produits ainsi que ses produits.
- ✓ Le **client** peut consulter les produits disponibles et effectuer des commandes.

Ce diagramme permet ainsi d'avoir une vision d'ensemble des cas d'utilisation offerts par l'application, en identifiant clairement les rôles de chaque type d'utilisateur.

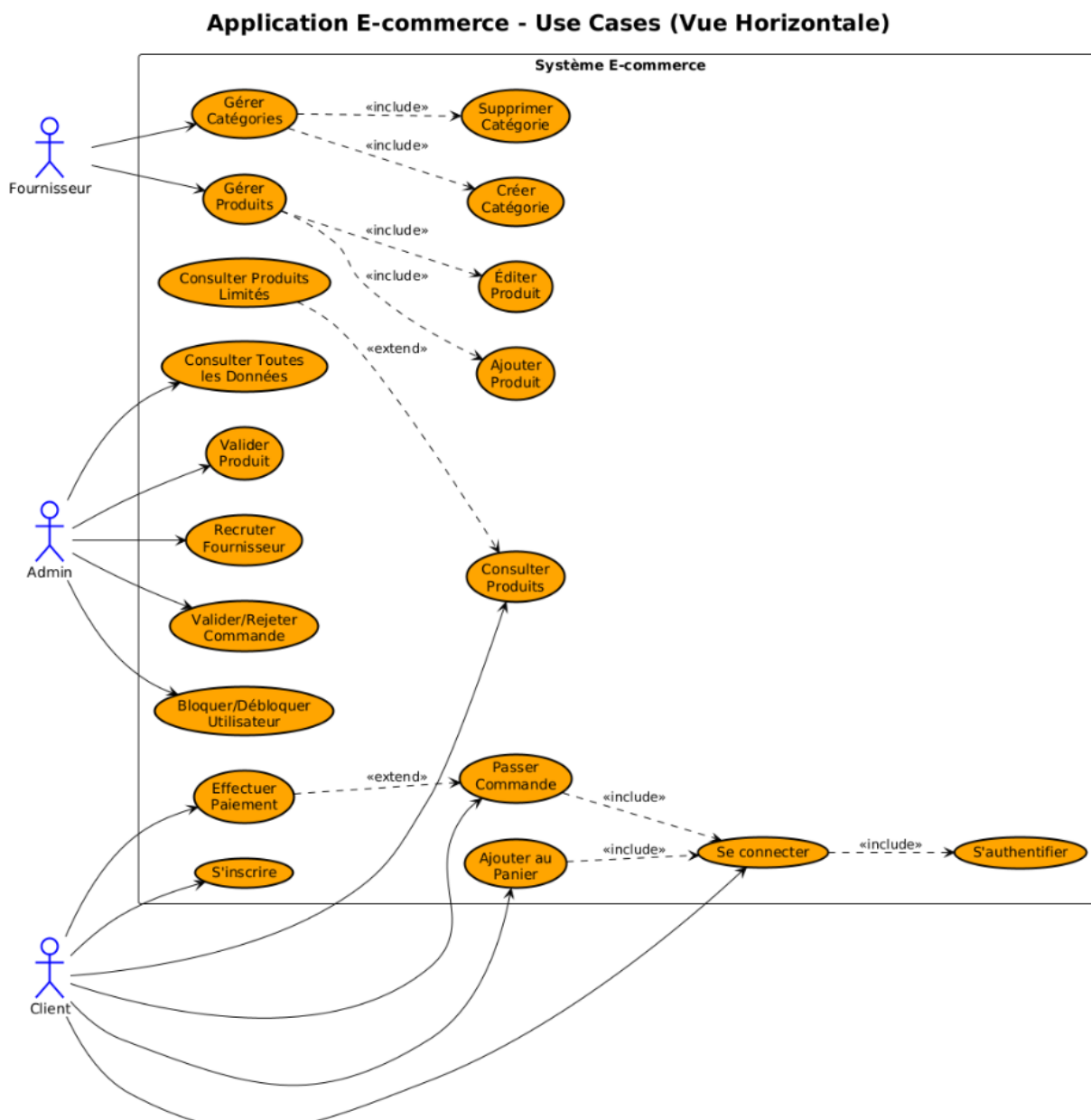


Figure 2: Diagramme de cas d'utilisation.

2. Diagramme de classes :

Le diagramme de classes ci-dessous représente la structure statique de l'application. Il illustre les différentes entités principales du système ainsi que leurs relations. L'application repose sur une hiérarchie d'utilisateurs (administrateur, fournisseur et client), chacun ayant des responsabilités spécifiques. Les classes métier telles que **Produit**, **Commande**, **Catégorie** et **Panier** permettent de modéliser les fonctionnalités principales du système. Ce diagramme facilite la compréhension de l'organisation des données et des interactions entre les objets dans l'application.

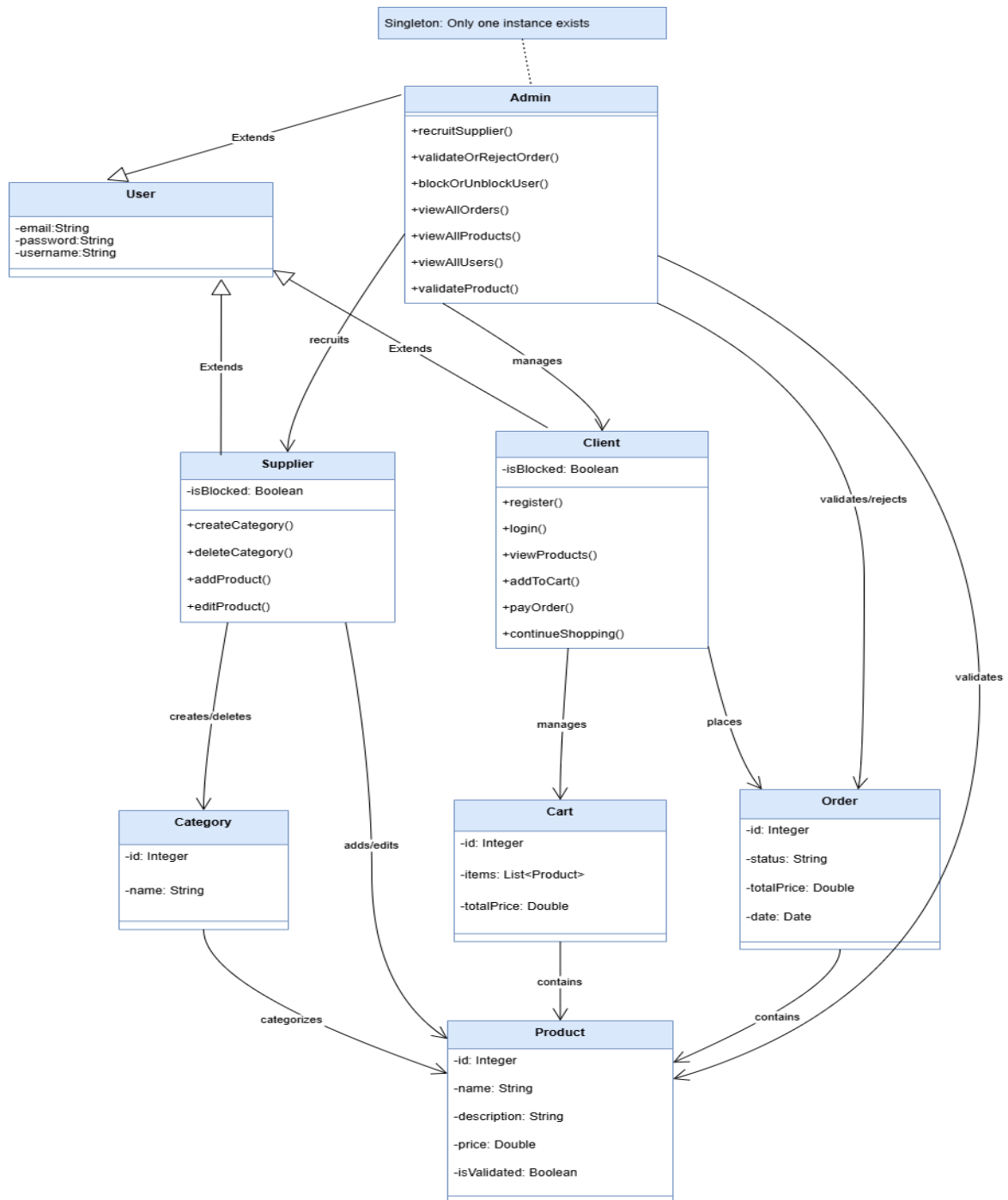


Figure 3 : Diagramme de classes.

3. Diagramme de séquence :

Ce diagramme illustre les interactions d'un **client** avec le système e-commerce, depuis la **connexion** jusqu'au **paiement**. Il montre comment le client consulte les produits, ajoute des articles au panier, passe une commande, puis effectue le paiement. Chaque action déclenche des échanges entre l'interface utilisateur, le contrôleur, le service métier et la base de données, reflétant le parcours d'achat classique dans l'application.

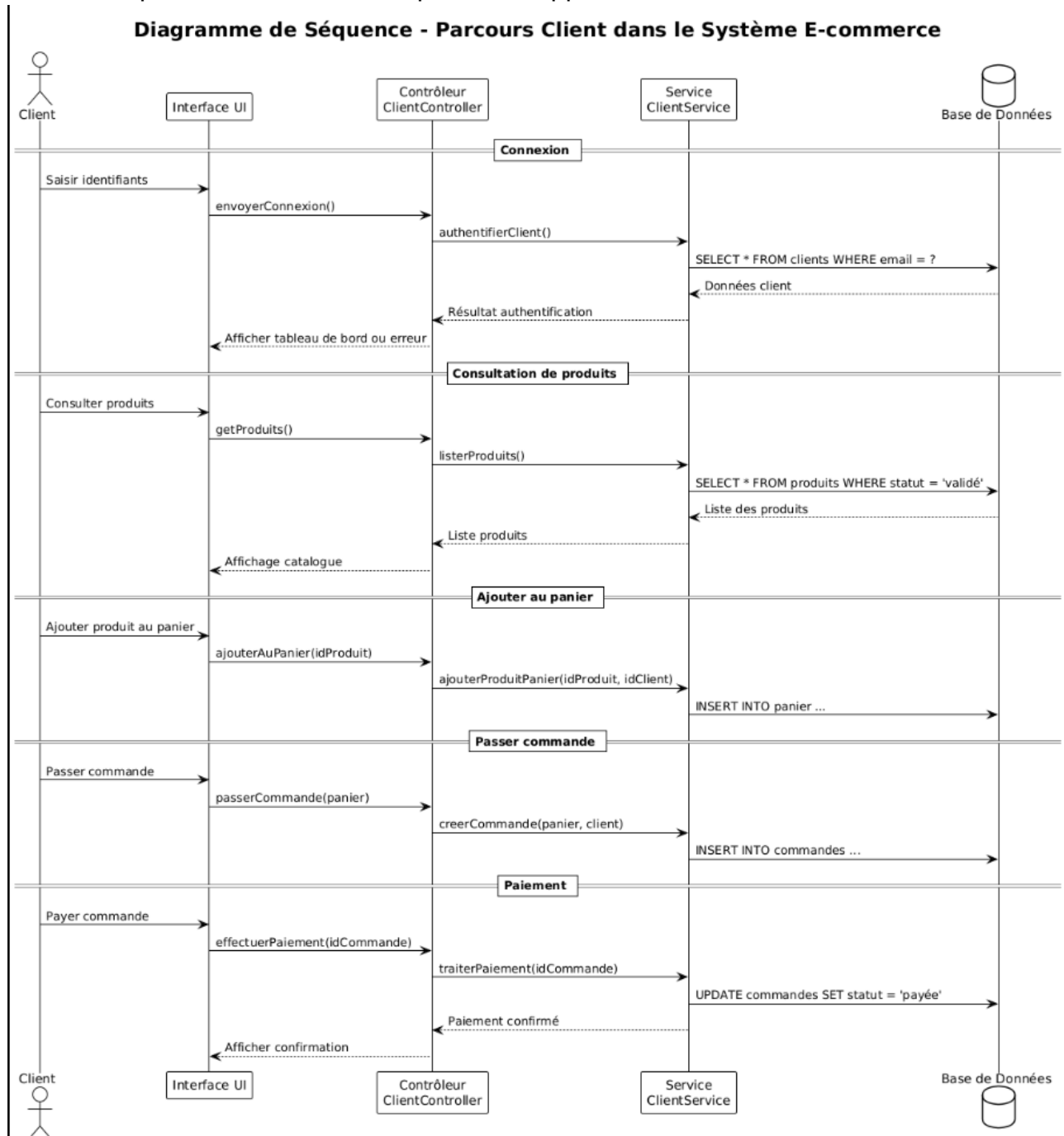


Figure 4 : Diagramme de séquence.

4.La base de données :

La base de données ordex_db fonctionne comme un système intégré pour gérer une application e-commerce ou de gestion de commandes, basé sur les informations dérivées du diagramme de classe impliqué. Elle repose sur sept tables interconnectées : app_role, categories, order_product, orders, products, user_roles, et users.

Le système commence par gérer les utilisateurs via la table users, qui stocke leurs informations personnelles et identifiants, tandis que app_role et user_roles définissent et attribuent des rôles (ex. : client, administrateur) pour contrôler les accès. Les produits disponibles sont organisés dans products, classés par catégories dans categories, permettant une navigation structurée. Lorsqu'un utilisateur passe une commande, celle-ci est enregistrée dans orders avec son identifiant, et order_product établit les liens détaillés entre les commandes et les produits, incluant les quantités. Ce flux permet un suivi efficace des transactions et une gestion centralisée des données, adaptée à une application web dynamique.

III. Conclusion :

En conclusion, la phase de conception et de mise en place de notre projet a été essentielle pour garantir son succès. Les différents éléments de conception abordés, tels que les diagrammes de cas d'utilisation, de classes et de séquence, ainsi que la base de données, ont joué un rôle central dans le développement de notre application. Grâce à une planification approfondie et une conception réfléchie, nous avons pu établir une vision claire du projet et identifier les étapes nécessaires pour sa réalisation. Ces éléments de conception ont servi de référence tout au long du processus de développement, nous permettant de rester sur la bonne voie et d'ajuster notre plan en fonction des besoins et des contraintes. Cette approche méthodique nous a permis de maximiser notre efficacité et de livrer une application fonctionnelle et conforme aux attentes. Pour l'avenir, nous envisageons d'explorer de nouvelles améliorations et extensions basées sur cette conception solide, afin de continuer à faire évoluer notre projet et à répondre aux besoins changeants de nos utilisateurs.

Chapitre 3 : Etude Technique Du Projet:

I. Architecture technique:

1-Modele Rest API :

Une **API REST (Representational State Transfer)** est une interface qui permet à différentes applications, même fonctionnant sur des systèmes d'exploitation distincts, de **communiquer entre elles** et d'**échanger des données** de manière simple et standardisée via le **protocole HTTP**. Elle facilite ainsi l'interopérabilité entre logiciels en exposant des services accessibles à distance.

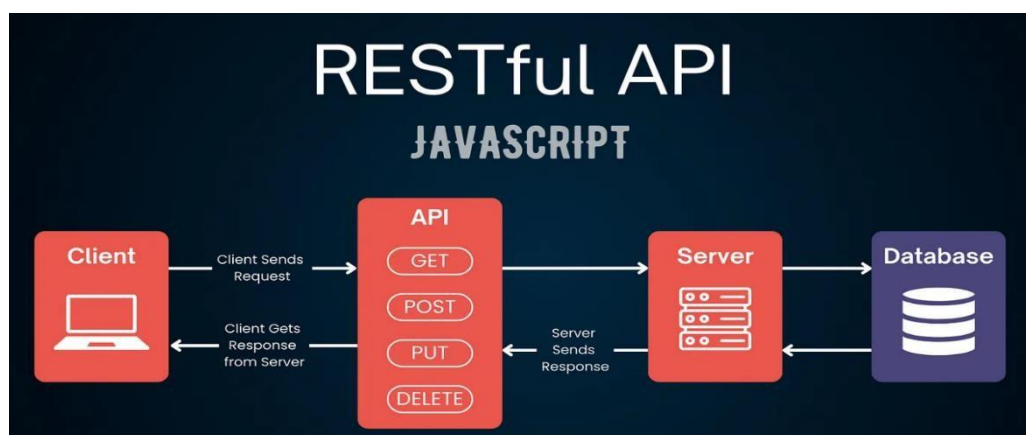


Figure 5 : Architecture REST API.

2- Spring Boot/Angular/MySQL:

Spring Boot / Angular / MySQL est une stack combinant trois technologies complémentaires : **Spring Boot** pour le développement backend en Java, **Angular** pour la création d'interfaces utilisateur dynamiques en TypeScript, et **MySQL** pour la gestion des données relationnelles. Cette combinaison permet de développer des applications web sécurisées, performantes et bien structurées.

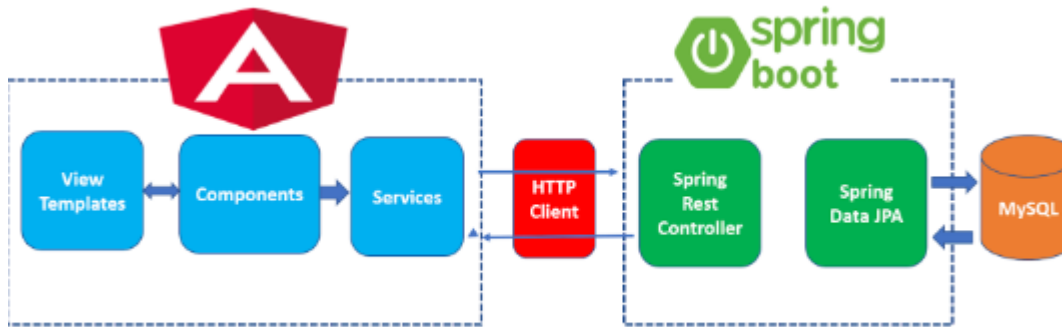


Figure 6: Architecture de combinaison Spring Boot/Angular/MySQL.

3-technologies et utiles utilisees:



MySQL est un système de gestion de base de données relationnelle (SGBDR) open source, basé sur le langage SQL (Structured Query Language). Il permet de stocker, organiser et interroger des données structurées selon un modèle relationnel. MySQL est développé par **Oracle Corporation** et est distribué sous une double licence : open source (GPL) et commerciale.



MySQL Workbench est un outil graphique officiel fourni par Oracle, permettant de modéliser, concevoir, interroger et gérer visuellement des bases de données MySQL.

Il facilite le déploiement, la gestion des connexions et l'exécution des requêtes SQL, tout en proposant des fonctionnalités de supervision, de conception E/R, et d'administration complète des bases de données locales ou distantes.



Spring Boot est un framework open source basé sur Java, qui simplifie la création d'applications backend robustes, évolutives et sécurisées.

Il est conçu pour faciliter le développement d'API REST, grâce à une configuration minimale, une intégration facile avec les bases de données, et un puissant écosystème Spring.

Spring Boot permet de créer des services web performants et prêts pour la production, avec un support intégré pour la sécurité, la validation, les tests, et plus encore.



Angular est un framework open source développé par Google pour la création d'applications web dynamiques côté client.



Tailwind est un framework CSS "utile d'abord" qui fournit un catalogue complet de classes et d'outils CSS qui vous permet de commencer facilement à styliser votre site Web ou votre application.



JSON Web Token est une norme Internet proposée pour la création de données avec une signature facultative et/ou un cryptage facultatif dont la charge utile contient JSON qui revendique un certain nombre de revendications. Les jetons sont signés à l'aide d'un secret privé ou d'une clé publique/privée.



Postman : Postman est un outil qui permet aux développeurs de tester et de documenter les API (Application Programming Interface). Il permet de créer des requêtes HTTP (GET, POST, PUT, etc.), de gérer les paramètres et les entêtes, de vérifier les réponses et les statuts de retour, et de sauvegarder

des requêtes pour une utilisation ultérieure. Il permet également de créer des collections de requêtes pour organiser et partager les tests avec d'autres développeurs. Il est disponible en tant qu'application de bureau et en tant qu'extension de navigateur.



StarUML est un outil de modélisation puissant utilisé pour la conception et la documentation de logiciels à l'aide de diagrammes UML (Unified Modeling Language). Il permet de créer divers types de diagrammes tels que les diagrammes de classes, de cas d'utilisation, d'activités, de séquences, et bien d'autres. Dans le cadre de ce projet, StarUML a été utilisé pour modéliser l'architecture logicielle, notamment les entités, leurs relations, ainsi que les interactions entre les différentes couches de l'application. Cet outil facilite la compréhension globale du système pour les développeurs et les parties prenantes, tout en garantissant une bonne structuration du code dès la phase de conception.



IntelliJ IDEA est un environnement de développement intégré (IDE) développé par JetBrains, conçu principalement pour le développement en Java. Il offre une interface moderne et complète pour créer des applications robustes, notamment avec le framework Spring Boot. Il dispose de nombreuses fonctionnalités comme l'autocomplétion intelligente, l'analyse de code, le débogage, la gestion de projets Maven ou Gradle, ainsi qu'une intégration avancée avec Git. IntelliJ IDEA est

disponible sur Windows, Linux et macOS, en version Community (gratuite) ou Ultimate (payante).



Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre et gratuit, créé en 2005 par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Le principal contributeur actuel de Git, et ce depuis plus de 16 ans, est Junio C Hamano. Depuis les années 2010, il s'agit du logiciel de gestion de versions le plus populaire dans le développement logiciel et web, qui est utilisé par des dizaines de millions de personnes, sur tous les environnements (Windows, Mac, Linux).



GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on Rails et Erlang par Chris Wanstrath, PJ Hyett et Tom PrestonWerner. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.



L'API Gmail est une interface proposée par Google permettant aux développeurs d'accéder aux fonctionnalités de Gmail de manière sécurisée via des applications. Elle permet d'envoyer, lire, supprimer ou gérer des messages, des brouillons, des libellés et des pièces jointes à l'aide de requêtes HTTP. L'API utilise le protocole OAuth 2.0 pour garantir une authentification sécurisée et respecte les autorisations accordées par l'utilisateur. Elle est souvent utilisée dans les applications web ou backend pour automatiser l'envoi d'e-mails ou interagir avec la boîte de réception Gmail.



EmailJS est un service en ligne qui permet d'envoyer des e-mails directement depuis des applications JavaScript sans avoir besoin de serveur backend. Il fonctionne en connectant votre frontend (comme Angular) à des services de messagerie (Gmail, Outlook, etc.) via des modèles personnalisés. EmailJS offre une interface simple et sécurisée pour gérer l'envoi de mails avec des données dynamiques, sans exposer de clés sensibles côté client. Il est très utilisé pour des formulaires de contact, des notifications, ou des confirmations automatiques dans les applications web.

II. Arborescente du projet:

1-Back-End :

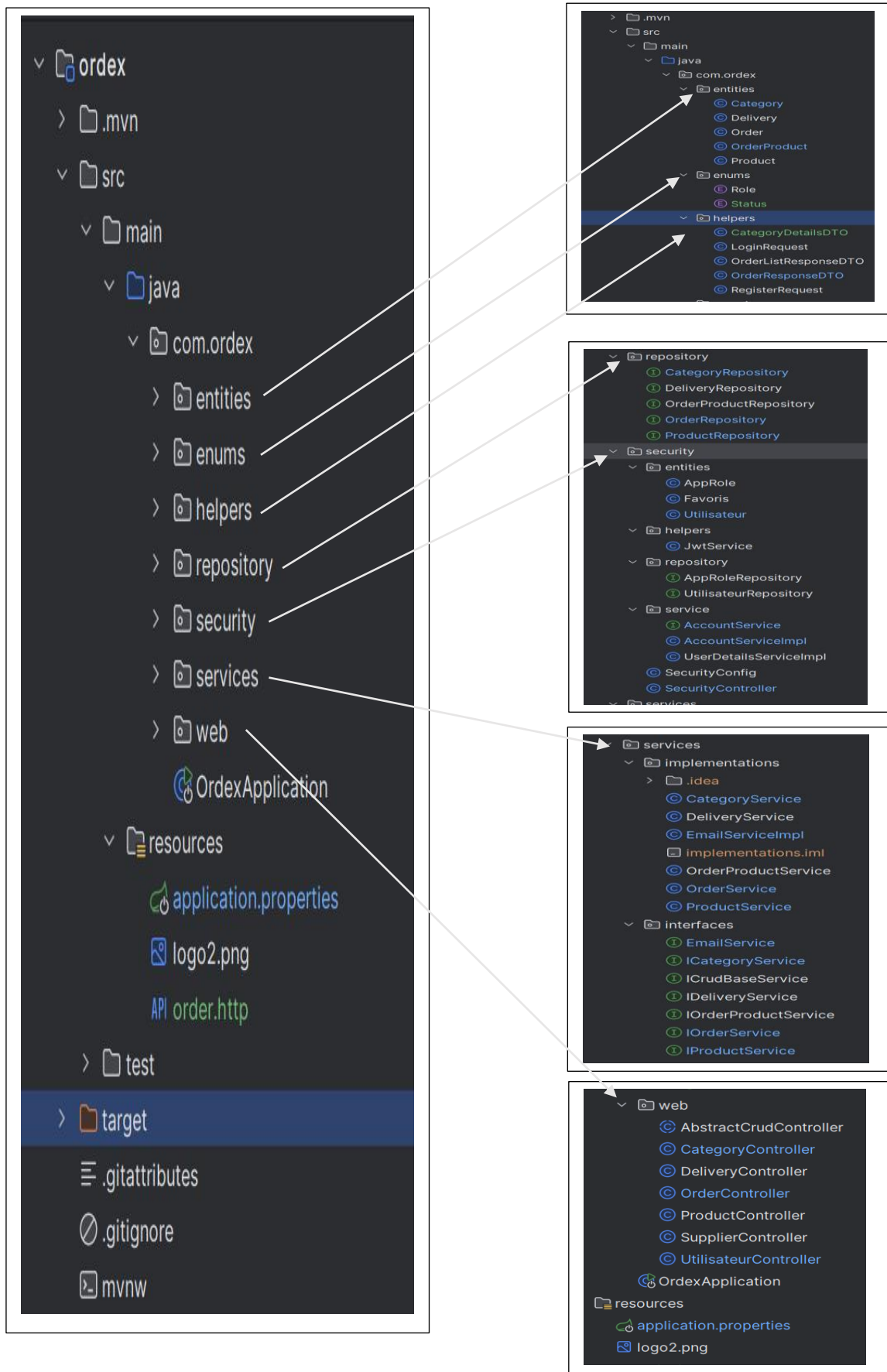


Figure 7 : Architecture du Back-End.

L'architecture du backend est organisée de manière modulaire pour favoriser la maintenabilité et la scalabilité. Elle comporte plusieurs dossiers clés :

- ✚ **entities** : Contient les classes de modèles de données comme Category, Delivery, Order, Product, etc. Ces entités sont mappées à la base de données à l'aide de JPA/Hibernate.
- ✚ **enums**: Définit des énumérations telles que Role et Status, utilisées pour représenter des constantes dans les entités et la logique métier.
- ✚ **helpers**: Regroupe les classes utilitaires et les DTOs (Data Transfer Objects) comme CategoryDetailsDTO, OrderListResponseDTO, ou RegisterRequest qui facilitent la communication entre les couches.
- ✚ **repository**: Contient les interfaces des repositories comme CategoryRepository, OrderRepository, UtilisateurRepository, etc. Ces interfaces héritent de JpaRepository et permettent les opérations CRUD sur les entités.
- ✚ **security**: Inclut les fichiers de routes (auth.routes.js, message.routes.js, user.routes.js, friendInvitation.routes.js) pour définir les points d'entrée de l'API pour l'authentification, les messages, et les utilisateurs.
- ✚ **socket** : Contient socket.js pour gérer la logique en temps réel avec les WebSockets.
- ✚ **utils** : Un dossier pour des utilitaires génériques.
- ✚ **server.js** : Le point d'entrée principal du serveur backend.
Cette structure modulaire permet une organisation claire et facilite les ajouts ou modifications de fonctionnalités.

2- Front-End :



Figure 8 : Architecture du Front-End.

L'architecture frontend de notre application de chat en temps réel est soigneusement structurée pour maximiser la modularité et la maintenabilité du code. Nous avons utilisé les technologies modernes de développement web, notamment React et les outils associés, pour construire une interface utilisateur réactive et performante. Voici un aperçu de l'organisation des fichiers et des répertoires principaux du projet frontend :

- + **assets:** Ce répertoire contient les ressources statiques utilisées par l'application.
- + **components:** Contient les composants organisés par domaine fonctionnel, comme: `admin-products` : pour la gestion des produits administrateur.
- + **categories** : pour l'affichage et la gestion des catégories.
- + **clients** : pour les interactions liées aux clients.
- + **contact** : pour la section contact avec ses fichiers `.css`, `.html` et `.ts`.
- + **dashboard** : pour la vue principale du tableau de bord.
- + **orders** : pour les commandes.
- + **models:** Contient les définitions des interfaces TypeScript, représentant les entités de l'application: `category.ts`, `categoryDetails.ts`, `delivery.ts`, `order.ts`, `order-product.ts`, `product.ts`, `user.ts`
- + **pages** : Contient les différentes pages de l'application (routées), classées par section :
 - + **admin-panel, supplier-panel** : interfaces dédiées aux administrateurs et fournisseurs.
 - + **checkout** : gestion du processus de paiement.
 - + **home** : page d'accueil.
 - + **login, register, mdpoublie** : pour la gestion de l'authentification.
 - + **verify-code, verify-register** : pour la vérification des comptes et codes.

- + services : Contient les services injectables Angular utilisés pour la logique métier et les appels HTTP :

abstract-crud.service.ts : un service générique pour les opérations CRUD.

auth.service.ts, user.service.ts, product.service.ts, order.service.ts, delivery.service.ts, client.service.ts, category.service.ts, cart.service.ts.

- + shared : Dossier destiné aux composants ou modules réutilisables dans toute l'application.

- + Fichiers racine Angular (dans src/) :

app.component.html, app.component.ts, app.config.ts, app.routes.ts : les fichiers principaux de l'application Angular.

- + main.ts : point d'entrée principal de l'application Angular.

- + index.html : le fichier HTML racine.

- + styles.css : le fichier global de style.

- + Autres fichiers et configurations :

- + tailwind.config.js, postcss.config.js : configuration de Tailwind CSS pour le design moderne.

- + angular.json, tsconfig*.json, package.json : fichiers de configuration Angular et des dépendances.

- + assets/ : contient les fichiers statiques comme favicon.ico.

Cette organisation modulaire permet de maintenir un code propre et bien structuré, facilitant ainsi les mises à jour et les ajouts de nouvelles fonctionnalités.

Chapitre 4 :Réalisation du projet:

1-Pages commune :

1.1 Accueil:

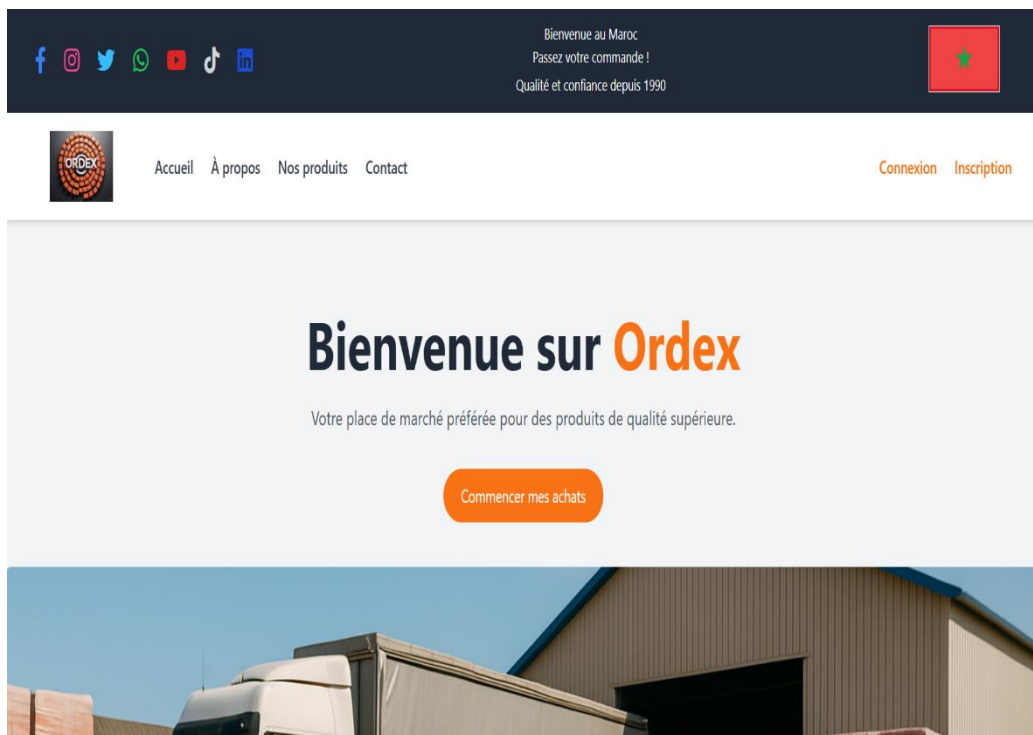


Figure 9 : Interface d'accueil.

La page d'accueil d'**Ordex** accueille les visiteurs avec une interface moderne, épurée et professionnelle. Elle met en avant le message principal de la plateforme : offrir des produits de **qualité supérieure** tout en inspirant **confiance** et **proximité**.

1.2 Connexion:



Connexion à Ordex

Nom d'utilisateur

Mot de passe

Se connecter

Vous n'avez pas de compte ? [Créer un compte](#)

Mot de passe oublié ? [Réinitialiser le mot de passe](#)

Figure 10: Interface de connexion.

Le formulaire de connexion de "**Ordex**" se compose de plusieurs éléments : un champ pour le "Nom d'Utilisateur" et un autre pour le mot de passe. En dessous des champs de saisie, un texte indique "Vous n'avez pas de compte?" avec un lien qui renvoie l'utilisateur à la page d'inscription.

1.3 Inscription:

1.3.1 Formulaire:



Connexion à Ordex

Nom d'utilisateur

Mot de passe

Se connecter

Vous n'avez pas de compte ? [Créer un compte](#)

Mot de passe oublié ? [Réinitialiser le mot de passe](#)

Figure 11: Interface d'inscription.

Le formulaire d'inscription de "**Ordex**" se compose de plusieurs éléments : un champ pour le "Nom d'utilisateur", un autre pour l'adresse e-mail, et un dernier pour le mot de passe. En dessous du bouton d'inscription, un texte invite les utilisateurs ayant déjà un compte à se connecter via un lien menant à la page de connexion.

1.3.2 Verification:



Vérifier le code d'inscription

Entrez le code de vérification que vous avez reçu par email pour confirmer votre inscription.

Code de vérification

Vérifier et confirmer

Vous n'avez pas reçu de code ? [Renvoyer un code](#)

Figure 12: Interface de verification d'inscription

Le formulaire de **vérification de compte** de "Ordex" permet à l'utilisateur de confirmer son inscription. Il comprend un champ pour saisir le **code de vérification** reçu par e-mail. En

dessous, un bouton permet de valider ce code. Si l'utilisateur n'a pas reçu le code, un lien est proposé pour **en demander un nouveau**.

1.3 Oubli de mot de passe:



Réinitialiser votre mot de passe

Entrez votre adresse email pour recevoir un code de réinitialisation de mot de passe.

Adresse email

Envoyer le code de réinitialisation

Vous vous souvenez de votre mot de passe ? **Se connecter**

Figure 13: Interface de réinitialisation de mot de passe.

Le formulaire de réinitialisation du mot de passe de "Ordex" permet aux utilisateurs de récupérer l'accès à leur compte. Il comporte un champ pour l'adresse email, afin d'envoyer un code de réinitialisation. Un bouton permet de lancer l'envoi. En bas de la page, un lien "Se connecter" est proposé pour ceux qui se souviennent de leur mot de passe.

2 Espace Admin:

2.1 Tableau de bords:

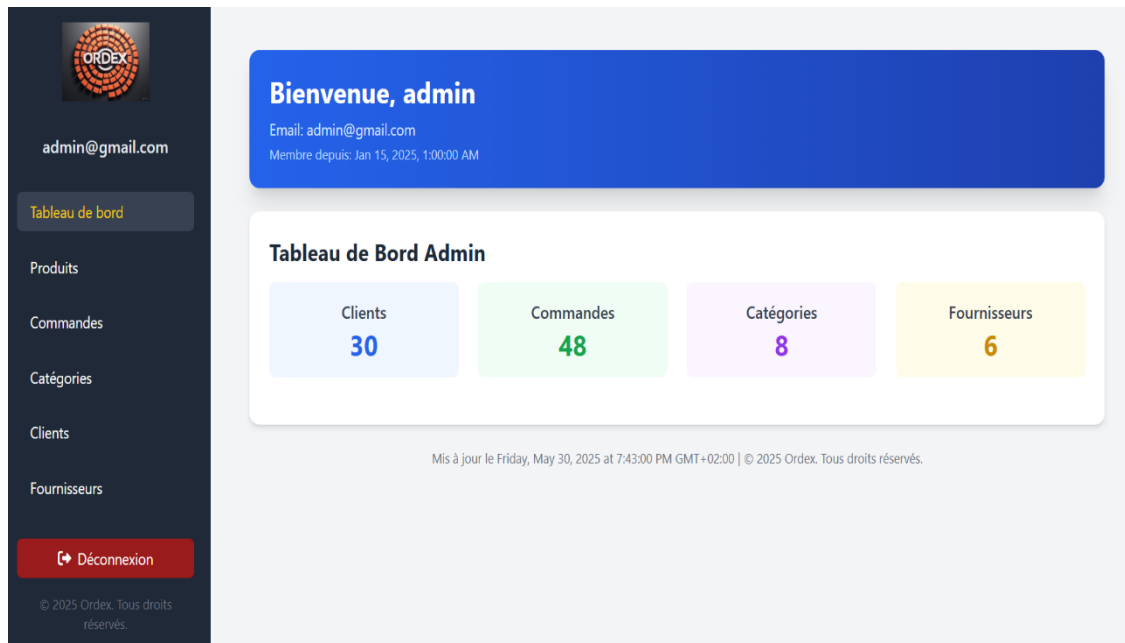


Figure 14: Interface tableau de bord pour admin.

Cette interface représente le **tableau de bord principal de l'administrateur** de la plateforme ORDEX. Elle a pour but de fournir une vue d'ensemble synthétique et claire sur les éléments clés de gestion : clients, commandes, catégories de produits et fournisseurs.

2.2 Interface de Gestion des Produits:

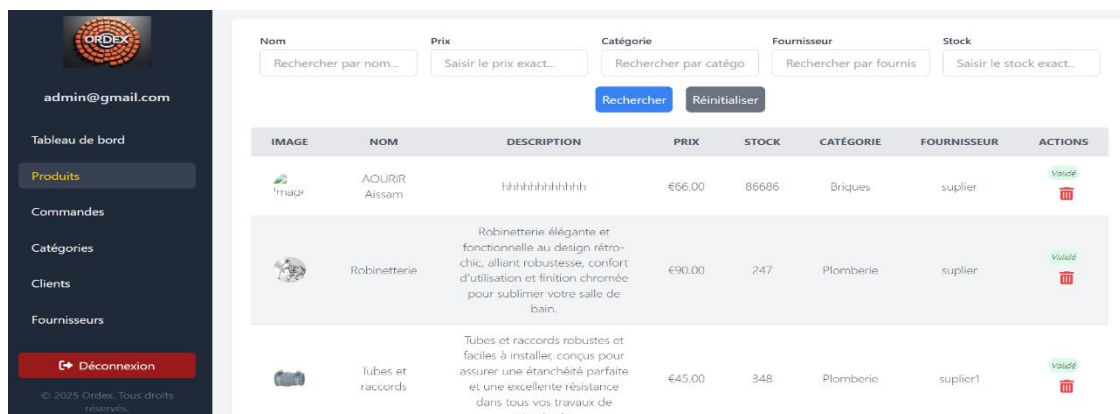


Figure 15: Interface de gestion des Produits.

Cette page permet à l'administrateur de **visualiser, rechercher, filtrer et gérer les produits** disponibles sur la plateforme ORDEX. Elle intègre des fonctionnalités avancées de filtrage et un tableau de visualisation claire et intuitive

2.3 Interface de Gestion des Commandes:

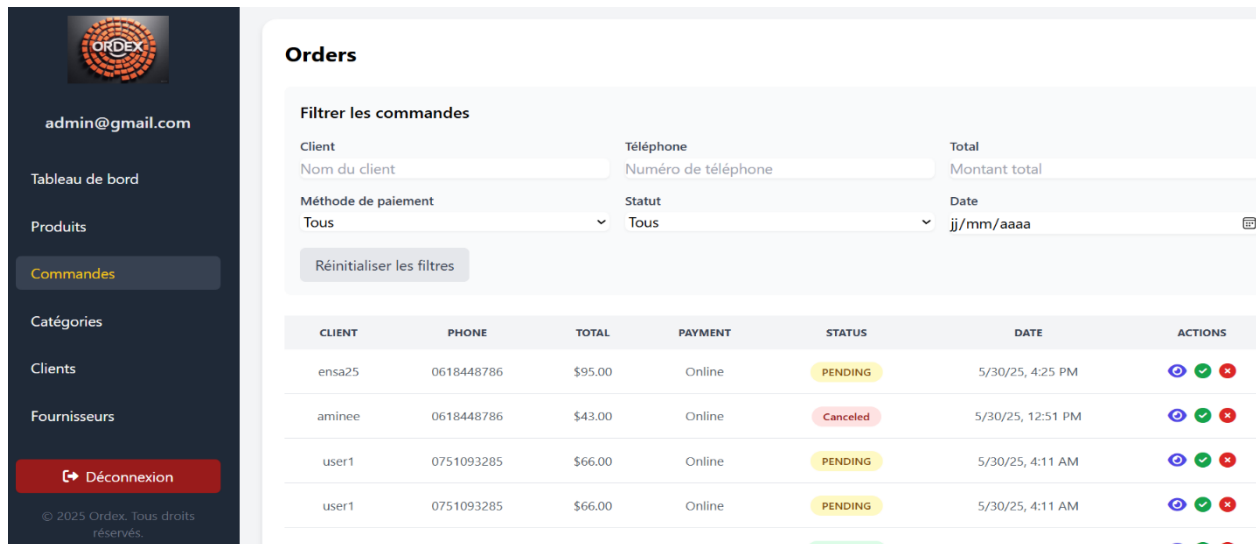


Figure 16: Interface de gestion des commandes.

Cette page permet à l'administrateur de **visualiser, filtrer, suivre et gérer les commandes** passées sur la plateforme ORDEX. Elle fournit des informations essentielles telles que le client, le montant, le statut et les actions à effectuer.

2.4 Interface de Gestion des Catégories de Produits:

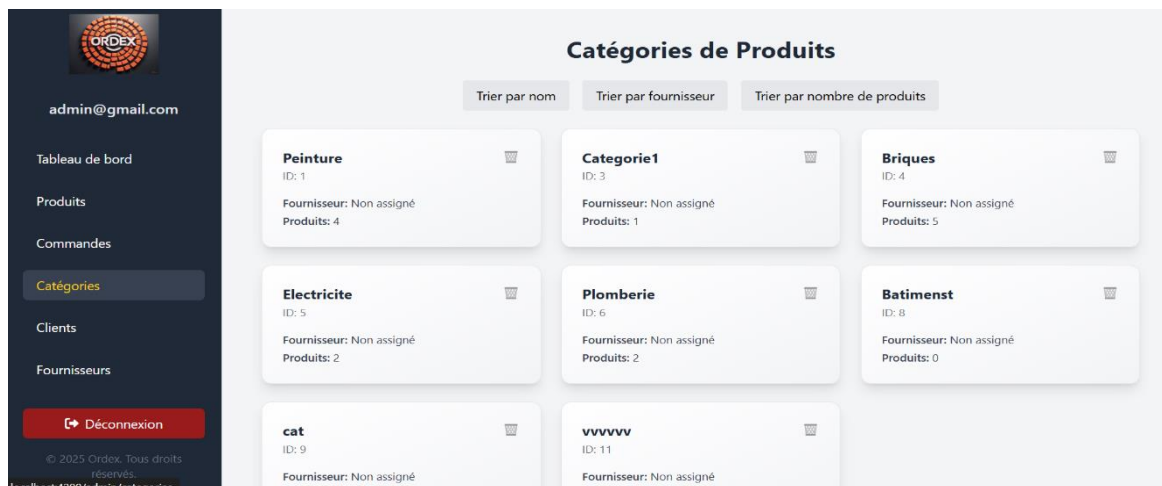


Figure 17: Interface de gestion des categories des produits.

Permettre à l'administrateur de **visualiser, trier et supprimer les catégories de produits** disponibles sur la plateforme. Chaque carte regroupe les informations clés d'une catégorie, y compris son fournisseur et le nombre de produits associés.

2.5 Interface de Gestion des Clients:

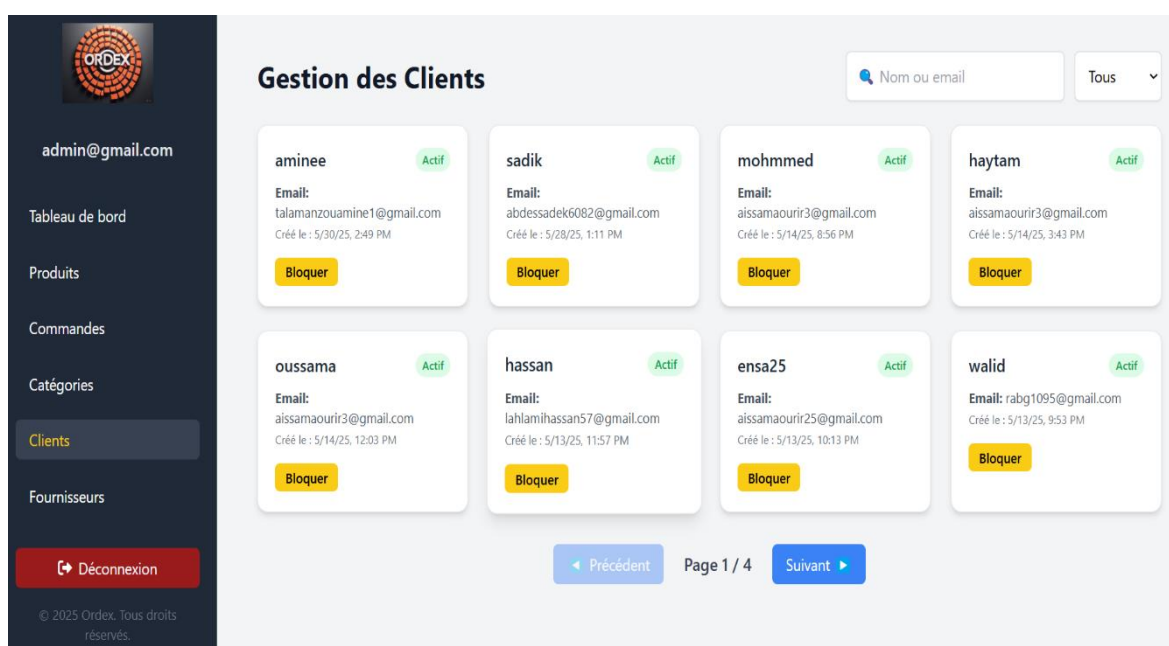


Figure 18: Interface de gestion des categories des produits

Cette interface permet à l'administrateur de consulter la liste des clients inscrits, avec leurs informations essentielles (nom, email, date d'inscription). Chaque client actif est affiché sous forme de carte claire et lisible, avec un bouton d'action permettant de le bloquer rapidement. Un système de recherche, de filtrage et de pagination facilite la navigation parmi les utilisateurs.

2.5 Interface de Gestion des Fournisseurs:

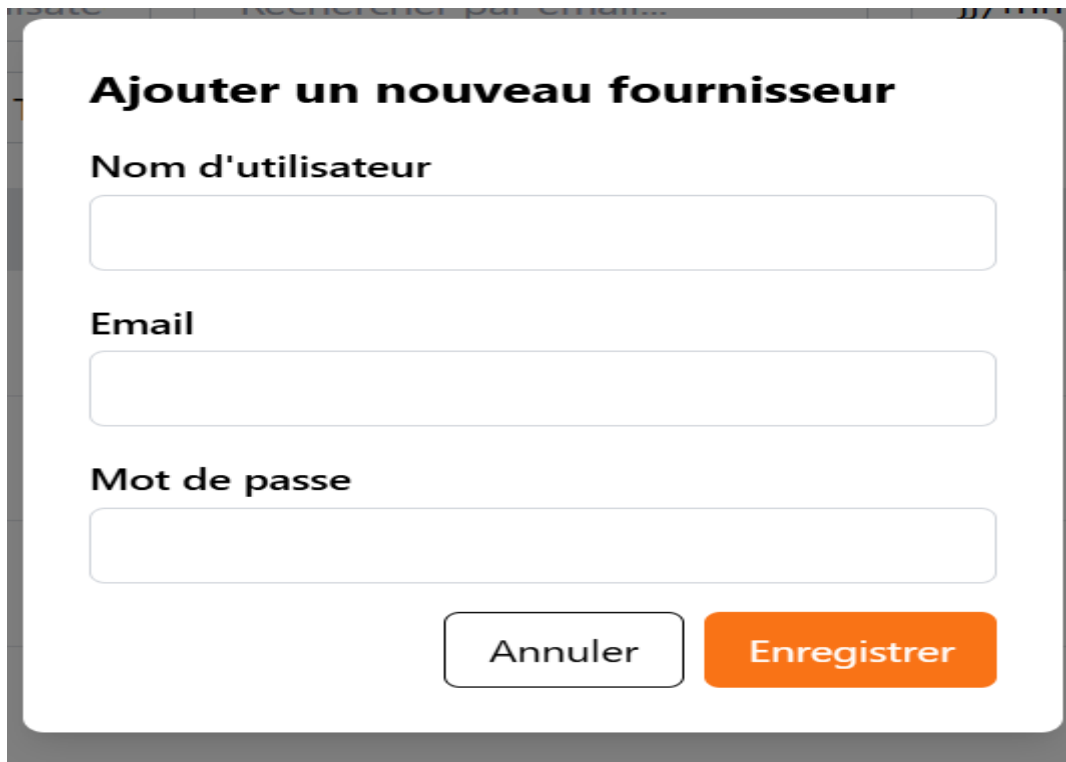
2.5.1: Interface d'affichage des fournisseurs.

NOM D'UTILISATEUR	EMAIL	DATE DE CRÉATION	ACTIONS
AOURIRENSA	aissamaourir3@gmail.com	May 20, 2025, 10:21:26 PM	🚫
suplier	aissamaourir33@gmail.com	May 6, 2025, 4:08:40 PM	🚫
suplier1	aissamaourir3@gmail.com	May 5, 2025, 8:07:10 PM	🚫
suplier2	aissamaourir2@gmail.com	May 6, 2025, 1:28:14 AM	🚫
suplier3	aissamaourir21@gmail.com	May 7, 2025, 5:57:46 PM	🚫
supplier	supplier@gmail.com	May 5, 2025, 7:45:31 PM	🚫

Figure 19: Interface de gestion des fournisseurs.

Cette interface permet à l'administrateur de visualiser, trier et gérer les comptes des fournisseurs de manière claire et efficace. Elle propose des filtres avancés (par nom, email, date et statut), un tri dynamique, et un bouton d'action pour bloquer un fournisseur si nécessaire. L'administrateur peut également ajouter rapidement un nouveau fournisseur via le bouton dédié.

2.5.2: Interface de creation des fournisseurs

The image shows a web form titled "Ajouter un nouveau fournisseur" (Add a new supplier). The form is contained within a light gray rounded rectangle. It has three input fields: "Nom d'utilisateur" (Username), "Email", and "Mot de passe" (Password). Each field is a white rectangle with a thin gray border. Below the fields are two buttons: "Annuler" (Cancel) in a white box with a gray border, and "Enregistrer" (Register) in an orange box. The text is in a clean, sans-serif font.

Ajouter un nouveau fournisseur

Nom d'utilisateur

Email

Mot de passe

Annuler Enregistrer

Figure 20: Interface de gestion des categories des produits

Ce formulaire permet à l'administrateur d'ajouter un nouveau fournisseur en saisissant son **nom d'utilisateur**, son **email** et un **mot de passe sécurisé**. L'interface est simple, intuitive et garantit une expérience fluide, avec des boutons clairs pour valider ou annuler l'opération.

3-Espace Fournisseur:

3.1 Interface de Gestion des Produits:

3.1.1 Interface d'affichage des Produits:

IMAGE	NOM	DESCRIPTION	PRIX	STOCK	CATÉGORIE	STATUT	ACTIONS
	Robinetterie	Robinetterie élégante et fonctionnelle au design rétro-chic, alliant robustesse, confort d'utilisation et finition chromée pour sublimer votre salle de bain.	\$90.00	247	Plomberie	Publié	
	Produit33	hhhhhhhhhh	\$66.00	86686	Briques	Publié	

Figure 21: Interface de gestion des produits par fournisseur.

Cette interface permet aux fournisseurs de gérer facilement leurs produits. Chaque ligne affiche les informations essentielles d'un produit : **image**, **nom**, **description**, **prix**, **stock disponible**, **catégorie**, **statut de publication** et **actions** (édition).

3.1.2 Interface de modification des produits:

Modifier le produit

Nom du produit	Catégorie
<input type="text" value="Robinetterie"/>	<input type="text" value="Plomberie"/>
Description	
<input type="text" value="Robinetterie élégante et fonctionnelle au design rétro-chic, alliant robustesse, confort d'utilisation et finition chromée pour sublimer votre salle de bain."/>	
Prix	Stock
<input type="text" value="90"/>	<input type="text" value="247"/>
URL de l'image	
<input type="text" value="https://www.lemag-arthurimmo.com/lemag-arthurimmo.com/wp-content/uploads/2017/0"/>	
<input type="button" value="Annuler"/> <input type="button" value="Enregistrer les modifications"/>	





Figure 22: Interface de modification de produits par un fournisseur.

Ce formulaire permet au fournisseur de **mettre à jour les détails d'un produit existant** de manière rapide et intuitive.

3.1.2 Interface de creation des categories:

Gérer les catégories

Ajouter

Peinture	
Categorie1	
Briques	
Electricite	
Plomberie	

Fermer

Figure 23: Interface de creation des categories par un fournisseur.

Cette interface permet au fournisseur de **gérer dynamiquement les catégories de produits** disponibles dans la plateforme.

3.1.2 Interface de creation des produits:

Créer un nouveau produit

Nom du produit

Catégorie

Description

Prix

Stock

URL de l'image

Annuler

Enregistrer le produit

Figure 24: Interface de creation des produits par un fournisseur.

Cette interface permet à un fournisseur de **renseigner un nouveau produit à intégrer dans la plateforme** de vente.

4 Espace des clients:

4.1 Interface pour afficher les commande sd'un client:

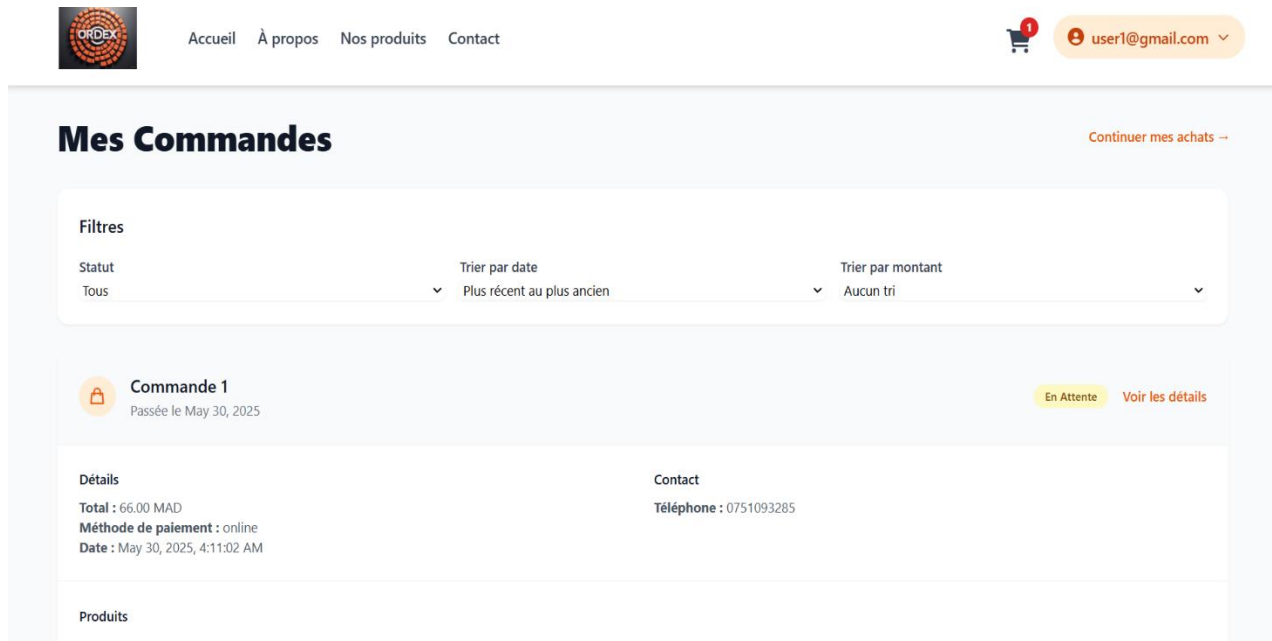


Figure 25: Interface d'affichage d'un client.

Cette page permet à le client connecté de consulter **l'historique de ses commandes passées** sur la plateforme ORDEX.

4.2 Interface pour afficher le profil d'un client:

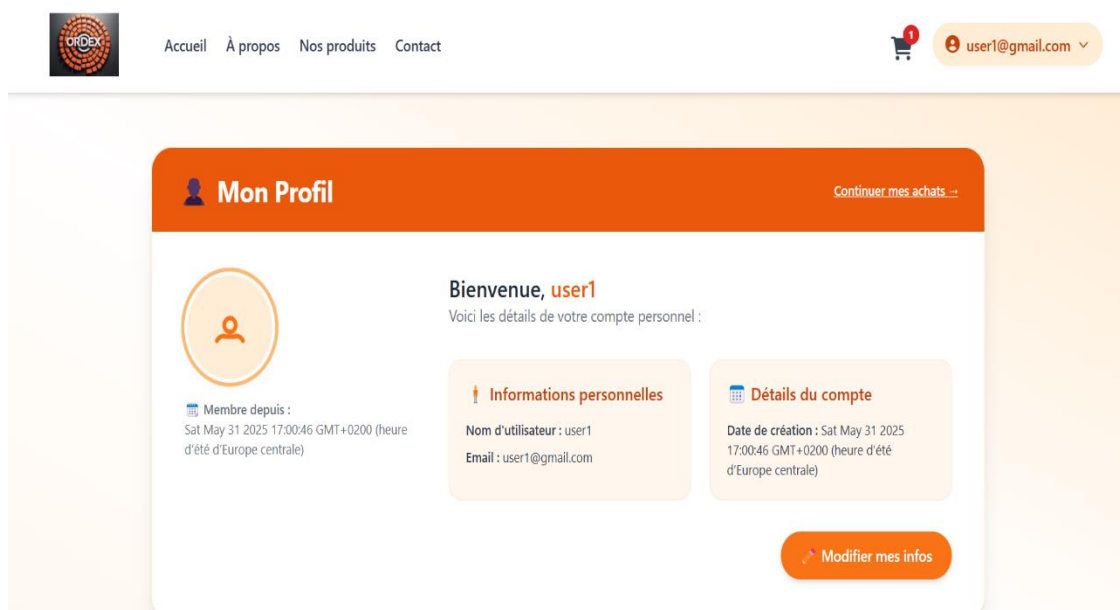


Figure 26: Interface d'affichage du profil d'un client.

Cette page permet l'affichage du profil d'un client sur la plateforme ORDEX.

Chapitre 5 : Synthèses et Orientation Futures:

I. Conclusion :

En conclusion, notre projet de développement d'une application web dédiée à la digitalisation du commerce des matériaux de construction et de bricolage constitue une solution innovante et adaptée aux besoins du marché marocain. En répondant aux enjeux de modernisation, de visibilité et d'efficacité commerciale, notre application permet aux commerçants de mieux gérer leurs activités tout en offrant aux clients une expérience d'achat fluide, rapide et intuitive.

Grâce à l'utilisation de technologies modernes telles que **Java**, **Spring Boot**, **Angular** et **MySQL**, nous avons mis en place une architecture robuste, modulaire et évolutive. Cette combinaison technologique nous a permis de construire une plateforme performante, sécurisée et facilement extensible, tant pour les vendeurs que pour les clients.

En résumé, notre solution représente une avancée majeure vers la digitalisation d'un secteur encore peu exploité numériquement, en apportant une réponse concrète aux besoins des commerçants et des consommateurs. Nous sommes convaincus que ce projet contribuera à améliorer la compétitivité des magasins marocains et à renforcer l'accessibilité aux produits de construction et de bricolage à travers tout le pays.

II. Perspectives :

Afin de renforcer davantage la valeur ajoutée de notre application web et de répondre de manière plus complète aux besoins du marché, plusieurs évolutions sont envisagées pour les prochaines phases du projet.

La première perspective concerne l'**intégration d'un module de gestion de la livraison**. Cette fonctionnalité permettra aux clients de commander les matériaux en ligne et de se faire livrer à domicile ou sur leur chantier. Elle inclura la sélection du mode de livraison, le suivi en temps réel du colis, ainsi que la gestion des délais et des coûts associés. Cela offrirait un service complet, depuis la consultation du produit jusqu'à sa réception, tout en augmentant la satisfaction client.

Par ailleurs, nous envisageons de mettre en place un système avancé d'**analyse des données de transaction**. Ce module permettra de collecter, structurer et visualiser les données relatives aux ventes, aux produits les plus demandés, aux périodes d'activité, ou encore aux comportements d'achat des clients. Ces informations seront mises à disposition des fournisseurs sous forme de tableaux de bord interactifs pour les aider à prendre des décisions stratégiques : ajustement des stocks, lancement de promotions ciblées, adaptation de l'offre aux préférences locales, etc.

En combinant **logistique intelligente** et **analyse de données**, notre solution évoluera vers une plateforme intelligente capable d'accompagner les commerçants dans l'optimisation de leur gestion et d'offrir aux clients un service fluide, rapide et personnalisé. Ces évolutions contribueront à faire de notre application un véritable levier de modernisation pour le secteur de la construction et du bricolage au Maroc.

III. References:

- Angular. (n.d.). *Angular documentation*. Angular. Consulté la première fois le 10 mars 2025, à l'adresse : <https://angular.io/docs>
- Spring. (n.d.). *Spring Boot Reference Documentation*. Spring.io. Consulté la première fois le 10 mars 2025, à l'adresse : <https://docs.spring.io/spring-boot/docs/current/reference/html/>
- Tailwind Labs. (n.d.). *Tailwind CSS Documentation*. Tailwind CSS. Consulté la première fois le 10 mars 2025, à l'adresse : <https://tailwindcss.com/docs>
- Oracle Corporation. (n.d.). *MySQL documentation*. MySQL. Consulté la première fois le 10 mars 2025, à l'adresse : <https://dev.mysql.com/doc/>
- Microsoft. (n.d.). *TypeScript Handbook*. TypeScript. Consulté la première fois le 10 mars 2025, à l'adresse : <https://www.typescriptlang.org/docs/>

- OpenJS Foundation. (n.d.). *Node.js documentation*. Node.js. Consulté la première fois le 10 mars 2025, à l'adresse : <https://nodejs.org/en/docs>
- Apache Software Foundation. (n.d.). *Maven Project Documentation*. Apache Maven. Consulté la première fois le 10 mars 2025, à l'adresse : <https://maven.apache.org/guides/>
- Bootstrap Team. (n.d.). *Bootstrap Documentation*. Bootstrap. Consulté la première fois le 10 mars 2025, à l'adresse : <https://getbootstrap.com/docs/>