

Classes Préparatoires Intégrées (CPI)

2ème année

2023/2024

Conception d'une plateforme de Freelance WorkWave

Réalisé par :

- Achelache Aymen
- Agsous Haroun
- Berguella Abderrahim
- Djedidi Hamza
- Laouici Heythem
- Saoual Zakarya

Encadré par :

Tagmouni Abdelkader

Remerciements

À Monsieur Abdelkader TAGMOUNI

Nous tenons à exprimer notre plus profonde gratitude à Monsieur Abdelkader TAGMOUNI pour son encadrement précieux et son soutien indéfectible tout au long de ce projet. Ses conseils avisés et sa bienveillance ont été des sources d'inspiration et de motivation constantes, sa patience et sa disponibilité ont permis de surmonter les obstacles et d'aboutir à un résultat de qualité.

Nous lui sommes reconnaissants pour son expertise, son sens du partage et sa foi en nos capacités, ses encouragements nous ont permis de repousser nos limites et d'accomplir de réels progrès.

Nous gardons un souvenir précieux de nos échanges et de l'expérience enrichissante que nous avons acquise à ses côtés et nous lui adressons nos plus sincères remerciements pour son rôle déterminant dans la réussite de ce projet.

Aux membres de l'équipe

Nous tenons également à remercier chaleureusement l'ensemble des membres de l'équipe qui ont contribué à la réalisation de ce projet.

Votre collaboration, votre engagement et votre esprit d'équipe ont été des éléments clés de la réussite de ce projet.

Aux anciens étudiants

Nous remercions également les anciens étudiants qui ont bien voulu nous prodiguer leurs précieux conseils et partager leur expérience. Vos encouragements et votre soutien ont été très appréciés.

Conclusion

Ce projet a été une expérience enrichissante qui nous a permis d'apprendre et de grandir. Nous sommes reconnaissants à tous ceux et celles qui ont contribué à sa réussite.

**Respectueusement,
WorkWave Team.**

Table des matières

Conception d'une plateforme de Freelance	1
WorkWave	1
Introduction	7
Chapitre 1 : Spécification des besoins	8
1.1 Introduction	8
1.2 Présentation du client	8
1.2.1 Les Freelancers (Utilisateurs Externes) :	8
1.2.2 Les Clients (Utilisateurs Externes) :	8
1.2.3 L'Équipe WorkWave (Utilisateurs Internes) :	8
1.3 Problématique et solutions	9
1.3.1 Problématique	9
1.3.2 Solutions	9
1.4 Étude de l'existence et critique	10
1.4.1 Sites Web similaires à notre projet	10
1.4.2 Critiques et commentaires des utilisateurs	10
1.5 Analyse des besoins	11
1.5.1 Besoins Fonctionnelles	11
1.5.2 Besoins non fonctionnels	13
Chapitre 2 : Conception	14
2 Introduction	14
2.1 Diagramme de cas d'utilisation	14
2.2 Diagramme de Séquence	20
2.2.1 Postuler pour un projet (Freelancer) :	21
2.2.2 Créer un compte (Freelancer/Client)	23
2.2.3 Acheter un service (Client) :	24
2.2.4 Gestion des Contrat (Admin)	25
2.3 Diagramme de Classes	27
2.4 Conclusion	29
Chapitre 3 : Realisation	30

3	Outils de développement :	30
3.1	Figma :	30
3.2	Adobe illustrator :	31
3.3	Adobe Photoshop :	31
3.4	JavaScript :	32
3.5	Express :	32
3.6	MongoDB :	33
3.7	React :	33
3.8	Tailwind CSS:	34
3.9	Postman:	34
3.10	Visual Studio Code (VSCode):	35
3.11	Lucychart:	35
	35
4	Présentation de l'Application :	36
4.1	Page d'accueil.....	37
4.2	Sign Up :	38
4.3	Mon Compte :	39
4.4	Mes Projets	40
4.4.1	Working on (projet en cours) :	41
4.4.2	Histrory (Historique) :	41
4.5	Mes Services	42
4.6	Page des services	43
	Conclusion Générale	44
	Bibliographie/webographie.	45

Table des figures

Figure 1: Diagramme de cas d'utilisation	15
Figure 2: Diagramme de Séquence : Postuler pour un projet	21
Figure 3: Diagramme de Séquence : Créer un compte.....	23
Figure 4: Diagramme de Séquence : Acheter un service	24
Figure 5: Diagramme de Séquence : Gestion des contrats.....	25
Figure 6: Diagramme de classes	27
Figure 7: Logo Figma.....	30
Figure 8: Logo Adobe Illustrator	31
Figure 9: Logo Adobe Photoshop.....	31
Figure 10: Logo JavaScript	32
Figure 11: Logo Express	32
Figure 12: Logo MongoDB	33
Figure 13: Logo React	33
Figure 14: Logo Tailwind CSS	34
Figure 15: Logo Postman	34
Figure 16: Logo VS Code	35
Figure 17: Logo Lucychart.....	35
Figure 18: Page d'accueil de l'application.....	36
Figure 19: Page Sign Up	38
Figure 20: Page Mon Compte	39
Figure 21: Page Mes Projets	40
Figure 22: Page Mes Services.....	42
Figure 23: Page Des Services	43

Liste des tableaux

Tableau 1 : Besoins Fonctionnelles.....	12
Tableau 2: Besoins Non Fonctionnelles.....	13

Introduction

L'objectif de ce rapport est de présenter le projet de création d'une plateforme de Freelance. Ce projet a été réalisé par notre équipe de développement. Le but du projet est de concevoir et développer un site Web facilitant les relations entre les gens spécialisé et les clients tout en répondant aux besoins de chacun des Freelancers et clients.

Le rapport est divisé en trois chapitres : Analyse des besoins, Conception et Implémentation.

Dans le premier chapitre, nous avons détaillé les besoins du client, les fonctionnalités requises pour le projet Plateforme Freelance et les contraintes techniques à prendre en compte. Nous avons effectué une analyse complète des besoins du client, en nous appuyant sur des entretiens avec le client et sur des études de marché pour déterminer les fonctionnalités indispensables à la réalisation efficace du projet. Nous avons également pris en compte les contraintes techniques telles que la sécurité, la performance et la maintenance.

Le deuxième chapitre décrit la conception de notre solution freelance. Nous avons présenté notre analyse fonctionnelle, notre architecture technique et notre modèle de données. Nous avons également décrit les différents choix technologiques que nous avons effectués pour la réalisation de ce projet, en concevant différents diagrammes tels que les diagrammes de séquences, de classes et de cas.

Le troisième chapitre décrit l'implémentation de notre solution. Nous avons présenté les différentes étapes de développement, les outils utilisés et les défis rencontrés. Nous avons également présenté les différentes fonctionnalités implémentées sur le projet final. Nous avons utilisé des méthodes de développement agiles pour permettre une flexibilité maximale dans la gestion du projet. De plus, nous avons utilisé des outils de développement modernes pour accélérer le processus de développement. Nous avons testé chaque fonctionnalité du projet pour garantir son bon fonctionnement et avons effectué des mises à jour régulières pour améliorer le projet.

Nous avons terminé ce rapport en récapitulant les différentes phases du projet Plateforme Freelance, en abordant les défis rencontrés, les résultats obtenus, ainsi que les perspectives à venir. Nous avons mis en avant l'importance de ce projet dans la réponse aux besoins actuels des Freelances/Clients et dans la proposition d'une solution novatrice pour la gestion des activités du Freelance. En outre, nous avons exposé les opportunités futures visant à améliorer et à élargir cette solution, en intégrant de nouvelles fonctionnalités pour s'adapter aux évolutions du marché de Freelance

Chapitre 1 : Spécification des besoins

1.1 Introduction

Dans ce chapitre, nous aborderons les spécifications des besoins pour la création de notre Plateforme. Il est essentiel de comprendre les exigences et les besoins de nos utilisateurs afin de développer une plateforme efficace et réussie. Nous commencerons par examiner la problématique qui a conduit à la création de cette plateforme, puis nous procéderons à l'analyse des besoins et à l'étude de l'existence ainsi que sa critique.

1.2 Présentation du client

Notre client n'est pas seulement quelqu'un qui a besoin d'un service, mais aussi quelqu'un qui en offre. Notre but est de toucher un maximum de clients afin de pouvoir offrir notre service à un nombre vaste d'individus. C'est pour ça notre plateforme est basée sur deux types de clients : externe, et un seul interne

1.2.1 Les Freelancers (Utilisateurs Externes) :

Les Freelancers constituent la base de notre plateforme. Ils sont des professionnels indépendants dans divers domaines tels que le graphisme, la rédaction, la programmation, etc. Ils utilisent WorkWave pour présenter leurs compétences, trouver des opportunités de travail, et collaborer avec des clients.

1.2.2 Les Clients (Utilisateurs Externes) :

Les clients, entreprises ou particuliers, cherchent des Freelancers qualifiés pour leurs projets. WorkWave leur offre une plateforme conviviale pour trouver les talents adaptés à leurs besoins, établir des collaborations fructueuses, et gérer efficacement leurs projets.

1.2.3 L'Équipe WorkWave (Utilisateurs Internes) :

Notre équipe interne gère et développe activement la plateforme. Les responsabilités comprennent la maintenance du site, l'amélioration continue des fonctionnalités, et la satisfaction globale des utilisateurs. L'équipe s'efforce de fournir une expérience transparente et productive tant pour les freelancers que pour les clients. Chaque catégorie d'utilisateurs a des besoins spécifiques, et WorkWave vise à offrir une plateforme flexible et efficace pour répondre à ces besoins et faciliter la collaboration entre freelancers et clients.

1.3 Problématique et solutions

1.3.1 Problématique

Chaque jour, les individus à travers le monde rencontrent des défis considérables pour identifier des professionnels qualifiés pour divers services, tandis que d'autres éprouvent des difficultés à trouver un emploi. Bien que des plateformes de freelance existent, les transactions en ligne et la confiance envers des prestataires inconnus ne sont pas encore pleinement établies dans de nombreuses régions. Ce manque de confiance et de sécurité dans les transactions en ligne entrave l'accès à des services fiables et à des opportunités d'emploi pour beaucoup. Ainsi, il devient crucial de comprendre et d'aborder ces obstacles afin de faciliter les connexions entre les prestataires de services et les demandeurs d'emploi, et de garantir des interactions en ligne sécurisées et dignes de confiance..

1.3.2 Solutions

La plateforme internationale WorkWave est une solution idéale à tous ces problèmes. Développée par des étudiants algériens, elle propose une large gamme d'opportunités d'emploi dans divers secteurs et domaines d'activité, permettant ainsi à nos utilisateurs de trouver la niche dans laquelle ils sont compétents et de démontrer leur valeur. WorkWave assurera également aux clients ayant besoin de services qu'ils trouveront les personnes qualifiées pour les tâches requises.

En outre, en tant que plateforme globale, WorkWave simplifiera les paiements en ligne en intégrant divers moyens de paiement sécurisés et en respectant les normes internationales de sécurité. Cela permettra de renforcer la confiance entre les utilisateurs, qu'ils soient du même pays ou de régions différentes.

WorkWave offrira des outils performants pour faciliter la gestion des projets clients. Grâce à une interface conviviale et à des fonctionnalités de collaboration avancées, les professionnels pourront facilement travailler sur des projets depuis chez eux, tout en restant connectés avec leurs clients en temps réel. Les utilisateurs pourront ainsi optimiser leur temps, améliorer leur productivité et générer des revenus sans les contraintes des déplacements physiques.

En somme, WorkWave vise à créer un environnement fiable et efficace où les freelances peuvent prospérer et où les clients peuvent trouver les services dont ils ont besoin en toute confiance et simplicité.

1.4 Étude de l'existence et critique

L'objectif de cette étude est d'analyser les sites Web concurrents similaires à notre projet, c'est-à-dire des sites Web de Freelance comme Fiverr , Upwork etFreelancer, ainsi que les critiques et les commentaires des utilisateurs pour comprendre leurs besoins et leurs préférences

1.4.1 Sites Web similaires à notre projet

1. Fiverr :

Fiverr est une plateforme de freelance très populaire qui permet aux professionnels de proposer leurs services sous forme de "gigs". Les utilisateurs peuvent trouver des freelances pour une grande variété de tâches, allant de la conception graphique à la rédaction, en passant par le marketing digital et le développement web. La simplicité d'utilisation de Fiverr et son modèle tarifaire flexible en ont fait une option attrayante pour les freelances et les clients du monde entier.

2. Upwork :

Upwork est l'une des plus grandes plateformes de freelance au monde, offrant une vaste gamme de services professionnels. Les freelances peuvent créer des profils détaillés et postuler à des projets publiés par des clients. Upwork propose également des outils de gestion de projet, de suivi du temps et de facturation, ce qui facilite la collaboration entre freelances et clients. La réputation solide d'Upwork et sa large base d'utilisateurs en font un concurrent sérieux.

3. Freelancer :

Freelancer est une autre plateforme internationale de freelance, permettant aux professionnels de proposer leurs compétences et de postuler à des projets dans divers domaines. Les clients peuvent publier des projets et organiser des concours pour trouver le meilleur freelance pour leurs besoins. Freelancer offre également des options de gestion de projet et de paiement sécurisées. Avec sa présence mondiale et sa grande variété de services, Freelancer est une plateforme bien établie dans le secteur du travail indépendant.

1.4.2 Critiques et commentaires des utilisateurs

Problèmes de communication : Les utilisateur rencontre des difficultés à utiliser le chat du site et des bugs peuvent survenir entre les clients et les prestataires, notamment en ce qui concerne les attentes, les délais et les révisions des projets.

Manque de qualité des profils : quelques utilisateurs négligent la configuration de leurs profile où créer des profils incomplets ou inexacts, ce qui peut entraîner des difficultés lors de la mise en relation avec des clients ou des prestataires.

Manque de soutien pour les nouveaux arrivants : Notre plateforme se développe à rythme élevé pour surpasser les concurrents, ce qui rend l'utilisation de notre plateforme un peu difficile aux nouveaux arrivants sur le domaine.

Problèmes de paiement: Certains utilisateurs ont signalé des difficultés avec les systèmes de paiement, notamment des retards dans les transactions et des problèmes de conversion de devises. Ces problèmes peuvent affecter la confiance des utilisateurs et leur satisfaction globale.

1.5 Analyse des besoins

1.5.1 Besoins Fonctionnelles

Les besoins fonctionnels sont les exigences spécifiques concernant les fonctionnalités et les performances de notre site. Ils décrivent ce que le système doit accomplir du point de vue des fonctionnalités et des actions qu'il doit permettre aux utilisateurs de réaliser

Besoin	Fonctionnalité	Description
Interface	Interface Freelancer <ul style="list-style-type: none"> • Permettre aux freelancers de créer et gérer leur profil professionnel. • Offrir la possibilité de rechercher et postuler à des projets correspondant à leurs compétences. • Permettre aux freelancers de communiquer avec les clients • Créer des services personnalisés pour des tâches spécifiques et permettre aux clients de soumettre des demandes pour ces services. 	Afin de mettre en avant leurs compétences, expériences, et projets précédents et de partager des fichiers liés au projet et aussi partager des services pour des tâches spécifiques.
	Interface Client <ul style="list-style-type: none"> • Permettre aux clients de créer et gérer leur profil, décrivant leurs besoins, projets, et budgets. • Offrir une fonction de recherche avancée • Faciliter le processus de sélection de freelancers et la gestion des projets. • Offrir la possibilité aux clients de soumettre des demandes pour des services spécifiques proposés par les freelancers. 	Afin de décrire leurs besoins, projets, et budgets et trouver des freelancers qualifiés, ou bien postuler pour des tâches de service proposées par le freelancer.
	Interface Administrateur <ul style="list-style-type: none"> • Gérer les inscriptions des utilisateurs et les problèmes soumis par ces derniers. • Surveiller l'activité sur la plateforme, y compris les transactions entre utilisateurs. • Fournir des outils de modération pour garantir le respect des politiques et des normes de la plateforme. 	Pour assurer pleinement la conformité avec les politiques de la plateforme, garantir un environnement sûr et éthique, ainsi que sécuriser les transactions effectuées sur la plateforme..
Gestion des Projets	<ul style="list-style-type: none"> • Intégrer un système de messagerie pour faciliter la communication en cours de projet et permettre aux freelancers et aux clients de créer, gérer, et suivre l'avancement des projets . 	Assurer le fonctionnement organisé de la plateforme
Commentaires et Évaluations	<ul style="list-style-type: none"> • Fournir la possibilité d'ajouter des commentaires et des évaluations pour chaque projet, contribuant à la transparence et à la confiance sur la plateforme. 	Permet aux utilisateurs de savoir avec qui ils travaillent

Tableau 1 : Besoins Fonctionnelles

1.5.2 Besoins non fonctionnels

Besoin	Description
Sécurité des Données	<ul style="list-style-type: none">• Mettre en place des mesures de sécurité robustes pour protéger les données personnelles et professionnelles des utilisateurs.
Performances du Site Web	<ul style="list-style-type: none">• Assurer des performances rapides et fiables du site web pour garantir une expérience utilisateur optimale.
Interface Utilisateur Intuitive	<ul style="list-style-type: none">• Concevoir une interface utilisateur intuitive et conviviale pour faciliter la navigation et l'utilisation de la plateforme.
Compatibilité Multiplateforme	<ul style="list-style-type: none">• Garantir que la plateforme soit compatible avec différents navigateurs web et systèmes d'exploitation pour assurer une accessibilité maximale.
Support Technique	<ul style="list-style-type: none">• Assurer la disponibilité du support technique pour résoudre les problèmes techniques, répondre aux questions des utilisateurs, et assister les freelancers et clients

Tableau 2: Besoins Non Fonctionnelles

Chapitre 2 : Conception

2 Introduction

Le processus de développement de WorkWave comprend de nombreuses étapes cruciales. Parmi celles-ci, la phase de conception revêt une importance particulière, car elle permet de définir les fonctionnalités et les interactions essentielles de la plateforme. De même, la modélisation joue un rôle clé en représentant visuellement ces éléments de manière structurée. Les diagrammes de conception sont des outils essentiels pour visualiser ces aspects de manière claire et organisée. Ce chapitre explorera les différents types de diagrammes de conception utilisés dans le développement de WorkWave, notamment le diagramme de classe et le diagramme de séquence.

2.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML (Unified Modeling Language) qui modélise le comportement d'un système et permet de capturer les exigences du système. Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée de notre plateforme, ils identifient également les interactions entre le système et ses acteurs.

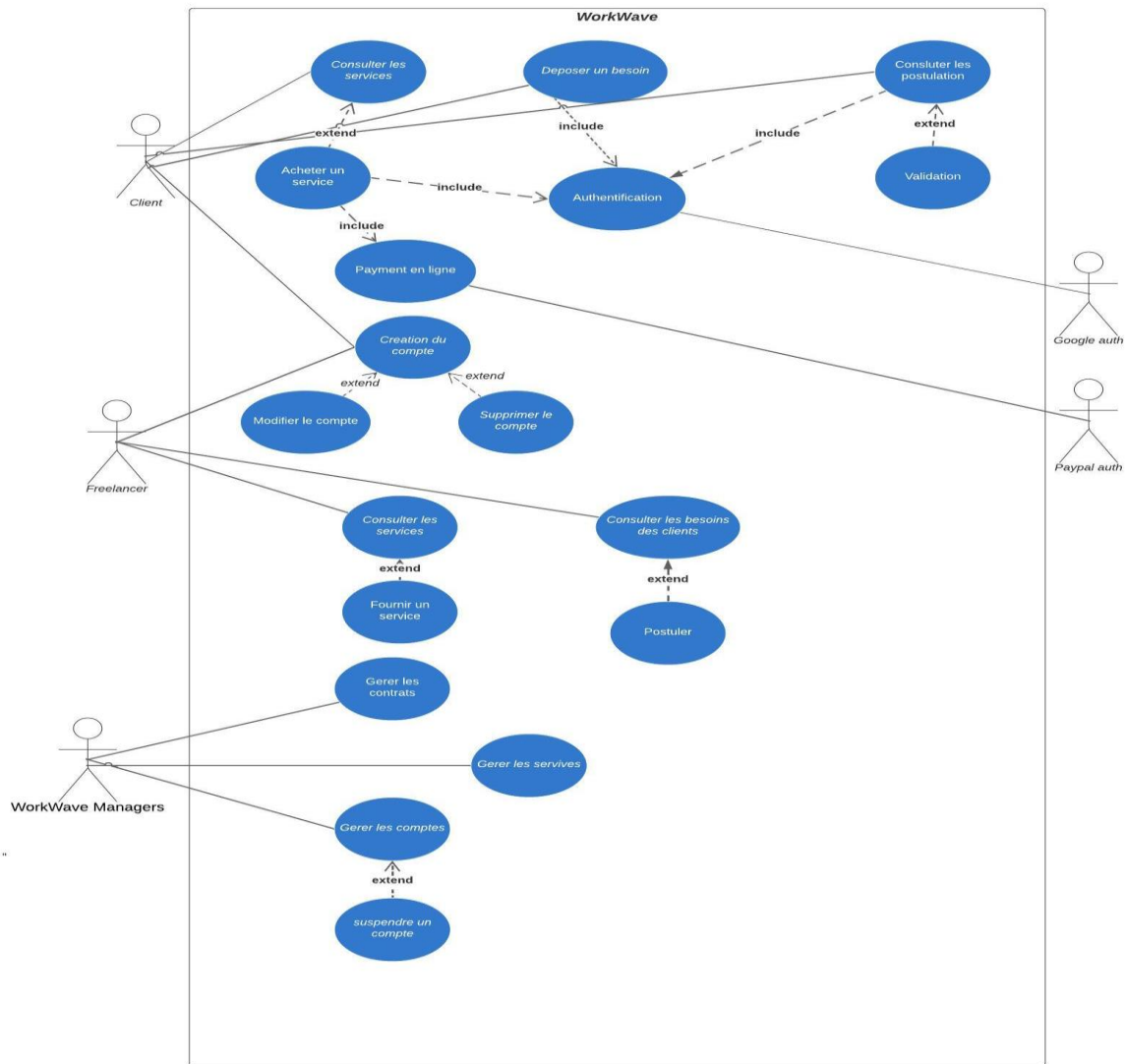


Figure 1: Diagramme de cas d'utilisation

Cas D'utilisation : Création d'un compte

Acteur Principale : Freelancer/Client

Objectif : S'inscrire sur la plateforme et présenter ses compétences et expériences

Scénario Principale : Le freelance accède à la page d'inscription de WorkWave. Il remplit un formulaire en fournissant ses informations personnelles et professionnelles. Il crée

un profil attractif en mettant en avant ses compétences et expériences. Il valide et soumet son inscription.

Scenario Alternatif : L'utilisateur a déjà un compte sur la plateforme.

Pré-Condition : -

Post-Condition : Le compte du freelance est créé et activé. Le profil du freelance est accessible aux clients.

Cas D'utilisation : Modifier le Compte

Acteur Principale : Freelancer/Client

Objectif : Mettre à jour son profil et ses informations personnelles.

Scenario Principale : Le freelance se connecte à son compte WorkWave. Il sélectionne l'option "Modifier le compte" dans son profil. Il modifie ses informations personnelles, ses compétences et ses expériences. Il enregistre les modifications.

Scenario Alternatif : Le freelance n'est pas connecté à son compte.

Pré-Condition : L'utilisateur possède un compte WorkWave actif.

Post-Condition : Le profil du freelance est mis à jour avec les nouvelles informations.

Cas D'utilisation : Supprimer le Compte

Acteur Principale : Freelancer/Client

Objectif : Désactiver son compte WorkWave.

Scenario Principale : Le freelance se connecte à son compte WorkWave. Il sélectionne L'option "Supprimer le compte" dans son profil. Il confirme la suppression de son Compte.

Scenario Alternatif : Le freelance n'est pas connecté à son compte.

Pré-Condition : L'utilisateur possède un compte WorkWave actif.

Post-Condition : Le compte du freelance est désactivé et ses informations ne sont plus Visibles sur la plateforme.

Cas D'utilisation : Consulter les Services

Acteur Principale : Freelancer/Client

Objectif : Explorer les services disponibles et identifier les opportunités.

Scenario Principale : Le Clients/Freelancer peut consulter les détails de chaque service et identifier les projets correspondants à ses compétences.

Scenario Alternatif : -

Pré-Condition : -

Post-Condition : -

Cas D'utilisation : Postuler à un Besoin

Acteur Principale : Freelancer/Client

Objectif : Répondre à un projet publié par un client ou publier un besoin.

Scenario Principale : l'utilisateur se connecte à son compte WorkWave. Il sélectionne un Besoin qui correspond à ses demandes. Il consulte les détails du projet et les exigences du client.

Scenario Alternatif : -

Pré-Condition : L'utilisateur possède un compte WorkWave actif.

Post-Condition : La candidature du freelance est envoyée au client. Le client peut consulter la candidature et évaluer le profil du freelance.

Cas D'utilisation : Consulter les Besoins des Clients

Acteur Principale : Freelancer

Objectif : Explorer les demandes publiés par les clients et identifier des opportunités.

Scenario Principale : Le Freelancer se connecte à son compte WorkWave. Il sélectionne l'option "Besoins des clients" dans le menu principal. La liste des projets

disponibles s'affiche, avec des filtres par Compétences, budget et localisation. Le freelance peut consulter les détails de chaque projet et Identifier des opportunités.

Scenario Alternatif : -

Pré-Condition : L'utilisateur possède un compte WorkWave actif.

Post-Condition : Le freelance a accès à la liste des demandes publiés par les clients. Le freelance peut Identifier des opportunités de missions en fonction de ses compétences et ses préférences.

Cas D'utilisation : Acheter un service

Acteur Principale : Freelancer/client

Objectif : Se mettre d'accord avec le Freelancer afin de valider l'accord.

Scenario Principale : Le client demande le service. Le Freelancer réalise sa demande et La termine. Le client doit payer le Freelancer.

Scenario Alternatif : -

Pré-Condition : L'utilisateur possède un compte WorkWave actif.

Post-Condition : Transaction se fait avec succès.

Cas D'utilisation : Gérer les services et les contrats

Acteur Principale : WorkWave Manager

Objectif : Supprimer/Modifier les services et garder des « Logs »

Scenario Principale : Un Freelancer publie une fausse annonce ou dans une fausse catégorie. Le Manager peut donc là supprimer/modifier

Scenario Alternatif : Un utilisateur a demandé de revoir les « Logs » d'une transaction déjà passer

Pré-Condition : -

Post-Condition : -

Cas D'utilisation : Gérer les comptes

Acteur Principale : WorkWave Manager

Objectif : Supprimer/Modifier les comptes

Scenario Principale : Un Freelancer ouvre un faux compte ou commit un acte interdit. Le Manager peut donc supprimer/modifier son compte

Scenario Alternatif : Le Manager peut expulser un utilisateur temporairement.

Pré-Condition : -

Post-Condition : -

2.2 Diagramme de Séquence

Un diagramme de séquence est un élément du langage UML (Unified Modeling Language) qui illustre la séquence chronologique des messages échangés entre les différents objets lors d'une interaction. Il se compose d'un ensemble d'objets représentés par des lignes de vie, ainsi que des messages échangés entre ces objets pendant leur interaction.

2.2.1 Postuler pour un projet (Freelancer) :

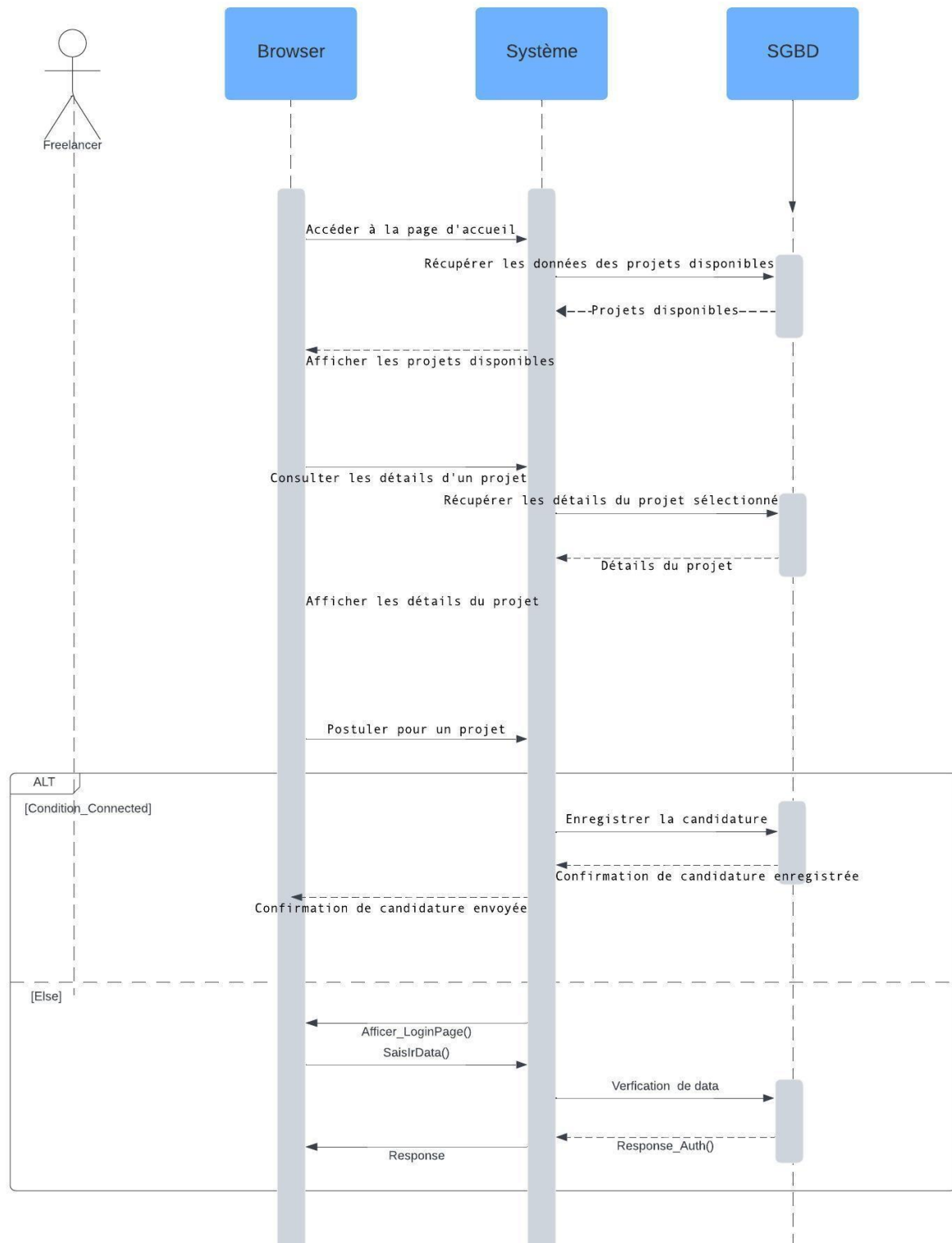


Figure 2: Diagramme de Séquence : Postuler pour un projet

Explication :

- Le freelance peut accéder à la page d'accueil et consulter les opportunités disponibles et leurs détails
- Le freelance veut donc postuler pour un projet, connexion au compte requise
- Le freelance saisit ses identifiants et les envoie au système.
- Le système vérifie les identifiants et envoie une réponse au freelance.
- Si les identifiants sont corrects :
 - Le système affiche le menu principal.
 - Le freelance sélectionne un besoin qui correspond à ses compétences.
 - Le système affiche les détails du projet et les exigences du client.
 - Le freelance rédige une candidature en mettant en avant ses qualifications et son expérience.
- Le freelance soumet sa candidature au client en envoyant le message Soumettre la candidature.
- Si les identifiants sont incorrects :
 - Le système affiche un message d'erreur au freelance.
 - Le freelance peut recommencer la connexion en envoyant un nouveau message Se connecter au compte.

2.2.2 Créer un compte (Freelancer/Client)

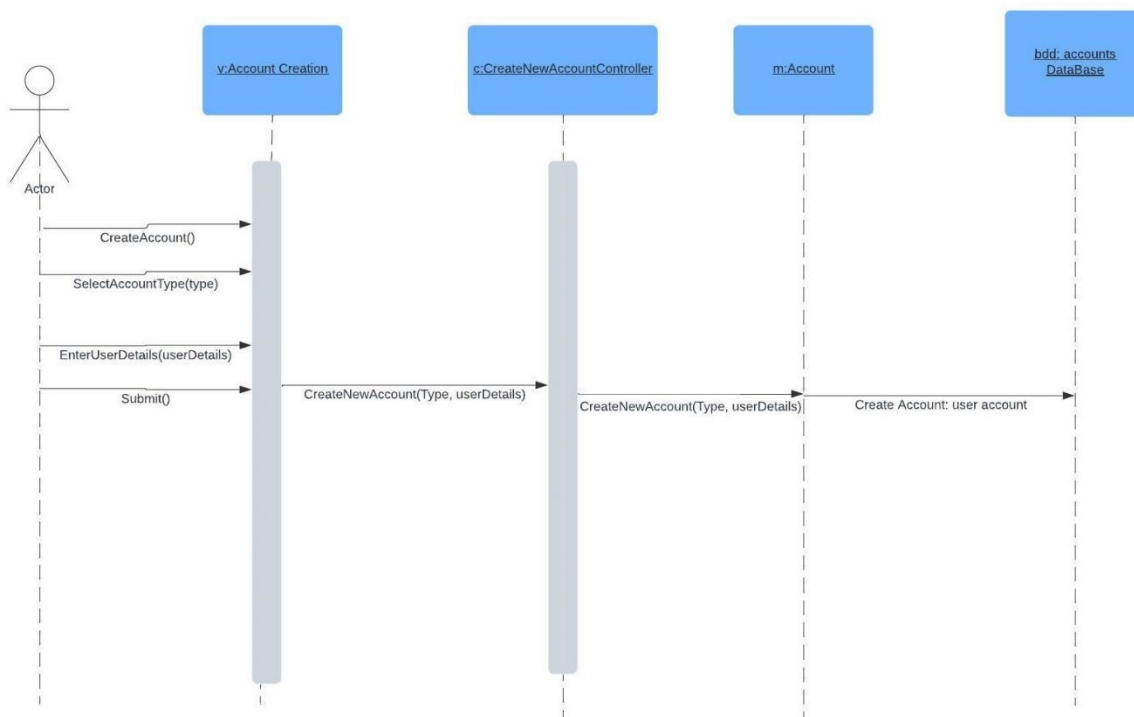


Figure 3: Diagramme de Séquence : Créer un compte

Explication :

- CreateAccount() : Cela signifie le début du processus où un utilisateur initie la création d'un compte.
- Sélectionner le type de compte : L'utilisateur a la possibilité de choisir le type de compte qu'il souhaite créer. Soit Freelancer, ou Client.
- Saisir les informations de l'utilisateur : L'utilisateur saisit ses informations personnelles nécessaires à la création du compte. Cela inclue des informations telles que le nom, l'adresse e-mail et le mot de passe.
- Soumettre : L'utilisateur soumet les informations qu'il a saisies à l'étape précédente.
- CréerNouveauCompte(Type, DétailsUtilisateur) : Cette étape représente une fonction du système déclenchée lors de la soumission par l'utilisateur. Elle prend deux paramètres :

Type : Cela correspond au type de compte sélectionné à l'étape 2

DétailsUtilisateur : Cela fait référence aux informations utilisateur saisies à l'étape 3.

- Création de compte : Ce point terminal signifie une création de compte réussie. L'utilisateur reçoit une confirmation et sera dirigé vers la page d'accueil d'un compte nouvellement créé.

2.2.3 Acheter un service (Client) :

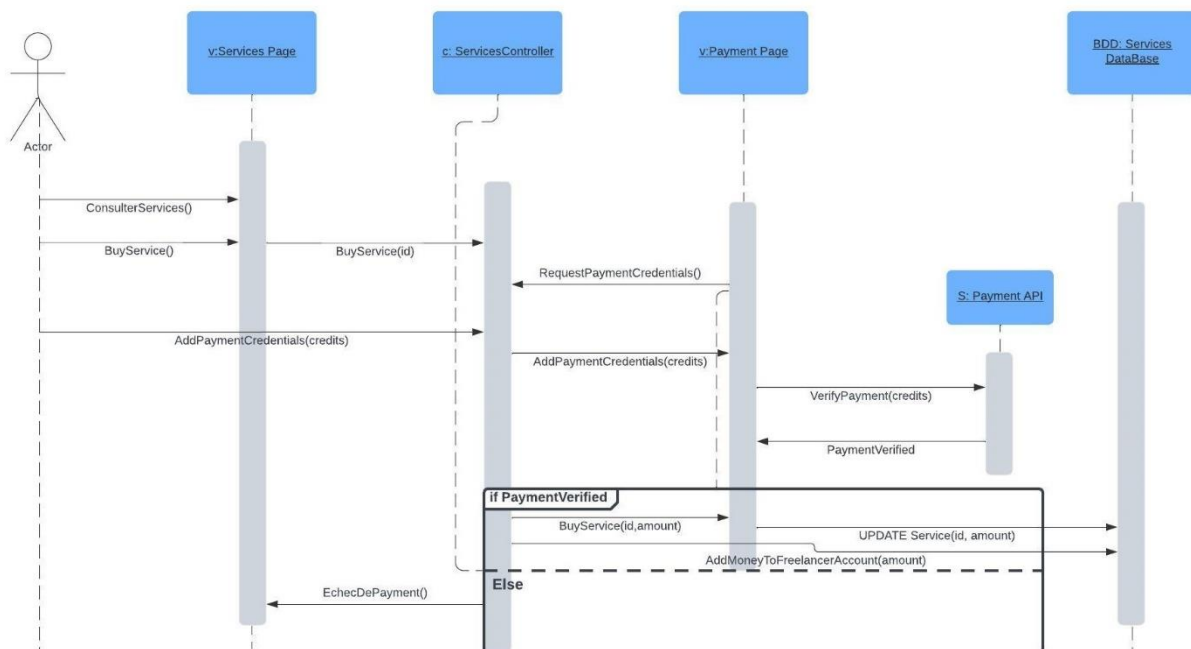


Figure 4: Diagramme de Séquence : Acheter un service

Explication :

- Le client sélectionne le service qu'il souhaite payer et clique sur le bouton "Payer".
- Le système de paiement affiche la page de paiement avec le montant à payer et les options de paiement disponibles.
- Le client sélectionne un mode de paiement et saisit ses informations de paiement.
- Le système de paiement envoie une requête d'autorisation au fournisseur de service.
- Le fournisseur de service vérifie la disponibilité du service et envoie une réponse au système de paiement.

- Si le service est disponible:
 - Le système de paiement envoie une requête de capture au fournisseur de service.
 - Le fournisseur de service capture le paiement et envoie une confirmation au système de paiement.
 - Le système de paiement affiche un message de confirmation au client et lui fournit un reçu.
- Si le service n'est pas disponible:
 - Le système de paiement affiche un message d'erreur au client.

2.2.4 Gestion des Contrat (Admin)

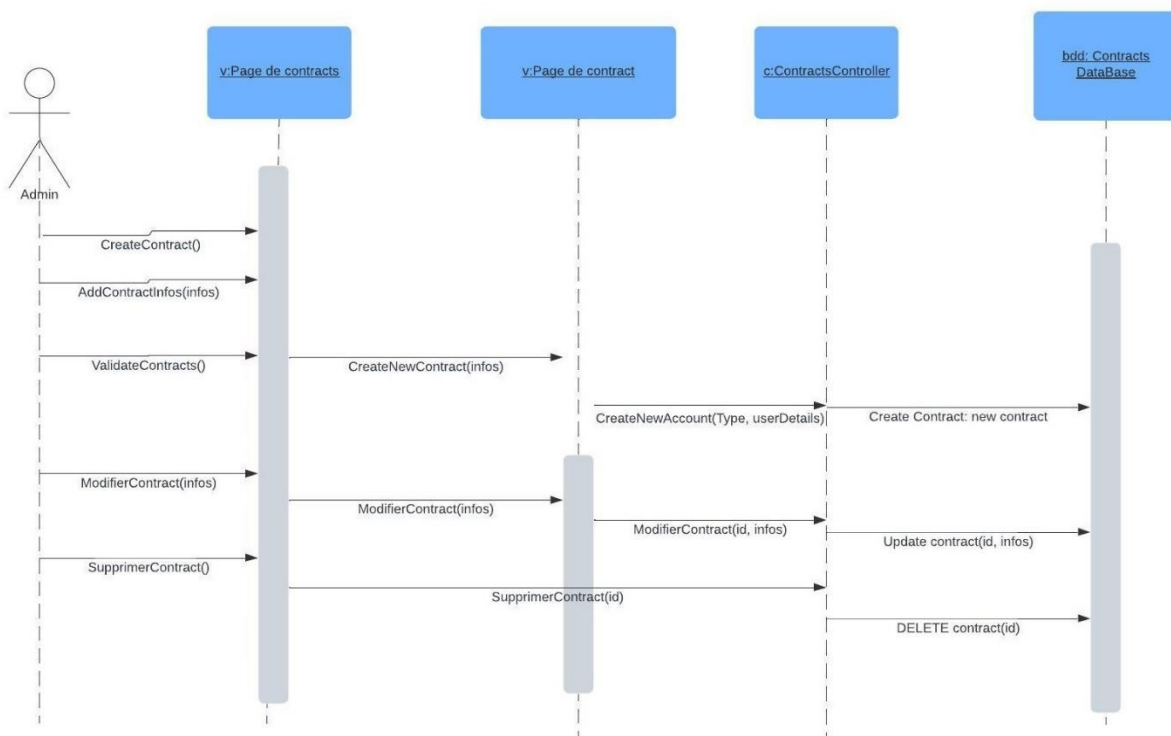


Figure 5: Diagramme de Séquence : Gestion des contrats

Explication :

- L'administrateur sélectionne un contrat dans la liste des contrats disponibles.
- Le système de gestion des contrats affiche les détails du contrat sélectionné.

- L'administrateur peut effectuer différentes actions sur le contrat :

- Valider la transaction :

- L'administrateur clique sur le bouton "Valider la transaction".

- Le système de gestion des contrats appelle la fonction CreateNewContract(Infos) pour créer le contrat.

- La fonction Create Contract : new Contract récupère les informations du contrat dans la base de données et les renvoie à l'administrateur.

- Le contrat est peut-être envoyé aux destinataires.

- Modifier le contrat :

- L'administrateur clique sur le bouton "Modifier le contrat".

- Le système de gestion des contrats appelle la fonction ModifierContrat(Infos).

- L'administrateur effectue les modifications nécessaires.

- Les modifications sont enregistrées et le contrat est mis à jour.

- Supprimer le contrat :

- L'administrateur clique sur le bouton "Supprimer le contrat".

- Un message de confirmation s'affiche à l'administrateur pour valider la suppression.

- L'administrateur confirme la suppression.

- Le système de gestion des contrats appelle la fonction DeleteContract(id) pour supprimer le contrat de la base de données.

- La fonction DeleteContract supprime le contrat de la base de données.

2.3 Diagramme de Classes

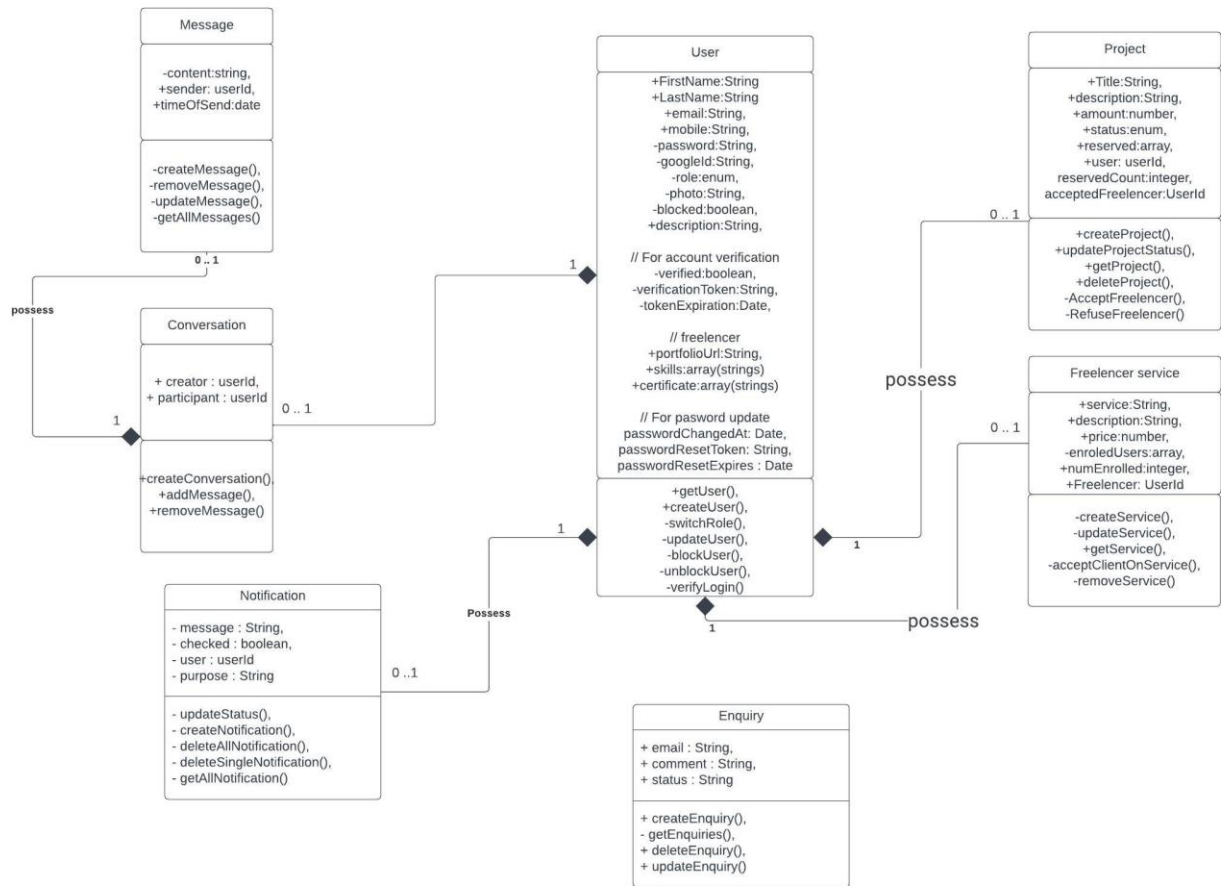


Figure 6: Diagramme de classes

Explication:

Le diagramme de classes fourni illustre les classes principales et leurs relations dans une application de gestion des utilisateurs et des projets. Il décrit les entités du système et leurs interactions.

Classes principales :

Utilisateur : Représente un utilisateur du système. Les attributs incluent le nom d'utilisateur, le mot de passe, l'adresse e-mail, le numéro de téléphone, le rôle (par exemple, administrateur, client, freelanceur) et l'état (par exemple, actif, bloqué). Les méthodes incluent la création d'un utilisateur, la mise à jour des informations de l'utilisateur, la suppression d'un utilisateur et la vérification de la connexion.

Projet : Représente un projet. Les attributs incluent le titre, la description, le montant, le statut (par exemple, en attente, en cours, terminé), la liste des utilisateurs réservés et l'utilisateur responsable. Les méthodes incluent la création d'un projet, la mise à jour des informations du projet, la récupération des informations du projet, la suppression d'un projet et l'acceptation d'un freelanceur sur le projet.

Message : Représente un message dans une conversation. Les attributs incluent le contenu, l'expéditeur, le destinataire, l'heure d'envoi et le statut (par exemple, lu, non lu). Les méthodes incluent la création d'un message, la suppression d'un message, la mise à jour d'un message et la récupération de tous les messages.

Conversation : Représente une conversation entre deux utilisateurs ou plus. Les attributs incluent le créateur, les participants et la liste des messages. Les méthodes incluent la création d'une conversation, l'ajout d'un message, la suppression d'un message et la récupération des informations de la conversation.

Notification : Représente une notification pour un utilisateur. Les attributs incluent le message, l'utilisateur cible, le statut (par exemple, lu, non lu) et l'objectif (par exemple, nouveau message, projet mis à jour). Les méthodes incluent la création d'une notification, la mise à jour du statut d'une notification, la suppression d'une notification et la récupération de toutes les notifications pour un utilisateur.

Enquête : Représente une demande de renseignements d'un client potentiel. Les attributs incluent le commentaire, l'adresse e-mail, le statut (par exemple, nouveau, en attente, traité) et l'utilisateur responsable. Les méthodes incluent la création d'une demande de renseignements, la récupération des demandes de renseignements, la suppression d'une demande de renseignements et la mise à jour des informations de la demande de renseignements.

Service : Représente un service proposé par un freelanceur. Les attributs incluent le nom du service, la description, le prix, la liste des utilisateurs inscrits et le freelanceur qui propose le service. Les méthodes incluent la création d'un service, la mise à jour des informations du service, la récupération des informations du service, la suppression d'un service et l'acceptation d'un client sur le service.

Relations :

- Un utilisateur peut créer et gérer plusieurs projets.
- Un projet peut avoir plusieurs participants (freelancers) et peut générer plusieurs conversations.

- Un freelancer peut créer et gérer plusieurs services.
- Un service peut avoir plusieurs participants (utilisateurs) et peut générer plusieurs conversations.
- Une conversation peut contenir plusieurs messages entre deux participants (utilisateurs).
- Un message appartient à une seule conversation et est envoyé par un seul utilisateur.
- Une notification peut être associée à un projet ou non et est envoyée à un seul utilisateur.

2.4 Conclusion

En résumé, ce chapitre consacré aux spécifications des besoins pour la création de WorkWave a souligné l'importance de comprendre à la fois les exigences du client et les besoins des utilisateurs. Nous avons examiné la problématique à l'origine de la création de cette plateforme, suivi d'une analyse approfondie des besoins et d'une étude critique des sites web similaires existants. Enfin, nous avons présenté le client et ses objectifs pour ce projet. Cette étape cruciale nous permettra de planifier de manière plus efficace et de concevoir un site web qui répondra pleinement aux besoins des utilisateurs et du client.

Chapitre 3 : Realisation

Introduction

La conception d'un site web pour une plateforme de freelances représente à la fois un défi passionnant et complexe. Un site bien conçu a le pouvoir de simplifier la gestion des inscriptions, d'améliorer la communication entre les clients et les freelances, tout en offrant une expérience utilisateur intuitive et agréable. Dans ce chapitre, plongeons dans le monde captivant du projet "WorkWave", une plateforme de freelances en ligne. Nous dévoilerons les outils de développement que nous avons judicieusement sélectionnés pour donner vie à cette application et présenterons l'interface de l'application à travers des captures d'écran détaillées. Nous expliquerons soigneusement les fonctionnalités clés de chaque élément. Enfin, nous mettrons en lumière les nombreux avantages de l'utilisation de la stack MERN (MongoDB, Express.js, React, Node.js) dans le développement de ce projet, tout en soulignant l'efficacité de l'utilisation d'un seul langage de programmation, JavaScript..

3 Outils de développement :

En combinant ces outils, nous avons pu créer une expérience visuelle cohérente et attrayante pour notre site web de freelance, tout en optimisant l'efficacité de notre processus de conception et de développement.

3.1 Figma :

Figma est une application de conception d'interface utilisateur (UI) et de prototypage collaboratif basée sur le cloud. Nous avons utilisé Figma pour créer et collaborer sur les maquettes de notre site web de freelance. Grâce à ses fonctionnalités de collaboration en temps réel, nous avons pu travailler efficacement en équipe, itérer rapidement sur nos designs et obtenir des retours instantanés. Figma nous a également permis de créer des prototypes interactifs pour visualiser et tester le flux utilisateur avant même de commencer le développement.

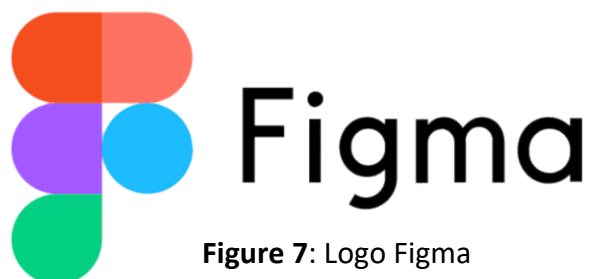


Figure 7: Logo Figma

3.2 Adobe illustrator :

Adobe Illustrator est un logiciel de création graphique vectorielle. Nous avons utilisé Illustrator pour créer des illustrations, des icônes et des éléments graphiques sur mesure pour notre site web. Sa capacité à travailler avec des vecteurs nous a permis de créer des éléments graphiques de haute qualité et de les dimensionner sans perte de qualité, ce qui est crucial pour assurer une apparence nette et professionnelle sur différents appareils et résolutions d'écran.



Figure 8: Logo Adobe Illustrator

3.3 Adobe Photoshop :

Adobe Photoshop est un logiciel de retouche d'images et de conception graphique. Nous avons utilisé Photoshop pour retoucher et optimiser les images que nous avons utilisées sur notre site web. De la retouche de photos de profil à l'optimisation des images pour le web, Photoshop nous a permis de manipuler les images avec précision et de garantir qu'elles correspondent parfaitement à notre esthétique globale.



Figure 9: Logo Adobe Photoshop

3.4 JavaScript :

JavaScript est un langage de programmation de premier ordre utilisé pour développer des applications web interactives et dynamiques. Nous avons utilisé JavaScript à la fois côté client et côté serveur dans notre projet. Côté client, nous l'avons utilisé pour ajouter de l'interactivité à notre interface utilisateur, comme la validation des formulaires et les animations. Côté serveur, nous l'avons utilisé avec Node.js pour développer la logique métier de notre application, manipuler les données et communiquer avec notre base de données.



Figure 10: Logo JavaScript

3.5 Express :

Express est un framework web minimaliste et flexible pour Node.js, conçu pour créer des applications web et des API robustes et performantes. Nous avons utilisé Express comme notre framework côté serveur pour gérer les requêtes HTTP, router les différentes parties de notre application et gérer la logique métier. Grâce à sa simplicité et à sa puissance, Express nous a permis de construire rapidement une API robuste pour notre plateforme de freelance, tout en offrant une grande flexibilité pour personnaliser notre backend selon nos besoins spécifiques.



Figure 11: Logo Express

3.6 MongoDB :

MongoDB est une base de données NoSQL flexible et évolutive, conçue pour stocker des données semi-structurées et non structurées. Nous avons utilisé MongoDB comme notre système de gestion de base de données pour stocker les informations des utilisateurs, les projets, les transactions et autres données pertinentes pour notre plateforme de freelance. Sa structure de données flexible nous a permis de modéliser nos données de manière efficace, en permettant des schémas dynamiques et évolutifs. De plus, son intégration transparente avec Node.js nous a permis d'interagir facilement avec la base de données depuis notre backend Express.



Figure 12: Logo MongoDB

3.7 React :

React est une bibliothèque JavaScript pour la construction d'interfaces utilisateur interactives et réactives. Nous avons utilisé React côté client pour structurer et rendre dynamiquement les composants de notre application web. Grâce à ses fonctionnalités de composants réutilisables et de gestion efficace de l'état, React nous a permis de créer une interface utilisateur intuitive et réactive pour nos utilisateurs. Son approche déclarative et sa capacité à isoler les composants ont grandement simplifié le processus de développement front-end de notre plateforme de freelance.

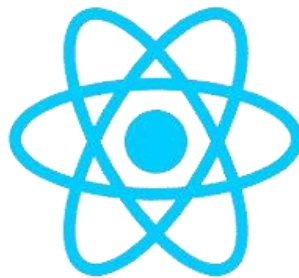


Figure 13: Logo React

3.8 Tailwind CSS:

Tailwind CSS est un framework CSS utilitaire qui permet de créer rapidement des interfaces utilisateur modernes et personnalisées. Nous avons utilisé Tailwind CSS pour styliser notre application web, en profitant de sa large gamme de classes prédéfinies pour simplifier le processus de conception. Grâce à son approche basée sur les classes, Tailwind CSS nous a permis de créer un design cohérent et réactif tout en minimisant le besoin de CSS personnalisé

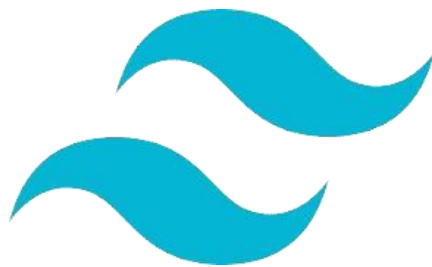


Figure 14: Logo Tailwind CSS

3.9 Postman:

Postman est un outil de développement d'API qui simplifie la création, le partage, les tests et la documentation d'API. Nous avons utilisé Postman pour tester et valider les différentes fonctionnalités de notre API RESTful développée avec Express.js. Grâce à ses fonctionnalités de test automatisé et de validation des réponses API, Postman nous a permis d'identifier et de résoudre rapidement les problèmes potentiels, assurant ainsi la fiabilité et la qualité de notre API. Son interface conviviale et ses fonctionnalités avancées ont facilité la collaboration entre les membres de notre équipe de développement, accélérant ainsi le processus de développement de notre plateforme de freelance.



Figure 15: Logo Postman

3.10 Visual Studio Code (VSCode):

Vscode est un éditeur de code source gratuit et open source, développé par Microsoft. Nous avons utilisé VSCode comme environnement de développement intégré (IDE) pour écrire, tester et déboguer notre application. Grâce à ses fonctionnalités avancées telles que l'autocomplétions, la coloration syntaxique, le débogage intégré et les extensions, VSCode nous a permis de développer efficacement notre plateforme de freelance.



Figure 16: Logo VS Code

3.11 Lucychart:

Lucidchart est une plateforme de création de diagrammes en ligne qui permet de concevoir facilement des organigrammes, des diagrammes de flux, des diagrammes UML et bien plus encore. Nous avons utilisé Lucidchart pour créer et partager des diagrammes d'architecture système, des diagrammes de flux de données et des diagrammes de conception d'interface utilisateur pour notre plateforme de freelance. Grâce à ses fonctionnalités conviviales et à sa facilité d'utilisation, Lucidchart nous a permis de visualiser nos idées, de communiquer efficacement avec les membres de l'équipe et de prendre des décisions éclairées tout au long du processus de développement de notre application.



Figure 17: Logo Lucychart

4 Présentation de l'Application :

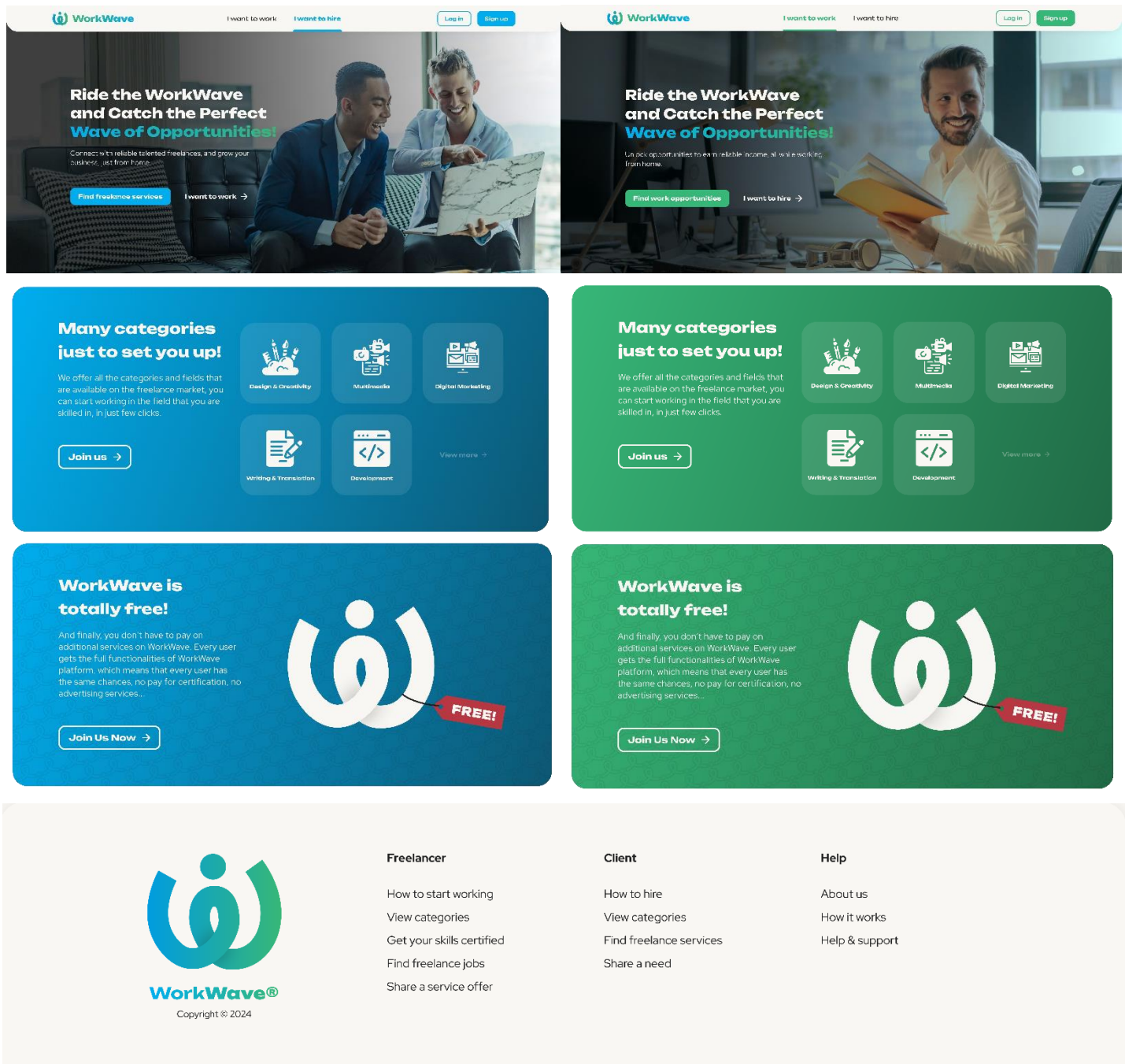


Figure 18: Page d'accueil de l'application

4.1 Page d'accueil

Notre application a été conçue avec un design simple et clair pour faciliter la navigation pour tous les utilisateurs. Des photos accrocheuses, des couleurs attrayantes et des boutons bien visibles améliorent l'expérience utilisateur.

En-tête :

Logo WorkWave

Menu de navigation :

I want to work | I want to hire | Sign up | Log in

Section principale :

Bannière avec image accrocheuse

Titre : "Ride the WorkWave and catch the perfect wave of opportunities"

Sous-titre : "Many categories to set you up" "WorkWave is totally free!"

Bouton : "Join us now !"

Sections supplémentaires : Sommaire.

4.2 Sign Up :

Nous permettons à nos utilisateurs de créer un compte afin qu'ils puissent rejoindre nos services, avec une procédure très simple.

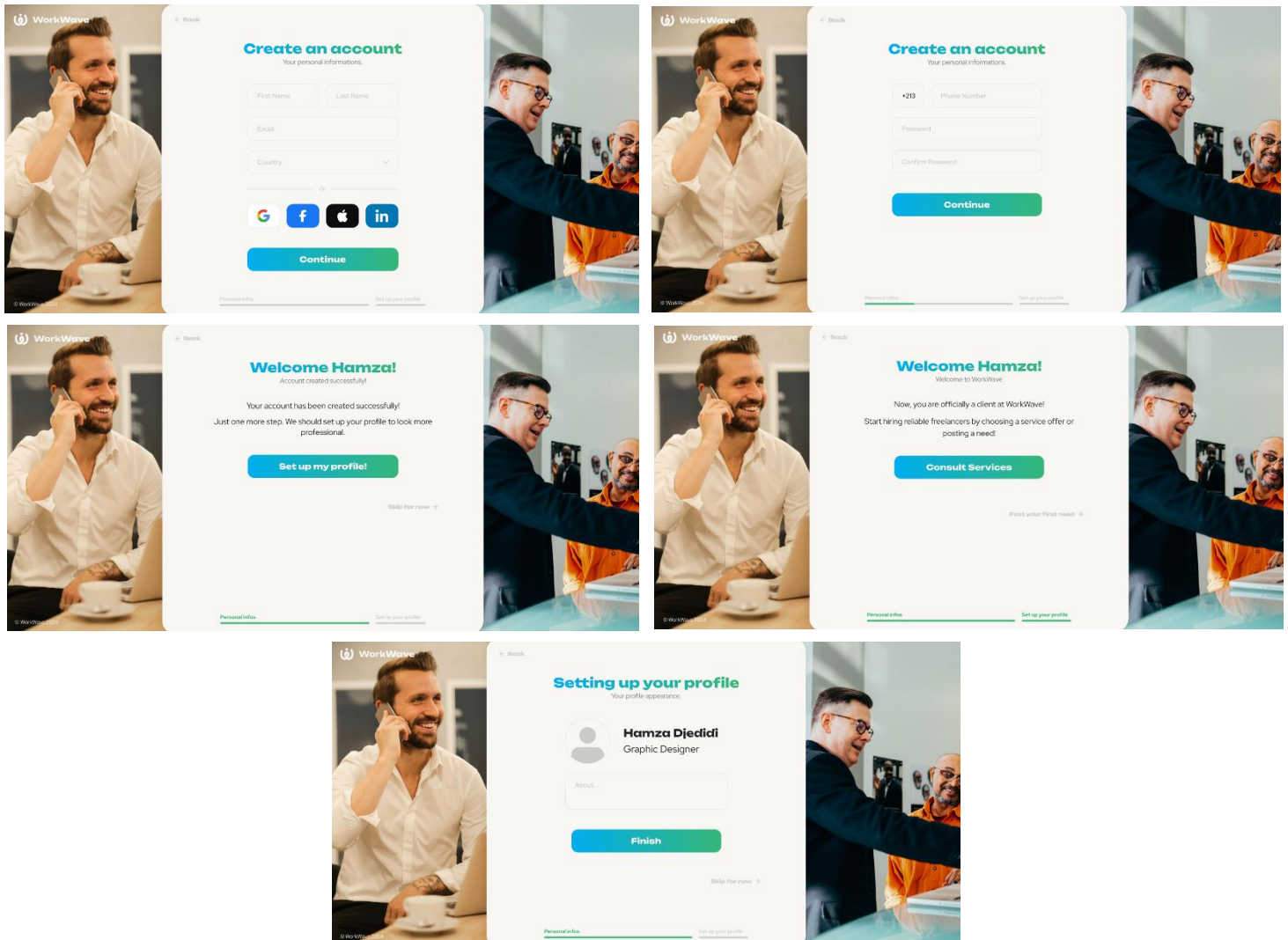


Figure 19: Page Sign Up

Choisir la méthode d'authentification : via Google ...

Informations personnelles : Nom | Prénom | Email | Pays

Contact et Sécurité : Numéro de téléphone | Mot de passe

Configurer mon profil : Déposer CV

Consulter les services : (Skippable)

4.3 Mon Compte :

Après avoir terminé l'inscription, l'utilisateur sera redirigé vers son profil.

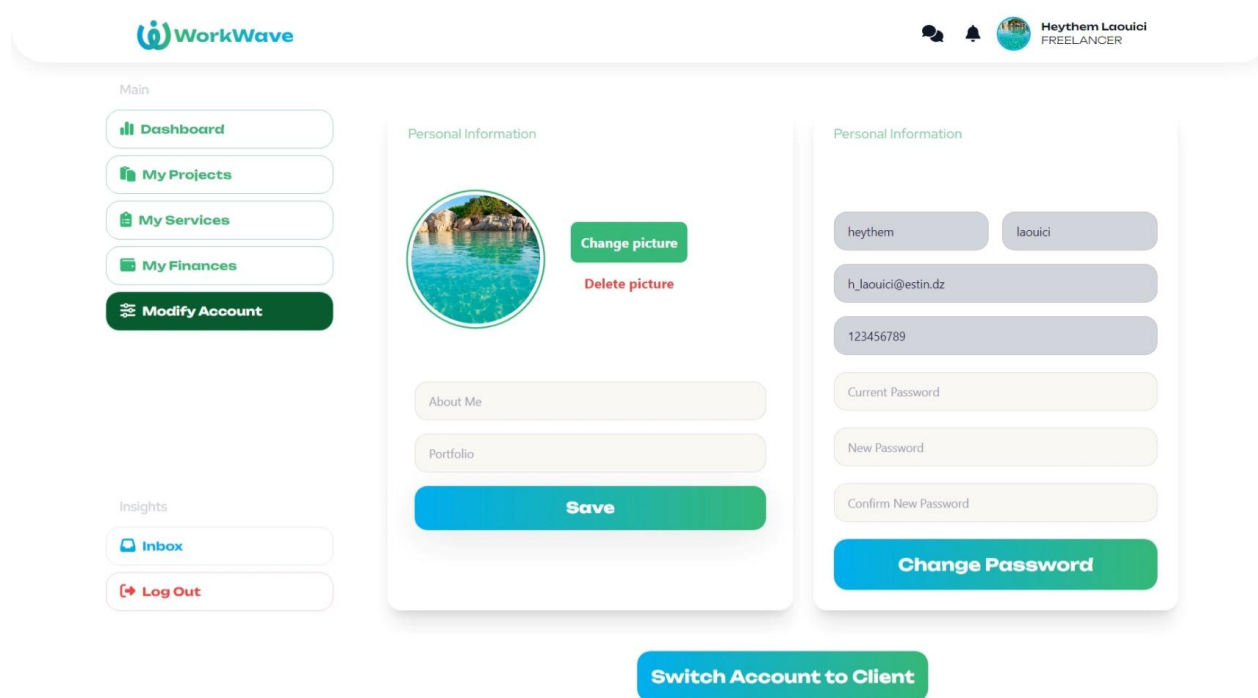


Figure 20: Page Mon Compte

Continuer la configuration : Changer la photo de profil | Ajouter des informations

Mettre à jours les informations personnelles : Changer le mot de passe, Email...

Switch account to client : Continuer d'utiliser notre platform comme un simple client

Inbox : Messagerie

Log out : Se déconnecter

4.4 Mes Projets

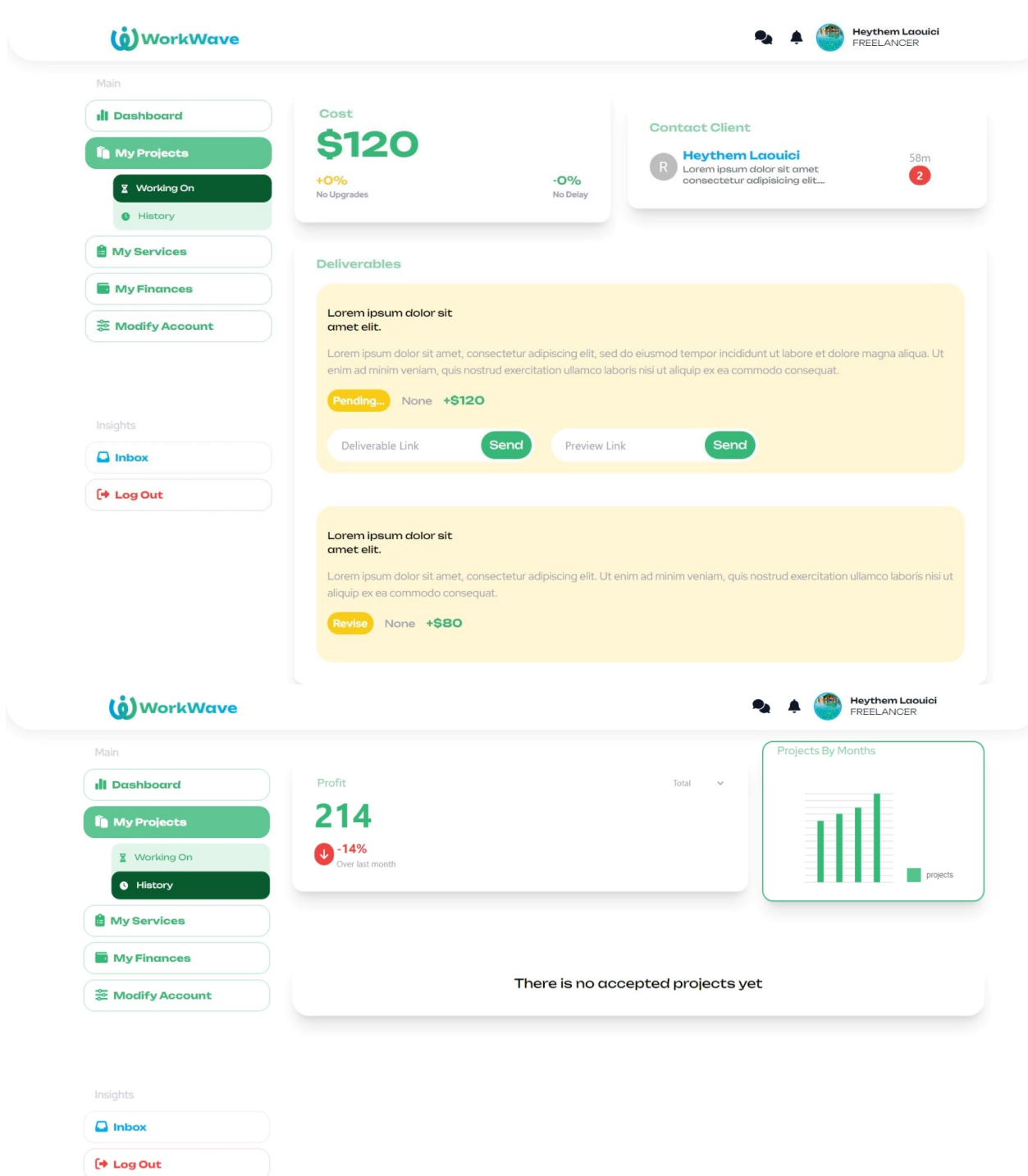


Figure 21: Page Mes Projets

Nous offrons à nos utilisateurs la possibilité de consulter et de revoir leurs projets en cours de manière très efficace et détaillée. Grâce à notre interface intuitive, ils peuvent facilement suivre l'avancement de chaque projet, examiner les détails spécifiques, et apporter les modifications nécessaires. Notre plateforme est conçue pour être à la fois simple et compréhensible, garantissant une gestion de projet fluide et accessible à tous, quelle que soit leur expérience technique.

4.4.1 Working on (projet en cours) :

Contient des informations utiles pour le freelance, telles que :

Les projets en cours.

L'état d'avancement détaillé

Le budget associé

Moyens de communication avec le client. (Messagerie)

Le freelance peut voir une liste complète de ses projets actifs, suivre les étapes accomplies et celles à venir, vérifier le montant total et les paiements effectués, ainsi que discuter directement avec les clients via une messagerie intégrée. Cette centralisation des informations permet une gestion optimisée et une meilleure organisation de leur travail.

4.4.2 History (Historique) :

Nous offrons à nos utilisateurs la possibilité de revisiter leurs projets précédents et de bénéficier d'une vision globale de tous les projets déjà réalisés. Cette fonctionnalité leur permet de récapituler leur parcours professionnel, d'analyser les succès passés et d'identifier les opportunités d'amélioration. Grâce à une interface conviviale, ils peuvent explorer chaque projet en détail, consulter les étapes franchies, les feedbacks des clients, ainsi que les résultats obtenus. Cette rétroaction constructive leur permet d'ajuster leur approche et d'affiner leurs compétences pour des projets futurs. En résumé, cette fonctionnalité offre une réflexion précieuse sur le parcours professionnel du freelance, favorisant ainsi une croissance continue et une évolution vers l'excellence.

4.5 Mes Services

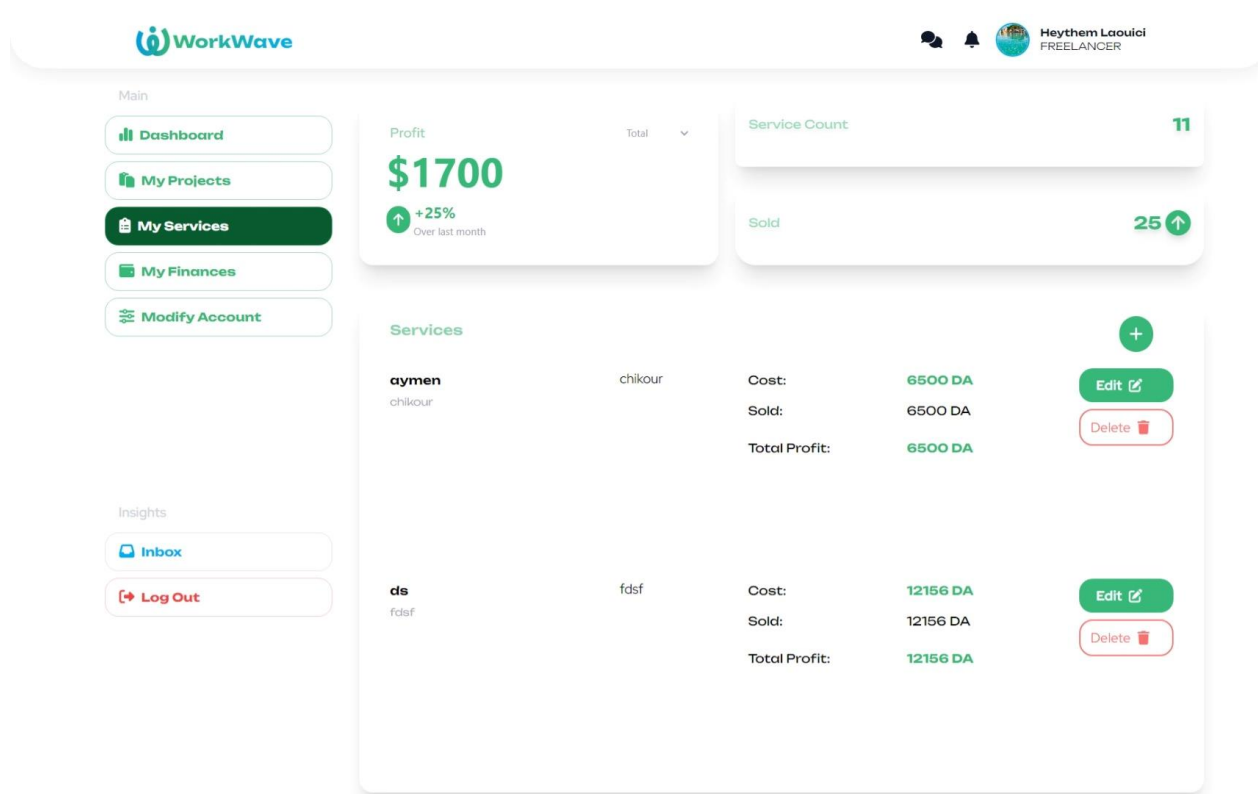


Figure 22: Page Mes Services

Sur cette page dédiée, le freelance bénéficie d'un contrôle total sur la gestion de ses services. Il peut aisément ajouter, supprimer ou modifier ses offres, que ce soit en ajustant les prix, en peaufinant les descriptions, en renommant les services ou en actualisant les dates de disponibilité. Cette flexibilité lui permet d'adapter son catalogue en fonction de l'évolution de ses compétences et des besoins du marché, assurant ainsi une pertinence constante de son profil professionnel.

En plus de la gestion des services, cette page fournit également des informations clés sur les projets réalisés, en cours et à venir. Le freelance peut ainsi avoir un aperçu détaillé de ses ventes passées, de ses services actuellement disponibles et des projets à venir, ce qui lui permet de mieux planifier ses activités et d'anticiper les demandes futures. De plus, il peut consulter son profit total généré grâce à ses services, lui offrant ainsi une visibilité claire sur sa performance financière et une base solide pour prendre des décisions éclairées pour son activité freelance. En somme, cette page constitue un outil essentiel pour la gestion stratégique et le développement proactif de la carrière du freelance.

4.6 Page des services

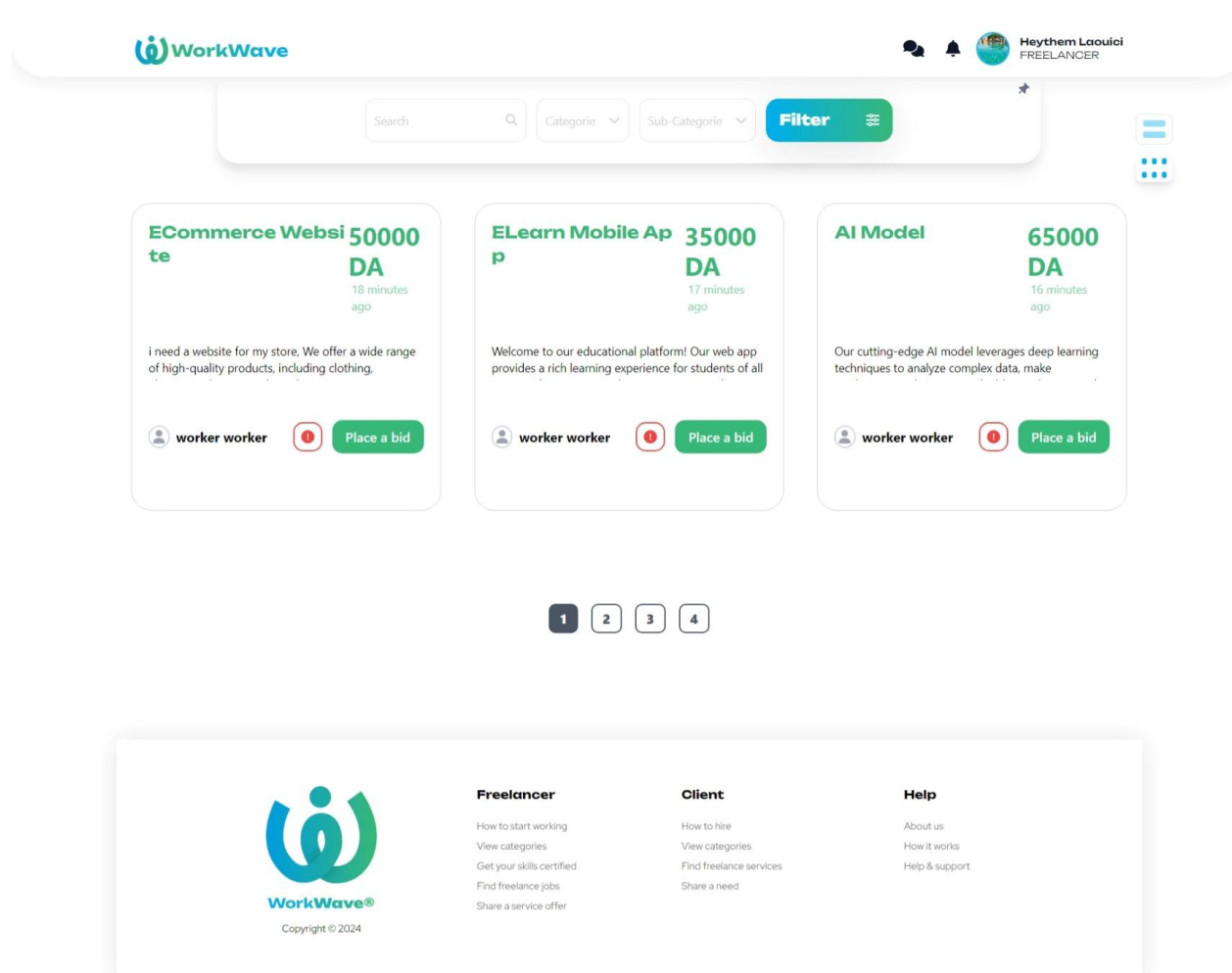


Figure 23: Page Des Services

Sur cette page, les utilisateurs bénéficient d'une expérience interactive et complète, leur offrant la possibilité de découvrir et d'explorer les services disponibles sur notre plateforme. Que ce soit en tant que client ou en tant que freelance chacun peut accéder aux détails des projets et même les acheter directement.

De plus, cette page est également accessible en mode visiteur, permettant à ceux qui ne sont pas encore inscrits de parcourir les offres disponibles. Toutefois, les visiteurs ne peuvent consulter que les détails des services ; ils doivent créer un compte pour effectuer un achat.

Conclusion Générale

En conclusion, le développement de l'application web WorkWave a été un processus bien planifié et exécuté avec succès. Les différentes phases du projet, y compris la spécification des besoins, la conception et la réalisation, ont été abordées avec soin et rigueur pour produire une application web de qualité dédiée à la gestion du e-marketing.

Comme on dit, "Le design n'est pas seulement ce à quoi ça ressemble et ce que ça ressent. Le design, c'est comment ça fonctionne." La conception de l'application a été réalisée avec une approche orientée objet, permettant une meilleure organisation et une maintenance facilitée. De plus, des technologies web modernes telles que HTML, CSS, JavaScript, Express et MongoDB ont été utilisées pour garantir des performances rapides et une stabilité accrue de l'application. La sécurité des données a également été prise en compte grâce à la mise en place de fonctionnalités de sécurité appropriées. Enfin, il convient de souligner que l'application a été testée et validée avec succès sur différentes plateformes et navigateurs. Les résultats des tests ont montré que l'application répondait à toutes les spécifications des besoins et qu'elle était facile à utiliser pour les freelancers et les clients.

En somme, le développement de l'application WorkWave pour la gestion de e-marketing a été une expérience enrichissante et réussie, mettant en évidence l'importance d'une planification et d'une exécution rigoureuses pour produire une application web de qualité.

Bibliographie/webographie.

- stackoverflow.com
- <https://chat.openai.com/>
- Google authentication
- Mongoose ORM (Communication between backend-database)
- Lucychart (building class diagram)
- DBdiagram (Building DB structure)
- Cloudinary (Image and file stocking)
- JWT (Generating secure tokens)
- API Request Limiter
- Socket.io (Real time chat and notification)
- Stripe (Payment)
- Docker (Building container for backend)