

Mémoire de projet de fin d'étude

Présenté par :

Chaimaa MAHFOUD

Ibrahim BENMAGHA

Amine TAJEDDINE

En vue de l'obtention du diplôme

Licence Universitaire Professionnelle

Développement WEB Full Stack et DevOps

Sujet :

**« Étude et développement d'une plateforme
de gestion des services Artisans et des
utilisateurs »**

Encadré par :

Mohamed-Amine HAMMA

Année : 2023-2024

Dédicace

On dédie cet humble et modeste travail avec un grand amour, sincérité et fidélité à nos chers parents, qui sont pour nous une source inépuisable d'encouragement, de tendresse, de noblesse et d'affection. Puisse cette étape représente pour vous un motif de grande satisfaction.

C'est grâce à vos encouragements que nous avons choisi cette profession, et c'est à travers vos critiques constructives que nous avons pu nous réaliser. Nous espérons avoir répondu aux attentes que vous avez placées en nous. Recevez cette dédicace comme l'expression de notre gratitude éternelle et de notre amour infini. Que Dieu, le Tout-Puissant, vous protège, vous accorde santé, bonheur et longue vie.

On dédie également ce travail à nos frères et à nos sœurs, en témoignage de notre fraternité, avec tous nos vœux de bonheur, de santé et de succès. On n'oublie pas non plus nos amis, nos collègues de travail et nos professeurs, dont le respect précieux nous a accompagné tout au long de nos études.

REMERCIEMENT

Nous tenons tout d'abord à remercier Dieu, le Tout-Puissant, qui nous a donné la force, la patience et la bienveillance nécessaires pour accomplir ce travail.

Nous souhaitons exprimer nos profondes gratitude et nos respectueuses reconnaissances à notre encadrant, **M. Mohamed-Amine HAMMA**, pour son encadrement durant la période de préparation de notre projet de fin d'études. Il n'a cessé de nous orienter et de nous encourager afin que nous puissions donner le meilleur de nous-mêmes. Son suivi régulier a été essentiel à la réalisation de ce travail.

Nous remercions également tous les membres du jury qui nous ont fait l'honneur d'accepter de juger notre travail.

Nos remerciements s'adressent aussi à tout le corps professoral et administratif du **CENTRE D'EXCELLENCE EN INGÉNIERIE ET MANAGEMENT INDUSTRIEL**, en particulier à **M. Ahmed ENNACIRI**, responsable de la Formation et du Développement, pour la qualité de l'enseignement et la pédagogie qui nous ont été dispensés durant l'année d'études.

Résumé

Le projet "Lmoqaf" vise à créer une plateforme en ligne dédiée aux services artisanaux, facilitant la recherche, la réservation et la prestation de services dans divers domaines comme la plomberie, l'électricité, et la menuiserie. La plateforme a pour objectif de répondre aux besoins croissants des consommateurs tout en offrant une opportunité aux artisans de promouvoir leurs compétences et d'élargir leur clientèle.

Objectifs du Projet :

- **Simplicité d'utilisation** : Conception d'une interface intuitive pour une navigation facile.
- **Variété de services** : Offrir une gamme diversifiée de services artisanaux.
- **Flexibilité pour les artisans** : Permettre aux artisans de personnaliser leurs profils et leurs services.
- **Qualité et confiance** : Établir un processus de vérification des artisans pour garantir des services de qualité.
- **Accessibilité géographique** : Couverture étendue pour trouver des artisans localement.
- **Interaction transparente** : Faciliter la communication entre clients et artisans.
- **Expérience utilisateur optimale** : Offrir une expérience fluide et cohérente.

Fonctionnalités Clés :

- Création de profils détaillés pour artisans et clients.
- Recherche avancée et réservation de services.
- Système de paiement sécurisé (optionnel).
- Messagerie intégrée pour la communication directe.
- Gestion des avis et des notations.

Exigences Techniques :

- **Frontend** : Utilisation de React.js avec Ant Design et React Router DOM.
- **Backend** : Développement avec Laravel et implémentation d'authentification basée sur JWT.
- **Base de données** : Utilisation de MySQL.
- **Déploiement** : Hébergement sur AWS ou Heroku avec conteneurisation via Docker.
- **Tests** : Tests unitaires et fonctionnels pour assurer la qualité et la performance.

En atteignant ces objectifs, "Lmoqaf" ambitionne de devenir une référence dans le domaine des plateformes en ligne dédiées aux services artisanaux, offrant une solution pratique, fiable et efficace pour les utilisateurs.

Abstract

The "Lmoqaf" project aims to create an online platform dedicated to artisanal services, facilitating the search, booking, and provision of services in various fields such as plumbing, electricity, and carpentry. The platform aims to meet the growing needs of consumers while providing artisans with the opportunity to promote their skills and expand their clientele.

Project Objectives:

- **Ease of Use:** Design an intuitive interface for easy navigation.
- **Variety of Services:** Offer a diverse range of artisanal services.
- **Flexibility for Artisans:** Allow artisans to customize their profiles and services.
- **Quality and Trust:** Establish a verification process for artisans to ensure quality services.
- **Geographical Accessibility:** Extensive coverage to find local artisans.
- **Transparent Interaction:** Facilitate communication between clients and artisans.
- **Optimal User Experience:** Provide a smooth and consistent user experience.

Key Features:

- Creation of detailed profiles for artisans and clients.
- Advanced search and service booking.
- Secure payment system (optional).
- Integrated messaging for direct communication.
- Management of reviews and ratings.

Technical Requirements:

- **Front-end:** Using React.js with Ant Design and React Router DOM.
- **Back-end:** Development with Laravel and implementation of JWT-based authentication.
- **Database:** Using MySQL.
- **Deployment:** Hosting on AWS or Heroku with Docker containerization.
- **Testing:** Unit and functional testing to ensure quality and performance.

By achieving these objectives, "Lmoqaf" aims to become a benchmark in the field of online platforms dedicated to artisanal services, offering a practical, reliable, and efficient solution for users.

Liste des abréviation

Abreviation	Définition
AWS	Amazon Web Services
JWT	JSON Web Token
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DB	DataBase
IDE	Integrated Development Environment
UI	User Interface
UX	User Experience
CMS	Content management System
CI/CD	Continuous Integration/Continuous Deployment
SSL	Secure Sockets Layer
JSON	JavaScript Object Notation
SQL	Structured Query Language
HTTPS	HyperText Transfer Protocol Secure
CDN	Software Development Kit

Table 1 Liste des abreviations

Liste des figures

Figure 1 la methode SCRUM	21
Figure 2Diagramme de Gantt	23
Figure 3 Diagramme usecase.....	32
Figure 4 illustration de shéma de déploiement.....	38
Figure 5 Design pattern MVC	39
Figure 6 usecase administrateur	53

Liste des tableaux

Table 1 Liste des abreviations	6
Table 2 Livrable.....	19
Table 3 Risques	20
Table 4 Equipe MOE	23
Table 5 description sprint gestion administrateur	52
Table 6 Règles gestion sprint administrateur	53
Table 7 Description cas d'utilisation admin	53

Table des matières:

Dédicace	2
REMERCIEMENT	3
Résumé	4
Abstract.....	5
Liste des abréviations.....	6
Liste des figures.....	7
Liste des tableaux	8
Introduction Générale.....	12
Cadre et objectif du projet.....	12
• Simplicité d'utilisation :	12
• Variété de services :	12
• Flexibilité pour les artisans :	13
• Qualité et confiance :	13
• Accessibilité géographique :	13
• Interaction transparente :	13
• Expérience utilisateur optimale :	13
• Chat privé entre les artisans et les clients (optionnel) :	13
Description des chapitres	13
Chapitre 1 :	15
Etude Générale du projet :	16
1-Perimetre du projet :	16
1.1- Problématique générale :	16
1.2-Missions du projet :	17
1.3-Livrables :	19
1.4- Risques:	19
2-Conduite du projet:	20
2.1-Methode Agile:	20

2.2-Pourquoi SCRUM :	21
2.3-Cycle de vie :	21
2.4-Planification du projet :	22
3- Organisation:	23
3.1- Equipe MOE:	23
3.2-Description des rôles :	24
Conclusion :	24
Chapitre 2 :	25
C'est quoi la plateforme Lmoqaf?	26
Rappel Problématique :	26
Objectif du projet :	26
Analyse de l'existant :	27
Définition du système.....	29
Les bénéfices du système :	30
Spécifications de besoins :	31
1-Roles des acteurs :	31
2-Specifications des besoins :	33
• Création de Profils :	33
• Recherche Avancée :	34
• Réservation de Services :	34
• Système de Paiement (optionnel) :	34
• Messagerie Intégrée :	34
• Gestion des Avis et des Notations :	34
• Gestion des Profils et des Services :	34
• Système de Notification :	34
3-Règles de gestion :	35
Création de Comptes Clients et Artisans :	35
Conversion de Comptes Clients en Artisans :	35
Facturation de l'Abonnement Mensuel :	35
Notation Réciproque :	35
Blocage de Comptes Clients :	35

Conclusion :	36
Chapitre 3 :	37
Architecture technique :	38
1-Diagramme de déploiement :	38
2-Outils, langages et Framework utilisés :	40
Pourquoi choisir un framework ?	40
3-Architecture applicative :	46
4-Arborescence projet :	47
Conclusion :	49
Chapitre 4 :	50
1-Methodes et outils de modélisation :	51
1.1-Langage de modélisation (UML) :	51
1.2-Les outils de modélisation :	51
2- Sprint 1 : Gestion Administrateur	52
2.1- Description du sprint :	52
2.2- Backlog du sprint:	52
2.3- Règles de gestion du sprint :	53
2.4- Analyse et conception :	53
3- Sprint 2 : Gestion Artisan	54
3.1- Description du sprint :	54
Conclusion :	57

Introduction Générale

La plateforme "lmoqaf" est un projet visant à créer un marché en ligne dédié aux artisans, offrant une solution centralisée pour la recherche, la réservation et la prestation de services dans divers domaines tels que la plomberie, l'électricité, la menuiserie et autres métiers spécialisés.

"lmoqaf" a pour objectif de répondre aux besoins croissants des consommateurs en offrant un accès pratique à une gamme diversifiée de services artisanaux, tout en fournissant une plateforme lucrative pour les artisans indépendants et les petites entreprises afin de promouvoir leurs compétences et d'élargir leur clientèle. Dans le cadre de ce projet, seront détaillées les exigences fonctionnelles, techniques et opérationnelles de la plateforme "lmoqaf". Seront également décrits les objectifs du projet, son public cible, les principales fonctionnalités à développer, ainsi que les contraintes et les considérations clés qui guideront le processus de développement.

En collaborant étroitement avec les parties prenantes concernées, l'objectif est de créer une plateforme robuste, conviviale et sécurisée qui réponde aux besoins des artisans et des clients, tout en offrant une expérience utilisateur exceptionnelle.

Cadre et objectif du projet

L'objectif principal du projet "lmoqaf" est de créer une plateforme en ligne conviviale et inclusive qui offre une solution efficace pour la recherche, la réservation et la prestation de services artisanaux. Nous visons à répondre aux besoins des consommateurs en simplifiant le processus de recherche et de réservation de services, tout en offrant aux artisans une plateforme flexible pour promouvoir leurs compétences et développer leur activité. Plus spécifiquement, les objectifs du projet "lmoqaf" sont les suivants :

- **Simplicité d'utilisation :** Concevoir une interface utilisateur intuitive et conviviale qui permet aux utilisateurs de naviguer facilement sur la plateforme, de rechercher des services, de communiquer avec les artisans et de réserver des prestations en quelques clics.
- **Variété de services :** Offrir une gamme diversifiée de services artisanaux dans différents domaines tels que la plomberie, l'électricité, la menuiserie, la rénovation, etc., pour répondre aux besoins variés des clients.

- **Flexibilité pour les artisans :** Permettre aux artisans de créer des profils détaillés présentant leurs compétences, leur expérience, leurs tarifs et leur disponibilité, ainsi que la possibilité de personnaliser leurs services et de gérer leurs calendriers selon leurs besoins.
- **Qualité et confiance :** Établir un processus de vérification des artisans pour garantir la qualité des services proposés et renforcer la confiance des clients dans la plateforme.
- **Accessibilité géographique :** Élargir la couverture géographique de la plateforme pour permettre aux utilisateurs de trouver des artisans dans leur région ou leur zone de préférence.
- **Interaction transparente :** Faciliter la communication entre les clients et les artisans à travers la plateforme, en fournissant des outils pour échanger des messages, partager des informations et finaliser les détails des prestations.
- **Expérience utilisateur optimale :** Offrir une expérience utilisateur fluide et cohérente à chaque étape du processus, depuis la recherche initiale jusqu'à la réalisation du service, afin de garantir la satisfaction des clients et des artisans.
- **Chat privé entre les artisans et les clients (optionnel) :** Offrir la possibilité aux artisans et aux clients d'interagir via un chat privé sur la plateforme, permettant une communication directe et rapide pour discuter des détails des services, poser des questions et clarifier les besoins. En atteignant ces objectifs, "Imoqaf" ambitionne de devenir la référence en matière de plateforme en ligne pour les services artisanaux, offrant une solution pratique, fiable et efficace pour les utilisateurs à la recherche de solutions pour leurs besoins en matière de réparation, d'entretien et de rénovation.

Description des chapitres

Le but de ce rapport est de donner un aperçu clair du travail effectué lors de la réalisation du projet, divisé en plusieurs chapitres :

⇒ **Chapitre 1** : Ce chapitre élabore le contexte général du projet en offrant une vue d'ensemble du domaine dans lequel le projet s'inscrit. Il décrit la problématique actuelle qui a motivé la mise en place de la solution proposée. Les principaux défis rencontrés et les raisons pour lesquelles ces défis doivent être relevés sont expliqués en détail. En outre, ce chapitre précise les objectifs du projet, qu'ils soient à court terme ou à long terme, ainsi que les méthodes utilisées pour atteindre ces objectifs. Ces méthodes incluent les approches théoriques et pratiques adoptées, les technologies sélectionnées, et les stratégies de développement mises en œuvre pour surmonter les obstacles identifiés.

⇒ **Chapitre 2** : Ce chapitre consiste à établir les besoins fonctionnels et non fonctionnels du projet. Les besoins fonctionnels couvrent les fonctionnalités essentielles que la plateforme doit offrir pour répondre aux attentes des utilisateurs. Cela inclut la description des processus métier, les interactions entre les différents acteurs, et les services spécifiques à fournir. Les besoins non fonctionnels, quant à eux, concernent les exigences de performance, de sécurité, de fiabilité, et de maintenabilité. Ce chapitre se base sur une analyse approfondie des règles de gestion pour définir ces besoins de manière précise et exhaustive. L'objectif est de s'assurer que tous les aspects cruciaux du projet sont pris en compte dès les premières phases de conception.

⇒ **Chapitre 3** : Ce chapitre couvre tous les aspects de la mise en œuvre du projet. Il détaille les outils de travail utilisés, tels que les environnements de développement, les frameworks, et les bibliothèques logicielles. Une description des différentes étapes du processus de développement de la plateforme est fournie, allant de la phase de conception initiale à la phase de déploiement final. Chaque étape est accompagnée d'une explication des techniques et des pratiques de développement adoptées, y compris la gestion des versions, les tests unitaires, l'intégration continue, et les déploiements automatisés. Ce chapitre met également en lumière les défis techniques rencontrés au cours du développement et les solutions apportées pour les résoudre, assurant ainsi la robustesse et l'efficacité de la plateforme.

En conclusion, ce rapport vise à fournir une documentation détaillée et structurée du projet, mettant en avant le travail accompli et les résultats obtenus. Chaque chapitre est conçu pour offrir une compréhension approfondie des différentes phases du projet, de la conception à la mise en œuvre, tout en soulignant les contributions techniques et méthodologiques qui ont permis de réaliser les objectifs fixés.

Chapitre 1 :

« CONTEXTE GENERAL DU PROJET »

Etude Générale du projet :

Ce chapitre décrit différentes fonctionnalités de la solution, à savoir les composants de l'application web.

Et enfin, l'arrivé à l'approche de conception choisie.

1-Perimetre du projet :

1.1- Problématique générale :

Le marché des plateformes en ligne dédiées aux services artisanaux est actuellement saturé, mais bon nombre de ces plateformes présentent des lacunes significatives. Les plateformes existantes souffrent souvent de deux problèmes majeurs : soit elles ne sont pas simples à utiliser pour les utilisateurs finaux, soit elles limitent les services disponibles pour les artisans et les clients. Premièrement, de nombreuses plateformes existantes sont caractérisées par une complexité excessive dans leur interface utilisateur, ce qui rend difficile la navigation et l'utilisation pour les clients à la recherche de services artisanaux.

Les processus de recherche, de réservation et de communication avec les artisans peuvent être laborieux et peu intuitifs, décourageant ainsi les utilisateurs potentiels d'utiliser ces plateformes. Deuxièmement, certaines plateformes existantes imposent des restrictions importantes aux artisans et aux clients en termes de types de services disponibles, de zones géographiques couvertes ou de tarification.

Ces limitations peuvent entraver la capacité des artisans à commercialiser leurs services et à développer leur clientèle, tandis que les clients peuvent se sentir frustrés par le manque de choix et de flexibilité dans la recherche de solutions adaptées à leurs besoins spécifiques.

Face à ces défis, la problématique principale du projet "Imoqaf" réside dans la création d'une plateforme simple, conviviale et inclusive qui résolve ces problèmes de manière efficace. Il s'agit de concevoir une solution qui offre une expérience utilisateur fluide et intuitive, tout en permettant aux artisans de promouvoir leurs services de manière flexible et en offrant aux clients une gamme variée d'options pour répondre à leurs besoins en matière de services artisanaux.

Comment pouvons-nous simplifier l'interface utilisateur pour rendre la recherche, la réservation et la communication avec les artisans aussi intuitives que possible pour les clients ?

Comment pouvons-nous garantir que les artisans ne se sentent pas limités dans la promotion de leurs services et la gestion de leur activité sur la plateforme ?

Quelles stratégies pouvons-nous mettre en place pour offrir aux clients une gamme variée d'options et de services adaptés à leurs besoins spécifiques ?

Comment pouvons-nous assurer que la plateforme "lmoqaf" se démarque des autres sur le marché et devienne une solution attrayante pour les artisans et les clients ?

Quels indicateurs de performance pouvons-nous utiliser pour mesurer le succès de la plateforme et son impact sur les utilisateurs ?

1.2-Missions du projet :

Dans le cadre de notre projet de développement de la plateforme **LMoqaf**, nous mettrons en œuvre une méthodologie rigoureuse de gestion de la qualité pour garantir la conformité aux exigences définies, depuis la phase de conception jusqu'à la maintenance continue de la plateforme. Cette méthodologie comprend les étapes suivantes :

1. Identification des exigences :

- Nous débuterons par une analyse approfondie des besoins et des attentes des utilisateurs de la plateforme, en collaboration avec les parties prenantes internes et externes.
- Nous documenterons clairement les exigences fonctionnelles et non fonctionnelles de la plateforme, en mettant l'accent sur la convivialité, la sécurité, la performance et la fiabilité.

2. Conception :

- Sur la base des exigences identifiées, nous élaborerons une architecture et une conception détaillées de la plateforme, en utilisant des outils et des méthodes de modélisation appropriés.

- Nous nous assurerons que la conception est alignée sur les meilleures pratiques de développement web et qu'elle répond aux besoins des utilisateurs ainsi qu'aux objectifs du projet.

3. Développement :

- Nous mettrons en œuvre la conception de la plateforme en utilisant des technologies modernes et des pratiques de développement éprouvées.
- Nous suivrons des normes de codage strictes et nous effectuerons des revues de code régulières pour garantir la qualité et la cohérence du code source.

4. Tests :

- Nous élaborerons un plan de tests complet couvrant tous les aspects fonctionnels, de performance, de sécurité et de convivialité de la plateforme.
- Nous réaliserons des tests unitaires, des tests d'intégration, des tests de système et des tests d'acceptation utilisateur pour identifier et corriger les anomalies éventuelles.

5. Déploiement :

- Nous planifierons et exécuterons le déploiement de la plateforme de manière méthodique et contrôlée, en minimisant les interruptions de service et en assurant une transition fluide vers l'environnement de production.
- Nous surveillerons attentivement le déploiement pour détecter tout problème éventuel et prendre des mesures correctives immédiates.

6. Maintenance :

- Après le déploiement, nous assurerons une maintenance continue de la plateforme, en appliquant des correctifs de sécurité, en effectuant des mises à jour logicielles et en répondant aux besoins évolutifs des utilisateurs.
- Nous recueillerons les commentaires des utilisateurs et surveillerons les performances de la plateforme pour identifier les opportunités d'amélioration continue.

1.3-Livrables :

Ce tableau des livrables du projet offre une vue d'ensemble de l'état des documents à livrer pour chaque phase, en incluant les dates de livraison et de validation correspondantes. Il garantit ainsi une vision globale des livrables du projet.

Phase	Livrables	Date de livraison	Date de validation
Etude préalable	Cahier des charges	07/04/24	15/04/24
	Plan de qualité	20/05/24	22/05/24
Analyse et conception	Dossier de spécification technique	22/05/24	22/05/24
	Dossier de spécification fonctionnelles	22/05/24	22/05/24
	Diagrammes UML	22/05/24	22/05/24
Mise en place	Sprint 0 : Création de la base de données		
	Sprint 1 : Développement d'espace admin		
	Sprint 2 : Développement d'espace utilisateur		
	Sprint 3 : Développement d'espace Artisan		
	Sprint 4 : Gestion des services user/artisan		
Finalisation	Rapport		
	Code source		

Table 2 Livrable

1.4- Risques:

Un risque est un danger potentiel plus ou moins prévisible qui peut affecter le résultat d'un projet. Tous les éliminer est impossible et le risque zéro n'existe pas.

Risque	Type	Impact	Probabilité	Action correctives
Mauvaise gestion du temps	Non bloquant	Le projet ne sera pas terminé dans les délais initialement fixés	Moyenne	Augmenté la masse horaire
Gestion projet	Non Bloquant	Mauvaise planification, lacunes dans la communication	Forte	Doubler l'effort et travailler un temps extra.
Refus d'application	Bloquant	Impacter le délai de réalisation de projet	Moyenne	Métriser la méthode agile afin de
Manque de ressources	Non bloquant	Ralentissement de la réalisation de projet	Forte	Doubler l'effort et travailler un temps extra.
Problème technique	Bloquant	Impacter le délai de réalisation de projet.	Forte	Problème à résoudre le plutôt possible

Table 3Risques

2-Conduite du projet:

2.1-Methode Agile:

Les méthodologies agiles caractérisent un mode de gestion de projet informatique qui met l'accent sur le dialogue entre toutes les parties prenantes, clients, utilisateurs, développeurs et autres experts du projet, la flexibilité de mise en œuvre, la capacité de changer les plans et la rapidité de livraison. Il s'agit de rompre avec la pratique plus traditionnelle d'être trop rigide et exigeant en matière de cahier des charges (contractuel). En ce qui concerne le processus de développement, il est important de privilégier la communication relationnelle et omniprésente.

La méthode agile favorise la communication entre toutes les parties prenantes, telles que les clients, les utilisateurs, les développeurs et les autres professionnels impliqués dans le projet. Elle se distingue par sa flexibilité en cours de réalisation, sa capacité à modifier les plans et sa volonté de livrer rapidement les résultats

2.2-Pourquoi SCRUM :

La méthodologie Scrum correspond aux principes de la méthodologie Agile. Comme toutes les méthodes Agiles, Scrum aime livrer rapidement des prototypes. Un prototype est par définition opérationnel pour l'évaluation par les clients, les entrepreneurs et les membres de l'équipe. La méthodologie Scrum est basée sur les sprints. Les sprints sont relativement courts et durent généralement de 2 à 4 semaines. À la fin de chaque sprint, l'équipe annonce ce qu'elle a ajouté au produit.

2.3-Cycle de vie :

SCRUM est une méthode de développement agile axée sur les projets informatiques, caractérisée par des mises à jour régulières des ressources. Son principe fondamental repose sur la progression collective de l'équipe et sa capacité à ajuster continuellement le projet au fur et à mesure de son avancement.

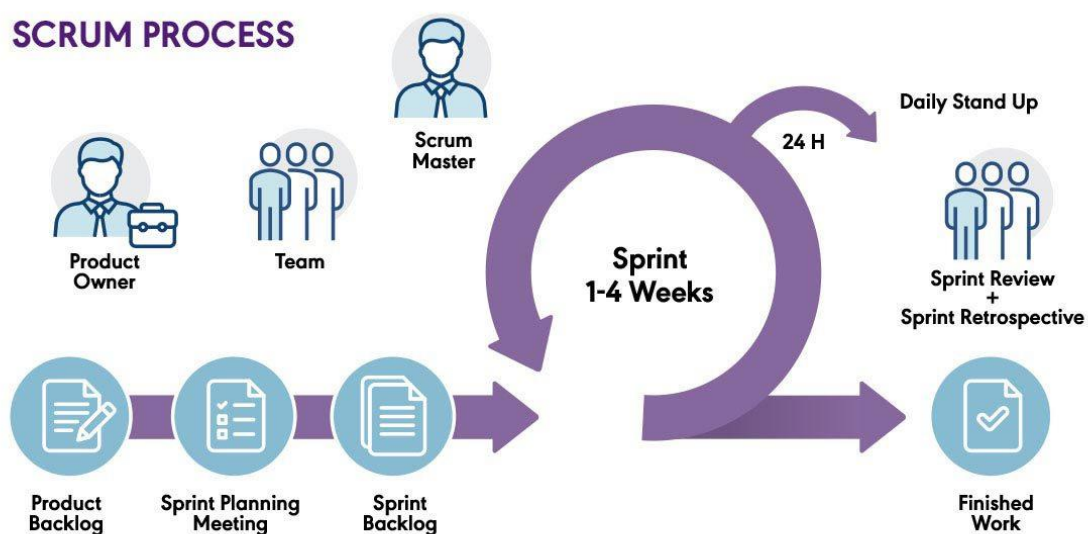


Figure 1 la methode SCRUM

⇒ Le Product Owner est chargé de la rédaction des User Stories et de leur intégration dans le Product Backlog.

⇒ Ensuite, le Product Owner établit les priorités des User Stories et organise le Product Backlog en conséquence.

⇒ l'équipe Scrum se réunit pour déterminer la liste des User Stories qui seront sélectionnées pour le Sprint. Ces User Stories constituent ensuite le Sprint Backlog et sont ensuite décomposées en tâches par l'équipe de développement.

⇒ Le sprint ensuite démarre pour une durée d'itération de 2, 3 ou 4 semaines.

2.4-Planification du projet :

La planification des projets de systèmes d'information consiste à planifier les opérations en termes de délais et d'utilisation des ressources. Diagramme de Gantt Le diagramme de Gantt est un outil de planification et de gestion de projet qui fournit une représentation visuelle des différentes tâches interdépendantes qui composent un projet au fil du temps. Vous pouvez visualiser graphiquement l'avancement de votre projet.

Pour créer un globe et un graphique des pertes, entrez une tâche avec des dates de début et de fin et une durée. Ce tableau montre les tâches à accomplir au cours du projet et les délais d'exécution approximatifs.

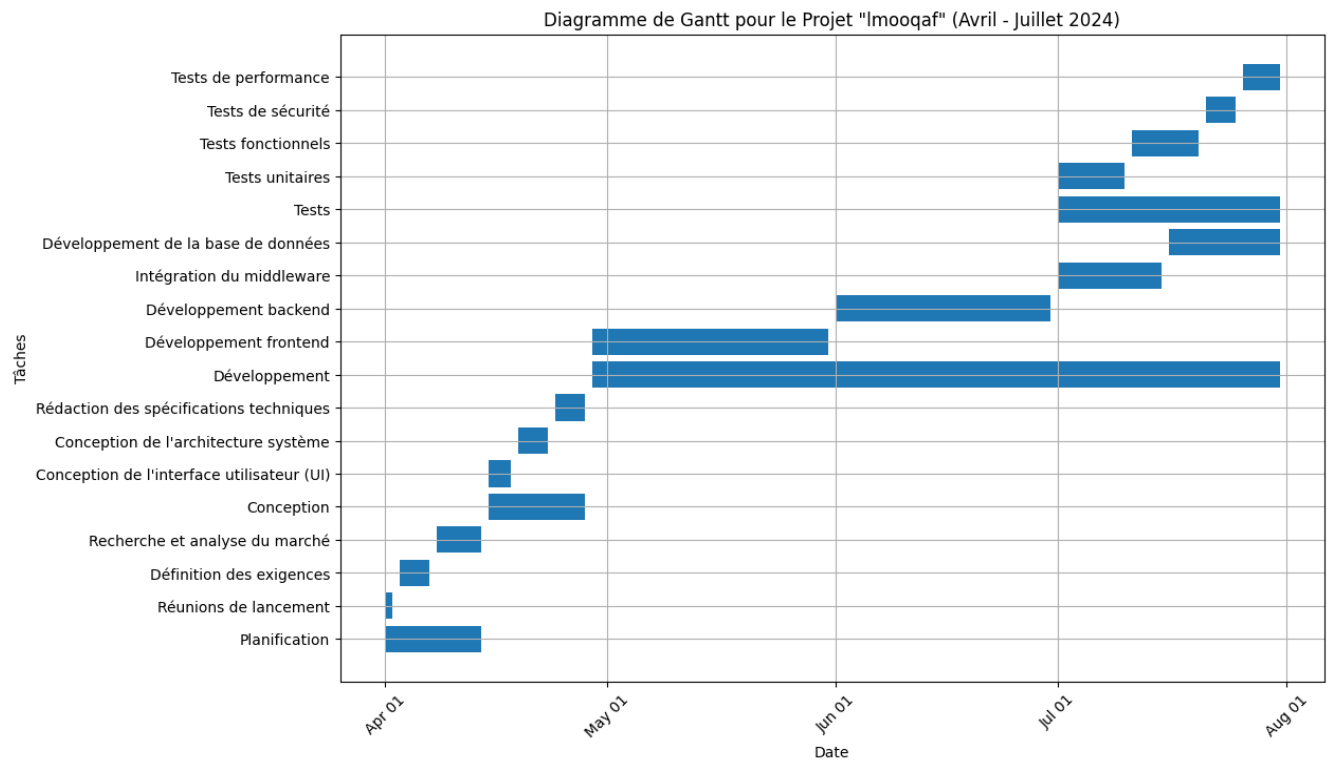


Figure 2 Diagramme de Gantt

3- Organisation:

3.1- Equipe MOE:

Le tableau ci-dessus fournit une présentation de l'équipe du projet, en indiquant le rôle et les tâches assignées à chaque membre de l'équipe.

Nom	Rôle	Email
Mohamed-Amine Hamma	Encadrant/ SCRUM master	medamine.hamma.9@gmail.com
Chaimaa Mahfoud	Développeur	Chaimaamahfoud681@gmail.com
Ibrahim Benmagha	Développeur	ibrahimbenmagha1337@gmail.com
Amine Tajeddine	Développeur	Amine.tajeddine1@gmail.com

Table 4 Equipe MOE

3.2-Description des rôles :

La réussite du projet **LMoqaf** exige une répartition claire des rôles et des responsabilités au sein de l'équipe projet. Les principaux rôles sont les suivants :

L'encadrant :

- Responsable de la constitution et de la coordination de l'équipe de projet, fournissant le soutien technique

Développeur :

- Identification et analyse des besoins et proposition des solutions adaptées
- Réalisation de cahier de spécifications technique
- Élaboration d'un plan de test et de vérification
- Correction de bug.

SCRUM Master :

- Guide l'équipe Scrum dans la mise en œuvre du projet.
- Fournir une visibilité sur l'avancement du projet.
- Aide l'équipe de développement à améliorer le rendement.

Conclusion :

Dans ce chapitre, nous avons abordé le cadre global du projet ainsi que la méthodologie pour sa mise en œuvre. Nous avons défini l'approche adoptée et les exigences du projet. Dans le prochain chapitre, nous nous concentrerons spécifiquement sur l'étude des besoins du projet.

Chapitre 2 :

« ETUDE DE BESOINS »

Dans ce chapitre, nous examinerons les résultats de l'analyse et de la conception du projet. Nous procéderons à une analyse détaillée des cas d'utilisation et des diagrammes de classe.

C'est quoi la plateforme Lmoqaf?

Lmoqaf est un projet qui vise à créer une plateforme innovante de mise en relation entre les clients et les freelancers, facilitant ainsi le processus de collaboration et de prestation de services.

Notre plateforme a pour objectif principal de simplifier le processus de recherche et de sélection de freelancers qualifiés pour répondre aux besoins spécifiques des clients. En offrant une interface conviviale et intuitive, nous visons à créer un environnement où les clients peuvent facilement trouver les services dont ils ont besoin et entrer en contact avec les freelancers les mieux adaptés à leurs projets.

Rappel Problématique :

Le marché des plateformes en ligne pour les services artisanaux est saturé, mais beaucoup de ces plateformes présentent des lacunes majeures, notamment une complexité excessive de l'interface utilisateur et des restrictions importantes sur les services disponibles. Ces problèmes rendent la navigation et l'utilisation difficiles pour les clients, décourageant ainsi leur engagement, et limitent les artisans dans la commercialisation de leurs services et la croissance de leur clientèle.

La plateforme "**lmoqaf**" vise à résoudre ces problèmes en créant une solution simple, conviviale et inclusive, offrant une expérience utilisateur fluide et intuitive, une grande flexibilité pour les artisans, et une variété d'options pour les clients, répondant ainsi efficacement à leurs besoins spécifiques en matière de services artisanaux.

Objectif du projet :

L'objectif principal du projet "**lmoqaf**" est de créer une plateforme en ligne conviviale et inclusive pour la recherche, la réservation et la prestation de services artisanaux. Cette plateforme vise à simplifier le processus pour les consommateurs tout en offrant aux artisans une flexibilité pour promouvoir leurs compétences et développer leur activité.

Les objectifs spécifiques incluent la conception d'une interface utilisateur intuitive, la diversification des services offerts, la personnalisation des profils des artisans, l'établissement d'un processus de vérification pour garantir la qualité et la confiance, l'extension de la couverture géographique, et la facilitation de la communication transparente entre clients et artisans. En outre, la plateforme promet une expérience utilisateur fluide, cohérente et optimale à chaque étape, avec une option de chat privé pour des interactions directes.

En atteignant ces objectifs, "**Imoqaf**" aspire à devenir la référence en matière de plateforme en ligne pour les services artisanaux, offrant une solution pratique, fiable et efficace pour la réparation, l'entretien et la rénovation.

Analyse de l'existant :

L'analyse de l'existant vise à évaluer les plateformes en ligne actuelles dédiées aux services artisanaux pour identifier leurs forces, leurs faiblesses, et les opportunités d'amélioration. Voici une analyse détaillée basée sur les éléments mentionnés dans le cahier des charges :

Faiblesses des Plateformes Existant

1. Complexité de l'Interface Utilisateur :

- La plateforme existante souffre d'une interface utilisateur trop complexe, rendant la navigation difficile pour les utilisateurs finaux. Cette complexité décourage les utilisateurs de continuer à utiliser la plateforme, augmentant ainsi le taux d'abandon.

2. Problèmes de Demande de Service :

- Les utilisateurs rencontrent fréquemment des difficultés lorsqu'ils demandent un service. Souvent, ces demandes restent sans réponse, ce qui entraîne frustration et perte de confiance. Cette absence de réactivité nuit gravement à la réputation de la plateforme et à la satisfaction des utilisateurs.

3. Manque de Fonctionnalité Générale:

- La plateforme manque de fonctionnalités essentielles qui sont cruciales pour une expérience utilisateur fluide et efficace. Par exemple, l'absence de fonctionnalités de suivi des demandes, de gestion des réservations, et de communication directe avec les artisans rend la plateforme moins utile et moins attrayante pour les utilisateurs.

4. Manque de Flexibilité pour les Artisans :

- Les artisans peuvent se retrouver contraints par des règles strictes en matière de tarification et de disponibilité, ce qui limite leur capacité à gérer leur activité de manière autonome.

5. Processus de Vérification des Artisans :

- Les mécanismes de vérification des compétences et de la fiabilité des artisans sont parfois insuffisants, entraînant une perte de confiance de la part des utilisateurs.

Opportunités d'Amélioration

1. Simplification de l'Interface Utilisateur :

- Concevoir une interface plus intuitive et conviviale, facilitant la navigation, la recherche de services, et la communication avec les artisans.

2. Élargissement de la Couverture Géographique :

- Offrir une plateforme avec une couverture géographique plus large pour permettre aux clients de trouver des artisans dans leur région de préférence.

3. Diversification des Services :

- Proposer une gamme diversifiée de services artisanaux, incluant des domaines tels que la plomberie, l'électricité, la menuiserie, la rénovation, etc.

4. Flexibilité et Autonomie pour les Artisans :

- Permettre aux artisans de personnaliser leurs profils, de définir leurs tarifs et disponibilités, et de gérer leurs calendriers selon leurs besoins spécifiques.

5. Systèmes de Vérification Rigoureux :

- Mettre en place un processus de vérification rigoureux pour garantir la qualité des services proposés et renforcer la confiance des clients.

6. Outils de Communication Améliorés :

- Faciliter les échanges entre clients et artisans avec des outils de messagerie intégrés et des options de chat privé pour une interaction directe et rapide.

7. Expérience Utilisateur Optimisée :

- Offrir une expérience utilisateur fluide et cohérente à chaque étape du processus, depuis la recherche initiale jusqu'à la réalisation du service, afin de garantir la satisfaction des clients et des artisans.

Conclusion

L'analyse de l'existant montre que bien que les plateformes actuelles pour les services artisanaux offrent certaines fonctionnalités essentielles, elles présentent aussi des lacunes importantes en matière de convivialité, de flexibilité, et de

couverture géographique. Le projet "lmoqaf" a l'opportunité de combler ces lacunes en créant une plateforme plus intuitive, flexible et inclusive, capable de répondre de manière efficace aux besoins des consommateurs et des artisans. En mettant l'accent sur une interface utilisateur simplifiée, une variété de services, et des outils de communication améliorés, "lmoqaf" peut devenir la référence en matière de plateforme en ligne pour les services artisanaux.

Définition du système

Le système Lmoqafest une plateforme innovante conçue pour faciliter la mise en relation entre les clients et les freelancers. Son objectif principal est de simplifier le processus de recherche et de sélection de freelancers qualifiés, permettant aux clients de trouver facilement les services nécessaires et d'entrer en contact avec les freelancers les mieux adaptés à leurs projets. Cette plateforme se caractérise par une interface utilisateur conviviale et intuitive, offrant une expérience de navigation simple et efficace. Elle inclut des fonctionnalités essentielles telles que la gestion des profils et des comptes, où les utilisateurs peuvent créer et gérer leurs profils personnels, incluant des informations sur leurs compétences, leur expérience et leurs évaluations.

De plus, Lmoqafdispose d'outils de recherche et de filtrage avancés, permettant aux clients de trouver des freelancers selon des critères spécifiques. Le système intègre également une messagerie pour faciliter la communication entre les clients et les freelancers, ainsi qu'un module de gestion des projets pour suivre la progression et les livrables des projets en cours. La sécurité et la confidentialité des données sont assurées par des mécanismes robustes, notamment le cryptage SSL, pour protéger les informations des utilisateurs. En outre, un système d'avis et d'évaluations permet aux clients de donner des retours sur les services des freelancers, contribuant à maintenir un niveau élevé de qualité et de confiance sur la plateforme. Enfin, des notifications en temps réel informent les utilisateurs des nouvelles offres de services, des messages reçus, et des mises à jour importantes concernant leurs projets. Ensemble, ces fonctionnalités font de LMoqafun environnement intégré et sécurisé, facilitant une collaboration efficace et transparente entre les clients et les freelancers.

Les bénéfices du système :

Le système Lmoqaf offre de nombreux avantages tant pour les clients que pour les freelancers. Voici les principaux bénéfices identifiés à partir des documents fournis :

1. **Interface Utilisateur Intuitive et Conviviale** : Lmoqaf propose une interface facile à utiliser, permettant aux utilisateurs de naviguer aisément, de rechercher des services, de communiquer avec les artisans et de réserver des prestations rapidement. Cette convivialité améliore l'expérience utilisateur globale, rendant la plateforme accessible même pour ceux ayant peu de compétences techniques.
2. **Large Gamme de Services** : La plateforme offre une variété de services artisanaux dans divers domaines tels que la plomberie, l'électricité, la menuiserie et la rénovation. Cette diversité permet de répondre aux besoins variés des clients, leur offrant un guichet unique pour toutes leurs exigences artisanales.
3. **Flexibilité pour les Artisans** : Les artisans peuvent créer des profils détaillés, mettant en avant leurs compétences, leur expérience, leurs tarifs et leur disponibilité. Ils ont également la possibilité de personnaliser leurs services et de gérer leurs calendriers selon leurs besoins, ce qui leur donne un contrôle total sur leur activité sur la plateforme.
4. **Qualité et Confiance** : Lmoqaf met en place un système d'avis et de notations qui permet aux clients de laisser des commentaires et des évaluations après avoir reçu des services. Cela aide à maintenir un haut niveau de qualité et de confiance, car les évaluations sont visibles sur les profils des artisans, aidant ainsi d'autres clients dans leur processus de sélection.
5. **Sécurité et Confidentialité** : La plateforme utilise des protocoles de sécurité robustes pour protéger les données des utilisateurs. Le cryptage SSL et d'autres mesures de sécurité sont mis en place pour assurer la confidentialité des informations sensibles et prévenir les attaques telles que le piratage ou la fraude.
6. **Notifications en Temps Réel** : Un système de notification intégré informe les utilisateurs des nouvelles réservations, des messages reçus, des mises à jour de profil, etc. Cela permet aux utilisateurs de rester informés en temps réel et de réagir rapidement aux changements ou aux nouvelles opportunités.
7. **Optimisation des Performances** : Lmoqaf assure des performances optimales en termes de vitesse de chargement des pages et de temps de réponse, même sous une charge élevée. Cette performance garantit une

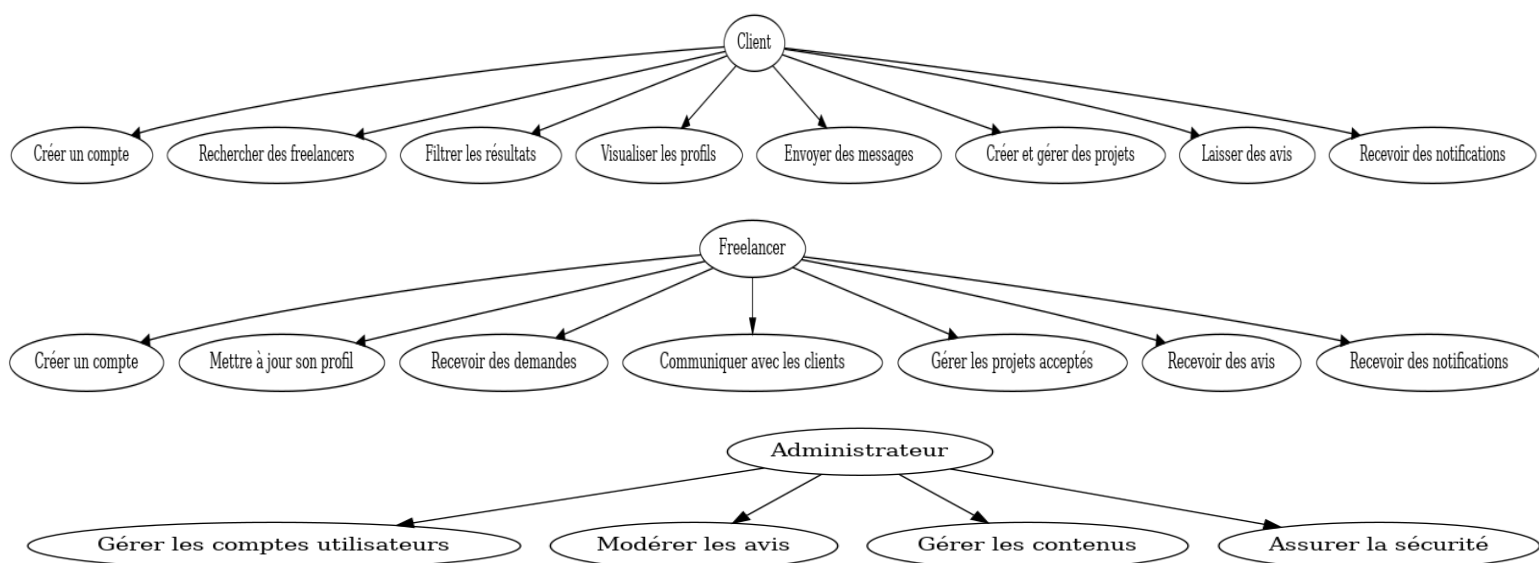
expérience utilisateur fluide et agréable, augmentant ainsi la satisfaction des utilisateurs.

8. **Adaptabilité et Maintenabilité** : La plateforme est conçue pour être adaptable à différents types d'appareils et de navigateurs, offrant une expérience cohérente sur les ordinateurs de bureau, les tablettes et les smartphones. De plus, le code source est bien structuré et modulaire, facilitant la maintenance et les mises à jour continues.

En somme, le système Lmoqaf offre une solution complète et intégrée pour la mise en relation entre les clients et les artisans, optimisant l'expérience utilisateur tout en assurant une gestion efficace et sécurisée des services et des données.

Spécifications de besoins :

1-Roles des acteurs :



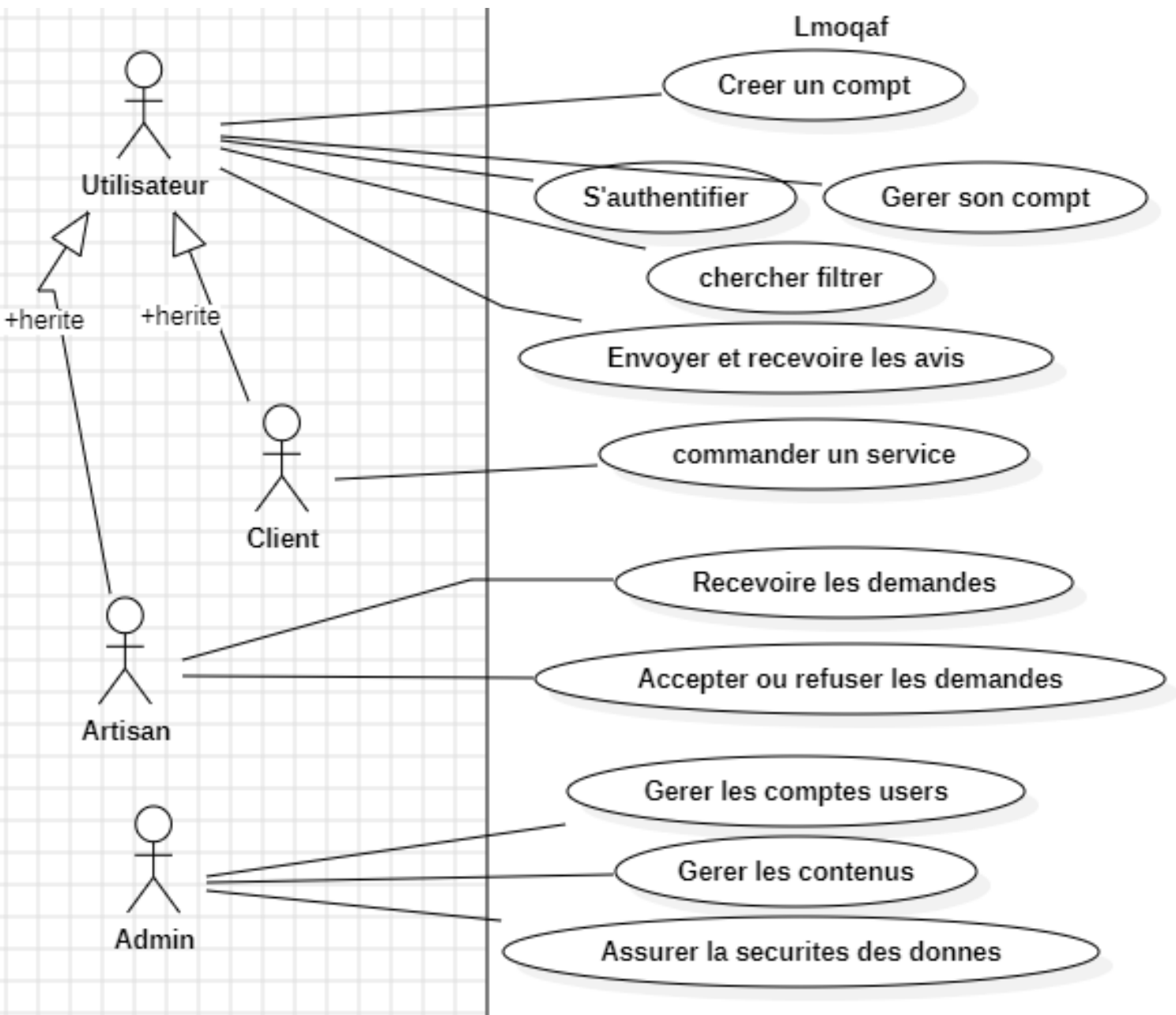


Figure 3 Diagramme usecase

Acteurs

1. Client
2. Freelancer
3. Administrateur

Cas d'utilisation

1. Client

- Créer un compte
- Rechercher des freelancers
- Filtrer les résultats
- Visualiser les profils des freelancers
- Envoyer des messages aux freelancers
- Créer et gérer des projets
- Laisser des avis et des évaluations
- Recevoir des notifications

2. Freelancer

- Créer un compte
- Mettre à jour son profil
- Recevoir des demandes de projet
- Communiquer avec les clients
- Gérer les projets acceptés
- Recevoir des avis et des évaluations
- Recevoir des notifications

3. Administrateur

- Gérer les comptes utilisateurs
- Modérer les avis et évaluations
- Gérer les contenus et les paramètres de la plateforme
- Assurer la sécurité et la maintenance du système

2-Specifications des besoins :

Les besoins non fonctionnels jouent un rôle essentiel en améliorant la qualité du logiciel de notre système. Ils agissent comme des contraintes pour les solutions proposées, mais les prendre en compte permet d'éviter de nombreuses incohérences dans le système. Pour atteindre cet objectif, notre système doit satisfaire les exigences suivantes :

- **Création de Profils :** Les artisans doivent pouvoir créer des profils détaillés présentant leurs compétences, leur expérience, leurs tarifs et leur

disponibilité. Les clients doivent pouvoir créer des profils pour accéder aux services de la plateforme et gérer leurs réservations.

- **Recherche Avancée :** Les utilisateurs doivent pouvoir effectuer des recherches avancées pour trouver des artisans selon différents critères tels que le type de service, la localisation, les avis, etc.
- **Réservation de Services :** Les clients doivent pouvoir réserver des services directement à partir des profils des artisans, en sélectionnant la date, l'heure et les détails spécifiques du service requis. Les artisans doivent recevoir des notifications des nouvelles réservations et avoir la possibilité de les accepter ou de les rejeter.
- **Système de Paiement (optionnel) :** La plateforme doit intégrer un système de paiement sécurisé permettant aux clients de payer les services réservés en ligne. Les artisans doivent pouvoir recevoir des paiements directement sur leur compte bancaire ou via d'autres méthodes de paiement acceptées.
- **Messagerie Intégrée :** Les clients et les artisans doivent pouvoir communiquer entre eux via un système de messagerie intégré sur la plateforme. Les utilisateurs doivent pouvoir échanger des messages pour discuter des détails des services, poser des questions et clarifier les besoins.
- **Gestion des Avis et des Notations :** Les utilisateurs doivent pouvoir laisser des avis et des notations sur les services fournis par les artisans après leur achèvement. Les avis et les notations doivent être visibles sur les profils des artisans pour aider les clients dans leur processus de sélection.
- **Gestion des Profils et des Services :** Les artisans doivent pouvoir gérer leurs profils, y compris la mise à jour des informations, la gestion des avis, la modification des tarifs et la gestion des disponibilités. Les utilisateurs doivent pouvoir gérer leurs réservations, y compris la visualisation de l'historique des réservations et la modification des détails des services.
- **Système de Notification :** La plateforme doit intégrer un système de notification pour informer les utilisateurs des nouvelles réservations, des messages reçus, des mises à jour de profil, etc.

En répondant à ces exigences fonctionnelles, la plateforme "Imoqaf" offrira une solution complète et conviviale pour la recherche, la réservation et la prestation de services artisanaux, visant à simplifier le processus pour les utilisateurs tout en offrant une plateforme lucrative pour les artisans.

3-Règles de gestion :

Création de Comptes Clients et Artisans :

- Les clients doivent créer leur compte client sur la plateforme "Imoqaf" pour accéder aux services proposés par les artisans.
- Les artisans doivent créer leur compte artisan pour proposer leurs services sur la plateforme "Imoqaf".
- Les artisans doivent s'abonner mensuellement à la plateforme "Imoqaf". Les trois premiers mois d'abonnement seront gratuits et seront déclenchés par chaque demande de service complétée (1 à 2 services présentés). Après les trois premiers mois, l'abonnement commencera avec des packs mensuels (1 à 2 services présentés ou 3 à 5 services présentés).

Conversion de Comptes Clients en Artisans :

Les clients ont la possibilité de convertir leur compte client en compte artisan s'ils souhaitent proposer leurs services sur la plateforme. Ils devront alors payer les mêmes frais d'abonnement mensuel que les artisans dès le début de leur activité en tant qu'artisans.

Facturation de l'Abonnement Mensuel :

Les frais d'abonnement mensuel pour les artisans seront déclenchés après les trois premiers mois d'activité sur la plateforme "Imoqaf". Les artisans auront le choix entre différents packs mensuels en fonction du nombre de services qu'ils souhaitent présenter sur la plateforme.

Notation Réciproque :

Les artisans peuvent noter les clients après avoir fourni leurs services. Cette notation est destinée à encourager les clients à adopter un comportement respectueux et responsable lors de l'utilisation de la plateforme.

Blocage de Comptes Clients :

Si un client reçoit plusieurs évaluations négatives de la part des artisans, son compte sera temporairement bloqué. Le client sera informé des raisons du blocage et pourra demander une révision de son compte après une période déterminée.

Conclusion :

Dans ce chapitre, nous avons minutieusement spécifié les besoins du client en présentant de manière organisée l'ensemble des fonctionnalités à travers les différents cycles de l'application, qu'il s'agisse des besoins fonctionnels ou non fonctionnels. Chaque aspect a été rigoureusement défini pour répondre aux attentes précises du projet, en tenant compte des exigences spécifiques des utilisateurs et des règles de gestion qui régissent leur interaction avec la plateforme. Les besoins fonctionnels ont été clairement identifiés pour couvrir toutes les opérations essentielles que l'application doit exécuter, allant de la gestion des utilisateurs et des profils à la manipulation des données transactionnelles et à la communication en temps réel. En parallèle, les besoins non fonctionnels ont été détaillés pour garantir des performances optimales, une sécurité renforcée, une maintenabilité accrue, et une convivialité maximale de l'interface utilisateur. Chaque élément a été aligné sur les objectifs stratégiques du projet, assurant ainsi une intégration harmonieuse des fonctionnalités et une expérience utilisateur enrichissante.

Chapitre 3 :

« ETUDE DE BESOINS »

Dans cette section, nous nous concentrerons sur l'aspect architectural de l'application, en prenant en compte les besoins identifiés lors de l'étape d'étude. Nous définirons également les outils et langages de développement qui seront utilisés pour la réalisation de notre projet.

Architecture technique :

Dans notre architecture technique, nous adopterons une approche basée sur une architecture en couches (ou architecture multi-tiers). Cette approche consiste à diviser l'application en plusieurs couches distinctes, chacune ayant des responsabilités spécifiques. Par exemple, nous pouvons avoir une couche de présentation qui gère l'interface utilisateur, une couche métier qui contient la logique applicative et une couche de persistance des données qui gère l'accès à la base de données.

1-Diagramme de déploiement :

Un diagramme de déploiement est une représentation statique qui permet de visualiser comment un système utilise l'infrastructure physique et comment ses composants sont répartis et interagissent entre eux. Il met en évidence les nœuds, les composants, les associations et les artefacts. Les stéréotypes peuvent être utilisés pour préciser les caractéristiques des ressources matérielles physiques et des supports de communication.

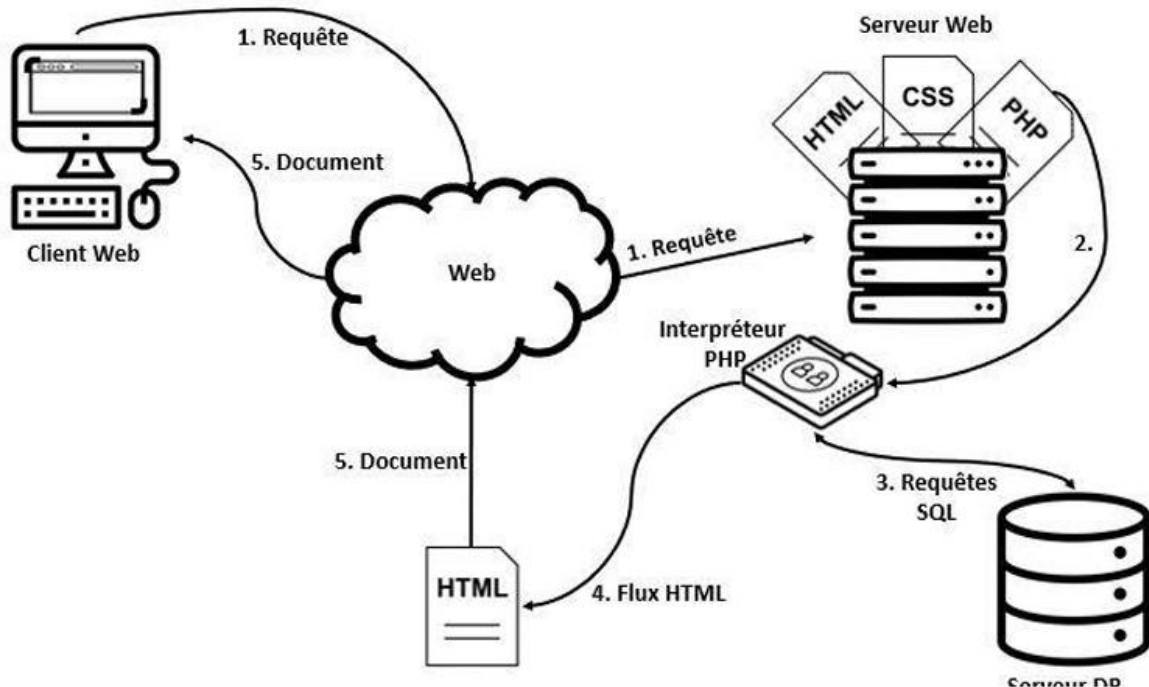


Figure 4 illustration de schéma de déploiement

Après l'analyse de la faisabilité du projet on a entamé la phase de développement de notre application web, vu la nature et le type de l'application, le model **MVC** (**M**odel, **V**iew, **C**ontroller) été le choix idéal pour la réalisation de l'application.

Le **MVC**, acronyme de **Model View Controller** (Modèle Vue Contrôleur), est une technique de développement avancée devenue un design pattern, qui découpe l'application en 3 couches principales, nommées Modèle, Vue et Contrôleur. La distinction de ces couches :

1. facilite l'organisation des sources du projet.
2. permet à chaque corps de métier de travailler en parallèle sur les sources qui leur sont dédiées ;
3. réduit l'impact des modifications pour minimiser les risques d'erreur et les régressions.

Le MVC évolue et améliore par la même occasion le cadre de travail des développeurs. Des Framework sont nés, apportant chacun leurs avantages et des inconvénients. Des idées engendrant d'autres idées ont permis le perfectionnement de cette technique.

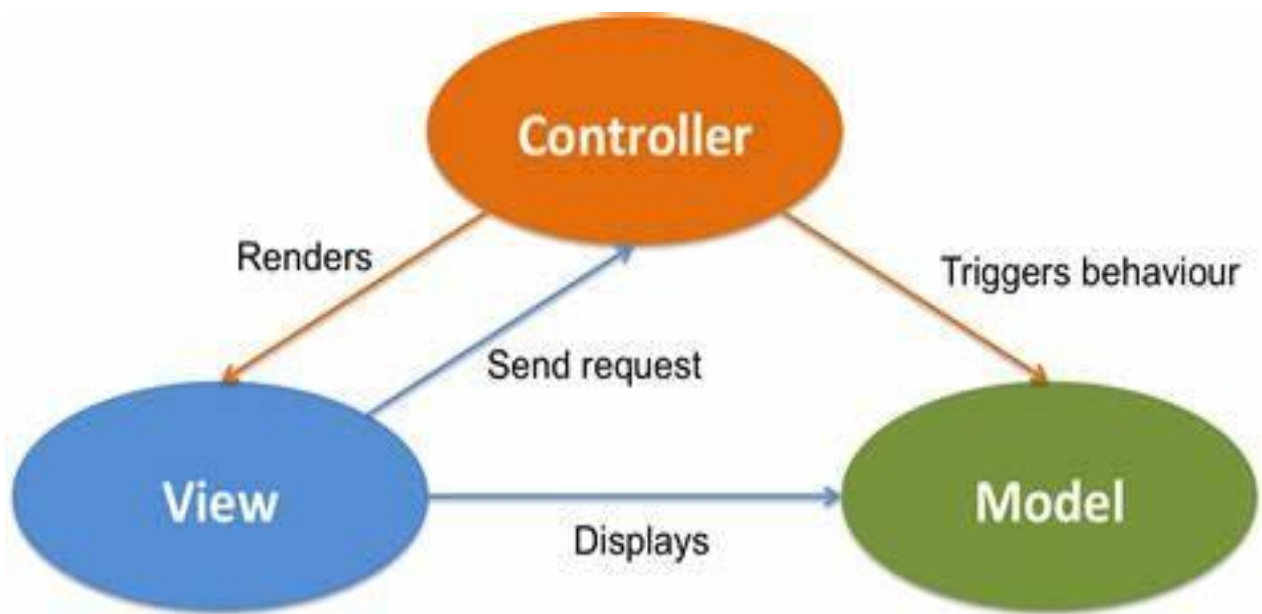
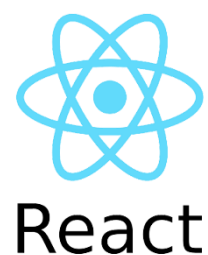


Figure 5 Design pattern MVC

2-Outils, langages et Framework utilisés :



Laravel est un framework web open-source écrit en PHP1 respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.

React est une bibliothèque qui ne gère que l'interface de l'application, cette interface étant considérée comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un Framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité2.

Pourquoi choisir un framework ?

L'utilisation d'un framework en entreprise maintenant est très répandue puisqu'elle délivre véritablement de nombreux avantages :

Un gain de temps pour le développeur web : les couches techniques étant déjà intégrées dans le Framework, le développeur peut pleinement se concentrer sur son développement. Le projet prend ainsi forme beaucoup plus vite.

Les maintenances et évolutions du Framework ne sont pas gérées par l'équipe de développement, mais par l'organisme fondateur. L'équipe chargée du développement gagnera ainsi un temps précieux pour développer le projet.

Facilitation pour collaborer : si l'équipe chargée du développement vient à recruter un nouveau membre, si ce dernier maîtrise le Framework en question, son intégration sera beaucoup plus simple grâce à l'utilisation d'un Framework.

Une communauté nombreuse et active : lorsque que vous allez rencontrer un bug ou si vous ressentez la nécessité de poser des questions précises, les

nombreux utilisateurs de Framework sauront vous guider. Des millions de développeurs utilisent :



Pour la gestion et la manipulation de base de données de notre application on s'est orienté vers **MySQL** puisqu'il s'agit d'une application de moyenne taille.

MySQL est un système de gestion de base de données relationnelle (SGBDR) complet et open source qui rivalise avec les goûts d'Oracle DB et de SQL Server de Microsoft.



Pour application nous avons choisi comme Framework CSS **Bootstrap** afin de le coupler avec le langage de balisage.

Bootstrap est une boîte à outils open source conçu et développé par twitter pour le développement avec HTML, CSS et JS.



PHPMysqlAdmin est un outil populaire utilisé pour le développement d'applications basées sur MySQL ou MariaDB. Voici quelques raisons pour lesquelles il est couramment utilisé dans le processus de développement.

Il est également accessible via un navigateur web, ce qui le rend pratique et utilisable sur différentes plateformes. Il offre une solution complète pour gérer les bases de données lors du développement d'applications, améliorant ainsi l'efficacité et la productivité des développeurs.



Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code (IntelliSense4.), les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Le code source de Visual Studio Code provient du projet logiciel libre et open source VS Code de Microsoft publié sous la licence MIT permissive, mais les binaires compilés constituent un freeware, c'est-à-dire un logiciel gratuit pour toute utilisation mais propriétaire.



Git est un système de contrôle de version distribué open source conçu pour gérer des projets de toute taille avec rapidité et efficacité. Créé par Linus Torvalds en 2005, il permet à plusieurs développeurs de travailler simultanément sur un projet sans se marcher sur les pieds. Les principales caractéristiques de Git incluent :

- **Versioning efficace** : Git suit les modifications de fichiers et permet de revenir à une version antérieure si nécessaire.
- **Branches et fusion** : Git permet de créer des branches pour développer des fonctionnalités isolées, qui peuvent ensuite être fusionnées avec la branche principale.
- **Distribution** : Chaque développeur a une copie complète du dépôt, ce qui permet de travailler hors ligne et de partager les modifications facilement.
- **Collaboration** : Git facilite la collaboration entre les développeurs en gérant les conflits de fusion et en permettant de revoir le code avant de l'intégrer à la base de code principale.



GitHub est une plateforme de développement collaboratif basée sur Git qui héberge des projets de code source. Elle offre des fonctionnalités supplémentaires pour faciliter la collaboration et la gestion de projets, telles que :

- **Référentiels hébergés** : GitHub héberge des dépôts Git, permettant aux développeurs de partager et de collaborer sur le code.
- **Gestion des versions** : GitHub permet de gérer les versions de projet, suivre les modifications et les contributions.

- **Pull Requests** : GitHub facilite la révision du code et la collaboration avec les Pull Requests, où les développeurs peuvent discuter et revoir les modifications avant de les fusionner.
- **Intégrations** : GitHub s'intègre avec divers outils et services CI/CD, facilitant l'automatisation des tests et des déploiements.
- **Pages GitHub** : Pour héberger des sites web statiques directement à partir d'un dépôt.



Jira est un outil de gestion de projet et de suivi des problèmes développé par Atlassian. Il est couramment utilisé pour le développement de logiciels agiles et offre des fonctionnalités robustes pour gérer les tâches et les projets, notamment :

- **Suivi des problèmes** : Jira permet de créer, suivre et résoudre des bugs et des tâches.
- **Gestion de projet Agile** : Jira supporte les méthodologies Scrum et Kanban, facilitant la planification des sprints, la gestion des backlogs et le suivi des progrès.
- **Tableaux de bord personnalisés** : Les utilisateurs peuvent créer des tableaux de bord personnalisés pour surveiller les métriques de performance et les progrès du projet.
- **Intégrations** : Jira s'intègre avec d'autres outils Atlassian comme Confluence, Bitbucket, ainsi que des outils tiers pour CI/CD, tests et gestion de version.
- **Rapports et Analyses** : Jira offre des capacités de reporting avancées, permettant de générer des rapports pour analyser les performances de l'équipe et l'état des projets.



Postman est un outil de collaboration pour le développement d'API, largement utilisé pour tester, documenter et partager les APIs. Il simplifie et accélère les processus de création, de test et de maintenance des APIs grâce à ses nombreuses fonctionnalités. Voici un aperçu de ses principales caractéristiques :

- **Interface utilisateur intuitive** : Postman offre une interface graphique conviviale permettant de créer et d'organiser des requêtes HTTP facilement, sans avoir besoin de ligne de commande.
- **Tests automatisés** : Les utilisateurs peuvent écrire des tests pour automatiser les validations des réponses d'API, en utilisant JavaScript pour vérifier les résultats attendus.
- **Collections** : Les requêtes peuvent être organisées en collections, facilitant le partage et la réutilisation. Les collections peuvent être exportées et importées, permettant une collaboration aisée entre les développeurs.
- **Environnements** : Postman permet de gérer plusieurs environnements (développement, test, production), facilitant la configuration des variables spécifiques à chaque environnement.
- **Documentation d'API** : Postman génère automatiquement une documentation pour les collections d'API, qui peut être partagée avec l'équipe ou rendue publique.
- **Mock Servers** : Les mock servers permettent de simuler les réponses des APIs, ce qui est utile pour le développement et les tests avant que l'API réelle ne soit disponible.
- **Intégrations** : Postman s'intègre avec des outils CI/CD comme Jenkins et Travis CI, ainsi qu'avec des plateformes de gestion de version comme GitHub et GitLab.
- **Collaboration en équipe** : Postman offre des fonctionnalités de collaboration en temps réel, permettant aux équipes de travailler ensemble sur des collections, des environnements et des tests.

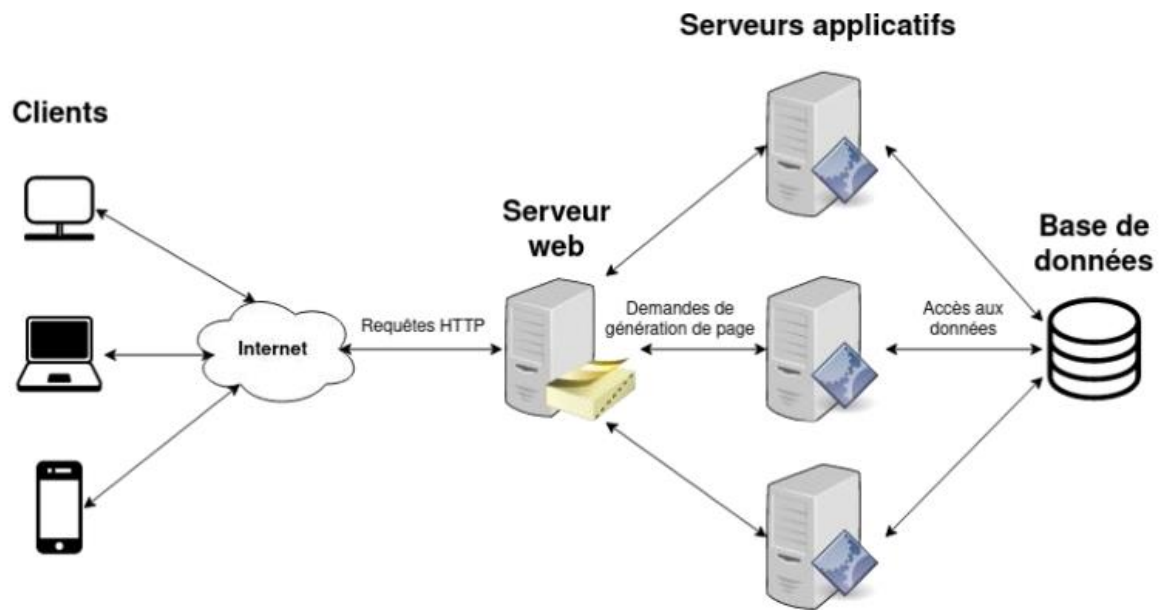
Postman est un outil puissant pour tout développeur travaillant avec des APIs, offrant une gamme complète de fonctionnalités pour simplifier et améliorer le processus de développement, de test et de documentation des APIs.



StarUML est un outil de modélisation UML (Unified Modeling Language) qui offre des fonctionnalités avancées pour la conception et la documentation de logiciels. Utilisé par les développeurs et les architectes logiciels, StarUML permet de créer divers types de diagrammes UML, tels que les diagrammes de cas d'utilisation, les diagrammes de classes, les diagrammes de séquence, et bien plus encore. Grâce à son interface intuitive et à ses outils puissants, StarUML facilite la visualisation des concepts et des structures logicielles, aidant ainsi les équipes à communiquer efficacement et à aligner leurs efforts tout au long du cycle de développement. Les fonctionnalités de personnalisation et les options d'exportation de StarUML permettent aux utilisateurs de créer des modèles précis et détaillés, essentiels pour la planification, l'analyse et la documentation des systèmes logiciels complexes.

3-Architecture applicative :

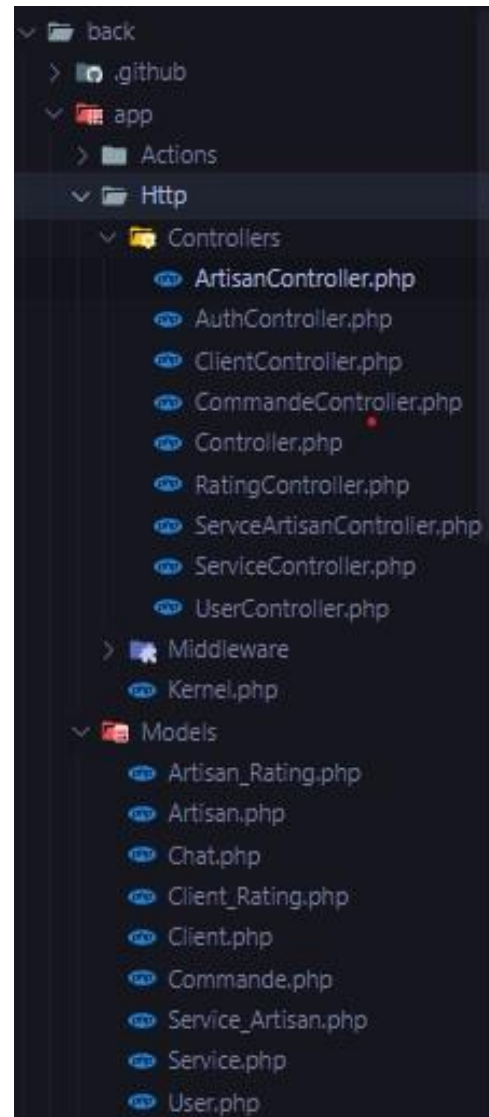
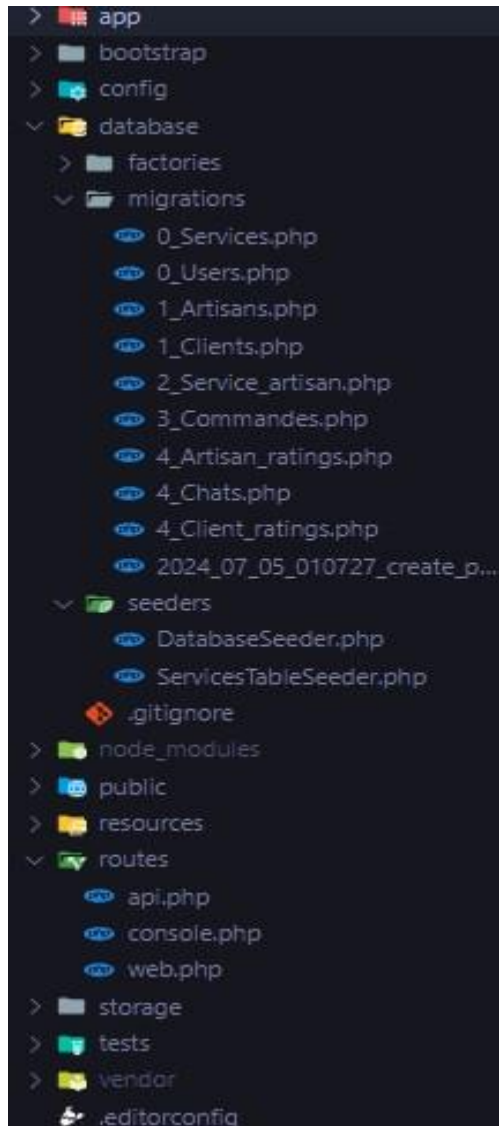
L'objectif d'une architecture applicative est de fournir une structure au système d'information en regroupant les applications en blocs interconnectés. Elle vise également à décrire techniquement les applications, les flux de données et les messages échangés entre ces applications.



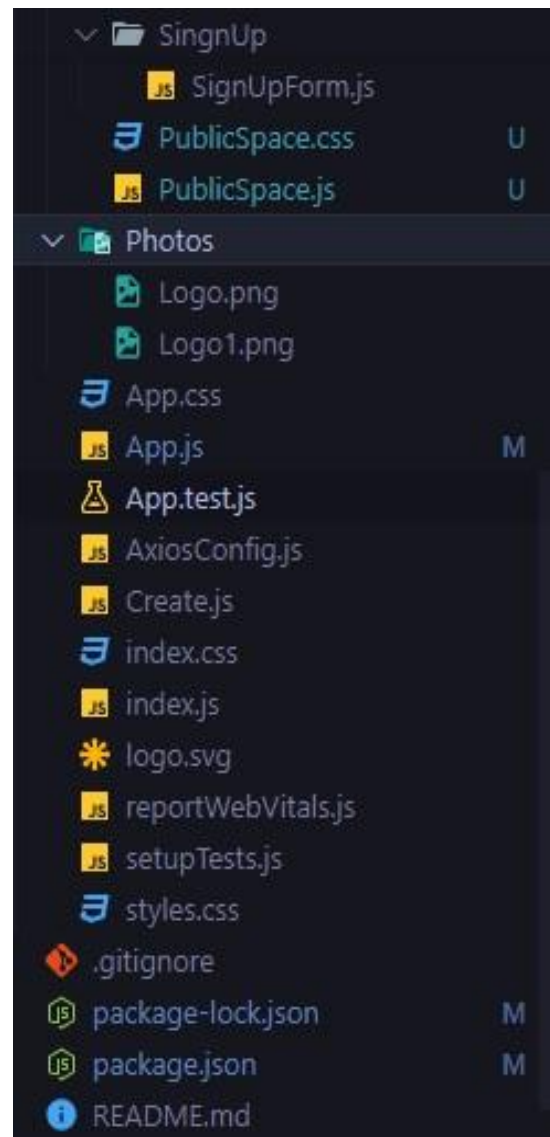
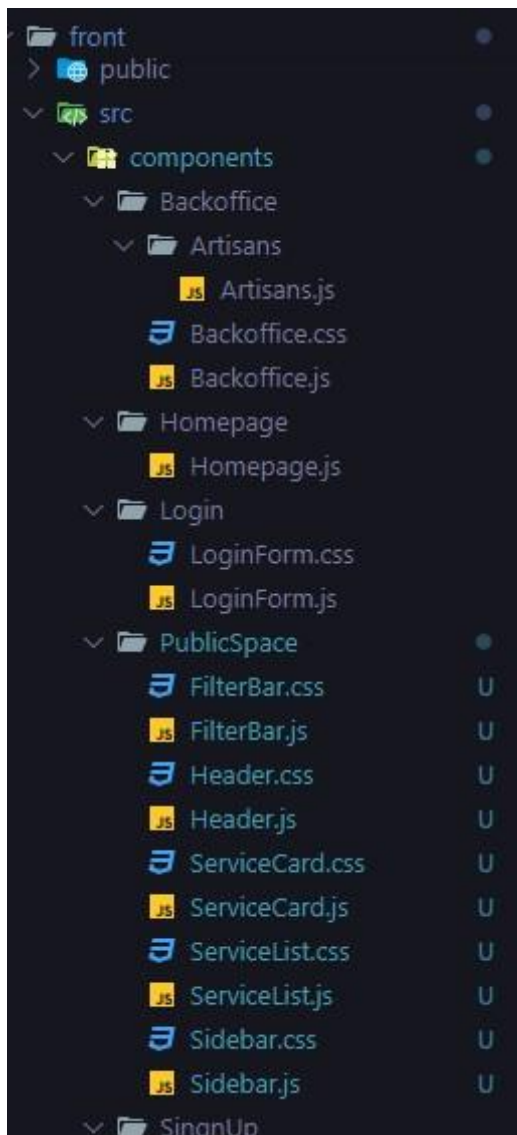
4-Arborescence projet :

L'IDE **Visual Studio Code** permet de visualiser et de structurer l'arborescence d'un projet de manière logique et hiérarchisée, en utilisant une structure algorithmique d'arbre. La figure suivante représente la représentation graphique de cette arborescence dans l'IDE **Visual Studio Code**.

Backend



Frontend



Conclusion :

En conclusion, l'étude technique est une étape cruciale dans le développement d'une application. Elle permet d'analyser et de définir les besoins, de concevoir l'architecture technique et de choisir les outils et langages appropriés.

Chapitre 4 :

« Modélisation et conception »

Dans ce chapitre, nous débutons par une introduction aux divers outils logiciels et langages de modélisation utilisés. Ensuite, nous approfondissons les concepts des diagrammes de cas d'utilisation, des diagrammes de classes et des diagrammes de séquences, tout en présentant certaines interfaces de notre application.

1-Methodes et outils de modélisation :

1.1-Langage de modélisation (UML) :

Dans la conception de notre système, nous avons privilégié une approche orientée objet, considérée comme incontournable dans le développement d'applications. Afin de présenter de manière optimale l'architecture de notre système, nous avons choisi d'utiliser UML, le langage de modélisation le plus largement utilisé. UML, en tant que norme de modélisation objet, permet de décrire et de documenter un système d'information. En adoptant ce langage, nous garantissons la réalisation des objectifs fondamentaux de la modélisation objet incluant les éléments suivants :

- Développer une conception structurée pour une application.
- Favoriser la communication entre les acteurs d'un projet informatique.
- Assurer la coordination des activités entre les différents intervenants.
- Gérer le processus d'évolution d'un projet informatique.
- Offrir des outils standardisés qui intègrent plusieurs aspects de la conception.

1.2-Les outils de modélisation :

Argo UML : ArgoUML est un outil open source de modélisation et création de diagrammes qui inclut le support de tous les diagrammes UML 1.4 standards. Il tourne sur n'importe quelle plateforme Java standard et il est disponible en 10 langues.

Star UML : StarUML est un logiciel de modélisation UML (Unified Modeling Language) open-source qui permet de concevoir et de visualiser les diagrammes de classes, les diagrammes de séquences et les diagrammes de cas d'utilisation. Nous avons Choisi cette solution pour sa facilité d'utilisation et son adaptabilité à notre projet.

2- Sprint 1 : Gestion Administrateur

2.1- Description du sprint :

Ce module permet un l'administrateur de gérer les demandes de la création du compte artisan, donne la possibilité à l'administrateur d'approuver des comptes ou supprimé un bien.

Description	
Objectif	Consulter les demandes de création et gérer le compte
Contrainte	Pas de contrainte
Prérequis	Cahier de charge
Etape de phase	Se référer au diagramme de GANTT du chapitre planification du projet.
Livrables de sortie	Exécutable du module.
Dépendance	N/A
Critère de fin de phase	Validation de l'encadrent.

Table 5 descripción sprint gestión administrate

2.2- Backlog du sprint:

- Authentifier.
- Afficher les informations du profil.
- Afficher les informations des utilisateurs.
- Consulter et gérer les demandes (approuver/ éditer ou supprimer)

2.3- Règles de gestion du sprint :

Règle	Description
RG01	Chaque administrateur ne peut avoir qu'un seul compte
RG02	Chaque administrateur doit se connecter avec son login et mot de passe fournis, afin d'accéder au backoffice.
RG03	L'administrateur ne peut pas réinitialiser son mot de passe, en cas d'oubli il contacte le super admin
RG04	L'administrateur est le seul utilisateur qui peut supprimer un bien.

Table 6 Règles gestion sprint administrateur

2.4- Analyse et conception :

La figure ci-dessous illustre le scénario d'utilisation du sprint "Gestion de l'administrateur", qui vise à identifier les différentes façons dont l'administrateur interagit avec le système.

1-Diagramme de cas d'utilisation :

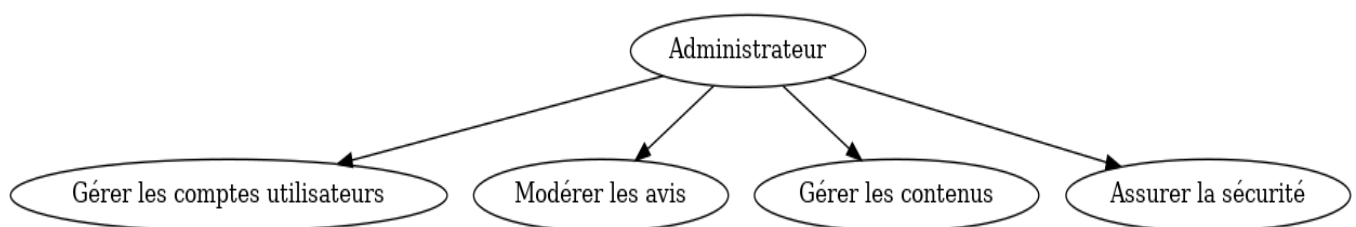


Figure 6 usecase administrateur

Libellé	Acteur	Description
S'authentifier	Admin	L'administrateur se connecte pour accéder à son backoffice pour gérer les biens.
Gestion des comptes	Admin	L'administrateur ajoute / supprime / modifie les informations des comptes.
Gestion des demandes	Admin	L'administrateur ajoute supprime ou modifie les demandes.

Table 7 Description cas d'utilisation admin

2-diagramme de classe :

Le schéma ci-dessous représente le diagramme de classe du sprint "Gestion d'Administrateur", qui présente les différentes classes du module :

3- Sprint 2 : Gestion Artisan

3.1- Description du sprint :

Ce module permet à l'artisan de se connecter et gérer son compte, tel que son profile, ainsi que la gestion de ces services et ses plans (ajout, modification, suppression).

Description	
Objectif	Tous les artisans doivent avoir un compte pour pouvoir ajouter des services et des plans Gestion de compte, et publication des services
Contrainte	Respecter les règles de gestion.
Prérequis	Accéder à la plateforme (landing page) du site de l'application via l'url
Etape de phase	Se référer au diagramme de GANTT du chapitre planification du projet.
Livrables de sortie	Exécutable du module.
Dépendance	N/A
Critère de fin de phase	Validation de l'encadrent.

Table 8 Description cas d'utilisation artisan

3.2-Backlog du sprint :

- S'authentifier.
- Afficher les informations du profil.
- Consulter/modifier le profil.
- Gérer les services.
- Gérer les plans.

3.3-Règles de gestion du sprint :

Règle	Description
RG01	Chaque artisan a la possibilité d'avoir un seul compte
RG02	Chaque artisan doit se connecter avec un login et mot de passe pour qu'ils puissent accéder à ces services.
RG03	Chaque artisan doit se connecter avec un login et mot de passe pour qu'ils puissent accéder à ces plan.

Table 9 Règles gestion sprint artisan

3.4-Analyse et conception :

1. Diagramme de cas d'utilisation :

Le schéma illustre le cas d'utilisation du sprint "Gestion de l'artisan"

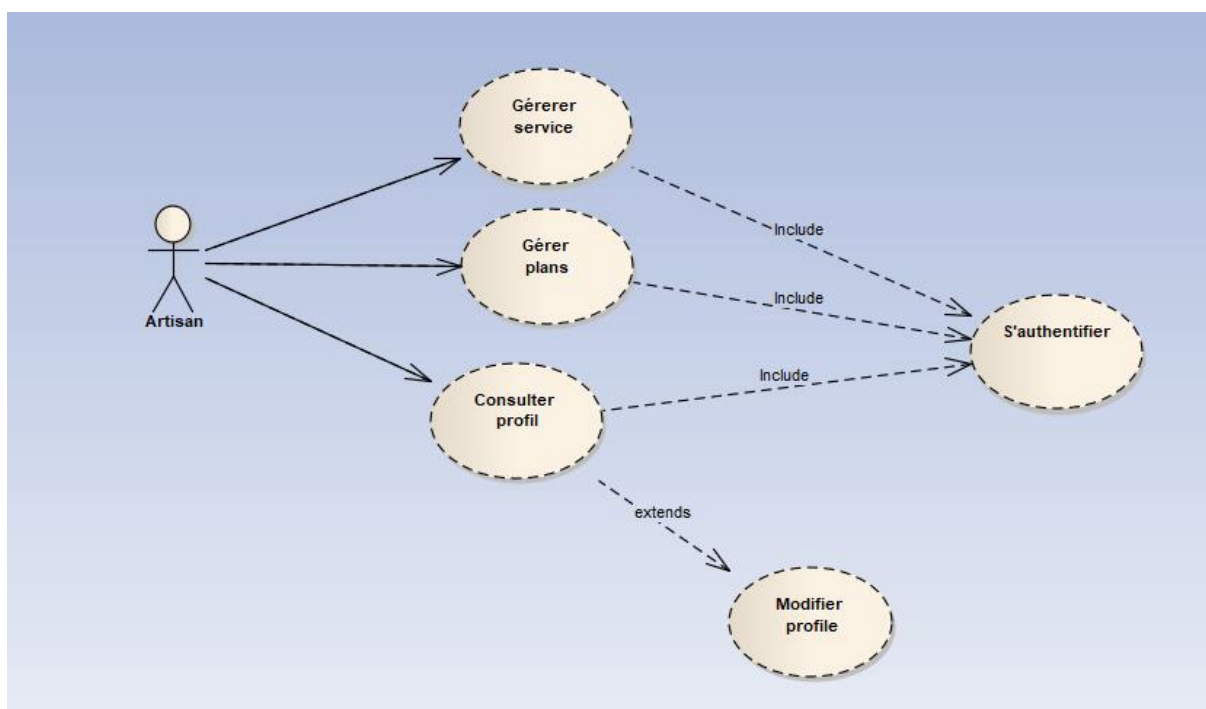


Figure 7 usecase artisan

Libellé	Acteur	Description
S'authentifier	Artisan	L'artisan se connecte pour accéder à la liste de ses services et plans
Gestion des informations	Artisan	L'artisan peut consulter et modifier ses informations personnels.
Ajout des services	Artisan	L'artisan ajoute un service.
Ajout des plans	Artisan	L'artisan ajoute un plan.
Suppression des services	Artisan	L'artisan supprime un service.
Suppression des plans	Artisan	L'artisan supprime un plan.

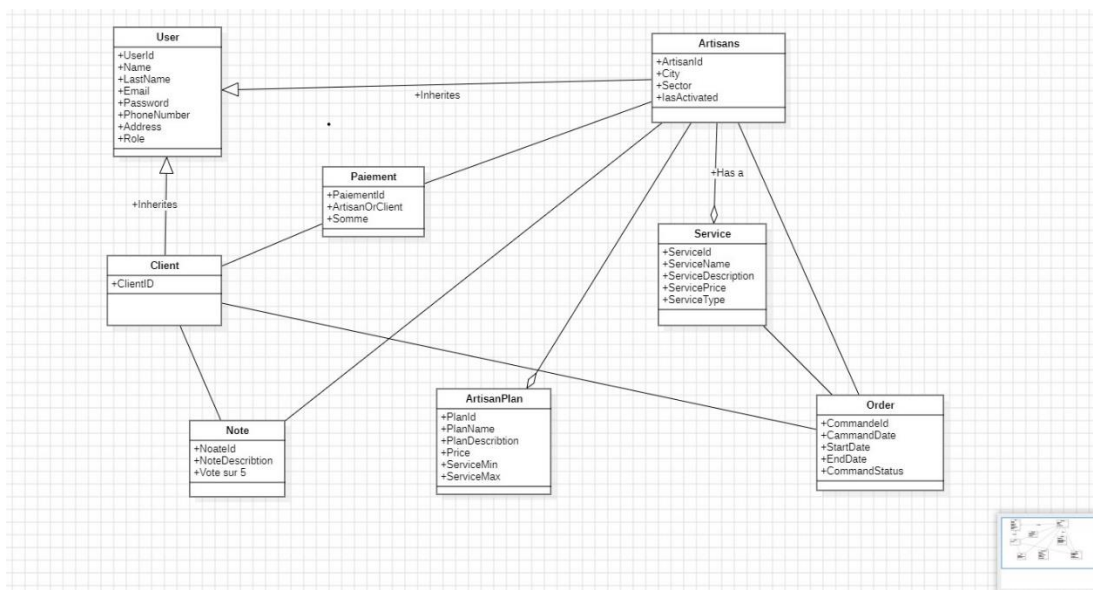
Table 10 Description de cas d'utilisation d'artisan

2-diagramme de classe :

Le diagramme de classe ci-dessous représente le sprint "Gestion de l'Artisan", mettant en évidence les classes du module ainsi que les différentes relations entre elles

4-Diagramme de classe Générale :

Outil de modélisation utilisé en génie logiciel pour représenter la structure statique d'un système logiciel. Il permet de visualiser les classes, les attributs, les méthodes et les relations entre les différentes entités du système. Dans notre cas, le diagramme de classe présenté illustre la gestion des services Artisans.



Conclusion :

En combinant ces outils et langages de modélisation avec une approche structurée de développement par sprints, nous pouvons garantir une planification minutieuse et une exécution efficace de chaque phase du projet. Cette méthodologie nous permet de répondre de manière agile aux besoins changeants du client, tout en maintenant une qualité élevée dans les livrables du projet.

Conclusion générale :

Le projet Lmoqaf a été développé pour offrir une plateforme robuste et conviviale qui facilite la mise en relation entre les clients et les artisans. Grâce à une interface intuitive et une large gamme de services artisanaux, Lmoqaf vise à transformer l'expérience utilisateur dans ce domaine. Notre architecture en couches permet de séparer clairement les responsabilités, ce qui facilite la maintenance et l'évolution du système. Nous utilisons des frameworks modernes comme Laravel et React, ainsi que des outils de modélisation UML, pour assurer une base solide au développement et à la gestion de l'application. Nous avons également mis un fort accent sur la sécurité des données, la qualité des services, et la flexibilité pour les artisans. Ceux-ci peuvent personnaliser leurs profils et gérer leurs disponibilités de manière autonome. Les fonctionnalités de messagerie intégrée, de notifications en temps réel, et de gestion des avis contribuent à une expérience utilisateur fluide et transparente. En plus de ces aspects, Lmoqaf intègre des mécanismes de protection avancés pour garantir la confidentialité et l'intégrité des informations personnelles et professionnelles échangées sur la plateforme. La conception ergonomique de l'application permet une navigation aisée et une accessibilité optimale pour tous les utilisateurs, indépendamment de leur niveau de compétence technologique.

Grâce à une équipe dédiée à l'innovation et à l'amélioration continue, Lmoqaf s'adapte constamment aux besoins évolutifs du marché et des utilisateurs. En intégrant des mises à jour régulières et des nouvelles fonctionnalités, nous nous assurons que la plateforme reste à la pointe de la technologie et répond aux attentes les plus exigeantes. Nous avons également établi des partenariats stratégiques avec divers acteurs du secteur pour enrichir notre offre de services et garantir une expérience utilisateur exceptionnelle.

Lmoqaf ne se contente pas de connecter les clients et les artisans; il vise également à créer une communauté dynamique et engagée. Les utilisateurs peuvent partager leurs expériences, recommander des artisans et participer à des

forums de discussion. Cette dimension communautaire renforce la confiance et la transparence sur la plateforme, tout en favorisant l'entraide et la collaboration.

Nous avons également mis en place un système de support client réactif et efficace pour répondre à toutes les questions et résoudre les problèmes rapidement. Les utilisateurs peuvent accéder à une base de connaissances complète, des tutoriels vidéo et un service d'assistance en direct pour obtenir de l'aide à tout moment.

La flexibilité de Lmoqaf permet aux artisans de s'inscrire facilement, de créer et de gérer leurs profils, de fixer leurs tarifs, et de planifier leurs disponibilités en fonction de leurs besoins. Les clients, quant à eux, peuvent facilement rechercher des artisans par catégorie, lire des avis, comparer des services et réserver des prestations en quelques clics.

En termes de sécurité, nous utilisons des protocoles de chiffrement de pointe pour protéger les données sensibles et garantir que toutes les transactions se déroulent de manière sécurisée. Nous respectons également les réglementations en matière de protection des données et nous nous engageons à maintenir les normes les plus élevées en matière de confidentialité.

En résumé, Lmoqaf se positionne comme une solution intégrée et efficace, répondant aux besoins variés des utilisateurs et créant un environnement fiable et sécurisé pour les transactions artisanales. Avec une interface utilisateur moderne, une architecture robuste, des fonctionnalités avancées et un engagement envers l'excellence, Lmoqaf est destiné à devenir le leader dans le domaine des services artisanaux en ligne.