[Date]

Rédigé et présenté par MOTSO Alix Audrey

[nom de la société]

Application de gestion d’un pressing

[Sous-titre du document]

# REMERCIEMENTS

Ce modeste travail n’aurait pu voir le jour sans l’aide et le soutien d’un grand nombre de personnes que nous souhaitons remercier. Tout d’abord, nous remercions le Seigneur tout

puissant, qui nous a donné la force et la santé pour la réalisation de ce projet. Nous pensons aussi à :

* Monsieur GUIMEZAP Paul, fondateur de l’Institut Universitaire de la Côte (IUC) pour les cadres et les infrastructures mis à notre disposition ;
* Monsieur LEUMOTIEU, encadreur académique pour son soutien et ses différents conseils ;
* Ma familles pour leur soutien inconditionnel tant moral que financier ;
* Tous nos camarades de classe pour leur soutien ;
* Tous les enseignants de l’institut universitaire de la côte (IUC) pour leur encadrement de qualité ;

Nous adressons également notre gratitude à tous ceux qui nous ont aidé de près ou de loin à la réalisation de ce rapport.

# ****SOMMAIRE****

[REMERCIEMENTS 1](#_Toc205328299)

[SOMMAIRE 2](#_Toc205328300)

[LISTE DE FIGURES 5](#_Toc205328301)

[LISTE DE TABLEAUX 6](#_Toc205328302)

[AVANT-PROPOS 7](#_Toc205328304)

[RESUME 11](#_Toc205328312)

[ABSTRACT 12](#_Toc205328313)

[INTRODUCTION 13](#_Toc205328315)

[CHAPITRE 1 : PRÉSENTATION DU PROJET 14](#_Toc205328316)

[I. DESCRIPTION (OU CADRAGE) DU PROJET 14](#_Toc205328318)

[II. SPÉCIFICATIONS FONCTIONNELLES 15](#_Toc205328319)

[III. SPÉCIFICATIONS NON-FONCTIONNELLES ET AUTRES CONTRAINTES 15](#_Toc205328320)

[IV. ESTIMATION DES RESSOURCES 16](#_Toc205328321)

[V. PLANIFICATION PREVISIONNELLE TEMPORELLE 17](#_Toc205328325)

[CHAPITRE 2 : ANALYSE ET CONCEPTION 19](#_Toc205328329)

[I. ANALYSE DU PROJET 19](#_Toc205328330)

[II. CONCEPTION DU PROJET 20](#_Toc205328331)

[CHAPITRE 3 : RESULTATS ET DISCUSSION 32](#_Toc205328333)

[I. ENVIRONNEMENT ET OUTILS DE DEVELOPPEMENT 32](#_Toc205328334)

[II. RESULTATS OBTENUS 36](#_Toc205328335)

[III. INTERPRETATION 40](#_Toc205328336)

[CONCLUSION 43](#_Toc205328337)

[BIBLIOGRAPHIE 44](#_Toc205328338)

# ****LISTE DE FIGURES****

[**Figure 1: Diagramme de Gantt** 17](file:///C:\Users\COMPUTER%20STORES\Desktop\App-Pressing\CHAPITRE%201%20rpt.docx#_Toc205382401)

[Figure 2: diagramme de classe 25](#_Toc205382402)

[Figure 3: diagramme de cas d'utilisation 26](#_Toc205382403)

[Figure 4: diagramme de sequence du processus de paiement 28](#_Toc205382404)

[Figure 5: diagramme d'activite pour la gestion d'une commande 30](#_Toc205382405)

[Figure 6: vs code 32](#_Toc205382406)

[Figure 7: html 32](#_Toc205382407)

[Figure 8: javascript 33](#_Toc205382408)

[Figure 9: sql 33](#_Toc205382409)

[Figure 10: tailwind css 33](#_Toc205382410)

[Figure 11: conposer 34](#_Toc205382411)

[Figure 12:wamp 34](#_Toc205382412)

[Figure 13: google chrome 34](#_Toc205382413)

[Figure 14: github 35](#_Toc205382414)

[Figure 15: interface administrateur 36](#_Toc205382415)

[Figure 16: interface administrateur 36](#_Toc205382416)

[Figure 17: interface client 37](#_Toc205382417)

[Figure 18: interface client 37](#_Toc205382418)

[Figure 19: interface du gerant 38](#_Toc205382419)

[Figure 20: page de connexion 39](#_Toc205382420)

# ****LISTE DE TABLEAUX****

[Table 1: tableau budgetaire 18](#_Toc205382291)

[Table 2: Analyse de l'existant 19](#_Toc205382292)

[**Table 3: Concept de base du diagramme de classe** 24](#_Toc205382293)

[**Table 4: concept de base du diagramme de cas d'utilisation** 26](#_Toc205382294)

[Table 5: test authentification des utilisateurs 40](#_Toc205382295)

[Table 6:test passer une commande 41](#_Toc205382296)

[Table 7: test gestion des statuts d'une commande 42](#_Toc205382297)

# AVANT-PROPOS

Dans le but d’assurer un développement durable et de fournir aux entreprises une main

d’œuvre compétente et compétitive dans divers domaines, le gouvernement Camerounais par le biais de Ministère de l’enseignement Supérieur a permis l’ouverture d’Instituts Privés

d’Enseignement Supérieur, donnant ainsi l’opportunité aux institutions privées, de contribuer à l’acquisition d’une formation académique et professionnelle en adéquation avec le monde professionnel.

C’est ainsi qu’est créé l’ISTDI (Institut Supérieur de Technologie et du Design Industriel) par arrêté N° 02/0094/MINESUP/DDES/ESUP du 13 septembre 2002 et autorisation d’ouverture N° 0102/MINESUP/DDES/ESUP du 18 septembre 2002. Située dans la région du littoral,

département du Wouri, arrondissement de Douala 5ème, au quartier Logbessou. L’ISTDI est ensuite érigée en Institut Universitaire de la Côte (IUC) par arrêté



*N°5/05156/N/MINESUP/DDES/ESUP/SAC/ebm, et comporte à ce jour quatre (4)*



*établissements notamment :*

**L’INSTITUT SUPERIEUR DE TECHNOLOGIES & DU DESIGN INDUSTRIEL (ISTDI) QUI FORME DANS LES CYCLES ET FILIERES SUIVANTS :**

* + **BTS Industriels :**
    - Maintenance des systèmes informatiques (MSI)
    - Informatique Industrielle (II)
    - Electrotechnique (ET)
    - Electronique (EN)
    - Froid et climatisation (FC)
    - Maintenance et après-vente automobile (MAVA)
    - Génie Civil (GC)
    - Génie Bois (GB)
    - Génie Logiciel (GL)
    - Chaudronnerie (CH)
    - Fabrication Mécanique (FM)

## LICENCES PROFESSIONNELLES INDUSTRIELLES et

**TECHNOLOGIQUES en partenariat avec l’université de Dschang :**

* + - Administration et Sécurité des Réseaux
    - Génie Logiciel
    - Automatique & Informatique Industrielle
    - Electrotechnique
    - Electronique Management des Services Automobiles
    - Maintenance et Expertise des Automobiles
    - Maintenance des Systèmes Industriels
    - Génie Energétique et Industriel
    - Génie Civil
    - Génie Bois
    - Génie Mécanique et Productique

## MASTER PROFESSIONNEL INDUSTRIEL en partenariat avec l’université de Dschang

* + - Génie Electrique & Informatique Industrielle Génie Télécommunications & Réseaux
    - Systèmes d’Information et Génie Logiciel

## L’Institut de Commerce et d’Ingénierie d’Affaires (ICIA) qui forme dans les cycles et filières suivantes :

* + **BTS Commerciaux :**
    - Assurance (AS)
    - Informatique de Gestion (IG)
    - Banque et Finance (BF)
    - Action Commerciale (ACO)
    - Commerce International (CI)
    - Communication d’Entreprise (CE)
    - Comptabilité et Gestion des Entreprises (CGE)
    - Logistique et Transport (LT)

## LICENCES PROFESSIONNELLES COMMERCIALES et GESTION en

**partenariat avec l’université de Dschang :**

* + - Marketing
    - Finance-Comptabilité
    - Banque
    - Banque - Assurance
    - Gestion des Ressources Humaines Logistique et Transport.
    - Contrôle et Audit
    - Gestion Qualité

## MASTER PROFESSIONNEL COMMERCIAL en partenariat avec l’université de Dschang

* + - Management des organisations
    - Finance-Comptabilité
  + **MASTER ISUGA-France en partenariat avec MBA France :**

1. **L’Institut d’Ingénierie Informatique d’Afrique Centrale (3IAC) qui forme dans les cycles et filières suivantes :**
   * **Cycle des TIC en partenariat avec CCNB-DIEPPE du Canada :**
     + Réseautique et sécurité
     + Programmation et Analyse

## MASTER EUROPEEN en partenariat avec 3IL en France :

* + - Génie Logiciel
    - Administration des Systèmes Réseaux

## MASTER PROFESSIONNEL en partenariat avec l’ENSP Yaoundé :

* + - Génie Civil
    - Génie Energétique et Environnement
    - Génie Industriel et Maintenance
  + **CYCLE INGENIEUR Informaticien (3IL-France)**
  + **Pôle de Recherche Innovation et Entreprenariat (PRIE)**

1. **Le Programmes Internationaux des Sciences et Technologies de l’Innovation (PISTI) qui forme dans les filières suivantes :**
   * **Cycle des Classes Préparatoires aux Grandes Ecoles d’Ingénieurs (CP) en partenariat avec l’université du Maine en France :**
     + Classes préparatoires
     + Licences Sciences & Techniques
   * **CYCLE INGENIEUR de Génie Industriel (ESSTIN-France)**

# ****RESUME****

Ce projet consiste à développer une plateforme de gestion destinée à un réseau de pressings. L’objectif principal est de digitaliser et centraliser les opérations courantes telles que le suivi des commandes, la gestion des stocks, la communication avec les clients et l’analyse des performances. Trois profils d’utilisateurs ont été identifiés : l’administrateur, les gérants de chaque pressing, et les clients. L’administrateur supervise l’ensemble du réseau, visualise des statistiques globales et gère les gérants. Chaque gérant dispose d’un contrôle total sur son pressing : il gère les commandes, envoie des notifications, échange avec les clients, et pilote les stocks. Les clients, quant à eux, reçoivent les notifications, suivent l’état de leurs commandes et peuvent dialoguer avec leur gérant. Le projet repose sur une architecture évolutive et modulaire, avec un souci particulier porté à l’ergonomie, la sécurité et la fiabilité du système. Il répond à une volonté d’amélioration continue de l’expérience client tout en optimisant les processus internes.

# ****ABSTRACT****

## This project aims to develop a management platform tailored for a network of dry-cleaning businesses. The main objective is to digitize and centralize daily operations such as order tracking, inventory management, customer communication, and performance analytics. Three user profiles are defined: administrator, branch manager, and customer. The administrator oversees the entire network, accesses global analytics, and manages the branch managers. Each manager has full control over their site, including order management, manual notifications, live chat with customers, and stock tracking. Customers receive notifications, passively follow the progress of their orders, and can chat directly with their manager. The project is built on a modular and scalable architecture, with a strong focus on usability, security, and system reliability. It supports a broader strategy to enhance customer experience while streamlining internal processes.

# INTRODUCTION

Dans un contexte de digitalisation croissante des services, la gestion moderne des pressings représente un enjeu stratégique pour garantir à la fois efficacité opérationnelle et satisfaction client. Les établissements de nettoyage à sec, souvent répartis sur plusieurs sites, font face à des défis liés à la coordination, au suivi des commandes, à la gestion des stocks, et à la communication entre les différents intervenants. Ce projet s’inscrit dans une démarche d’optimisation des processus métiers au sein d’un réseau de pressings. Il vise à concevoir une plateforme numérique centralisée, accessible aux administrateurs, gérants et clients, afin de fluidifier la gestion quotidienne, d’améliorer la qualité du service rendu et de renforcer l’expérience utilisateur. L’approche retenue repose sur l’analyse fine des besoins spécifiques de chaque acteur, suivie du développement d’un système modulaire, sécurisé et évolutif. Ce rapport présente l’ensemble des étapes de la réalisation du projet, depuis la définition du besoin jusqu’aux aspects techniques et organisationnels de sa mise en œuvre.

# ****CHAPITRE 1 : PRÉSENTATION DU PROJET****

# Le premier chapitre a pour objectif de présenter de manière globale le projet dans son contexte, en définissant ses finalités, les fonctionnalités attendues, les contraintes éventuelles et les ressources nécessaires à sa mise en œuvre. Il constitue une base de compréhension pour la suite du rapport et permet de cadrer le projet dans une vision structurée et réaliste.

## DESCRIPTION (OU CADRAGE) DU PROJET

Ce projet vise à concevoir et développer une plateforme numérique complète pour la gestion d’un réseau de pressings. Le contexte est celui d’un groupe d’enseignes de pressing réparties sur plusieurs sites, nécessitant un système centralisé de pilotage et de suivi des activités. La problématique à résoudre est de fluidifier la gestion des commandes, le suivi des clients, la gestion des stocks et la communication entre les différents acteurs (admin, gérants, clients) afin d’améliorer la qualité de service et l’efficacité globale.

Les objectifs visés sont :

* offrir un outil de gestion centralisé pour l’administrateur,
* permettre aux gérants de gérer efficacement leur pressing,
* apporter plus de transparence et d’information aux clients finaux.

L’intérêt de ce projet est d’optimiser les processus métier tout en améliorant l’expérience client et la rentabilité de l’organisation. Les attentes concernent notamment la livraison d’un système fiable, ergonomique, sécurisé, ainsi que des livrables documentés et testés.

## SPÉCIFICATIONS FONCTIONNELLES

Les principales fonctionnalités attendues sont :

**Pour l’administrateur (ADMIN) :**

* Gestion de plusieurs pressings (multi-sites)
* Consultation d’analytics globaux
* Gestion et affectation des gérants

**Pour les gérants :**

* Gestion des commandes et des statuts des vêtements
* Envoi manuel de notifications aux clients
* Module de chat avec les clients
* Gestion des stocks de produits de nettoyage
* Gestion du personnel (fonctionnalité à prévoir)
* Rapports et analytics locaux

**Pour les clients :**

* Réception des notifications envoyées par le gérant
* Suivi passif de l’état des commandes
* Chat en temps réel avec le gérant
* Accès à l’historique personnel des commandes

## SPÉCIFICATIONS NON-FONCTIONNELLES ET AUTRES CONTRAINTES

Le système devra répondre à plusieurs contraintes non fonctionnelles :

* **Ergonomie** : interface intuitive adaptée aux profils des utilisateurs
* **Sécurité** : protection des données personnelles et des transactions
* **Performance** : temps de réponse rapides, même en cas de forte charge
* **Fiabilité** : disponibilité élevée et tolérance aux pannes
* **Budget et délais** : à préciser dans le planning prévisionnel
* **Compatibilité** : utilisation sur ordinateur et appareils mobiles

## ESTIMATION DES RESSOURCES

**Ressources humaines :**

* 1 développeur

**Ressources matérielles :**

* Ordinateur personnel
* Accès à Internet

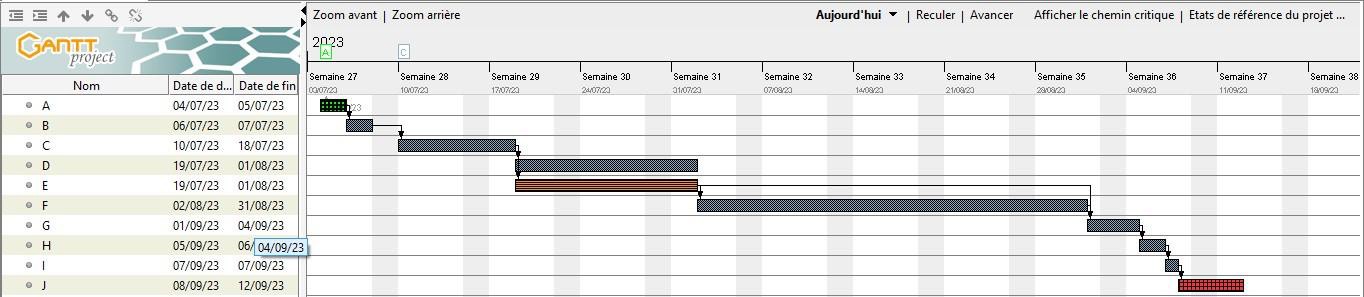
**Ressources logicielles :**

* Serveur local : XAMPP ou WAMP
* Éditeur de code : VS Code ou équivalent
* Navigateur web

## PLANIFICATION PREVISIONNELLE TEMPORELLE

Le diagramme de Gantt est un outil utilisé (souvent en complément d'un réseau PERT) en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet.

À partir de résultats obtenus du réseau PERT, plus les hypothèses sur la ressource disponible, on construit un planning (calendrier) sous forme de diagramme dont l’axe des abscisses représente le temps et l’axe des ordonnées représente les tâches.



**Figure 1: Diagramme de Gantt**

### Enveloppe budgétaire prévisionnelle

Tout projet, aussi petit soit-il, a un prix. Cependant, le coût d’un projet informatique n’est pas donné au hasard ; de nombreux facteurs sont pris en compte dans l’estimation de celui-ci. Pour déterminer le coût de notre solution, nous avons recensé les ressources nécessaires à sa réalisation.

Table 1: tableau budgetaire

| **Prestations** | **Description** | **Durée estimée** | **Montant (FCFA)** |
| --- | --- | --- | --- |
| **1. Analyse des besoins et conception** | Étude des exigences, maquettes UI/UX, architecture technique | 1 semaine | 40 000 |
| **2. Développement – Interface ADMIN** | Multi-pressing, gestion gérants, dashboards globaux | 2 semaines | 60 000 |
| **3. Développement – Interface GÉRANT** | Gestion commandes, notifications, chat client, stocks, rapports locaux | 3 semaines | 60 000 |
| **4. Développement – Interface CLIENT** | Suivi passif des commandes, chat, historique, notifications | 2 semaines | 40 000 |
| **5. Base de données & Sécurité** | Conception, sécurisation des données utilisateurs, connexions sécurisées | Parallèle (1 semaine) | 35 000 |
| **6. Tests, validation et débogage** | Tests fonctionnels, correction d’anomalies, optimisation | 1 semaine | 20 000 |
| **7. Déploiement & formation** | Hébergement, mise en production, formation de l’équipe du client | 1 semaine | 25 000 |

### ****III. Conditions générales****

* **Délai total estimé** : 6 à 8 semaines (selon retours du client).
* **Modalités de paiement** :
  + 40 % à la signature du contrat
  + 30 % après livraison de la version beta
  + 30 % à la mise en production
* **Maintenance évolutive** : possible sous forme de contrat mensuel ou au forfait.

### ****Montant total**** : ****280 000 FCFA****

# ****CHAPITRE 2 : ANALYSE ET CONCEPTION****

## ****I. ANALYSE DU PROJET****

#### **1. État de l’art et comparaison avec les solutions existantes**

De nombreuses solutions numériques pour la gestion de pressing existent sur le marché, notamment sous forme de logiciels SaaS (Software as a Service). Cependant, ces solutions sont souvent orientées vers des structures de grande taille et ne sont pas toujours adaptées aux petits pressings locaux. Voici une comparaison entre deux solutions populaires et la solution proposée dans ce projet :

Table 2: Analyse de l'existant

| **Critères** | **CleanCloud** | **Quick Dry Cleaning Software** | **ManohPressing** |
| --- | --- | --- | --- |
| **Type** | Application SaaS (cloud) | Logiciel à installer | Application Web en local ou hébergée |
| **Fonctionnalités principales** | gestion client, commandes, app mobile, livraison | Gestion stock, client, facturation, livraison | Gestion client, commande, interface client,multi-pressing, option tracking |
| **Suivi client** | Application mobile (clients) | App mobile ou SMS | Suivi en ligne via QR Code |
| **Prix** | À partir de 60 €/mois | À partir de 500 € licence + abonnements | Payement du logiciel et de la maintenance mensuel |
| **Accessibilité** | Nécessite connexion Internet | Option cloud ou locale | Fonctionne en local et compatible hébergement gratuit |
| **Personnalisation** | Faible (modèle SaaS figé) | Moyenne (modèles préconçus personnalisables) | Haute : développée sur mesure selon les besoins du client |
| **Langue et support** | Interface en anglais principalement | Interface multilingue selon versions | Interface en français, adapté au contexte local |

#### **2. Limites des solutions existantes**

* **Coût élevé** pour les pressings à petite échelle.
* **Fonctionnalités surdimensionnées** ou non pertinentes pour un petit pressing.
* **Peu de flexibilité** en termes de personnalisation ou d’évolution fonctionnelle.
* **Nécessité de connexion constante à Internet**, ce qui peut être contraignant en zone à faible débit.

#### **3. Valeur ajoutée de notre solution**

* **Simplicité d’utilisation** pour les employés non technophiles.
* **Suivi client innovant** par QR Code, sans application mobile requise.
* **Indépendance** vis-à-vis des prestataires ou abonnements mensuels.
* **Adaptabilité complète** aux besoins du bénéficiaire (fonctionnalités modulables).
* **Déploiement rapide** et coût nul.

## II. CONCEPTION DU PROJET

1. **METHODOLOGIE CHOISIE**

La méthodologie **Agile** est une approche de gestion de projet centrée sur l’adaptabilité, la collaboration et la livraison continue de valeur. Contrairement aux méthodes traditionnelles dites « en cascade », Agile privilégie des cycles courts appelés **sprints**, permettant d’itérer rapidement, de recueillir les retours du client en cours de développement, et d’ajuster les priorités en fonction des besoins réels. Cette méthode favorise une meilleure communication entre l’équipe et le client, une grande réactivité face aux changements, et une livraison progressive de fonctionnalités testées et opérationnelles.

Pour le développement de l'application web de gestion de pressing, la **méthodologie Agile** a été adoptée. Cette approche est particulièrement adaptée aux projets de taille moyenne nécessitant une **implication régulière du client** (ici, le gérant du pressing) et une **livraison progressive** des fonctionnalités.

#### **Principes appliqués de l’Agile dans ce projet :**

1. **Développement itératif et incrémental** : le projet a été découpé en **sprints hebdomadaires**, chacun produisant une version partielle mais fonctionnelle de l'application.
2. **Collaboration étroite avec le client** : le bénéficiaire a été impliqué à la fin de chaque sprint pour valider les fonctionnalités livrées, ajuster les besoins ou suggérer des améliorations.
3. **Réactivité au changement** : certains ajustements ont été intégrés en cours de développement (par exemple, format du QR Code ou présentation du reçu) sans remettre en cause la structure globale du projet.
4. **Livrables fréquents** : au lieu d’attendre la fin du projet pour livrer un produit complet, chaque sprint a livré une partie utilisable
5. **Tests continus et retours rapides** : après chaque sprint, des tests ont été effectués pour garantir la stabilité, la sécurité et l’utilisabilité des modules livrés.

### ****Pourquoi cette méthode est adaptée à ce projet ?****

* Le périmètre fonctionnel est **modulable et bien découpé**.
* Le client n’a pas de cahier des charges strict, ce qui permet **des ajustements en cours de route**.
* Le développement est réalisé par une **seule personne**, ce qui facilite l’agilité et la gestion des priorités sans bureaucratie.
* L’objectif est d’obtenir un **produit immédiatement exploitable**, même s’il reste perfectible.

**2. CHOIX TECHNOLOGIQUES :**

**Serveur de développement**

Pour ce projet, nous avons utilisé **XAMPP**, une solution tout-en-un permettant de simuler un environnement serveur local. XAMPP inclut notamment :

* Une **application multiplateforme**, compatible avec Windows, Linux et macOS,
* Le **serveur Apache**, qui permet d’héberger l’application localement pendant le développement,
* Le moteur **MySQL**, utilisé pour gérer la base de données et exécuter des requêtes SQL côté serveur.  
  Cela nous a permis de tester l’application Web dans un environnement de développement local, sans avoir besoin d’un serveur distant.

**Modélisation**

La conception du système a été réalisée à l’aide de **Power AMC**, un outil de modélisation permettant de créer les différents **diagrammes UML** (cas d’utilisation, classes, séquences, etc.), nécessaires à la structuration et à la compréhension du système avant le développement.

**Éditeur de code**

Le développement a été effectué à l’aide de **Visual Studio Code**, un éditeur de texte léger, polyvalent et largement utilisé pour les projets web et mobiles. Il offre de nombreuses extensions facilitant le codage, le débogage et la gestion du projet.

**Langage**

L’application a été développée en utilisant les langages **HTML**, **CSS**, **JavaScript** et **PHP**.

* **HTML** a servi à structurer les pages web,
* **CSS** à les styliser et les rendre visuellement attractives,
* **JavaScript** à gérer l’interactivité côté client,
* **PHP** à assurer la logique côté serveur, la communication avec la base de données et le traitement des formulaires.  
  Ce choix de technologies classiques mais robustes permet d’obtenir une application Web fonctionnelle, compatible avec la majorité des navigateurs, et facilement déployable.

**3.MODELISATION UML**

La modélisation UML (Unified Modeling Language) est une méthode normalisée permettant de représenter visuellement l’architecture, les comportements et les interactions d’un système informatique. Elle facilite la compréhension du fonctionnement d’une application avant son développement.

Dans le cadre de ce projet, UML a été utilisé pour structurer les besoins fonctionnels et techniques à travers des diagrammes tels que :

* le **Diagramme de cas d’utilisation**
* le **Diagramme de classes**
* le **Diagramme de séquence**
* le **Diagramme de composants**
* le **Diagramme de déploiement**
* le **Diagramme d’objets**
* le **Diagramme de packages**
* le **Diagramme de structure composite**

Cette étape de modélisation a permis de clarifier la conception du système, de prévenir les erreurs de logique, et de guider le développement de manière structurée. Afin d’assurer une conception claire et structurée de l’application, plusieurs types de diagrammes UML ont été réalisés. Chacun permet de visualiser un aspect spécifique du système, de sa logique fonctionnelle à son organisation technique.

Les principaux diagrammes présentés dans les sections suivantes sont :

* **Le diagramme de cas d’utilisation**, qui illustre les différentes fonctionnalités accessibles par les utilisateurs selon leur rôle (administrateur, gérant, client).
* **Le diagramme de classes**, qui définit la structure des données et les relations entre les différentes entités du système.
* **Le diagramme de séquence**, qui décrit le déroulement chronologique des interactions entre les composants lors d’un scénario précis (ex. : validation d’une commande ou envoi de notification).

Ces représentations visuelles ont servi de base tout au long du développement, en assurant une compréhension partagée entre le concepteur et les différentes parties prenantes du projet.

* 1. **Diagramme de Classe**

**Table 3: Concept de base du diagramme de classe**

|  |  |
| --- | --- |
| **Elément du diagramme** | **Description** |
| **Classe** | Modèle de donnée qui définit la structure commune à tous les éléments qui lui appartiennent. |
| **Classe d’association** | Classe qui fait partie d’une relation d’association entre deux autres classes. |
| **Cardinalité** | Permet de représenter le nombre minimum et maximum d’instance qui sont autorisées à  participer à la relation. |
| **Héritage** | Mécanisme qui permet lors de la création d’une nouvelle classe (classe fille) d’y inclure les caractéristiques d’une autre (classe mère) |
| **Agrégation** | Relation de contenant à contenu ou la  destruction du contenant n’engendre pas celle du contenu (couplage faible). |
| **Composition** | Relation de contenant à contenu ou la destruction du contenant engendre  automatiquement celle du contenu (couplage  fort). |

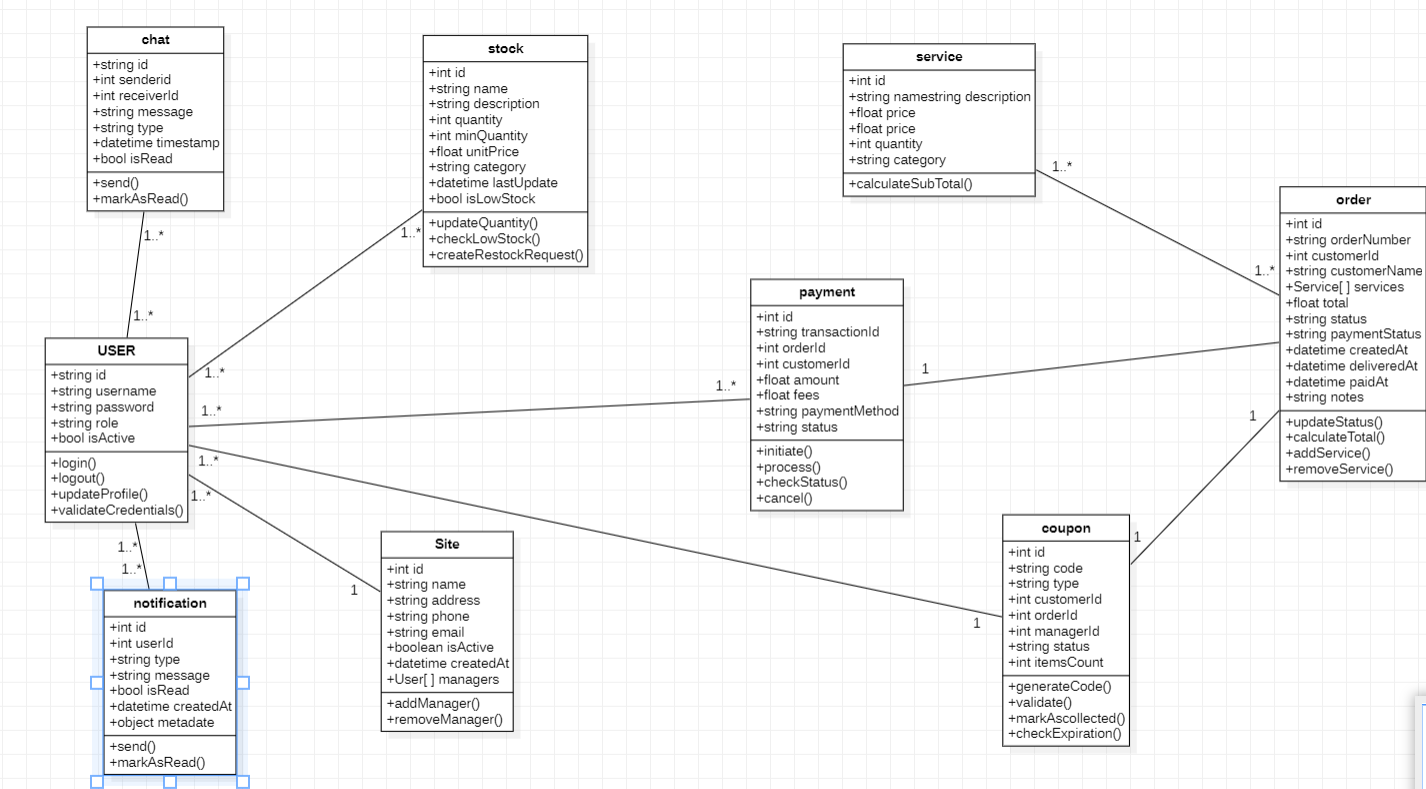


Figure 2: diagramme de classe

* 1. **Diagramme de Cas d’Utilisation**

**Table 4: concept de base du diagramme de cas d'utilisation**

|  |  |  |
| --- | --- | --- |
| **Elément** | **Description** | **Représentation graphique** |
| **Acteur** | Toute entité qui interagit avec le système | |
|  |  | **Acteur** |
| **Cas d’utilisation** | Cas  Fonctionnalité réalisable par le d’utilisation  système | |
| **Association** | Lien entre un acteur et les cas  d’utilisation auxquels il a  accès | |

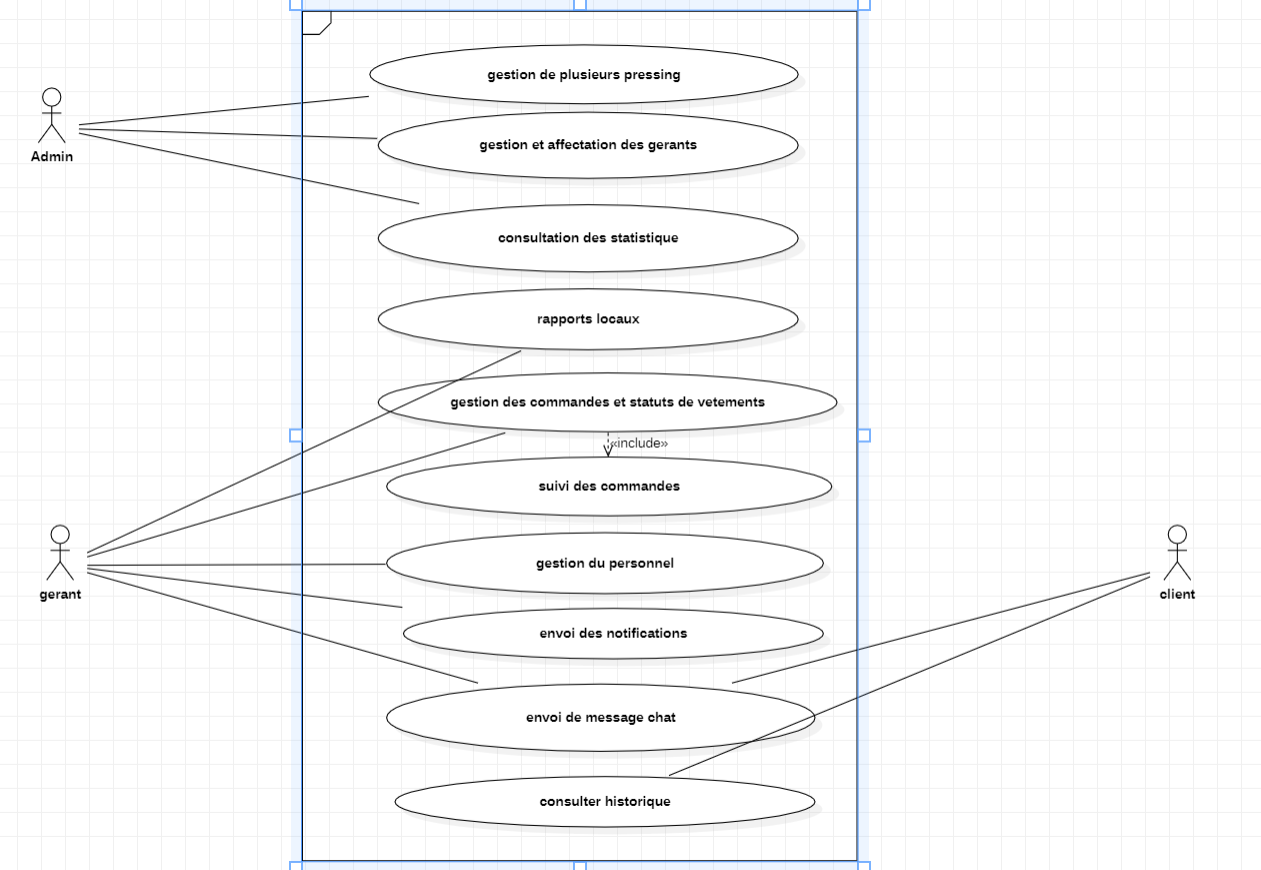


Figure 3: diagramme de cas d'utilisation

**3. diagramme de sequence**

Tableau : concept de base du diagramme de sequence

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Eléments du diagramme** | **Description** | **Représentation graphique** | | | |
| **Objet** | Il s’agit des instances de classe | |  |  |  |
| **Acteur** | Toute entité qui interagit avec le système | |  |  |  |
| **Ligne de vie** | Ligne verticale qui identifie l’existence de l’objet par  rapport au temps | |  |  | |
|  | | | | | |
|  | | | |  | |
| Boite rectangulaire sur une | | |  |  |  |
| **Activation** ligne de vie marquant la durée | | | |  | |
| d’une action | | | | | |
| **Message** | Flèche horizontale portant un texte qui indique les  communications objet-acteur  et objet-objet | |  |  |  |

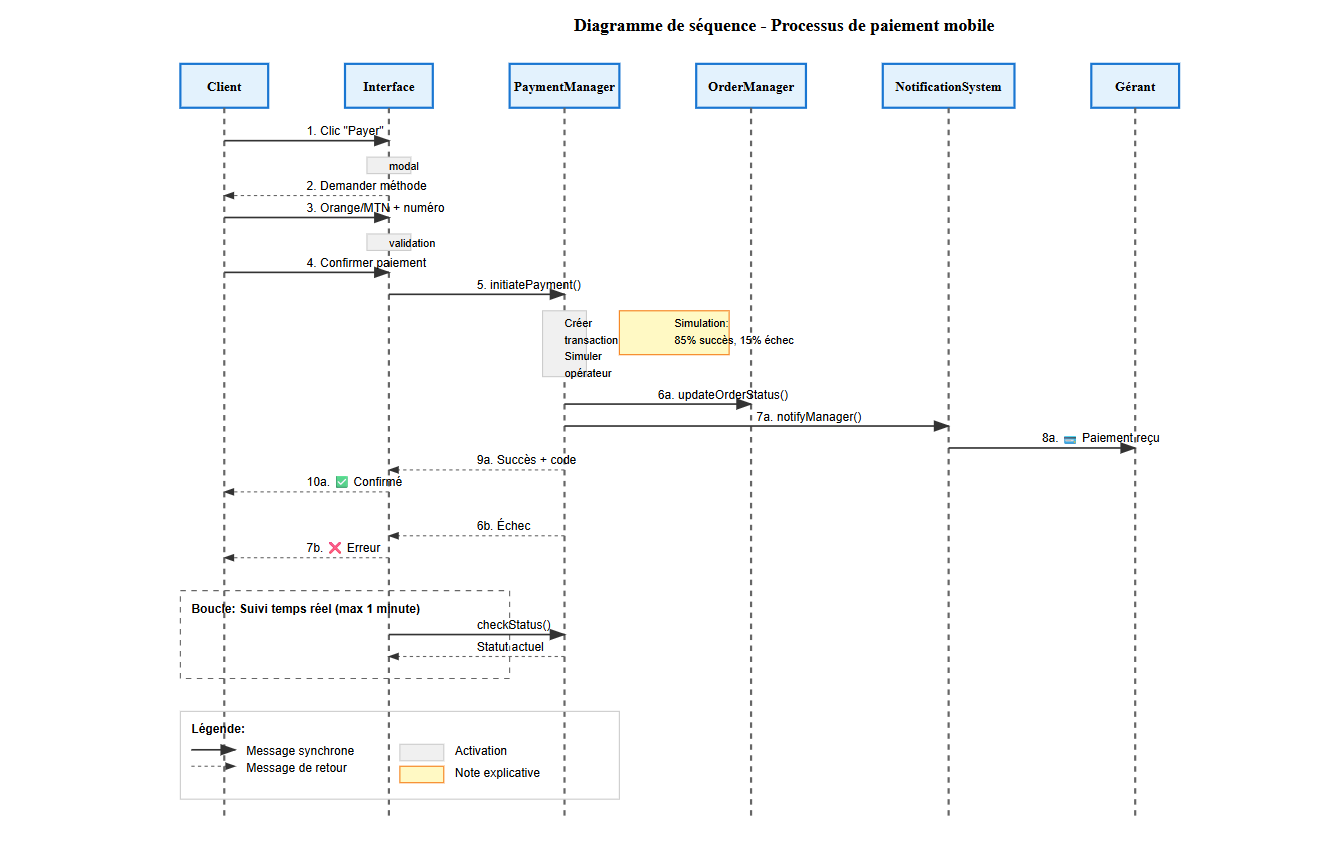
****

Figure 4: diagramme de sequence du processus de paiement

* 1. **Diagramme d’activité**

Tableau 1: concept de base du diagramme d'activité

|  |  |
| --- | --- |
| **Elément** | **Représentation** |
| Nom de l’activité  **Activité** | |
| **Transition** | |
| Décision  **Nœud de décision** | |
| **Nœud de début** | |
| **Nœud de fin** | |

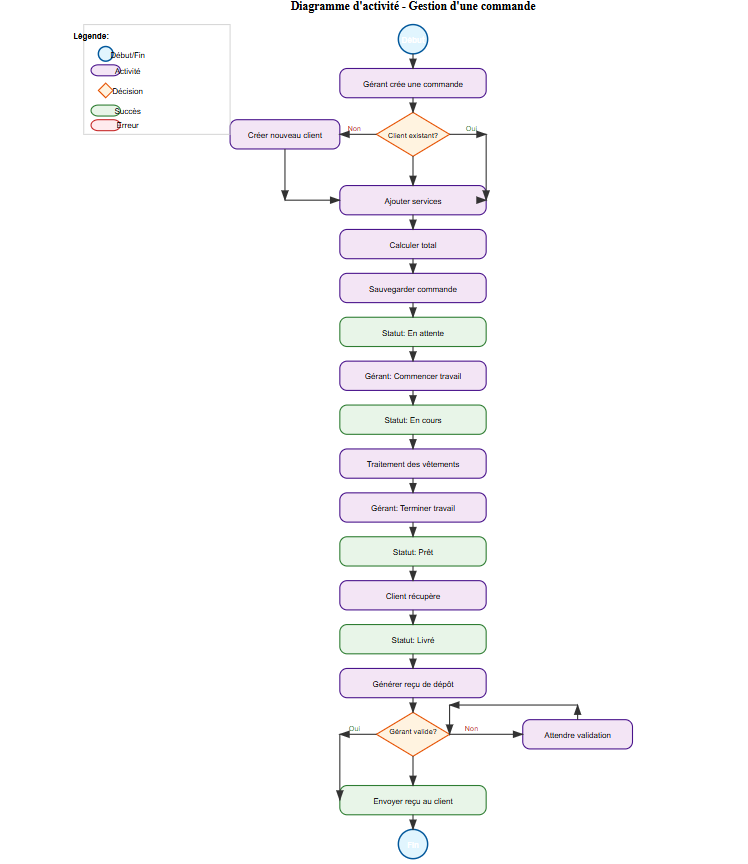


Figure 5: diagramme d'activite pour la gestion d'une commande

**4. ARCHITECTURE DE L’APPLICATION :**

L’architecture **MVC** (Modèle – Vue – Contrôleur) est un modèle de conception logicielle très utilisé dans le développement d’applications web. Elle permet de **séparer les responsabilités** d’un programme en trois couches distinctes :

* **Modèle (Model)** : il représente la **logique métier** et la gestion des **données** (lecture, écriture, traitement). C’est lui qui interagit avec la base de données.
* **Vue (View)** : elle correspond à la **partie visible de l’application**, c’est-à-dire l’interface utilisateur. Elle affiche les données fournies par le modèle.
* **Contrôleur (Controller)** : il sert d’**intermédiaire** entre la vue et le modèle. Il reçoit les actions de l’utilisateur, les traite, puis met à jour le modèle et/ou la vue en conséquence.

Cette séparation rend l’application **plus claire, modulaire et maintenable**, car chaque couche à un rôle précis et indépendant des autres.

**5. CONDITIONS DE BON FONCTIONNEMENT :**

Pour garantir le bon fonctionnement de l'application, certaines conditions doivent être réunies. Tout d'abord, le serveur local doit être opérationnel afin d'assurer le bon déroulement des échanges entre les différentes composantes du système.

Ensuite, la base de données doit être correctement initialisée, avec toutes les tables, relations et données nécessaires prêtes à l'emploi. Cela permet d'éviter les erreurs lors de l'accès ou de la manipulation des informations.

Enfin, il est recommandé d'utiliser un navigateur web récent et à jour. Cela garantit une compatibilité optimale avec les technologies utilisées dans l'application, ainsi qu'une meilleure sécurité et performance lors de la navigation.

# ****CHAPITRE 3 : RESULTATS ET DISCUSSION****

## ENVIRONNEMENT ET OUTILS DE DEVELOPPEMENT

## EDITEURS DE CODE

## Vs code

## 

Figure 6: vs code

## Visual Studio Code, souvent appelé VS Code, est un éditeur de code source gratuit développé par Microsoft. Il est largement utilisé par les développeurs web pour écrire et organiser leur code dans des langages comme HTML, JavaScript, PHP ou encore Tailwind CSS. Grâce à son interface simple, rapide et extensible, il propose une large variété d’extensions utiles, telles que l’auto complétion, le débogage, le contrôle de version avec Git, ou encore l'intégration avec des serveurs locaux. C’est un outil central dans le quotidien des développeurs.

## Les LANGAGES

## 

Figure 7: html

## HTML, ou HyperText Markup Language, est le langage fondamental pour créer la structure d’une page web. Il permet de disposer les éléments comme les titres, les paragraphes, les images ou les liens, en les organisant dans une hiérarchie claire. On peut le comparer au squelette d’un site web, car sans HTML, aucun contenu ne pourrait s’afficher dans le navigateur.

## 

Figure 8: javascript

## JavaScript est le langage de programmation qui rend les pages web interactives. Il permet de créer des fonctionnalités dynamiques comme des menus déroulants, des formulaires qui réagissent sans recharger la page, des animations, ou encore des jeux. JavaScript est exécuté directement dans le navigateur, ce qui le rend indispensable pour améliorer l’expérience utilisateur.

## 

Figure 9: sql

SQL, ou Structured Query Language, est le langage utilisé pour communiquer avec les bases de données relationnelles. Il permet d’enregistrer, modifier, consulter ou supprimer des données. Par exemple, lorsqu’un utilisateur se connecte à un site, SQL permet de vérifier ses informations dans la base de données. Il est souvent utilisé avec PHP pour créer des systèmes de gestion d’utilisateurs, de contenu, ou de produits.

## FRAMEWORK

## 

Figure 10: tailwind css

## Pour styliser les éléments HTML, de plus en plus de développeurs utilisent Tailwind CSS. Il s’agit d’un framework CSS utilitaire qui permet de créer rapidement des interfaces modernes et réactives en utilisant des classes prédéfinies directement dans le code HTML. Plutôt que d’écrire des feuilles de style séparées, Tailwind permet de construire le design au fur et à mesure, avec des classes . Cela améliore la productivité et la cohérence du design, tout en simplifiant la maintenance du code.

## 

Figure 11: conposer

## Composer est un gestionnaire de dépendances pour le langage PHP. Il permet d’installer, de mettre à jour et de gérer automatiquement les bibliothèques nécessaires à un projet. Par exemple, au lieu d’écrire soi-même une fonction complexe, on peut utiliser une bibliothèque existante via Composer, comme une solution d’authentification ou un outil de gestion des requêtes SQL. Composer facilite donc la modularité et l’organisation du code dans les projets PHP modernes.

## LOGICIEL SERVEUR

## 

Figure 12:wamp

## WAMPP est un logiciel qui permet de créer un serveur web local sur son propre ordinateur. Il regroupe plusieurs services essentiels comme Apache (le serveur web), MySQL (la base de données), et PHP (le langage de script). Avec WAMPP, on peut développer et tester des sites dynamiques en local, sans avoir besoin d’héberger le site sur internet. C’est un outil très pratique pour le développement PHP et SQL.

## Le NAVIGATEUR

## 

Figure 13: google chrome

## Google Chrome est un navigateur web rapide et populaire, qui permet d’afficher et de tester des sites internet. Il est particulièrement apprécié par les développeurs pour ses outils de développement intégrés. Grâce à l’inspecteur, on peut analyser le code HTML, tester des modifications en temps réel, surveiller les erreurs JavaScript, ou mesurer les performances d’un site.

## GITHUB

## 

Figure 14: github

## GitHub est une plateforme de partage et de gestion de code qui repose sur le système de versionnement Git. Elle permet de stocker son code en ligne, suivre les modifications, collaborer avec d’autres développeurs, et gérer différentes versions d’un projet. GitHub est aujourd’hui indispensable dans les projets professionnels, car il facilite le travail en équipe et garantit une meilleure organisation du code source.

## 

## RESULTATS OBTENUS

Nous avons mis sur pied la première version de notre application, avec fonctionnalités fars réalisées .

## Interface administrateur

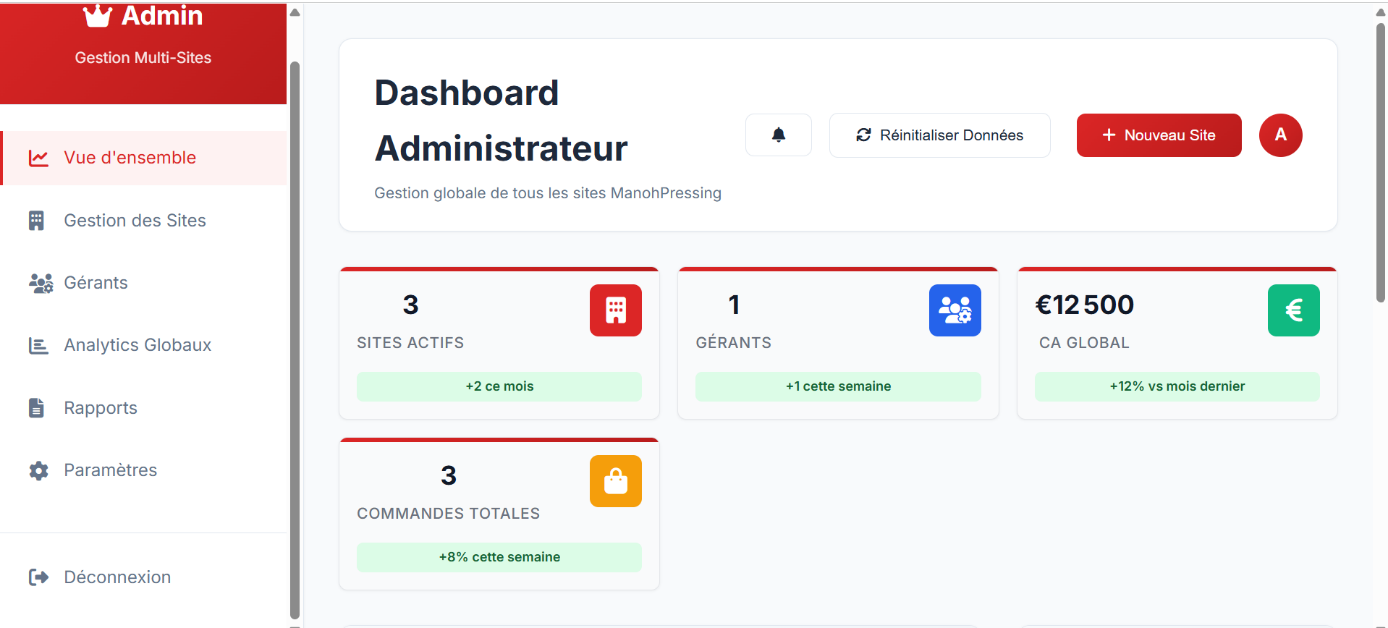


Figure 15: interface administrateur

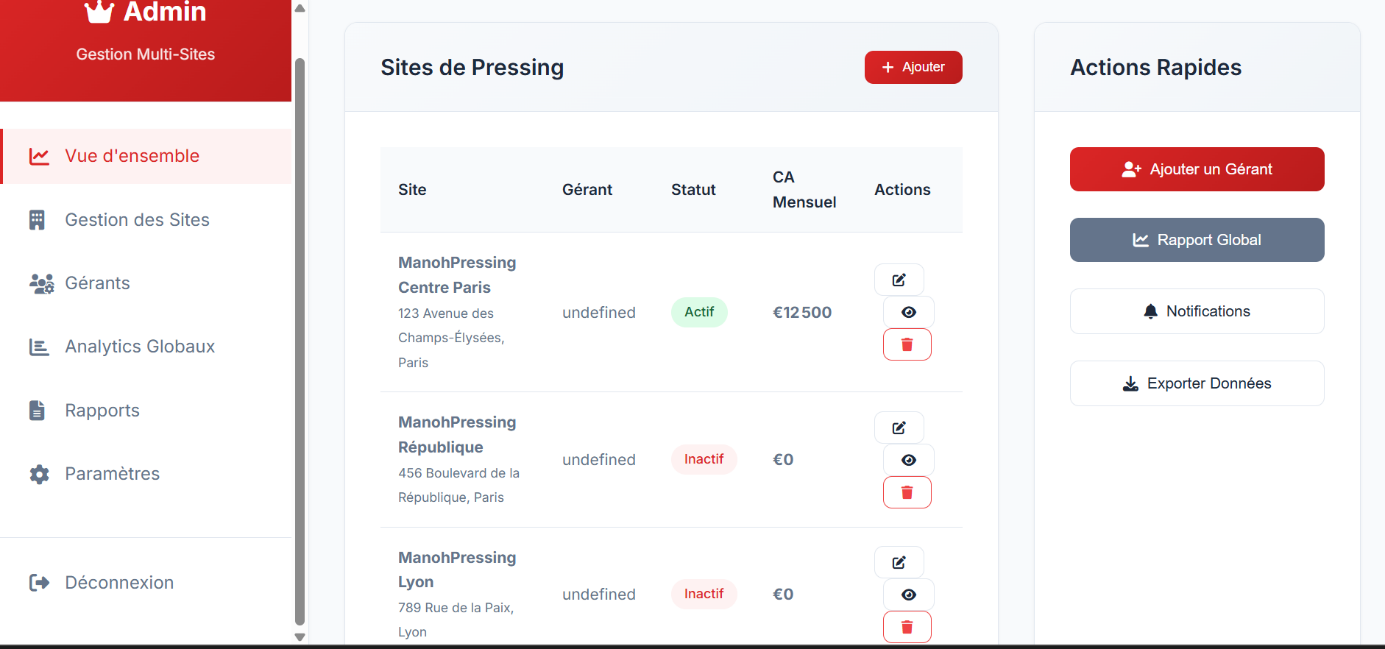


Figure 16: interface administrateur

1. **Interface client**

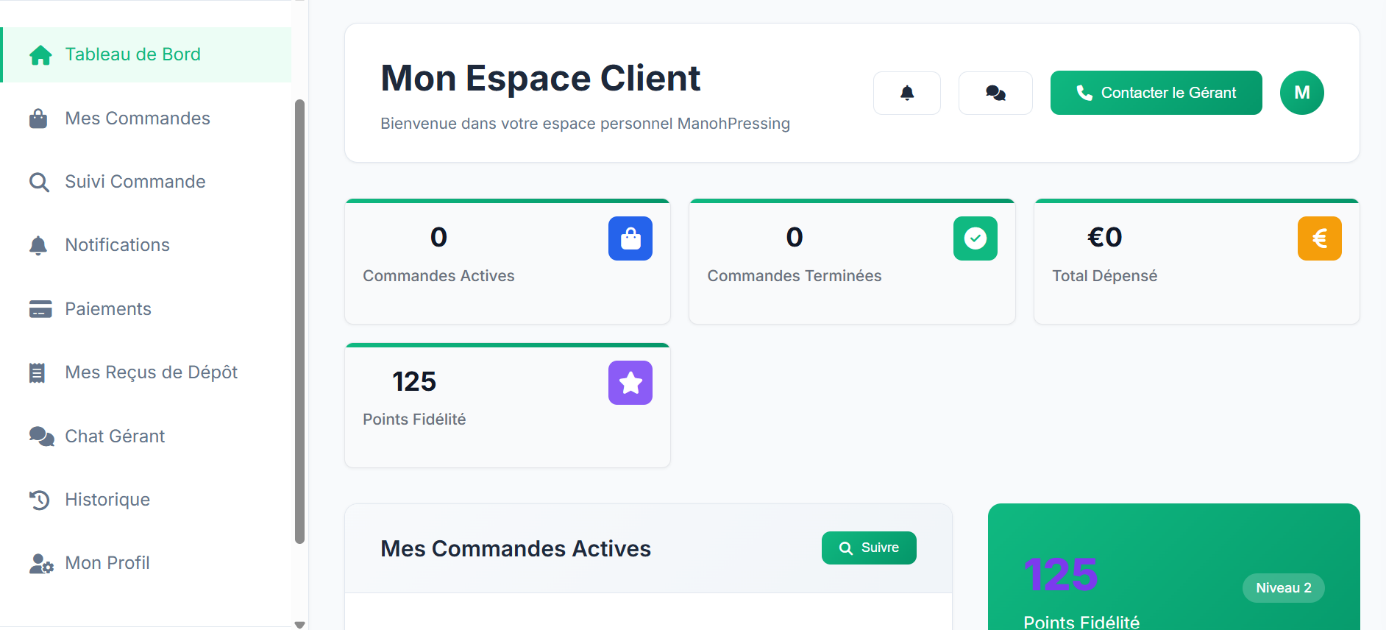


Figure 17: interface client

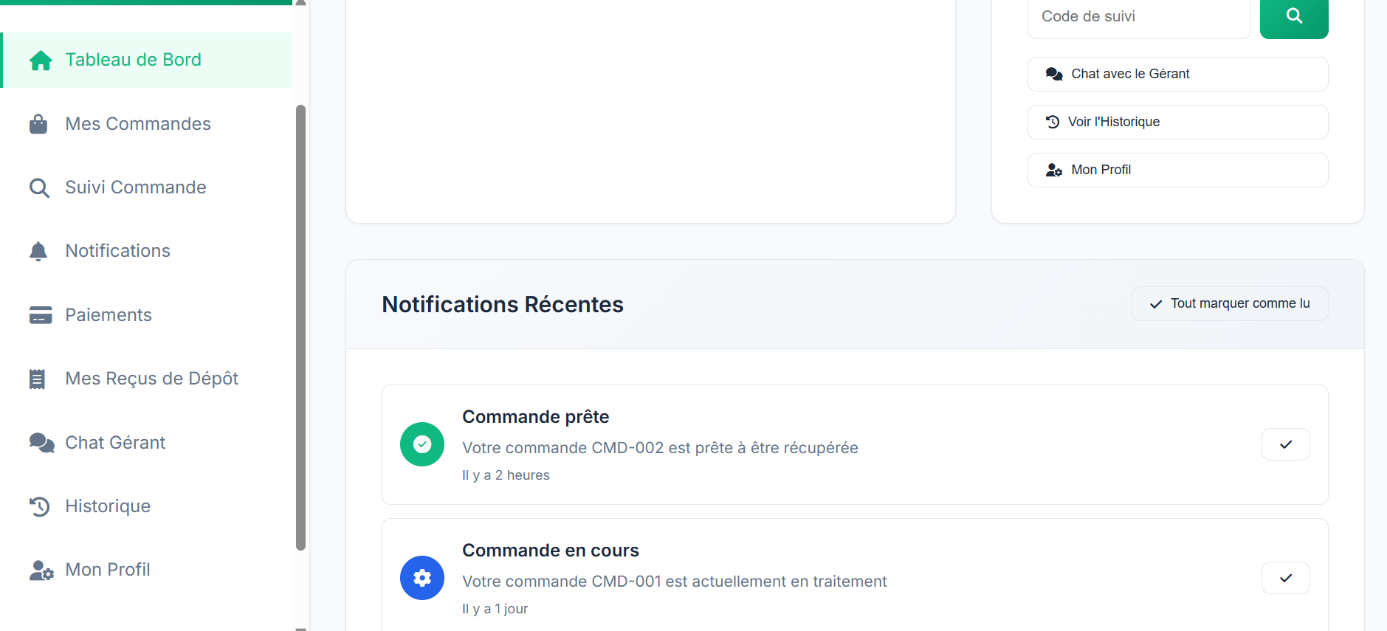


Figure 18: interface client

1. **Interface du gérant**

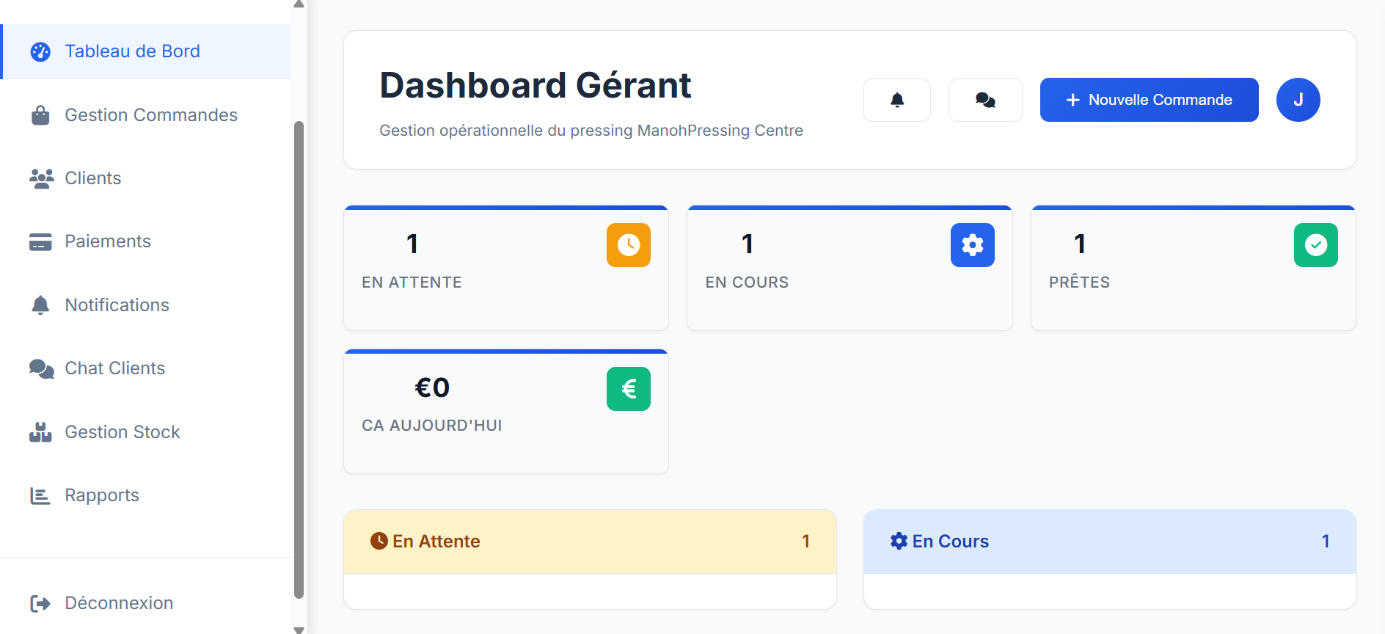
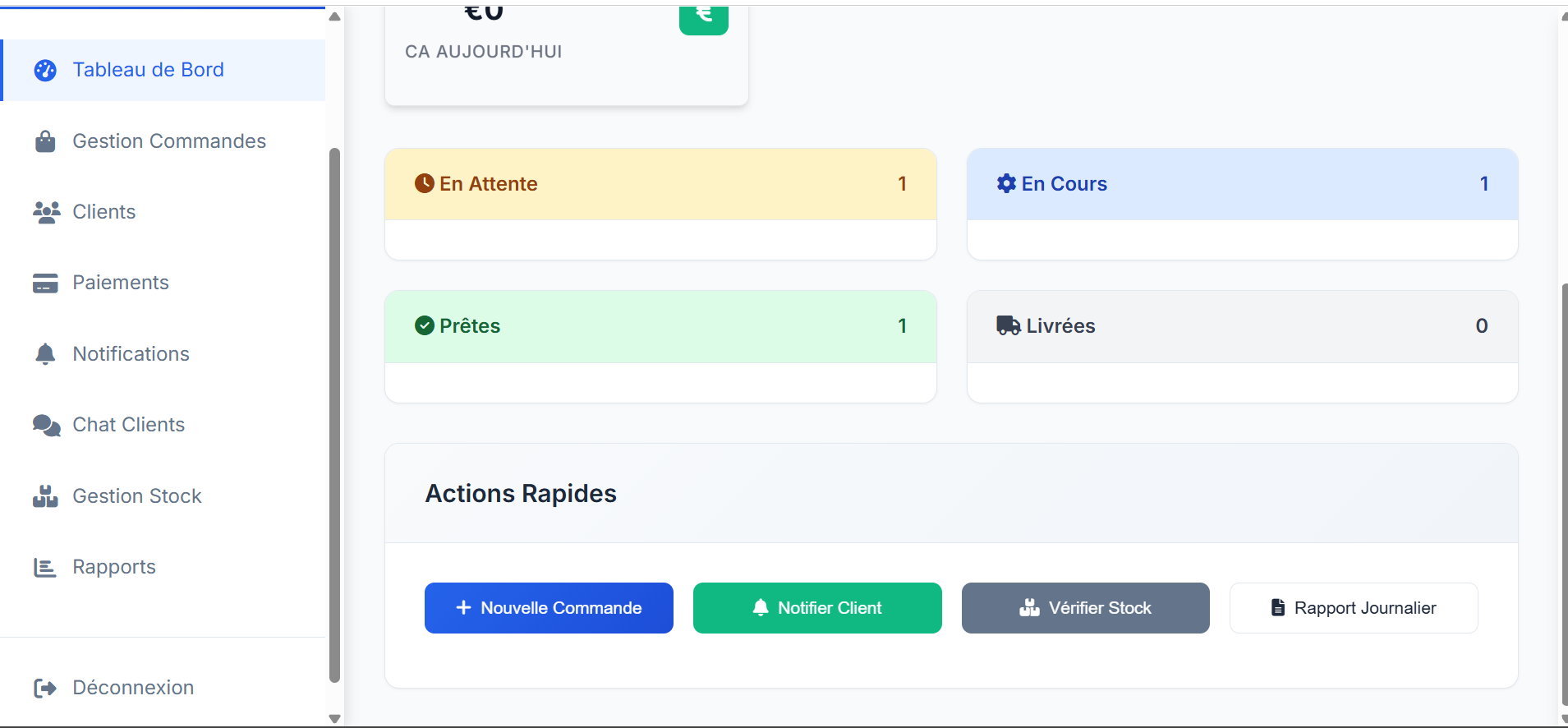


Figure 19: interface du gerant



1. **Interface de connexion**

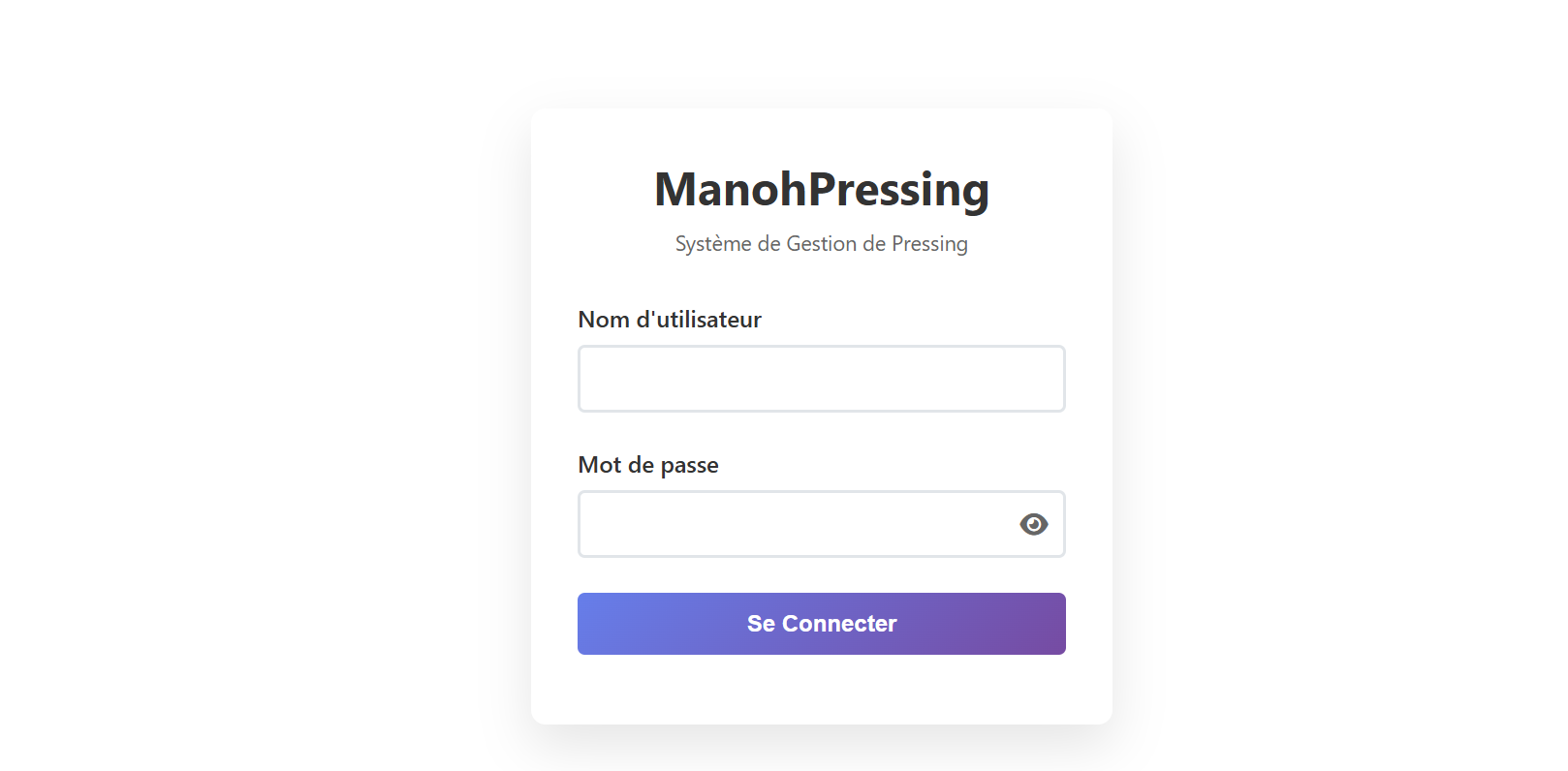


Figure 20: page de connexion

## 

## INTERPRETATION

Dans le cadre de la phase de validation, plusieurs tests fonctionnels ont été réalisés afin de s'assurer du bon comportement des principales fonctionnalités de l'application. Ci-dessous, deux fonctionnalités majeures sont présentées avec leurs objectifs, scénarios de test et résultats observés.

Table 5: test authentification des utilisateurs

|  |  |
| --- | --- |
| **Authentification des utilisateurs** | |
| **objetif** | Vérifier que le système autorise uniquement les utilisateurs enregistrés à accéder à l’espace de gestion correspondant a l’acteur qui se connecte. |
| **procedure** |  Saisir un identifiant et un mot de passe valides → vérifier l’accès.   Saisir un identifiant invalide → accès refusé avec message d’erreur.   Laisser les champs vides → message d’erreur. |
| **Resultat attendu** | Accès autorisé uniquement pour les identifiants valides. Message d’erreur ou de validation sinon redirection vers le Dashboard correspondant. |
| **Resultat obtenu** | Conforme aux attentes. |
| **remarque** | aucun dysfonctionnent observés |

Table 6:test passer une commande

|  |  |
| --- | --- |
| **Creation des commandes** | |
| **objetif** | Vérifier que le système permet l’enregistrement correct d’une commande client avec les bons vêtements, services et calculs associés. |
| **procedure** |  Depuis le tableau de bord employé, accéder au formulaire de création de commande.   Sélectionner un client existant.   Choisir différents types de vêtements (ex. : chemise, costume) avec les services associés (nettoyage, repassage).   Saisir les quantités et vérifier le calcul automatique du prix.   Valider la commande. |
| **Resultat attendu** | La commande est enregistrée dans la base de données avec un identifiant unique, un prix total correct. |
| **Resultat obtenu** | Fonctionnalité conforme. L'ensemble du processus de création fonctionne comme attendu, y compris le calcul automatique |
| **remarque** | Des tests supplémentaires avec des entrées invalides (quantité nulle, champs vides) ont permis de vérifier le bon fonctionnement des contrôles de validation. |

Table 7: test gestion des statuts d'une commande

|  |  |
| --- | --- |
| **Gestion des statuts d’une commande** | |
| **objetif** | S'assurer que chaque commande peut évoluer dans son cycle de vie selon les statuts prédéfinis ("reçu", "en cours", "prêt", "livré"). |
| **procedure** |  Accéder à la liste des commandes.   Sélectionner une commande en cours.   Modifier successivement son statut à chaque étape du traitement. |
| **Resultat attendu** | Le statut de la commande évolue correctement à chaque étape et est visible aussi bien côté employé que côté client. |
| **Resultat obtenu** | Test concluant. Les changements de statut sont bien enregistrés en base de données et reflétés sur l’interface publique de suivi. |
| **remarque** | Le système empêche toute régression |

# 

# CONCLUSION

Le présent projet a permis la conception et la mise en œuvre d’une application web dédiée à la gestion complète d’un pressing. Face aux limites des méthodes traditionnelles encore largement utilisées dans ce secteur, cette solution numérique vise à optimiser les processus internes tout en améliorant l’expérience client. À travers une interface simple et intuitive, l'application offre des fonctionnalités robustes. L'approche agile adoptée a favorisé une progression structurée du développement, permettant une adaptation rapide aux retours du bénéficiaire. Chaque itération a apporté des améliorations tangibles et a contribué à affiner la solution en fonction des besoins réels du terrain. Cette méthodologie s’est révélée particulièrement efficace pour garantir la pertinence fonctionnelle et la fiabilité technique de l’application. La réussite de cette application témoigne de la capacité à transformer un besoin métier concret en un outil numérique stable, fonctionnel et évolutif. À l’avenir, des pistes d’amélioration pourront être envisagées, notamment l’intégration d’une version mobile, l’ajout de fonctionnalités statistiques avancées ou encore la mise en place de notifications automatisées pour renforcer la réactivité du service. Ce projet marque ainsi une étape importante dans la digitalisation des services de proximité, et souligne l’intérêt de solutions sur mesure pour accompagner efficacement les petites structures dans leur transition numérique.

# BIBLIOGRAPHIE