



République Tunisienne
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université de Tunis El Manar
Faculté des Sciences de Tunis
Département des Sciences de l'Informatique



RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du
Diplôme National d'Ingénieur en :
Informatique
par
Helmi HAMDI

**Création d'un système complet de gestion de données et
d'interface utilisateur pour les experts travaillant sur la
plateforme E-Tafakna**

Organisme d'accueil :



Président du jury : Mme. Asma AMDOUNI
Rapporteur : Mme. Salwa ZAMMALI
Encadrante Académique : Mme. Hanen BOUHADDA
Encadrants Professionnels : Mr.Fares SGHAIER
Mme.Nourchen MEZNI

Section IF5 - N° ID du Projet : **1005**

Année Universitaire : 2023-2024

J'autorise l'étudiant Helmi HAMDI à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrants professionnels :

M. Fares SGHAIER

Mme. Nourchen MEZNI

Signature



J'autorise l'étudiant Helmi HAMDI à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrante académique :

Mme. Hanen BOUHADDA

Signature



Dédicaces

De tout mon cœur, je souhaite dédier ce travail :

À mes chers parents

Les fondations de mon existence, les lumières qui illuminent mon chemin et m'entourent de douceur et d'amour. Je leur suis infiniment reconnaissante pour leur soutien inébranlable et les sacrifices qu'ils ont consentis tout au long de mon parcours éducatif.

À mes frères

Leurs blagues incessantes et leur humour sans fin apportent une joie inestimable à ma vie, égayant chaque moment partagé. Je prie pour que Dieu veille sur eux constamment, les protégeant et les guidant vers un avenir rempli de succès et de sécurité.

À mes amis les plus proches

Dont l'affection et le soutien indéfectibles ont été des piliers inestimables de ma vie, et avec qui j'ai partagé des moments d'une joie inégalée. Les mots peinent à exprimer toute la profondeur de ma gratitude et l'immensité de l'affection que je ressens pour vous. Votre présence constante a été une source inépuisable de bonheur et de réconfort.

Pour tout cela, je vous en serai éternellement reconnaissante.

À mes professeurs

Qui ont généreusement partagé leur savoir et m'ont soutenu de manière inconditionnelle tout au long de ma quête de réussite académique. Leur dévouement et leur patience ont été des éléments cruciaux de mon parcours. Mon respect pour chacun d'entre vous est profond et inébranlable. Que Dieu vous protège et vous bénisse abondamment pour votre inestimable contribution à mon éducation.

Remerciements

À la fin de cette expérience, je souhaite exprimer ma gratitude en premier lieu à :

Mon Dieu de m'avoir donné le courage, la force et la volonté pourachever ce travail.

*Mon encadrante académique, Mme. **Hanen Bouhadda**, pour son encadrement attentif, sa disponibilité sans faille, ses suggestions constructives qui ont enrichi mon projet, ainsi que ses conseils inestimables qui m'ont été d'une grande aide tout au long de cette expérience.*

*Également, je tiens à exprimer ma gratitude envers mes encadrants du stage, **Fares Sghaier** et **Nourchen Mezni**, pour leur orientation, leur soutien, leur disponibilité, leurs conseils précieux et surtout leur accompagnement attentif tout au long de cette période de stage.*

Au corps professoral et administratif de la FST, je souhaite exprimer mes vifs remerciements pour leur engagement inébranlable et leur soutien précieux tout au long de ma formation.

Finalement, avec un grand honneur, j'adresse mes remerciements à tous les membres de jury, d'avoir accepté l'évaluation de ce modeste travail. Vos vastes connaissances, compétences, gentillesse et qualités humaines imposent le respect.

Table des matières

Introduction Générale	9
I Contexte général	11
I.1 Introduction	12
I.2 Contexte générale du projet	12
I.3 Présentation de L'organisme	12
I.3.1 Organisme d'accueil	12
I.3.2 Domaine d'expertise	13
I.3.3 Les valeurs	13
I.4 Présentation du projet	14
I.4.1 Problématique	14
I.5 Étude et Critique de l'existant	15
I.5.1 Étude de l'existant	15
I.5.2 Critique de l'existant	17
I.6 Solution et Objectif	18
I.7 Méthodologie	18
I.8 Résultats attendus	19
I.9 Conclusion	19
II Analyse et spécification des besoins	20
II.1 Introduction	21
II.2 Démarche à suivre	21
II.2.1 Définition d'un cycle de vie	21
II.2.2 Modèle de cycle de vie en Cascade	21
II.2.3 Lien de cycle choisi avec les diagrammes UML	22
II.3 Identification des acteurs de l'application	22
II.4 Identification des besoins	23
II.4.1 Besoin fonctionnels	23
II.4.1.1 Administrateur	23
II.4.1.2 Client	23
II.4.1.3 Utilisateur	23
II.4.2 Besoin non fonctionnels	24
II.5 Modélisation des Besoins	25
II.5.1 Diagramme du cas d'utilisation global	25

II.6 Diagramme de classe	26
II.7 Conclusion	27
III Conception	28
III.1 Introduction	29
III.2 Diagramme des cas d'utilisation	29
III.2.1 Cas d'utilisation s'authentifier	29
III.2.2 Description textuelle	30
III.2.2.1 S'authentifier	30
III.2.3 Cas d'utilisation gérer client	31
III.2.3.1 Supprimer client	31
III.2.4 Cas d'utilisation gérer calendrier	33
III.2.4.1 Consulter évènement	33
III.2.5 Cas d'utilisation gérer réunion	34
III.2.5.1 Supprimer réunion	35
III.2.6 Cas d'utilisation gérer profil	36
III.2.6.1 Gérer profil	37
III.2.7 Cas d'utilisation gérer fichier	38
III.2.7.1 Gérer fichier	39
III.2.8 Cas d'utilisation gérer note	41
III.2.8.1 Gérer note	41
III.2.9 Cas d'utilisation Engager une conversation de chat	43
III.2.9.1 Conversation du chat	43
III.2.10 Cas d'utilisation Visualiser les graphiques	45
III.2.10.1 Visualisation des graphiques	45
III.3 Diagramme de séquence	47
III.3.0.1 Diagramme de séquence d'authentification	47
III.3.0.2 Diagramme de séquence consulter liste client	48
III.3.0.3 Diagramme de séquence consulter liste de réunion	49
III.3.0.4 Diagramme de séquence supprimer réunion	50
III.3.0.5 Diagramme de séquence modifiée profil	51
III.3.0.6 Diagramme de séquence de chat entre les experts	52
III.4 Conclusion	53
IV Étude de faisabilité et architecture	54
IV.1 Introduction	55
IV.2 Architecture logique	55
IV.2.1 Communication avec HTTP et REST	55

IV.2.2 Architecture de l'application E-Tafakna	56
IV.2.2.1 Architecture 3-tiers	57
IV.2.2.2 Le design pattern MVVM (Modèle-Vue-VueModèle)	57
IV.3 Architecture physique	58
IV.4 Environnement de travail	59
IV.4.1 Environnement matériel	60
IV.4.2 Environnement logiciel	60
IV.4.2.1 Outil Visual Studio	61
IV.4.2.2 Outil Visual Paradigme	61
IV.4.2.3 Thunder Client	62
IV.5 Choix technologiques	62
IV.5.1 HTML	62
IV.5.2 CSS	63
IV.5.3 JavaScript	63
IV.5.4 Bootstrap	63
IV.5.5 ReactJS	64
IV.5.6 Vitest	64
IV.5.7 NodeJS	65
IV.5.8 OpenAI	65
IV.5.9 Express.js	66
IV.5.10 MySQL Workbench	66
IV.6 Conclusion	67
V Interface et Architecture	68
V.1 Introduction	69
V.2 Interface d'authentification	69
V.3 Interface d'accueil	70
V.4 Interface de la carte géographique	73
V.5 Interface gérer clients	73
V.6 Interface de réunion	74
V.7 Interface de calendrier	76
V.8 Interface de profile	77
V.9 Interface de fichier	78
V.10 Interface de note	82
V.11 Interface de sélection le type de chat	83
V.12 Interface de chatbot	84
V.13 Interface de chatroom	84
V.14 Conclusion	86

VI Tests et Résultats	87
VI.1 Introduction	88
VI.2 Les types de tests	88
VI.3 Test utilisé	88
VI.4 Configuration et Structure des Répertoires	89
VI.5 Cas de test	89
VI.5.1 Détails sur les cas de test	90
VI.6 Résultats	91
VI.7 conclusion	92
Conclusion Générale	93

Table des figures

I.1	Logo E-Tafakna[1]	13
I.2	Capture d'écran de l'application Clio[2]	16
I.3	Capture d'écran de l'application Lexicata[3]	16
I.4	Capture d'écran de l'application Jarvis Legal[4]	17
II.1	Représentation du modèle en cascade[5]	22
II.2	Diagramme du cas d'utilisation général	26
II.3	Diagramme de classe de l'application web	27
III.1	Cas d'utilisation S'authentifier	29
III.2	Cas d'utilisation Gérer client	31
III.3	Cas d'utilisation Gérer calendrier	33
III.4	Cas d'utilisation Gérer réunion	35
III.5	Cas d'utilisation Gérer profil	37
III.6	Cas d'utilisation Gérer fichier	38
III.7	Cas d'utilisation Gérer note	41
III.8	Cas d'utilisation Engager une conversation de chat	43
III.9	Cas d'utilisation Visualiser les graphiques	45
III.10	Diagramme de séquence du scénario « S'authentifier»	48
III.11	Diagramme de séquence du scénario « Consulter liste client»	49
III.12	Diagramme de séquence du scénario « Consulter liste de réunion»	50
III.13	Diagramme de séquence du scénario « Supprimer réunion»	51
III.14	Diagramme de séquence du scénario « Modifier le profile»	52
III.15	Diagramme de séquence du scénario «Envoyer message»	53
IV.1	REST API[6]	56
IV.2	Architecture 3-tiers[7]	56
IV.3	Modèle MVVM[8]	58
IV.4	Architecture 3-tiers	59
IV.5	Environnement de travail	60
V.1	Capture d'écran de l'interface login	69
V.2	Capture d'écran de l'interface d'accueil	70
V.3	Capture d'écran de l'interface graphique en secteurs	71
V.4	Capture d'écran de l'interface graphique 0	71

TABLE DES FIGURES

V.5	Capture d'écran de l'interface graphique 1	72
V.6	Capture d'écran de l'interface graphique 2	72
V.7	Capture d'écran de l'interface de la carte géographique	73
V.8	Capture d'écran de l'interface de gestion des clients	74
V.9	Capture d'écran de l'interface des réunions	74
V.10	Capture d'écran de l'interface des réunions	75
V.11	Capture d'écran de l'interface des réunions	75
V.12	Capture d'écran de l'interface calendrier par mois	76
V.13	Capture d'écran de l'interface calendrier par semaine	76
V.14	Capture d'écran de l'interface calendrier par mois	77
V.15	Capture d'écran de l'interface profile	78
V.16	Capture d'écran de l'interface gestion de fichiers	78
V.17	Capture d'écran de l'interface ajouter dossier	79
V.18	Capture d'écran de l'interface créer fichier	80
V.19	Capture d'écran de l'interface de modification	80
V.20	Capture d'écran de l'interface de suppression	81
V.21	Capture d'écran de l'interface d'ajout d'une fichier	81
V.22	Capture d'écran de l'interface d'ajout des fichiers	82
V.23	Capture d'écran de l'interface note	82
V.24	Capture d'écran de sélection	83
V.25	Capture d'écran de l'interface chatbot	84
V.26	Capture d'écran de l'interface d'ivitation 1	85
V.27	Capture d'écran de l'interface d'ivitation 2	85
V.28	Capture d'écran de l'interface de chat 1	86
V.29	Capture d'écran de l'interface de chat 2	86
VI.1	Liste de dépendances des outils pour les tests	89
VI.2	Structure du dossier de test	89
VI.3	Exemple du cas de test	90
VI.4	Résultat de test	91

Liste des tableaux

III.1 Description du cas d'utilisation "S'authentifier"	30
III.2 Description du cas d'utilisation "Supprimer client"	32
III.3 Description du cas d'utilisation "Consulter évènement"	34
III.4 Description du cas d'utilisation "Supprimer réunion"	36
III.5 Description du cas d'utilisation "Gérer profil"	37
III.6 Description du cas d'utilisation "Gérer fichier"	39
III.7 Description du cas d'utilisation "Gérer note"	42
III.8 Description du cas d'utilisation "Gérer une conversation de chat"	44
III.9 Description du cas d'utilisation "Visualiser les graphiques"	46

Liste des abréviations

ISIAD Ingénierie des Systèmes d’Information et d’Aide à la Décision

UML Unified Modeling Language - Langage de modélisation unifié.

HTTP Hypertext Transfer Protocol - Protocole de transfert hypertexte.

REST Representational State Transfer - Style d’architecture pour les systèmes web.

JSON JavaScript Object Notation.

XML Extensible Markup Language - Langage de balisage extensible

API Application Programming Interface - Interface de programmation d’application.

MVVM Model View ViewModel - Modèle-Vue-Vue Modèle .

MySQL My Structured Query Language.

SQL Structured Query Language.

VS Code Visual Studio Code.

HTML HyperText Markup Language - langage de balises pour l’hypertexte

CSS Cascading Style Sheets - Feuilles de style en cascade.

JS JavaScript

DOM Document Object Model - Modèle d’objet de document.

IA Artificial Intelligence - Intelligence Artificielle.

Introduction Générale

Avec l'essor de la technologie de l'information, la nécessité d'optimiser la gestion des contrats est devenue primordiale pour les clients, qui cherchent à économiser du temps et à résoudre les problèmes liés à cette tâche. Pour répondre à ce besoin croissant, notre société a développé l'application web et mobile "E-Tafakna". Cette plateforme révolutionnaire permet aux utilisateurs de créer, personnaliser, gérer, stocker et envoyer des contrats en ligne, tout en offrant la possibilité de planifier des réunions virtuelles avec des experts juridiques et comptables pour des conseils sur mesure.

Dans cette optique, nous avons mis l'accent sur le développement de l'application web, notamment sur un tableau de bord dédié aux avocats et comptables. Ce tableau de bord offre une gamme complète de fonctionnalités, permettant à l'utilisateur, qu'il soit avocat ou comptable, de s'authentifier, de gérer ses clients, ses réunions en ligne, ses fichiers, ses notes, son profil, son calendrier et visualiser ses graphiques. De plus, il offre la possibilité d'engager des conversations en direct avec d'autres experts du domaine, favorisant ainsi la collaboration et l'échange d'expertise ou avec un assistant IA. En somme, "E-Tafakna" représente une solution intégrale et innovante pour simplifier la gestion contractuelle tout en encourageant la collaboration professionnelle.

Ce rapport sera divisé en six chapitres distincts, chacun apportant une contribution spécifique à la compréhension et à la réalisation de notre projet. Le premier chapitre, intitulé « Contexte général », se concentrera sur une présentation approfondie de l'entreprise dans laquelle notre projet prend place. Il exposera également le cadre général de notre initiative, en mettant en lumière la problématique à laquelle nous sommes confrontés, ainsi qu'une analyse approfondie de l'existant. De plus, ce chapitre détaillera la solution que nous proposons et la méthodologie de travail que nous envisageons d'adopter pour sa mise en œuvre.

Le deuxième chapitre, intitulé « Analyse et spécification des besoins », se penchera sur les étapes cruciales de la démarche à suivre pour comprendre les exigences fonctionnelles et non fonctionnelles de notre projet. Il identifiera également les acteurs clés impliqués et présentera de manière claire et concise le diagramme du cas d'utilisation général avec le diagramme de classe, permettant ainsi une vision d'ensemble du système.

Le troisième chapitre, « Conception », approfondira davantage la planification en dé-

taillant les diagrammes du cas d'utilisation pour chaque fonctionnalité spécifique, ainsi que les diagrammes de séquence. Cette section inclura également une analyse approfondie de la conception et de l'implémentation du projet, offrant ainsi un aperçu complet du processus de développement.

Le quatrième chapitre, « Étude de faisabilité et architecture », se concentrera sur l'évaluation de la viabilité de notre projet, en examinant de près l'architecture logique et physique de l'application. De plus, il abordera l'environnement de travail dans lequel le projet sera réalisé, ainsi que les choix technologiques effectués pour garantir son succès.

Le cinquième chapitre , « Interface et architecture », fournira des éléments visuels clés en présentant des captures d'écran représentatives des interfaces de l'application E-Tafakna développées tout au long de la période de stage.

Enfin, dans le dernier chapitre, « Tests et Résultats », nous aborderons différents types de tests et mettrons l'accent sur les tests de performance. Nous détaillerons également un exemple spécifique de test de performance et en présenterons les résultats.

En conclusion, une synthèse générale récapitulera l'ensemble du travail réalisé et soulignera les efforts déployés pour atteindre les objectifs fixés.

Chapitre I

Contexte général

Sommaire

I.1	Introduction	12
I.2	Contexte générale du projet	12
I.3	Présentation de L'organisme	12
I.3.1	Organisme d'accueil	12
I.3.2	Domaine d'expertise	13
I.3.3	Les valeurs	13
I.4	Présentation du projet	14
I.4.1	Problématique	14
I.5	Étude et Critique de l'existant	15
I.5.1	Étude de l'existant	15
I.5.2	Critique de l'existant	17
I.6	Solution et Objectif	18
I.7	Méthodologie	18
I.8	Résultats attendus	19
I.9	Conclusion	19

I.1 Introduction

Dans ce chapitre, nous présenterons d'abord l'entité d'hébergement « E-Tafakna » et le contexte du projet. Ensuite, nous passons brièvement en revue les défis rencontrés, en nous concentrant sur les enjeux du projet. Nous analysons et évaluons ensuite la situation actuelle, proposons des solutions similaires et proposons des solutions innovantes. Enfin, nous détaillerons la méthodologie qui a guidé notre travail tout au long du projet.

I.2 Contexte générale du projet

Ce projet marque la fin de mon cycle d'ingénieur informatique spécialisé en ISIAD. Il représente l'aboutissement de mon stage de projet de fin d'études, qui a été intégré à mon cursus académique débouchant sur un Diplôme de Cycle Ingénieur Informatique spécialité ISIAD. Ce stage a été une étape importante dans mon évolution de carrière, m'apportant une certification officielle et un ensemble de compétences essentielles à mon domaine d'études. Cela m'a également permis de découvrir les responsabilités du monde professionnel et d'acquérir une expérience précieuse pour ma future intégration sur le marché du travail.

Au cours de mon stage, mon objectif était de me familiariser avec le milieu professionnel, de me plonger dans un secteur spécifique, de m'engager activement dans les opérations de l'entreprise et de m'adapter aux attentes professionnelles. Cette expérience pratique m'a permis de mettre en pratique les connaissances théoriques acquises tout au long de mon parcours académique. Je suis immensément reconnaissant d'avoir été choisi pour un stage à E-TAFAKNA, situé à Tunis, du 12 février 2024 au 12 juin 2024. Je crois fermement que cette opportunité inestimable améliorera de manière transparente mes bases académiques.

I.3 Présentation de L'organisme

Dans cette section, nous décrirons d'abord l'organisation d'accueil, y compris ses domaines d'activité et ses valeurs. Nous introduisons ensuite ce sujet dans la section suivante, en abordant les enjeux qui animent ce projet.

I.3.1 Organisme d'accueil

Fondée en 2022, E-Tafakna propose une plateforme de gestion de contrats. Les utilisateurs peuvent utiliser la plateforme pour créer, personnaliser, gérer, stocker et envoyer des contrats en ligne. De plus, E-Tafakna permet de planifier des réunions virtuelles avec des experts juridiques pour des conseils personnalisés. La plateforme fournit donc une

solution complète pour les besoins de gestion des contrats des utilisateurs. Ainsi, il simplifie les processus liés aux contrats et aux consultations juridiques en ligne. Elle est à la pointe de l'innovation technologique juridique, proposant des applications mobiles complètes qui simplifient la création, la gestion et l'exécution de contrats et de documents juridiquement contraignants.



Figure I.1 : Logo E-Tafakna[1]

- Site :

- <https://www.e-tafakna.com/>

I.3.2 Domaine d'expertise

- **Développement de sites web** : E-Tafakna propose des solutions sur mesure de conception et de développement de sites Web adaptées aux besoins des entreprises.
- **Développement d'applications mobiles** : E-Tafakna se spécialise dans le développement d'applications mobiles pour les plateformes iOS et Android, offrant aux entreprises la possibilité d'atteindre leur public sur les appareils mobiles.
- **Le marketing** : E-Tafakna se démarque dans le secteur du marketing en fournissant des connaissances et des compétences approfondies dans divers domaines tels que l'image de marque et la gestion des médias sociaux.
- **Intelligence artificielle** : Entre autres domaines d'expertise, E-Tafakna se concentre sur l'intelligence artificielle, en fournissant des solutions innovantes et des services de pointe à cette industrie en pleine croissance.

I.3.3 Les valeurs

- **Accords signables électroniquement** : Les capacités de signature électronique permettent de préparer et de signer facilement des documents juridiques, offrant ainsi une accessibilité à tout moment et depuis n'importe quel endroit.
- **Mobilité** : travaillez avec une vitesse inégalée sur n'importe quelle distance.
- **Interface conviviale** : personnalisez facilement vos contrats avec l'assistance d'un expert.
- **Abordable** : une alternative rentable aux services juridiques traditionnels.
- **Fiabilité** : Documents élaborés par des cabinets d'avocats de premier plan.

- **Traçabilité** : signez, envoyez et stockez en toute confiance.
- **Environnemental** : Adoptez un avenir durable et sans papier.
- **Axé sur la technologie** : l'accès aux soins juridiques de base simplifié.

Essentiellement, E-Tafakna se spécialise dans la fourniture de solutions juridiques innovantes qui vous permettent de générer, stocker et superviser sans effort vos contrats. Cette société propose une gamme complète de services, englobant la conception de sites Web, L'intelligence artificielle, le développement d'applications mobiles et diverses solutions marketing.

I.4 Présentation du projet

E-Tafakna prévoit de développer une plateforme en ligne et une application mobile qui fourniront des contrats et des documents juridiquement contraignants personnalisables. Ces documents peuvent être gérés, négociés, signés et envoyés aux parties intéressées par voie électronique. L'initiative vise à fournir aux utilisateurs un moyen simple et efficace de gérer leurs services. Un accent particulier sera mis sur l'application web, qui fournira un tableau de bord complet pour la plateforme E-Tafakna. Le tableau de bord est spécialement conçu pour permettre aux avocats et aux comptables d'accéder facilement aux informations clés, de recevoir des notifications sur les réunions à venir, de suivre les réunions programmées et de gérer efficacement les données des clients. En résumé, le projet vise à créer un système complet de gestion de données et une interface utilisateur intuitive pour les experts sur la plateforme E-Tafakna.

I.4.1 Problématique

La problématique de la conception d'un tableau de bord exhaustive pour les experts juridiques et comptables peut être abordée de diverses manières. Voici quelques exemples de questions centrales qui pourraient être envisagées :

- Comment répondre aux exigences et aux attentes des avocats et des comptables pour assurer un service efficace aux clients ?
- À quels obstacles les avocats et les comptables sont-ils confrontés dans l'exercice de leurs fonctions ?. Quelles solutions pourraient-ils envisager pour surmonter ces obstacles ?
- Comment mesurer la satisfaction des clients des avocats et des comptables, et comment pouvons-nous améliorer leur expérience utilisateur ?
- Comment instaurer un cadre réglementaire efficace pour protéger les parties prenantes et encadrer les activités des avocats et des comptables ?

- Comment mettre en place les nouvelles technologies afin d'améliorer les services juridiques et d'optimiser les processus de réservation avec les experts en droit pour une plus grande efficacité ?

I.5 Étude et Critique de l'existant

Dans cette section, nous allons analyser l'étude de l'existant et en faire une évaluation critique.

I.5.1 Étude de l'existant

Dans cette section, nous évaluons les offres présentes sur le marché, puis nous en faisons une critique. Pour cela, nous avons analysé trois applications web dédiées à l'offre d'emploi.

Clio : La solution de gestion de cabinet juridique est conçue pour répondre aux besoins des cabinets d'avocats, qu'ils soient de grande, moyenne ou petite taille. Elle offre une gamme complète de fonctionnalités intégrées pour optimiser la gestion quotidienne du cabinet. Parmi ces fonctionnalités, on trouve des outils pour suivre les échéances critiques, garantissant ainsi que les délais importants ne sont jamais manqués. La plate-forme permet également de gérer efficacement les factures, en facilitant la facturation et en assurant un suivi précis des paiements. En ce qui concerne la gestion des documents et des dossiers, la solution offre des options robustes pour organiser, stocker et accéder facilement à toutes les informations nécessaires. En outre, elle inclut des outils avancés de suivi du temps, permettant aux avocats de gérer leurs tâches et leurs horaires de manière flexible, qu'il s'agisse de périodes mensuelles, hebdomadaires ou quotidiennes. Cette fonctionnalité est particulièrement utile pour garantir un suivi précis du temps facturable, ce qui est essentiel pour la facturation et la rentabilité du cabinet.

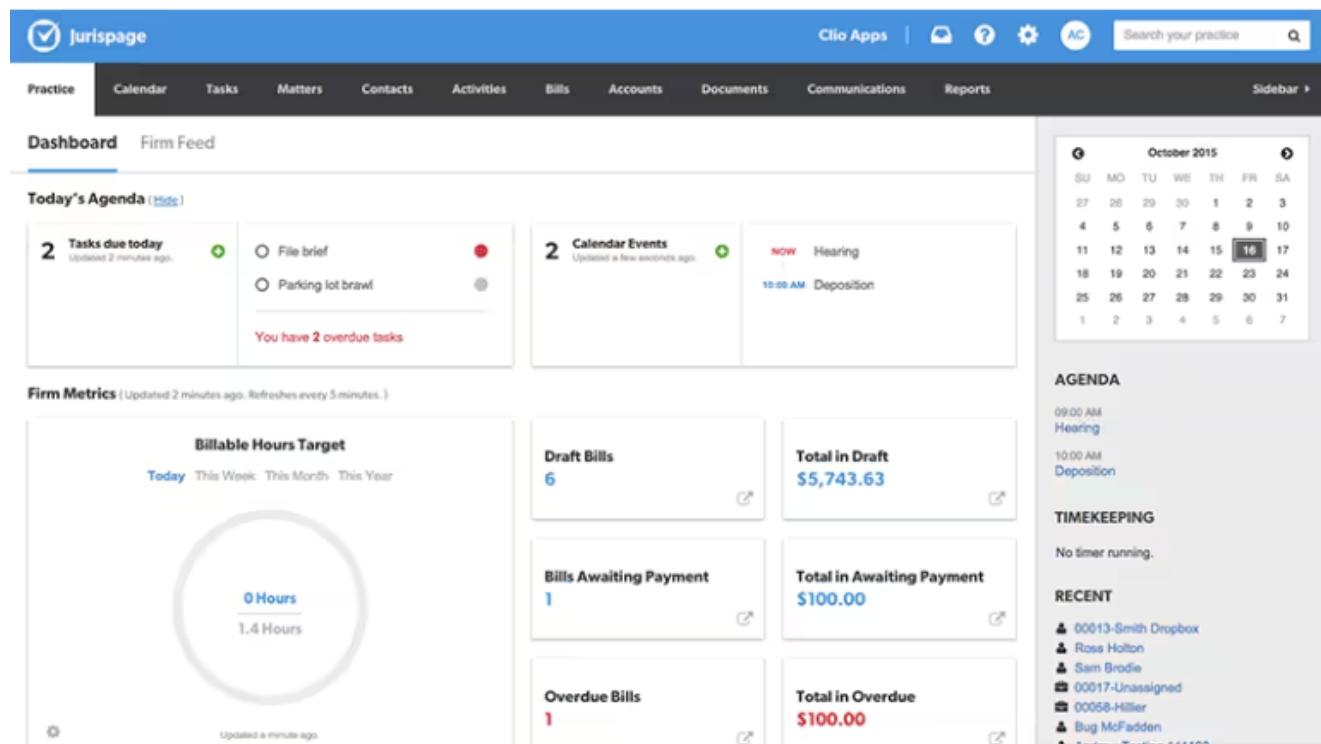


Figure I.2 : Capture d'écran de l'application Clio[2]

Lexicata : se distingue en tant que logiciel de gestion de cabinet spécialement conçu pour gérer les communications et suivre les prospects au sein d'un cabinet d'avocats. Il prend en charge la manipulation de divers documents en ligne et favorise l'augmentation des taux de conversion, tout en maintenant une surveillance sur la clientèle existante.

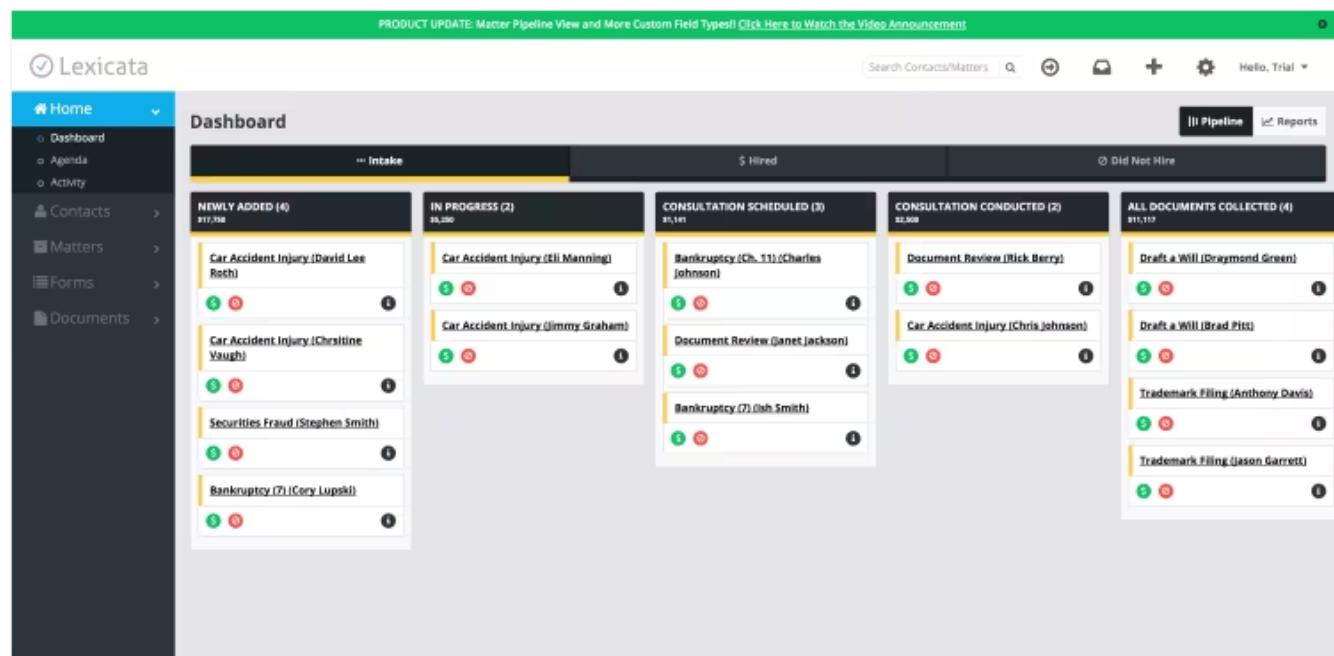


Figure I.3 : Capture d'écran de l'application Lexicata[3]

Jarvis Legal : Jarvis Legal se démarque par son logiciel de gestion de cabinet de bureau, intégrant un ensemble complet d'outils, tels que l'automatisation des documents, la comptabilité fiduciaire, la facturation, les rapports, le suivi du temps, la gestion des dossiers et la planification budgétaire.

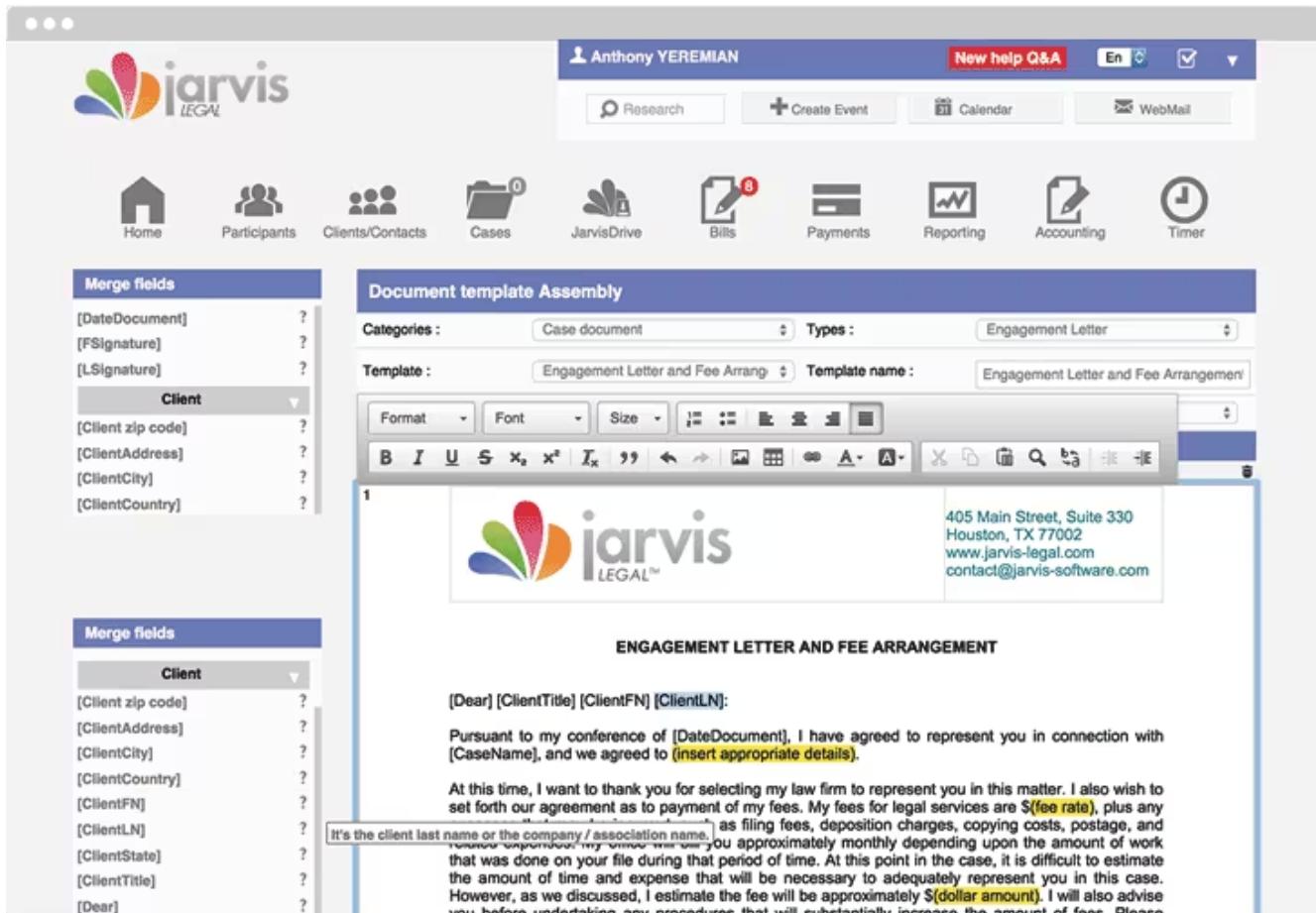


Figure I.4 : Capture d'écran de l'application Jarvis Legal[4]

I.5.2 Critique de l'existant

Malgré les nombreux avantages pratiques qu'offrent ces applications aux particuliers et aux prestataires de services, certaines critiques peuvent également être formulées à leur encontre. Voici quelques exemples de ces critiques :

- Manque de réglementation
- Manque de la planification des réunions
- Absence d'intégration d'un système de chat
- Intégration du courrier électronique compliquée
- Processus de relevés mensuels redondant avec de nombreuses étapes

- Automatisation limitée
- Moins de fonctions par rapport à d'autres programmes

I.6 Solution et Objectif

L'objectif de ce projet consiste à créer une plateforme web et mobile qui facilitera la mise en relation entre les prestataires de services juridiques , comptables et les utilisateurs. Cette plateforme permettra aux utilisateurs de créer, personnaliser, gérer, stocker et envoyer des contrats en ligne. De plus, elle offrira la possibilité de planifier des réunions en ligne avec des experts juridiques et comptables pour obtenir des conseils personnalisés. Par ailleurs, l'application web, sous forme de tableau de bord, sera conçue pour permettre aux avocats et comptables d'accéder facilement à des informations essentielles, de recevoir des notifications concernant les réunions à venir, d'engager une conversation de chat , de gérer des notes et des fichiers spécifiques liés aux clients , Visualiser des graphiques , de suivre les réunions programmées, ainsi que de gérer efficacement les données des clients.

I.7 Méthodologie

Pour élaborer un site web pour un service juridique et comptable, plusieurs étapes s'avèrent essentielles, suivant une méthodologie générale :

- **Planification** : Définir clairement les objectifs du site, cibler le public visé, délimiter les services à offrir, choisir les fonctionnalités à intégrer, établir un budget et fixer des échéances.
- **Conception** : Concevoir l'aspect et la structure du site en harmonie avec la charte graphique de l'entreprise, en utilisant des logiciels de conception ou en faisant appel à un designer professionnel.
- **Développement** : Implémenter le code et programmer le site web en veillant à son bon fonctionnement et à sa compatibilité avec différents navigateurs.
- **Tests** : Effectuer des tests approfondis pour assurer la convivialité du site, son bon fonctionnement et son accessibilité à tous les utilisateurs, y compris ceux ayant des besoins spécifiques en matière d'accessibilité.
- **Lancement** : Mettre en ligne le site web et vérifier son accessibilité à tous les utilisateurs, tout en veillant à son bon référencement sur les moteurs de recherche.
- **Maintenance** : Assurer régulièrement la mise à jour du site pour maintenir sa sécurité, sa pertinence et sa fonctionnalité. Il est également possible d'ajouter de nou-

velles fonctionnalités et du contenu pour améliorer l’expérience utilisateur et répondre aux besoins changeants de l’audience.

I.8 Résultats attendus

Les résultats de cette étude devraient fournir une compréhension plus approfondie des types de services juridiques et comptables les plus courants et les plus rentables pour les prestataires de services, ainsi que leur pertinence pour les clients. Les recommandations pourront aider les prestataires de services juridiques et comptables à mieux comprendre les attentes des clients et à adapter leurs offres de service en conséquence. Les résultats pourront également aider les entreprises à développer de nouveaux services ou à améliorer ceux existants pour mieux répondre aux besoins des clients.

I.9 Conclusion

Ce chapitre a établi les fondations du projet en exposant le contexte, la problématique, l’analyse de l’existant, la solution envisagée , la méthodologie de travail et les résultats attendus. Ces éléments constitueront le socle sur lequel se basera la réalisation du projet et orienteront les étapes suivantes du développement du tableau de bord pour les experts travaillant sur la plateforme E-Tafakna

Chapitre II

Analyse et spécification des besoins

Sommaire

II.1	Introduction	21
II.2	Démarche à suivre	21
II.2.1	Définition d'un cycle de vie	21
II.2.2	Modèle de cycle de vie en Cascade	21
II.2.3	Lien de cycle choisi avec les diagrammes UML	22
II.3	Identification des acteurs de l'application	22
II.4	Identification des besoins	23
II.4.1	Besoin fonctionnels	23
II.4.1.1	Administrateur	23
II.4.1.2	Client	23
II.4.1.3	Utilisateur	23
II.4.2	Besoin non fonctionnels	24
II.5	Modélisation des Besoins	25
II.5.1	Diagramme du cas d'utilisation global	25
II.6	Diagramme de classe	26
II.7	Conclusion	27

II.1 Introduction

Le succès d'un projet repose largement sur la clarté de sa compréhension des exigences. C'est pourquoi la phase de spécification des besoins revêt une importance capitale dans le cycle de développement des projets informatiques. Ce chapitre se focalisera sur l'identification des parties prenantes du projet, l'élaboration détaillée des spécifications fonctionnelles et non fonctionnelles du système, ainsi que la représentation des fonctionnalités à l'aide d'un diagramme UML d'analyse.

II.2 Démarche à suivre

Cette section traitera d'abord de la définition d'un cycle de vie, ensuite nous examinerons le modèle de cycle de vie en Cascade, et enfin, nous analyserons le lien entre le cycle de vie choisi et les diagrammes UML.

II.2.1 Définition d'un cycle de vie

Le concept de "cycle de vie d'un logiciel" englobe toutes les étapes du développement d'un logiciel, de sa conception à sa désuétude. Cette subdivision vise à établir des étapes intermédiaires pour évaluer le développement du logiciel, notamment sa conformité aux besoins exprimés et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes utilisées. Cette approche découle de la constatation que les erreurs ont un impact financier plus important lorsqu'elles sont détectées tardivement dans le processus de développement. Par conséquent, le cycle de vie permet de détecter les erreurs le plus tôt possible, de garantir la qualité du logiciel, et de maîtriser les délais et les coûts associés à sa réalisation.

II.2.2 Modèle de cycle de vie en Cascade

Dans ce modèle de développement, l'approche est simple : à chaque étape, fixée à une date précise, des livrables spécifiques, qu'il s'agisse de documents ou de logiciels, sont produits. Les avancées dépendent des interactions entre les différentes phases et font l'objet d'une évaluation rigoureuse. Seule une performance jugée satisfaisante permet de passer à la phase suivante, comme indiqué dans la **Figure II.1**.

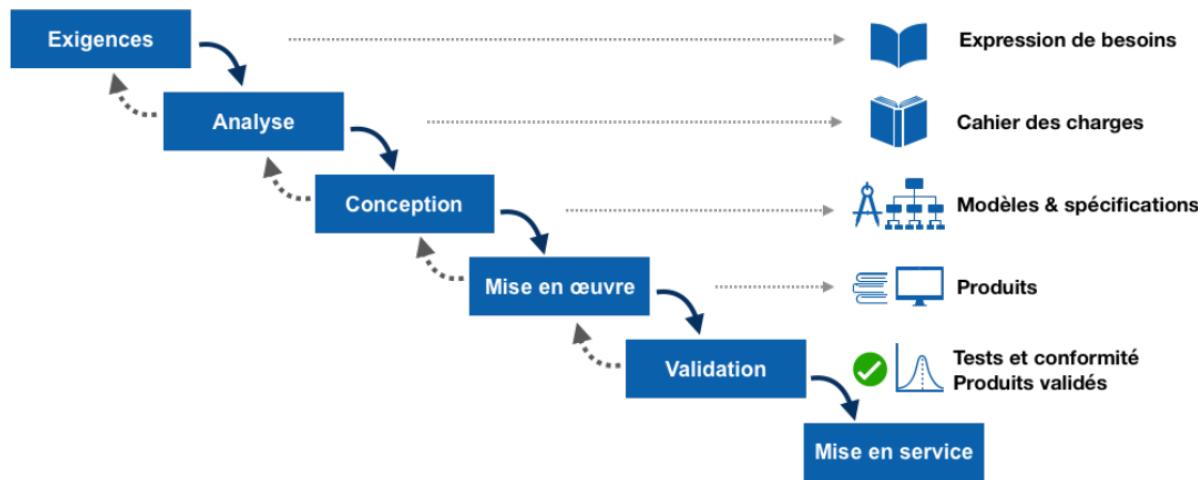


Figure II.1 : Représentation du modèle en cascade[5]

— Pourquoi le modèle en Cascade ?

- Notre projet bénéficie du modèle en cascade, car il découpe le processus de développement en étapes claires. Cela nous permet d'évaluer la faisabilité du projet, de définir la solution et de concevoir en détail avant de passer à l'implémentation. Chaque phase est soigneusement examinée pour garantir des résultats satisfaisants avant de progresser. Cette approche séquentielle assure une gestion efficace du développement du projet.

II.2.3 Lien de cycle choisi avec les diagrammes UML

Les diagrammes UML seront intégrés à toutes les phases de ce modèle pour améliorer la clarté des résultats obtenus. Ces schémas, regroupant treize types distincts, offrent une variété de perspectives pour illustrer les concepts spécifiques à chaque étape du processus.

II.3 Identification des acteurs de l'application

Pour amorcer, nous devons déterminer les acteurs ainsi que leurs interactions avec le système afin de cerner des cas d'utilisation précis et d'envisager toutes les éventualités. Ces acteurs peuvent être des entités externes au système, physiques ou non, qui interagissent avec lui dans le but d'utiliser les services qu'il propose.

II.4 Identification des besoins

Les besoins fonctionnels définissent les actions que le système doit effectuer en réponse à des demandes spécifiques. Comme notre application est structurée en plusieurs modules distincts, chacun de ces modules a ses propres exigences fonctionnelles. Par conséquent, nous allons examiner les fonctionnalités de chaque module de manière distincte.

II.4.1 Besoin fonctionnels

Les exigences fonctionnelles se définissent comme les divers services que l'utilisateur espère obtenir de l'application après avoir soumis une requête. Dans cette partie, nous organisons les besoins fonctionnels par acteur pour notre application web et mobile. Dans notre cas, nous distinguons : Administrateur, Client , Utilisateur(Avocat , Comptable)

II.4.1.1 Administrateur

- Envoyer des notifications

II.4.1.2 Client

- S'authentifier
- Gérer profile
- Sélectionner des contrats
- Scanner des documents
- Gérer son archives
- Gérer des abonnements
- Gérer des rendez-vous en ligne avec un expert juridique ou bien un expert comptable
- Engager une conversation de chat
- Gérer des signatures électroniques

II.4.1.3 Utilisateur

- S'authentifier
- Gérer Client
 - Consulter Client

- Supprimer Client
- Gérer Profil
 - Consulter Profil
 - Modifier Profil
- Gérer Calendrier
 - Consulter Calendrier
 - Consulter listes des évènements
- Gérer Réunion
 - Consulter Réunion
 - Supprimer Réunion
- Gérer Fichier
 - Consulter Fichiers
 - Supprimer Fichiers
 - Ajouter Fichiers
 - Créer Fichiers
 - Modifier Fichiers
- Gérer Notes
 - Consulter Notes
 - Supprimer Notes
 - Ajouter Notes
 - Modifier Notes
- Engager une conversation de chat :
 - Entre les experts juridiques ou comptables
 - entre un expert et un agent intelligent

Dans notre cas nous nous concentrerons sur les besoins des applications web et plus particulièrement les besoins des avocats et comptables.

II.4.2 Besoin non fonctionnels

Pour assurer le bon fonctionnement de l'application et la satisfaction de l'utilisateur, il est crucial de tenir compte de diverses exigences tout au long du développement du projet. Ces exigences incluent :

- **Utilisabilité et convivialité** : L’interface de l’application doit être intuitive et esthétique pour faciliter l’expérience utilisateur.
- **Adaptabilité, maintenance et évolutivité** : L’application doit être compatible avec plusieurs plateformes et opérationnelle sur tous les systèmes d’exploitation. Le code doit être bien structuré, modulaire et facilement compréhensible pour permettre la flexibilité, l’évolution et la maintenance de la solution, adaptée aux besoins changeants du marché.
- **Performance et fiabilité** : L’application doit garantir une réactivité rapide et une performance fiable. Les utilisateurs doivent pouvoir postuler facilement à des offres d’emploi, tandis que les administrateurs doivent pouvoir sélectionner efficacement les candidatures pertinentes de manière rapide et fiable.
- **Sécurité** : Pour assurer la confidentialité des données, l’application doit être munie de contrôles de sécurité pour limiter l’accès aux seuls utilisateurs autorisés. Il est donc essentiel de définir les rôles et les autorisations d’accès de manière à protéger les données sensibles.
- **Temps de réponse** : Pour garantir une expérience utilisateur optimale, il est essentiel de minimiser les délais de réponse et d’assurer une réactivité constante de l’application, quelles que soient les circonstances.

II.5 Modélisation des Besoins

Dans cette partie, nous illustrerons les fonctionnalités déjà discutées précédemment à travers l’utilisation du diagramme de cas d’utilisation. Ces schémas, créés en utilisant le langage UML, offrent une vision claire des interactions entre l’utilisateur et le système, favorisant ainsi une meilleure appréhension du fonctionnement de ce dernier.

II.5.1 Diagramme du cas d’utilisation global

Les cas d’utilisation sont un outil pour comprendre les exigences des utilisateurs d’un système. En utilisant un diagramme de cas d’utilisation, on peut représenter les interactions entre les différents acteurs et le système lui-même, ainsi que les fonctionnalités disponibles pour chaque type d’utilisateur . Dans cette section, nous décrivons les diverses fonctionnalités du système à travers un diagramme de cas d’utilisation global, comme indiqué dans la **figure II.2**. Ce dernier permet de formaliser les besoins de chaque acteur et offre une vision claire du fonctionnement de notre système du point de vue fonctionnel.

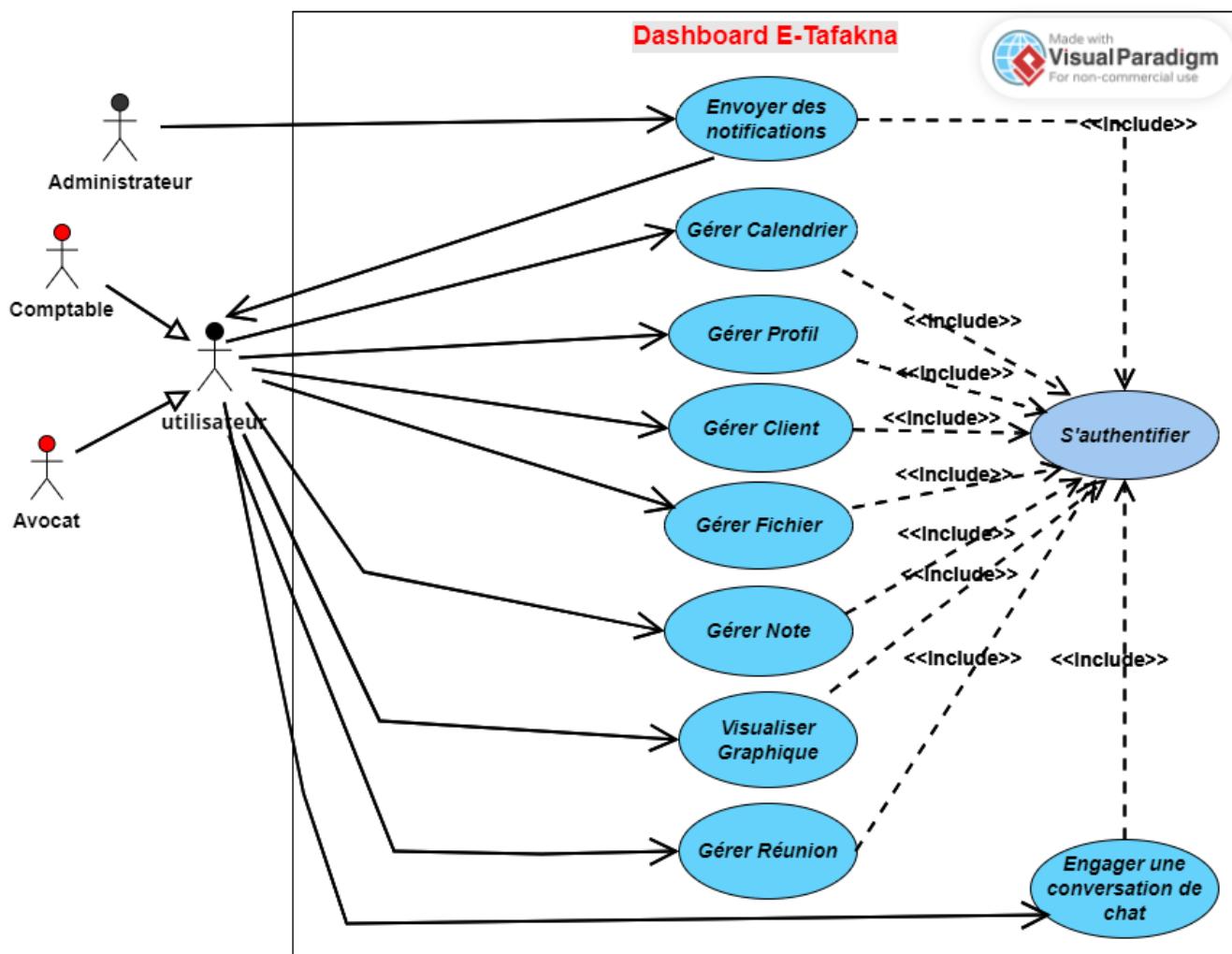


Figure II.2 : Diagramme du cas d'utilisation général

II.6 Diagramme de classe

Les diagrammes de classes servent à visualiser la structure statique d'un système en montrant les classes et les relations qui les relient. Ils offrent une perspective abstraite sur les objets du système et leurs interactions nécessaires pour réaliser les cas d'utilisation. Ces diagrammes organisent les différentes classes d'objets et illustrent leurs relations et associations de manière claire. Voici le diagramme de classe de l'application représenté dans la **figure II.3**.

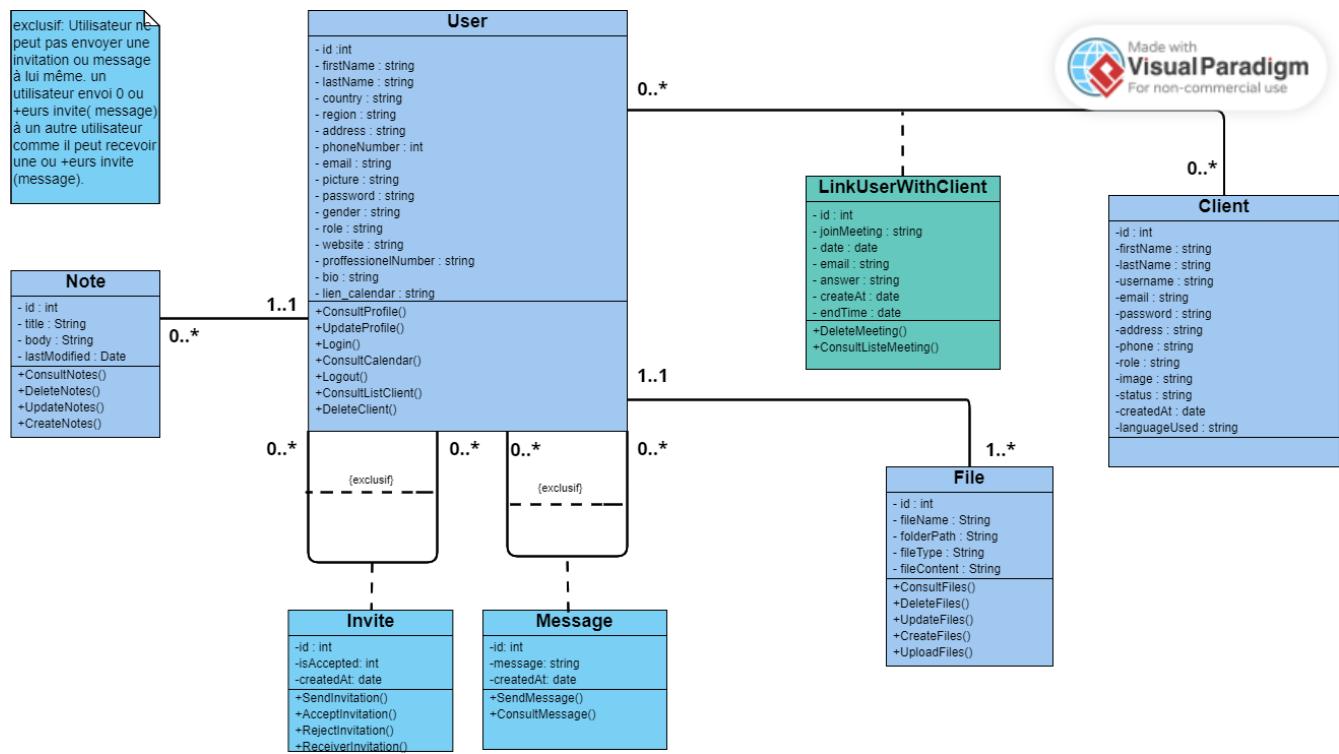


Figure II.3 : Diagramme de classe de l'application web

II.7 Conclusion

Dans cette section, nous avons d'abord examiné le processus de développement de notre projet, puis examiné en détail les exigences en matière de fonctionnalités et de contraintes non fonctionnelles de l'application. Ces exigences sont ensuite représentées à l'aide de schémas UML. Cette analyse nous prépare à aborder en détail l'architecture de l'application dans le chapitre suivant.

Chapitre III

Conception

Sommaire

III.1	Introduction	29
III.2	Diagramme des cas d'utilisation	29
III.2.1	Cas d'utilisation s'authentifier	29
III.2.2	Description textuelle	30
III.2.2.1	S'authentifier	30
III.2.3	Cas d'utilisation gérer client	31
III.2.3.1	Supprimer client	31
III.2.4	Cas d'utilisation gérer calendrier	33
III.2.4.1	Consulter évènement	33
III.2.5	Cas d'utilisation gérer réunion	34
III.2.5.1	Supprimer réunion	35
III.2.6	Cas d'utilisation gérer profil	36
III.2.6.1	Gérer profil	37
III.2.7	Cas d'utilisation gérer fichier	38
III.2.7.1	Gérer fichier	39
III.2.8	Cas d'utilisation gérer note	41
III.2.8.1	Gérer note	41
III.2.9	Cas d'utilisation Engager une conversation de chat	43
III.2.9.1	Conversation du chat	43
III.2.10	Cas d'utilisation Visualiser les graphiques	45
III.2.10.1	Visualisation des graphiques	45
III.3	Diagramme de séquence	47
III.3.0.1	Diagramme de séquence d'authentification	47
III.3.0.2	Diagramme de séquence consulter liste client	48
III.3.0.3	Diagramme de séquence consulter liste de réunion	49
III.3.0.4	Diagramme de séquence supprimer réunion	50
III.3.0.5	Diagramme de séquence modifiée profil	51
III.3.0.6	Diagramme de séquence de chat entre les experts	52
III.4	Conclusion	53

III.1 Introduction

Après avoir établi les exigences et les buts dans la section précédente, notre attention se tourne désormais vers la structure architecturale de l'application. Cette phase consiste à créer des diagrammes globaux qui servent à représenter et à expliquer de façon précise le fonctionnement désiré de l'application.

III.2 Diagramme des cas d'utilisation

Dans cette section, nous exposons les fonctionnalités du système à travers un schéma global de cas d'utilisation. Ce diagramme vise à formaliser les exigences de chaque utilisateur de façon structurée et à offrir une perspective sur le comportement fonctionnel de notre système.

III.2.1 Cas d'utilisation s'authentifier

La **figure III.1** offre un aperçu du processus d'authentification. Une fois authentifiés, les utilisateurs sont en mesure d'accéder à des fonctionnalités spécifiques de l'application, en fonction des autorisations qui leur sont conférées. Chaque utilisateur doit fournir un identifiant et un mot de passe lors de chaque tentative de connexion pour accéder au système.

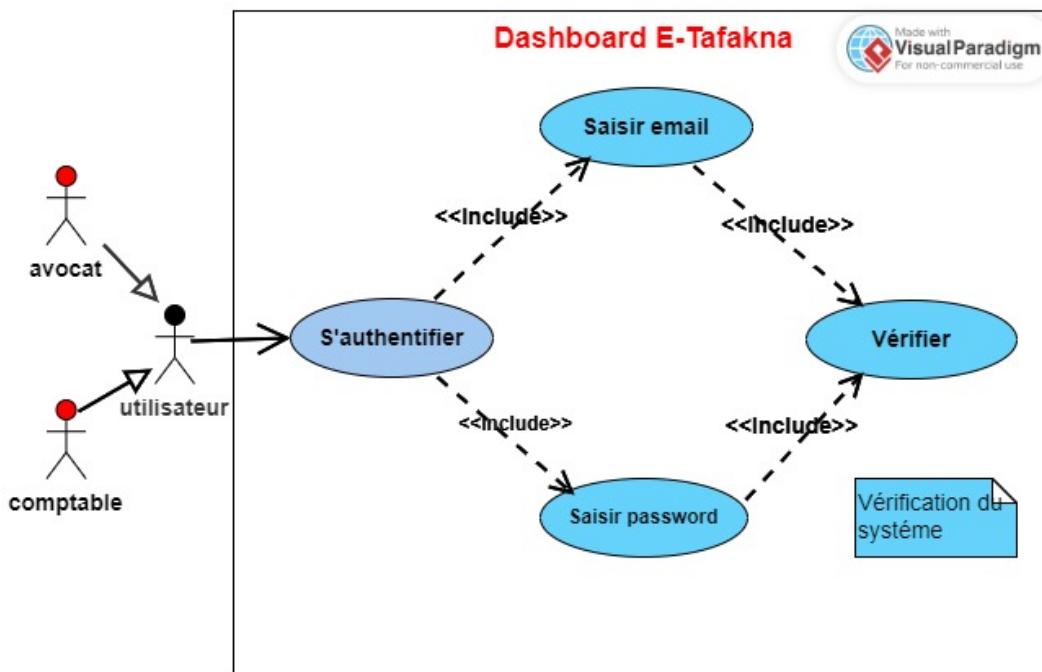


Figure III.1 : Cas d'utilisation S'authentifier

III.2.2 Description textuelle

III.2.2.1 S'authentifier

Dans ce processus d'analyse, nous allons plonger en profondeur dans le cas d'utilisation "S'authentifier", qui constitue une étape fondamentale dans de nombreux systèmes informatiques. L'authentification vise à vérifier l'identité d'un utilisateur pour lui accorder l'accès aux fonctionnalités ou aux ressources appropriées

Table III.1 : Description du cas d'utilisation "S'authentifier"

Cas d'utilisation	Authentification
Acteur	Utilisateur.
But	S'authentifier.
Pré-conditions	Compte doit être existant. Accéder à l'interface login.
Post-condition	L'utilisateur est dirigé vers la page d'accueil.
Scénario nominal	L'utilisateur atteint la page principale de l'application. Le système présente la page d'authentification. L'utilisateur doit remplir ces données (email et mot de passe) puis il clique sur le bouton "Login". Le système affiche donc la page d'accueil.
Alternatives	Si les informations saisies comportent des erreurs ou sont incomplètes, le système informe l'utilisateur de l'échec et lui demande de rectifier ces données.

III.2.3 Cas d'utilisation gérer client

La **figure III.2** met en lumière le diagramme du cas d'utilisation spécifique aux avocats et aux comptables, offrant une vue détaillée sur la procédure de suppression des clients dans leur système. Ce diagramme illustre les actions et les interactions nécessaires pour réaliser cette tâche, offrant ainsi une compréhension claire du processus pour ces utilisateurs spécifiques.

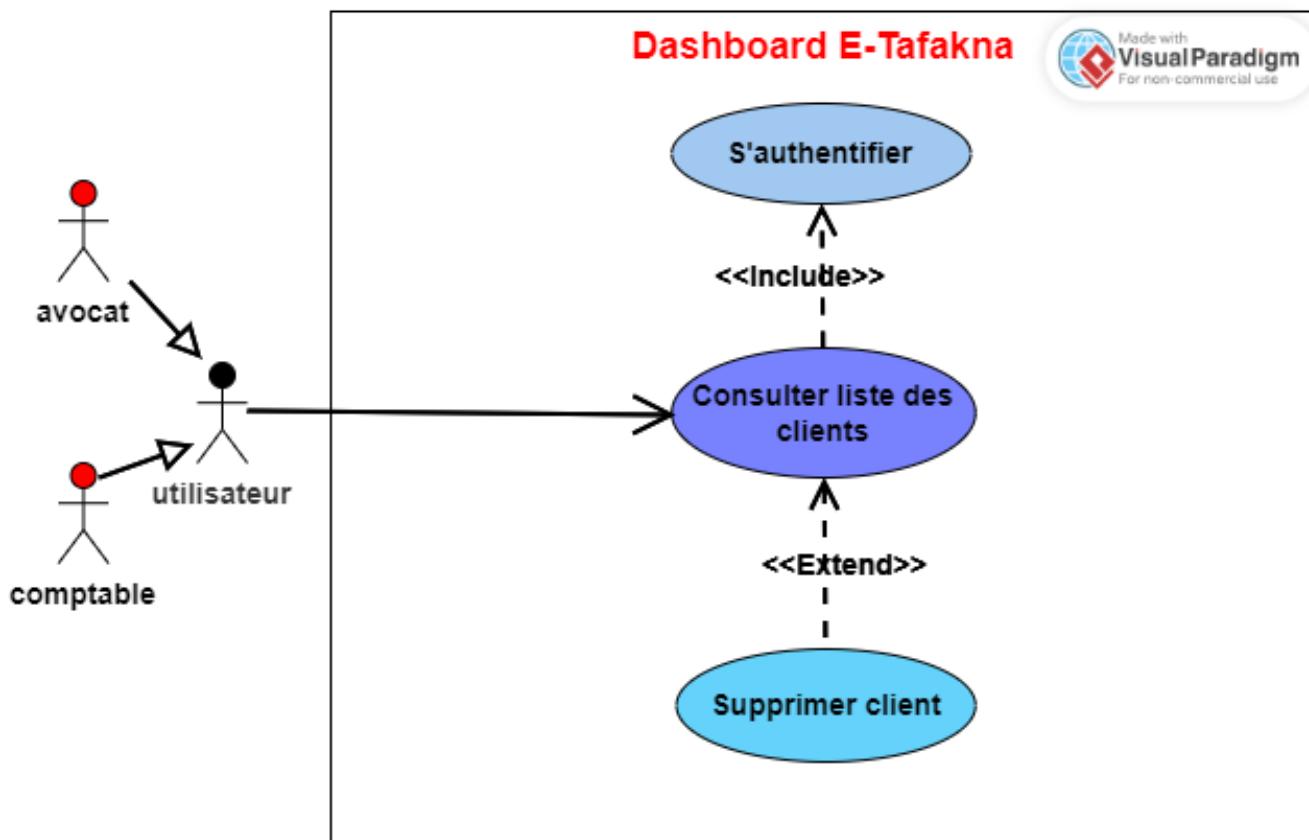


Figure III.2 : Cas d'utilisation Gérer client

III.2.3.1 Supprimer client

Dans cette analyse détaillée, nous examinerons minutieusement le processus de suppression de clients, en décomposant chaque étape avec précision afin d'acquérir une compréhension approfondie de son fonctionnement. Notre objectif est d'identifier les actions spécifiques nécessaires à cette opération, ainsi que les implications potentielles à chaque étape, afin d'optimiser ce processus et d'en assurer la fiabilité.

Table III.2 : Description du cas d'utilisation "Supprimer client"

Cas d'utilisation	Supprimer client
Acteur	Utilisateur.
But	Supprimer client .
Pré-conditions	S'authentifier. Accéder à l'interface de gestion des clients. Consulter la liste des clients. Supprimer un client.
Post-condition	Pouvoir supprimer un nouvel client.
Scénario nominal	L'acteur appuie sur le bouton marqué "Clients". La liste des clients apparaît à l'écran. L'utilisateur clique sur le bouton "delete".
Alternatives	En cas d'erreur un message s'affiche.

III.2.4 Cas d'utilisation gérer calendrier

La **figure III.3** montre le diagramme du cas d'utilisation relatif aux utilisateurs, leur offrant la possibilité de consulter leurs événements afin d'accéder à la date de la prochaine réunion.

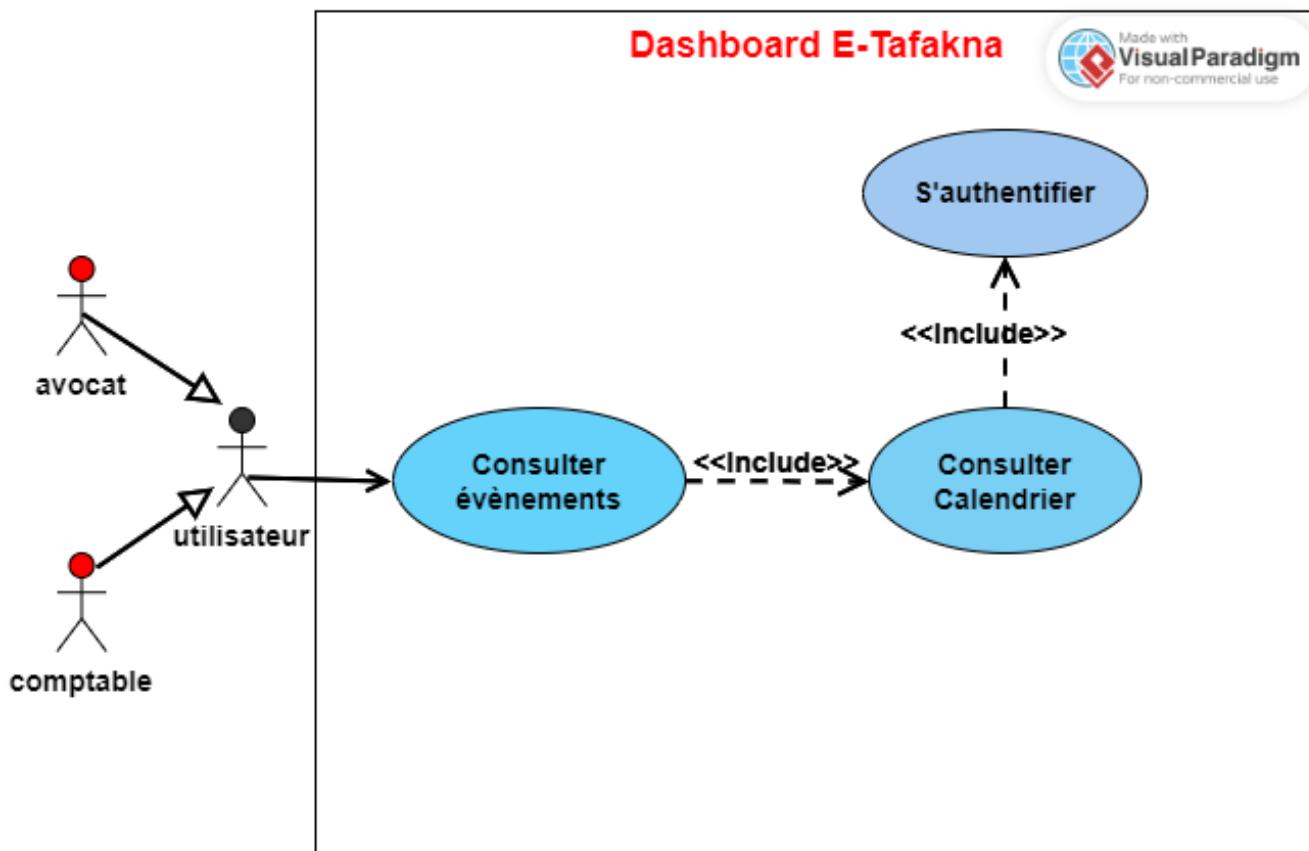


Figure III.3 : Cas d'utilisation Gérer calendrier

III.2.4.1 Consulter évènement

Nous allons explorer en détail le processus de consultation d'événements dans ce tableau, en mettant en lumière les étapes et les mécanismes impliqués dans ce cas d'utilisation spécifique.

Table III.3 : Description du cas d'utilisation "Consulter évènement"

Cas d'utilisation	Consulter évènement
Acteur	Utilisateur.
But	Consulter évènement .
Pré-conditions	S'authentifier. Accéder à l'interface de gestion calendrier. Consulter la liste des évènements.
Post-condition	Pouvoir avoir d'autres évènements.
Scénario nominal	L'utilisateur clique sur le bouton "Calendar" Un calendrier est affiché, montrant la liste des événements avec leur date associée. Le participant examine la liste des événements.

III.2.5 Cas d'utilisation gérer réunion

Dans la **figure III.4**, nous allons examiner de près le diagramme du cas d'utilisation "Gérer réunion", dans le but de fournir une explication détaillée de son fonctionnement.

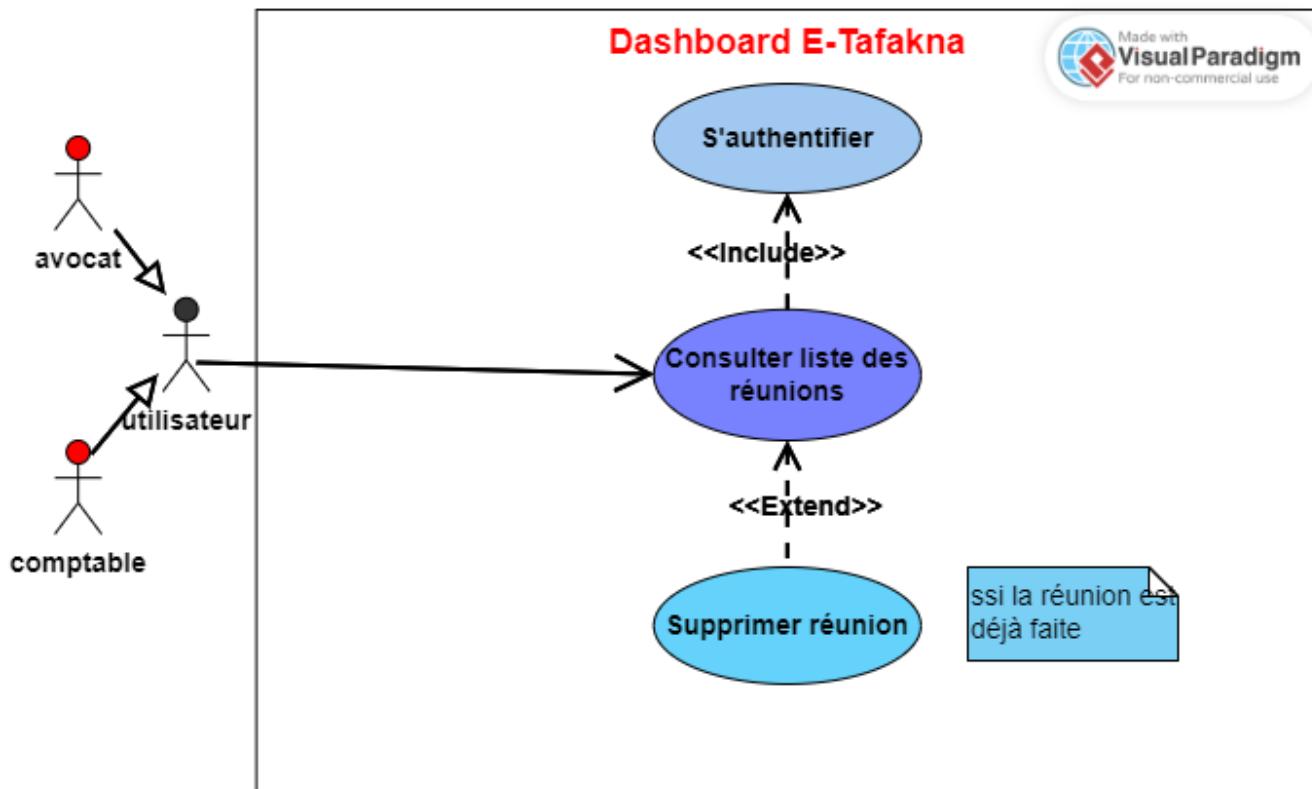


Figure III.4 : Cas d'utilisation Gérer réunion

III.2.5.1 Supprimer réunion

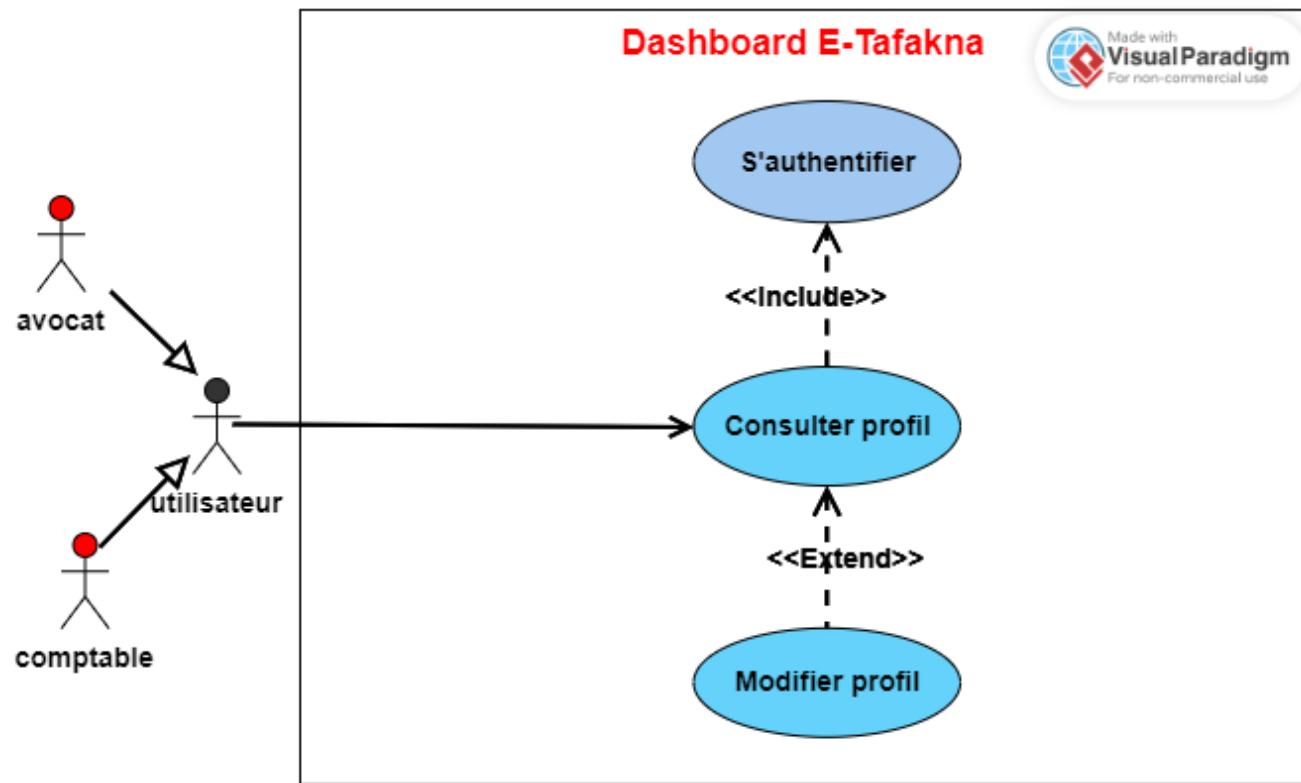
Nous examinerons en détail le processus de suppression d'une réunion, en identifiant les étapes nécessaires pour initier et confirmer la suppression. Ensuite, nous analyserons les actions spécifiques effectuées lors de la suppression, telles que la mise à jour des données et la notification des participants. Nous aborderons également les mesures de sécurité mises en place pour prévenir les suppressions non autorisées. Enfin, nous étudierons les stratégies de gestion des erreurs en cas de problèmes pendant la suppression.

Table III.4 : Description du cas d'utilisation "Supprimer réunion"

Cas d'utilisation	Supprimer réunion
Acteur	Utilisateur.
But	Supprimer la réunion .
Pré-conditions	S'authentifier. Accéder à l'interface de gestion de réunion. Consulter la liste des réunions.
Post-condition	Pouvoir supprimer d'autres réunions.
Scénario nominal	L'utilisateur clique sur le bouton "Meetings". La liste des réunions apparaît à l'écran. L'utilisateur clique sur le bouton "delete". Le système procède à une vérification de la suppression des réunions en fonction de la date. Si la validation est confirmée, la réunion est supprimée et un message de succès s'affiche.
Alternatives	En cas d'erreur un message s'affiche "Link can only be deleted after the meeting or date has passed".

III.2.6 Cas d'utilisation gérer profil

La **figure III.5** présente le diagramme de cas d'utilisation relatif aux actions disponibles pour les utilisateurs afin de modifier ou supprimer leur profil. Cette partie se concentre spécifiquement sur ces fonctionnalités.

**Figure III.5 :** Cas d'utilisation Gérer profil

III.2.6.1 Gérer profil

Nous allons examiner de près le processus de gestion de profil dans ce tableau

Table III.5 : Description du cas d'utilisation "Gérer profil"

Cas d'utilisation	Gérer profil
Acteur	Utilisateur.
But	Modifier profil.
Pré-conditions	S'authentifier. Accéder à l'interface de gestion de profil. Consulter son profil.
Post-condition	Avoir la capacité de mettre à jour son profil.

- **Modifier le profil d'un utilisateur :**

♠ **Description du scénario nominal :**

- N1 : L'utilisatur clique sur le bouton « Profile Form ».
- N2 : Le profil est affiché
- N3 : L'utilisatur clique sur le bouton « edit ».
- N4 : La personne choisit les informations à modifier.
- N5 : Il appuie sur le bouton « update ».

♠ **Description du scénario exceptionnel :** Si une erreur de saisie survient, un message est généré pour signaler le champ manquant ou incorrect.

III.2.7 Cas d'utilisation gérer fichier

Le diagramme de cas d'utilisation de la **figure III.6** présente de manière détaillée les multiples options disponibles aux utilisateurs pour interagir avec leurs fichiers ou dossiers. Il met en évidence les fonctionnalités spécifiques telles que la modification, la suppression et l'ajout de fichiers ou de dossiers dans le système. Chaque élément du processus est examiné en profondeur, offrant ainsi une vision complète des actions et des scénarios possibles. Ce diagramme vise à fournir une compréhension exhaustive des capacités offertes aux utilisateurs pour gérer efficacement leurs données..

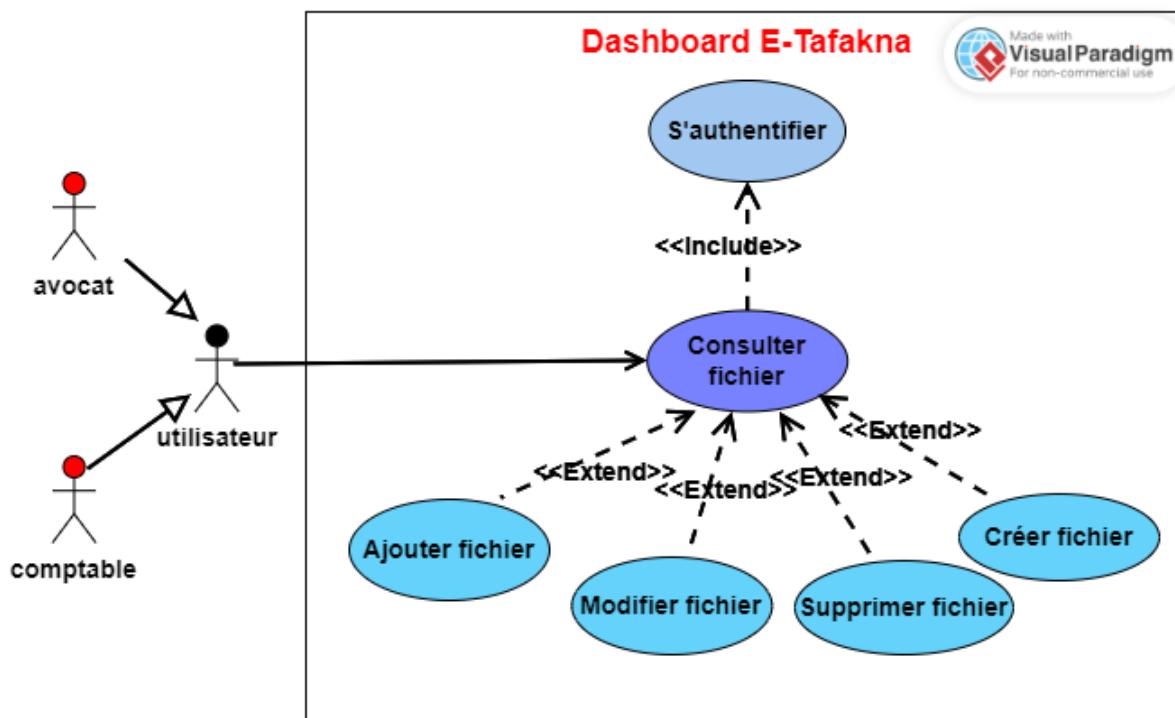


Figure III.6 : Cas d'utilisation Gérer fichier

III.2.7.1 Gérer fichier

Nous avons élaboré une description complète du cas d'utilisation "Gérer fichier" dans le tableau présenté. Cette présentation détaillée couvre les diverses étapes et fonctionnalités associées à ce processus, offrant ainsi une vue d'ensemble claire de ses implications. Chaque aspect de la gestion des fichiers, de leur création à leur modification ou leur suppression, a été examiné en profondeur pour une compréhension complète des actions impliquées. Cette analyse approfondie vise à fournir une base solide pour la mise en œuvre efficace de ce cas d'utilisation dans le système.

Table III.6 : Description du cas d'utilisation "Gérer fichier"

Cas d'utilisation	Gérer fichier
Acteur	Utilisateur.
But	Ajouter fichier. Modifier fichier. Supprimer fichier. Créer fichier
Pré-conditions	S'authentifier. Consulter l'interface pour la gestion des fichiers et des dossiers.
Post-condition	Ajouter d'autres fichiers et dossiers. Pouvoir supprimer ou modifier des fichiers et dossiers.

- **Créer un dossier ou fichier :**

♠ Description du scénario nominal :

N1 : L'utilisateur clique sur le bouton « Files ».

N2 : Maintenant, la page des dossiers et fichiers est visible.

N3 : L'utilisateur appuie sur le bouton «add Folder».

N4 : Alert s'affiche pour ajouter le nom du dossier

N5 : L'utilisateur clique sur le bouton «save»

N6 : L'utilisateur clique sur le dossier créé.

N7 : Le dossier s'ouvre.

N8 : L'utilisateur a la possibilité de créer d'autres sous-dossiers ou de créer des fichiers.

N9 : L'utilisateur appuie sur le bouton «add Files ».

N10 : Le fichier s'ouvre.

N11 : L'utilisateur saisit ces informations, puis appuie sur le bouton « save ».

- **Modifier un fichier :**

♠ **Description du scénario nominal :**

N1 : L'utilisateur clique sur le bouton « Files ».

N2 : La liste des dossiers et fichiers s'affiche

N3 : L'utilisateur clique sur le bouton « edit ».

N4 : L'utilisateur choisit les informations à modifier

- **Supprimer un fichier ou dossier :**

♠ **Description du scénario nominal :**

N1 : L'utilisateur clique sur le bouton « Files ».

N2 : La liste des dossiers et fichiers s'affiche

N3 : L'utilisateur clique sur le bouton « delete ».

N4 : Une Alert s'affiche

N5 : L'utilisateur clique sur "Yes" pour confirmé l'opération ou sur "No" pour annuler.

- **Ajouter un fichier :**

♠ **Description du scénario nominal :**

N1 : L'utilisateur clique sur le bouton « Upload file ».

N2 : Alert s'affiche pour choisir un fichier

N3 : L'utilisateur clique sur le bouton « save » ou sur « cancel» pour annuler l'opération.

III.2.8 Cas d'utilisation gérer note

La **figure III.7** illustre le schéma du cas d'utilisation "Gérer note" pour les avocats et les comptables qui ont la possibilité d'ajouter, de modifier ou de supprimer une note.

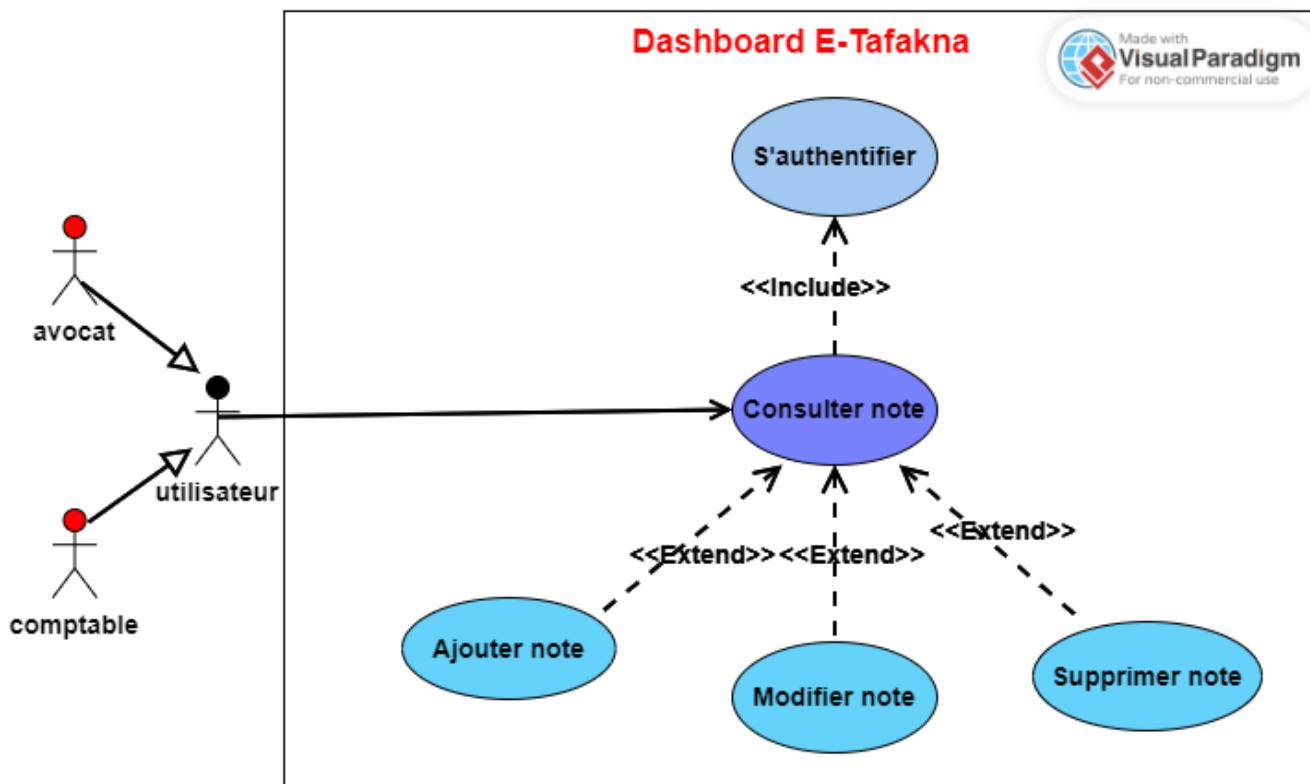


Figure III.7 : Cas d'utilisation Gérer note

III.2.8.1 Gérer note

Dans ce tableau, nous devons détailler la réalisation du cas d'utilisation "Gérer note". Cela comprend les étapes pour créer, modifier, afficher et supprimer une note, avec des descriptions claires et précises de chaque action. Nous inclurons les acteurs impliqués, les conditions préalables, et les résultats attendus. Des exemples visuels et les règles métiers associées seront également fournis pour une compréhension complète.

Table III.7 : Description du cas d'utilisation "Gérer note"

Cas d'utilisation	Gérer note
Acteur	Utilisateur.
But	Ajouter note. Modifier note. Supprimer note.
Pré-conditions	S'authentifier. Consulter notes.
Post-condition	Ajouter d'autres notes Pouvoir supprimer ou modifier des notes.

- **Ajouter un note :**

♠ **Description du scénario nominal :**

- N1 : L'utilisateur clique sur le bouton « Notes ».
- N2 : Maintenant, la page des notes s'affiche.
- N3 : L'utilisateur appuie sur le bouton «add Note».
- N4 : L'utilisateur rédige une note.

- **Modifier un note :**

♠ **Description du scénario nominal :**

- N1 : L'utilisateur clique sur le bouton « Notes ».
- N2 : Maintenant, la page des notes s'affiche.
- N3 : L'utilisateur clique sur la note qu'il souhaite modifier.
- N4 : L'utilisateur effectue les mises à jour nécessaires.

- **Supprimer un fichier ou dossier :**

♠ **Description du scénario nominal :**

- N1 : L'utilisateur clique sur le bouton « Notes ».

- N2 : La liste des notes s'affiche
- N3 : L'utilisateur clique sur le bouton « delete ».
- N4 : Une Alert s'affiche
- N5 : L'utilisateur peut confirmer ou annuler la suppression.

III.2.9 Cas d'utilisation Engager une conversation de chat

Dans cette **figure III.7** on va présenter le schéma du cas d'utilisation "Engager une conversation de chat" pour les experts juridiques et comptables qui ont la possibilité d'envoyer et consulter les messages.

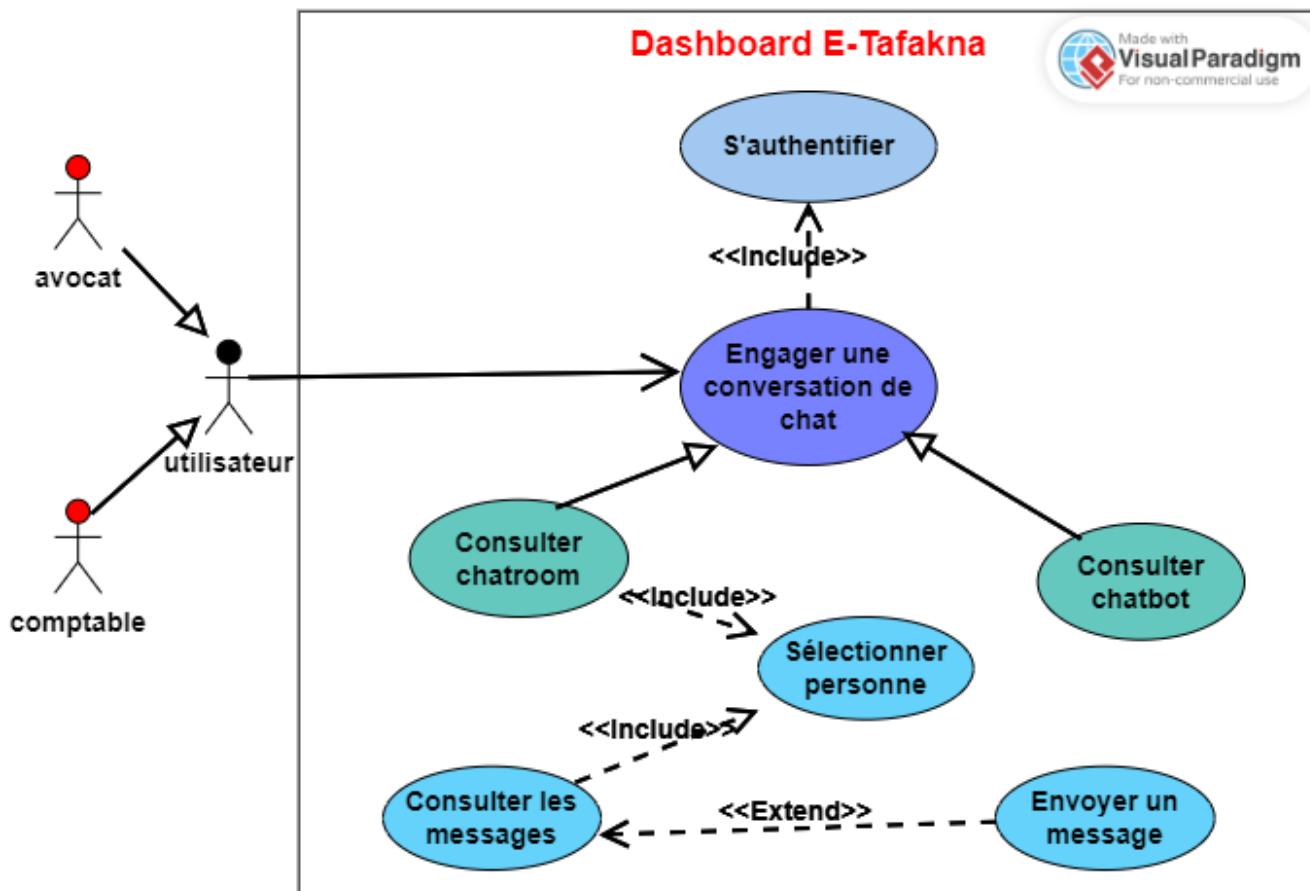


Figure III.8 : Cas d'utilisation Engager une conversation de chat

III.2.9.1 Conversation du chat

Dans ce tableau, nous avons minutieusement examiné et décrit les différentes caractéristiques, les spécificités ainsi que les scénarios possibles liés à ce cas d'utilisation. Nous avons également inclus des exemples concrets et des explications approfondies pour chaque aspect pertinent, afin de fournir une compréhension complète et détaillée de la situation analysée.

Table III.8 : Description du cas d'utilisation "Gérer une conversation de chat"

Cas d'utilisation	Avoir une conversation de chat
Acteur	Utilisateur.
But	Sélectionner personne ou un agent intelligent Envoyer message.
Pré-conditions	S'authentifier. Consulter l'interface de chatRoom. Consulter l'interface de chatbot.
Post-condition	Sélectionner d'autres personnes ou l'agent intelligent avec la possibilité d'envoyer des messages.

- **Sélectionner personne ou un agent intelligent :**

♠ **Description du scénario nominal :**

N1 : L'utilisateur clique sur le bouton « Chat ».

N2 : Maintenant, La page de sélection de chat s'affiche.

N3 : L'utilisateur choisir un type de chat avec un autre utilisateur ou avec un assistant intelligent

N4 : L'interface des messages s'affiche.

- **Ajouter un message :**

♠ **Description du scénario nominal :**

N1 : Si l'utilisateur clique sur le bouton « Go to chatRoom ».

N2 : La page de chatRoom s'affiche.

N3 : L'utilisateur choisit un ami.

N4 : L'utilisateur entre un message puis, il appuie sur le bouton .

N5 : Si l'utilisateur clique sur le bouton « Go to chatBot ».

N6 : La page de chatBot s'affiche.

N7 : L'utilisateur entre un message puis, il appuie sur le bouton « Send ».

III.2.10 Cas d'utilisation Visualiser les graphiques

La **figure III.9** suivante illustre le cas d'utilisation "Visualiser Graphiques", où un expert juridique ou comptable peut accéder à son tableau de bord et consulter la carte géographique des clients par pays.

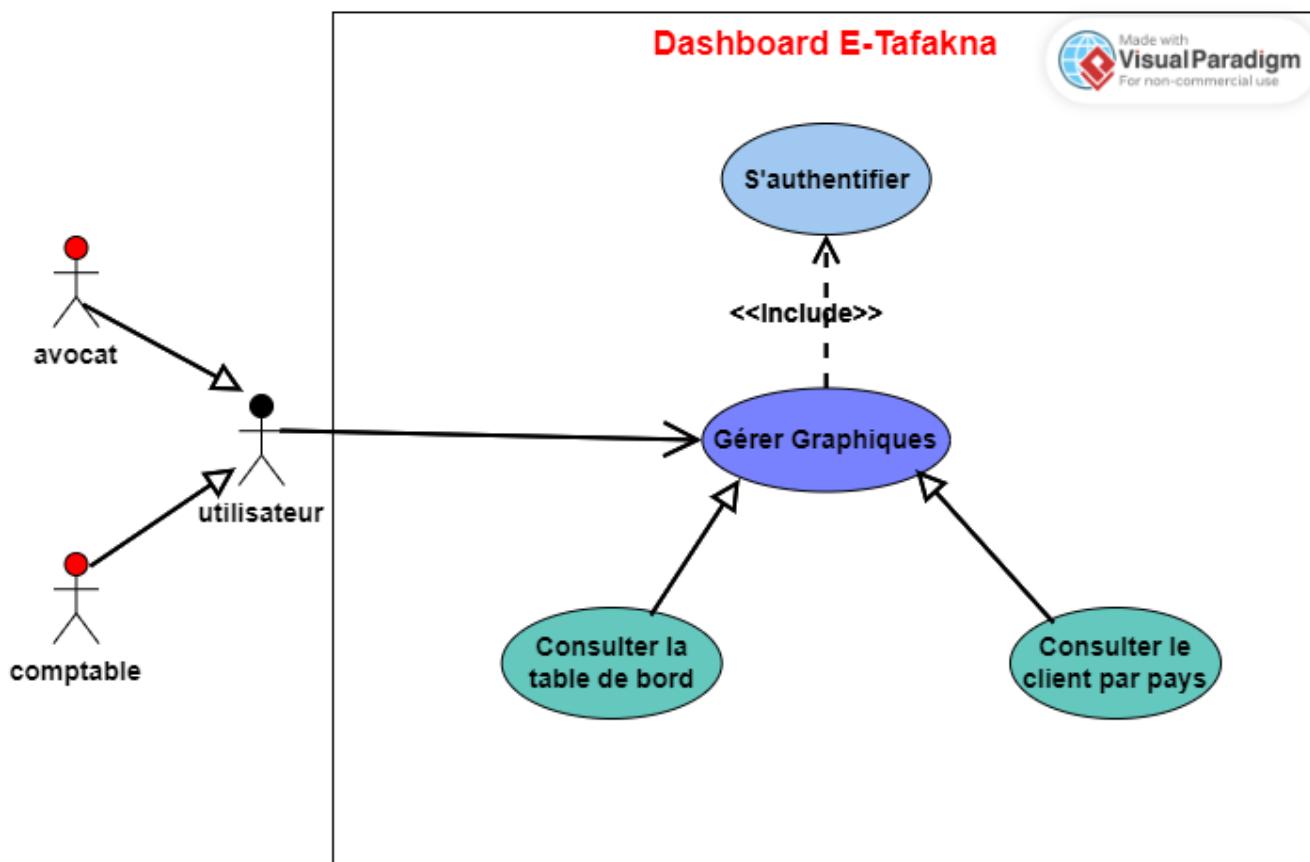


Figure III.9 : Cas d'utilisation Visualiser les graphiques

III.2.10.1 Visualisation des graphiques

Dans ce tableau, nous allons décrire en détail la réalisation du cas d'utilisation "Visualiser Graphiques". Cela inclut les étapes pour afficher le tableau de bord et la carte géographique, accompagnées de descriptions claires et précises de chaque action. Nous préciserons les acteurs impliqués, les conditions préalables et les résultats attendus. Des exemples visuels et les règles métier associées seront également fournis pour garantir une compréhension complète.

Table III.9 : Description du cas d'utilisation "Visualiser les graphiques"

Cas d'utilisation	gérer graphique
Acteur	Utilisateur.
But	Voir des courbes. Voir le diagramme en secteur . Voir le diagramme en bâtons. Voir la courbe la carte géographique.
Pré-conditions	S'authentifier. Consulter la table de bord . Consulter la carte géographique.
Post-condition	possibilité de visualiser à nouveau les diagrammes et les courbes avec leurs mises à jour.

- **Consulter des courbes :**

♠ **Description du scénario nominal :**

- N1 : L'utilisateur clique sur le bouton « dashboard ».
- N2 : Maintenant, La page de table de bord s'affiche.
- N3 : L'utilisateur clique sur l'icône de réunion.
- N4 : L'interface des courbes s'affiche.

- **Consulter le diagramme en secteur :**

♠ **Description du scénario nominal :**

- N1 : L'utilisateur clique sur le bouton « dashboard ».
- N2 : La page de table de bord s'affiche.
- N3 : Pour obtenir plus de détails, l'utilisateur clique sur le bouton « pie chart ».
- N4 : La page s'affiche.

- **Consulter le diagramme à barres :**

♠ **Description du scénario nominal :**

- N1 : L'utilisateur clique sur le bouton « dashboard ».
- N2 : La page de table de bord s'affiche.
- N3 : L'utilisateur clique sur l'icône de client.
- N4 : L'interface s'affiche.

- **Consulter la carte géographique :**

- ♠ **Description du scénario nominal :**

- N1 : L'utilisateur clique sur le bouton « geography chart ».
 - N2 : La page de la carte s'affiche.

III.3 Diagramme de séquence

Le diagramme de séquence est un outil utilisé pour illustrer les interactions entre les différents objets impliqués dans un scénario spécifique d'un diagramme de cas d'utilisation. Son but est de fournir une description détaillée des opérations effectuées, y compris les messages échangés et leur ordre chronologique. Dans cet exemple, nous examinerons le diagramme de séquence associé aux cas d'utilisation suivants : "Authentification", "Consulter liste client", "Supprimer réunion", "Modifier profil" et "Engager une conversation de chat".

III.3.0.1 Diagramme de séquence d'authentification

Dans la **figure III.10**, nous explorerons comment l'utilisateur interagit avec le système pendant le processus d'authentification. Le diagramme de séquence détaillera les principales étapes, telles que la soumission des informations d'identification par l'utilisateur, la vérification de ces informations par le système, et la réponse correspondante de celui-ci.

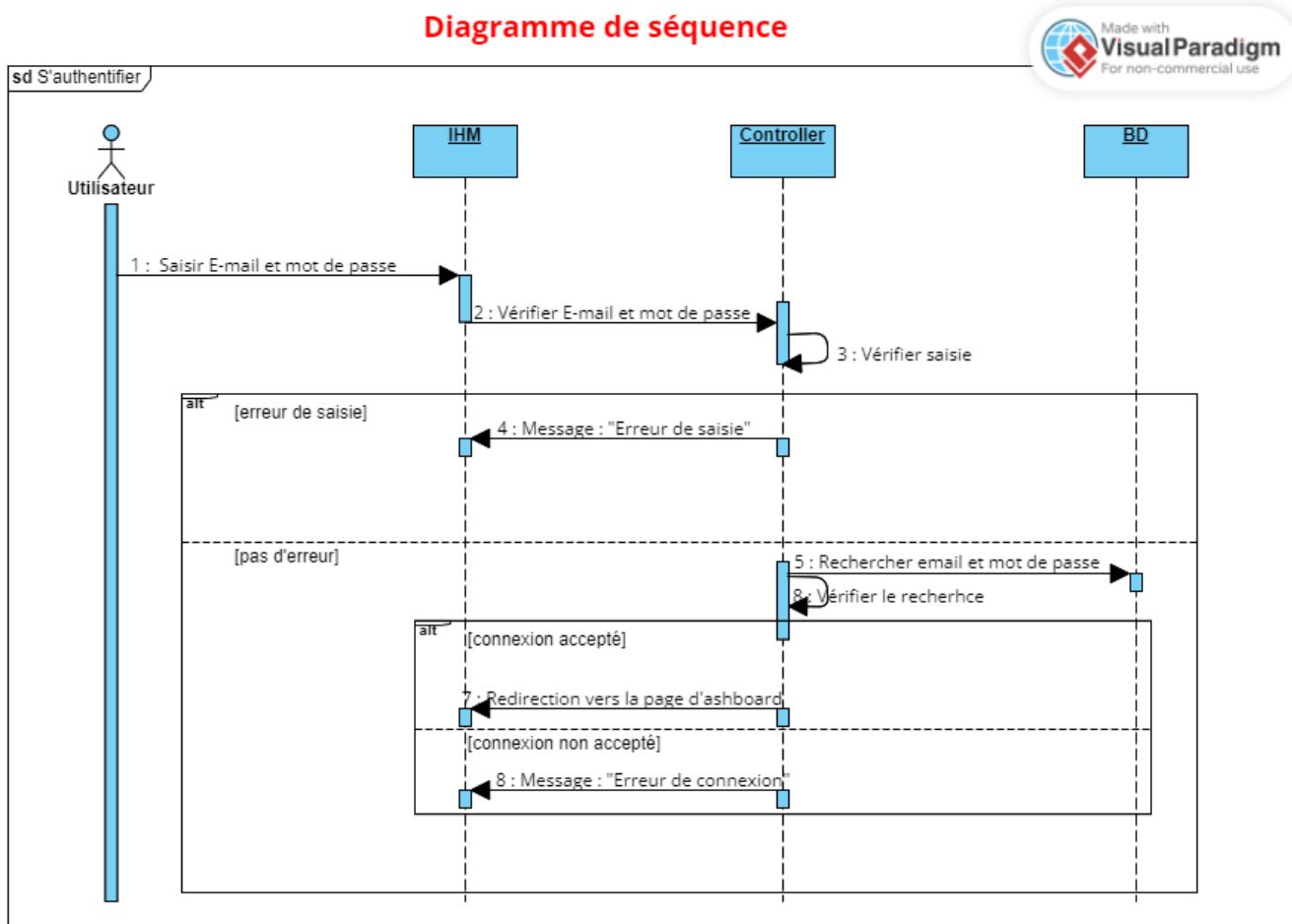


Figure III.10 : Diagramme de séquence du scénario « S'authentifier »

III.3.0.2 Diagramme de séquence consulter liste client

Après que l'avocat ou le comptable se soit authentifié, la séquence suivante est déclenchée. Une fois que la page d'accueil est chargée, l'avocat ou le comptable clique sur le bouton "Clients", ce qui entraîne l'affichage de la liste des clients, comme indiqué dans la **figure III.11**.

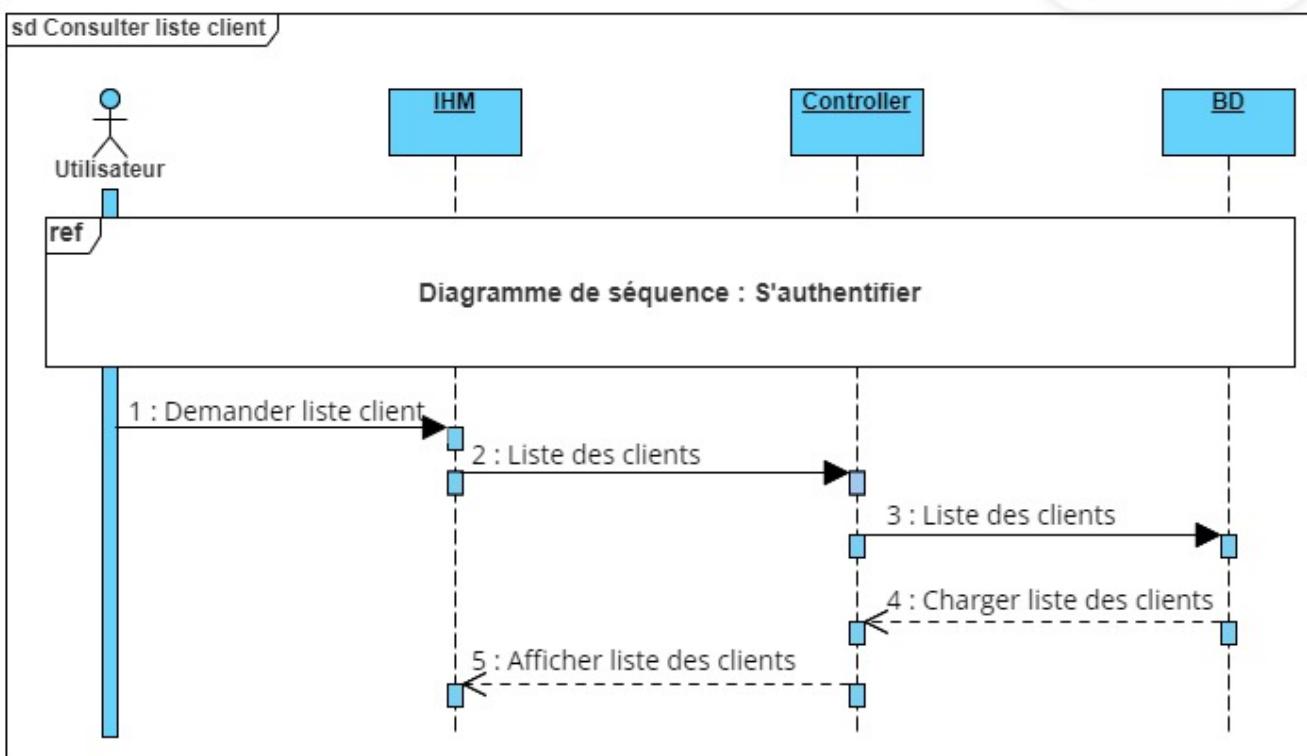


Figure III.11 : Diagramme de séquence du scénario « Consulter liste client»

III.3.0.3 Diagramme de séquence consulter liste de réunion

Après que l'avocat ou le comptable se soit authentifié, la séquence suivante est déclenchée. Une fois que la page d'accueil est chargée, l'avocat ou le comptable clique sur le bouton "Meetings", ce qui entraîne l'affichage de la liste des réunions, comme indiqué dans la **figure III.12**.

Diagramme de séquence

Made with
Visual Paradigm
For non-commercial use

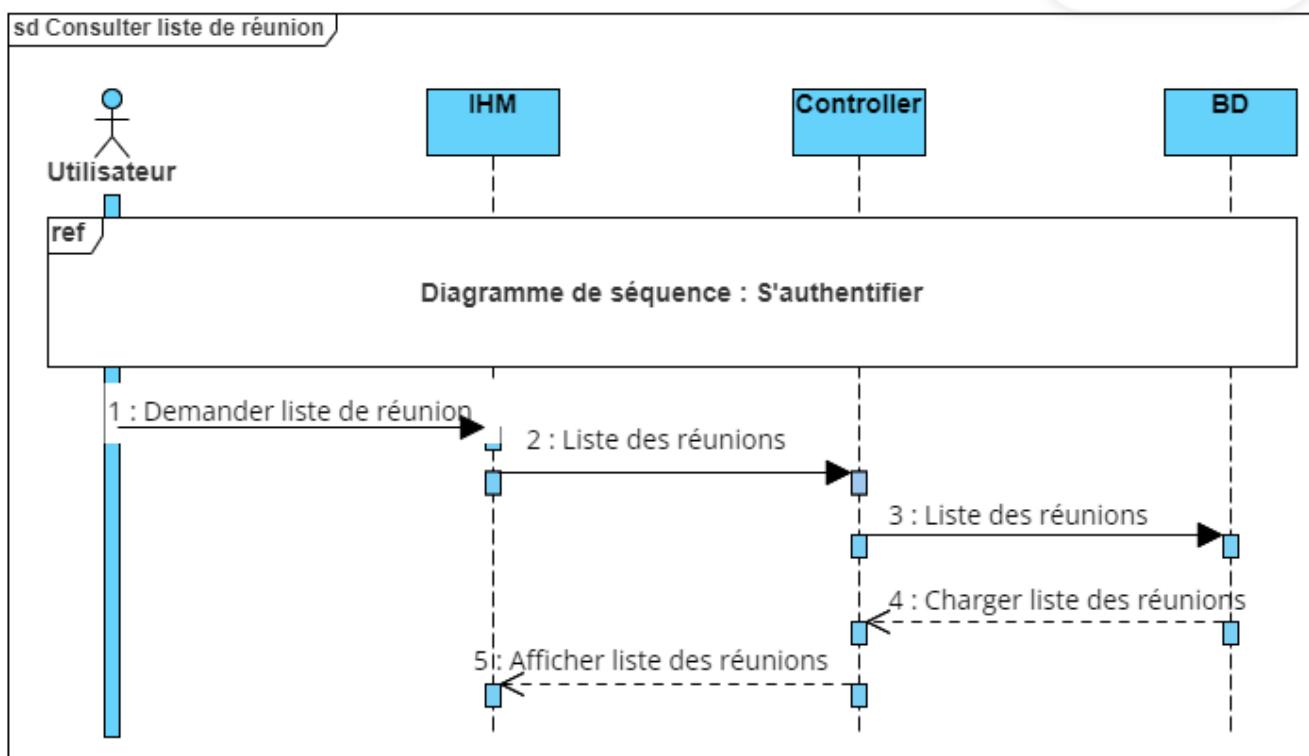


Figure III.12 : Diagramme de séquence du scénario « Consulter liste de réunion»

III.3.0.4 Diagramme de séquence supprimer réunion

Dans la **figure III.13**, nous avons exploré en détail un diagramme de séquence qui représente les interactions entre les différentes composantes de notre système. L'accent a été mis spécifiquement sur la gestion de la suppression d'une réunion. Ce diagramme offre une vue claire de la séquence précise d'actions et de messages échangés entre les différents objets impliqués lors de cette opération, permettant ainsi une meilleure compréhension du processus de suppression des réunions dans notre système.

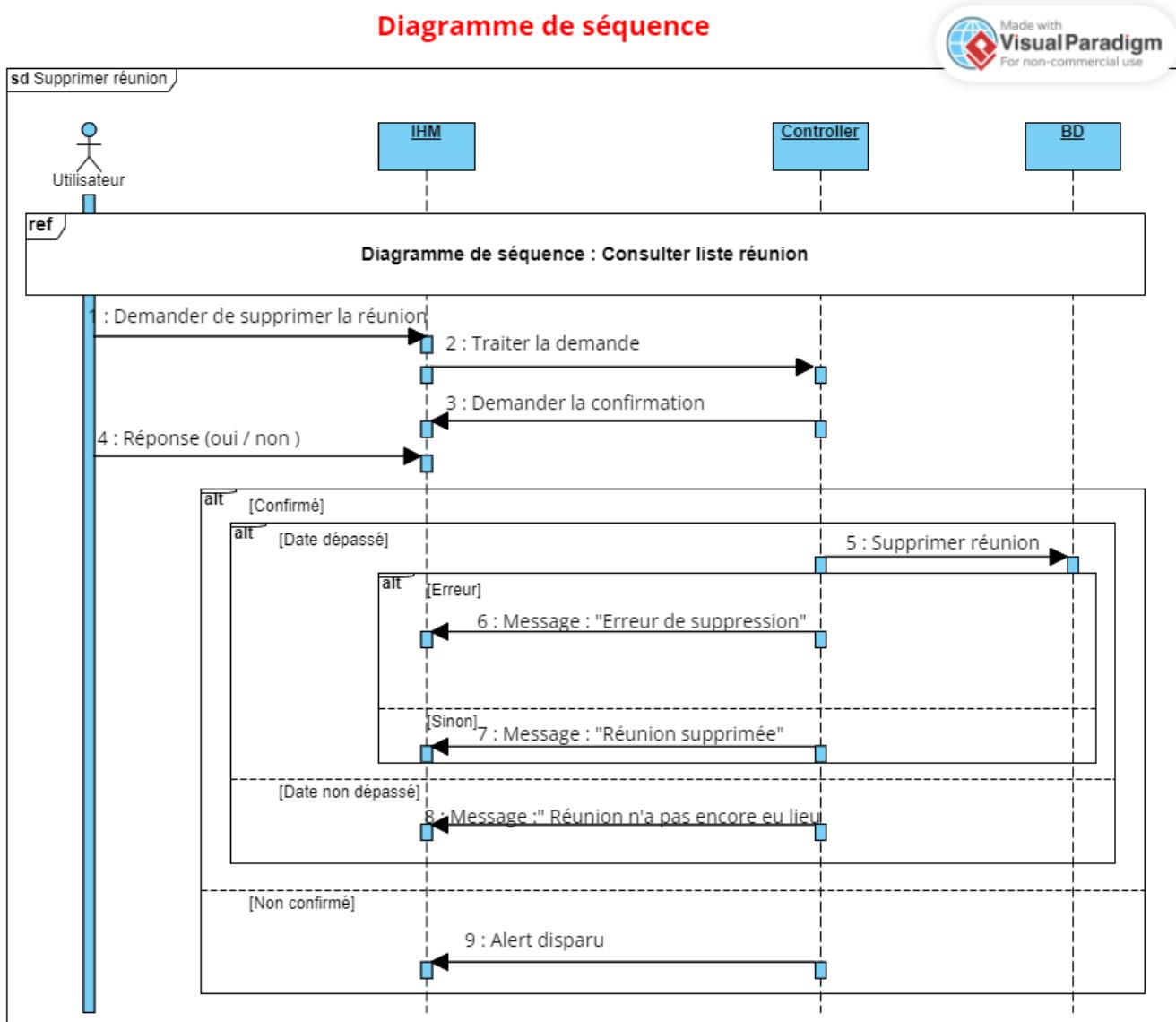


Figure III.13 : Diagramme de séquence du scénario « Supprimer réunion»

III.3.0.5 Diagramme de séquence modifiée profil

Dans la **figure III.14**, nous avons minutieusement détaillé le diagramme de séquence concernant le cas d'utilisation "Modifier le profil". Après que l'utilisateur (qu'il soit avocat ou comptable) se soit authentifié, il déclenche l'action en appuyant sur le bouton "Update". En réponse, une carte de formulaire s'affiche, permettant à l'utilisateur de saisir les informations à modifier. Une fois les modifications effectuées, l'utilisateur confirme en cliquant sur le bouton "Save".

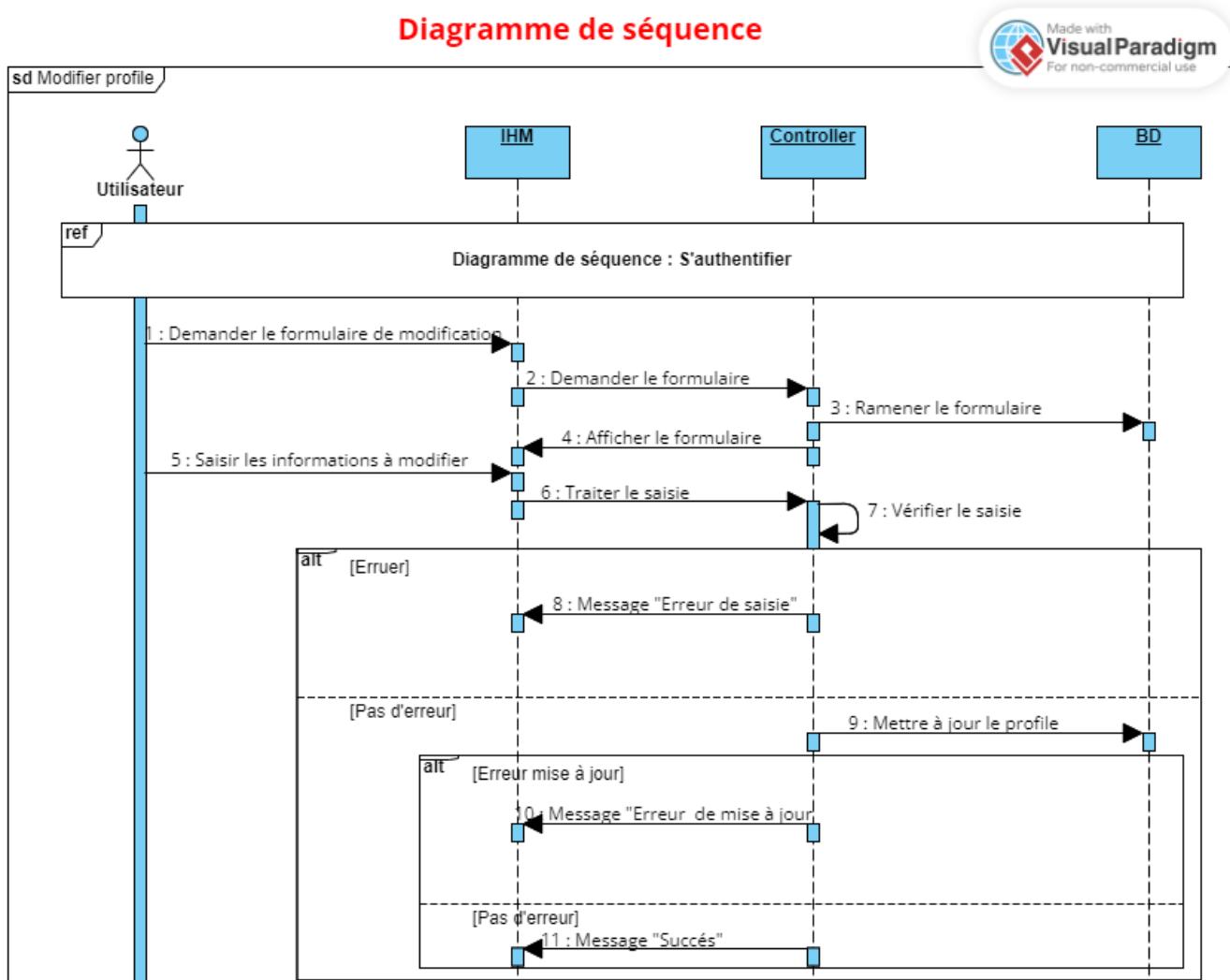


Figure III.14 : Diagramme de séquence du scénario « Modifier le profile»

III.3.0.6 Diagramme de séquence de chat entre les experts

Ce diagramme de séquence de la **figure III.15** détaille le cas d'utilisation « Envoyer un message » entre deux experts, montrant l’interaction entre les utilisateurs et les composants du système. L’utilisateur 1 demande à afficher la page de discussion, à sélectionner des amis et à envoyer un message. Le contrôleur vérifie le message et le stocke dans la base de données avant de le transmettre à l’utilisateur 2. En cas de panne, un message d’erreur s’affiche pour l’utilisateur 1. Ce processus garantit une communication fluide tout en gérant efficacement les erreurs.

Diagramme de Séquence

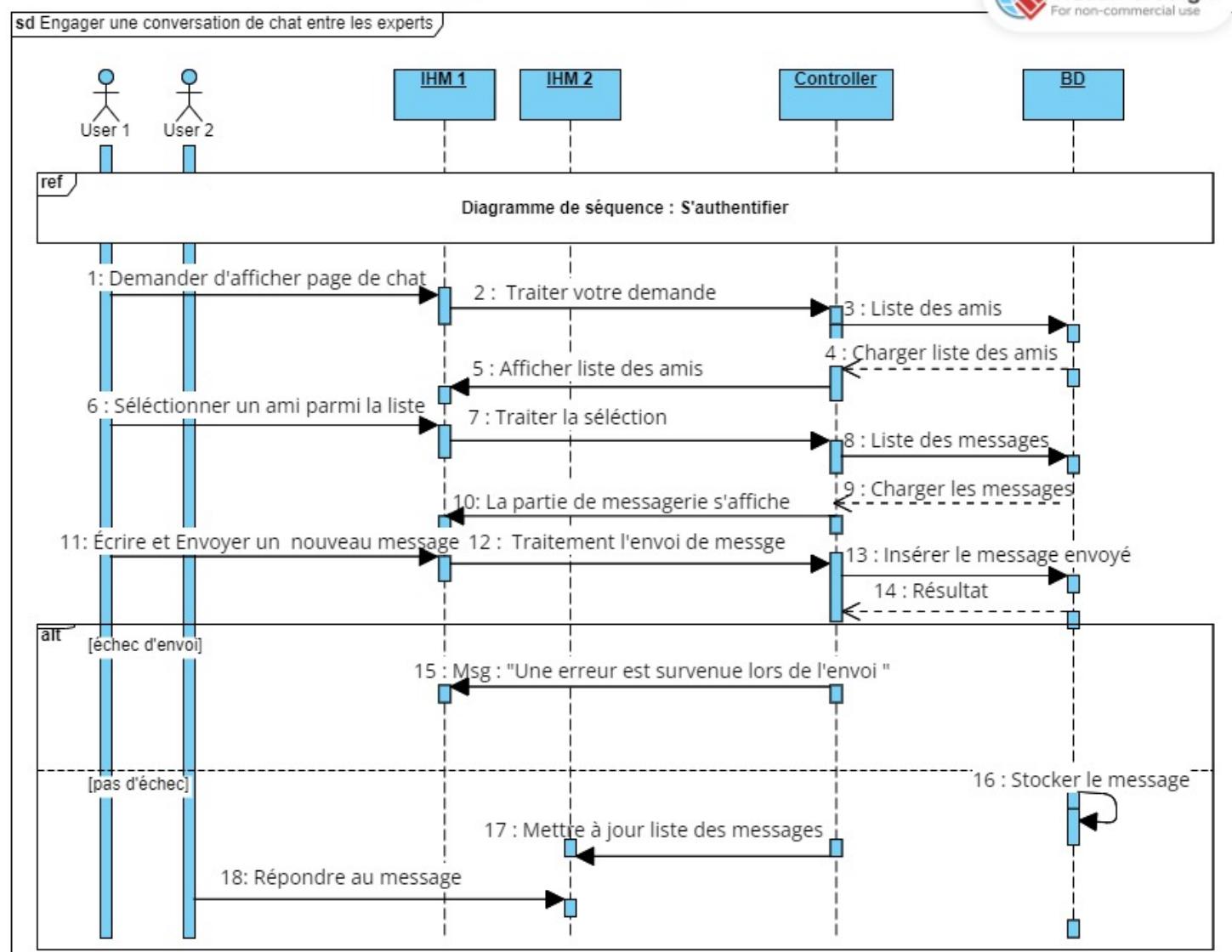


Figure III.15 : Diagramme de séquence du scénario «Envoyer message»

III.4 Conclusion

Dans ce chapitre, nous avons utilisé la modélisation UML pour présenter la conception de notre application de manière efficace, incluant le diagramme de classe et des diagrammes de séquence. Cette phase de conception détaillée facilitera sa réalisation et sa maintenance, et nous vous donnerons un aperçu de la mise en œuvre de l'environnement matériel et logiciel dans la prochaine étape.

Chapitre IV

Étude de faisabilité et architecture

Sommaire

IV.1	Introduction	55
IV.2	Architecture logique	55
IV.2.1	Communication avec HTTP et REST	55
IV.2.2	Architecture de l'application E-Tafakna	56
IV.2.2.1	Architecture 3-tiers	57
IV.2.2.2	Le design pattern MVVM (Modèle-Vue-VueModèle)	57
IV.3	Architecture physique	58
IV.4	Environnement de travail	59
IV.4.1	Environnement matériel	60
IV.4.2	Environnement logiciel	60
IV.4.2.1	Outil Visual Studio	61
IV.4.2.2	Outil Visual Paradigme	61
IV.4.2.3	Thunder Client	62
IV.5	Choix technologiques	62
IV.5.1	HTML	62
IV.5.2	CSS	63
IV.5.3	JavaScript	63
IV.5.4	Bootstrap	63
IV.5.5	ReactJS	64
IV.5.6	Vitest	64
IV.5.7	NodeJS	65
IV.5.8	OpenAI	65
IV.5.9	Express.js	66
IV.5.10	MySQL Workbench	66
IV.6	Conclusion	67

IV.1 Introduction

Après avoir établi les besoins et les objectifs dans la section précédente, nous nous concentrerons désormais sur l'aspect architectural de l'application. L'objectif à ce stade est de concevoir des schémas généraux permettant une représentation précise et sans ambiguïté du fonctionnement désiré de l'application. Ce chapitre présente deux perspectives conceptuelles : la première donne un aperçu global de l'architecture, tandis que la seconde décrit en détail les composants architecturaux à l'aide du langage UML (Unified Modeling Language).

IV.2 Architecture logique

IV.2.1 Communication avec HTTP et REST

Dans cette conception architecturale, on s'appuie sur une interaction entre un client et un serveur, comme il est marqué dans la **figure IV.1**. Le client émet des requêtes pour obtenir ou ajuster des ressources, ces dernières pouvant être structurées en JSON ou XML. En retour, le serveur utilise le protocole HTTP pour répondre à ces requêtes, en utilisant l'un des quatre verbes HTTP disponibles pour manipuler les ressources du système REST :

- Avec GET : on peut obtenir une ressource (en utilisant son identifiant) ou un groupe de ressources.
- En utilisant POST : on peut créer une nouvelle ressource.
- En employant PUT/PATCH : on peut mettre à jour une ressource spécifique (en utilisant son identifiant).
- En utilisant DELETE : on peut supprimer une ressource spécifique (en utilisant son identifiant).

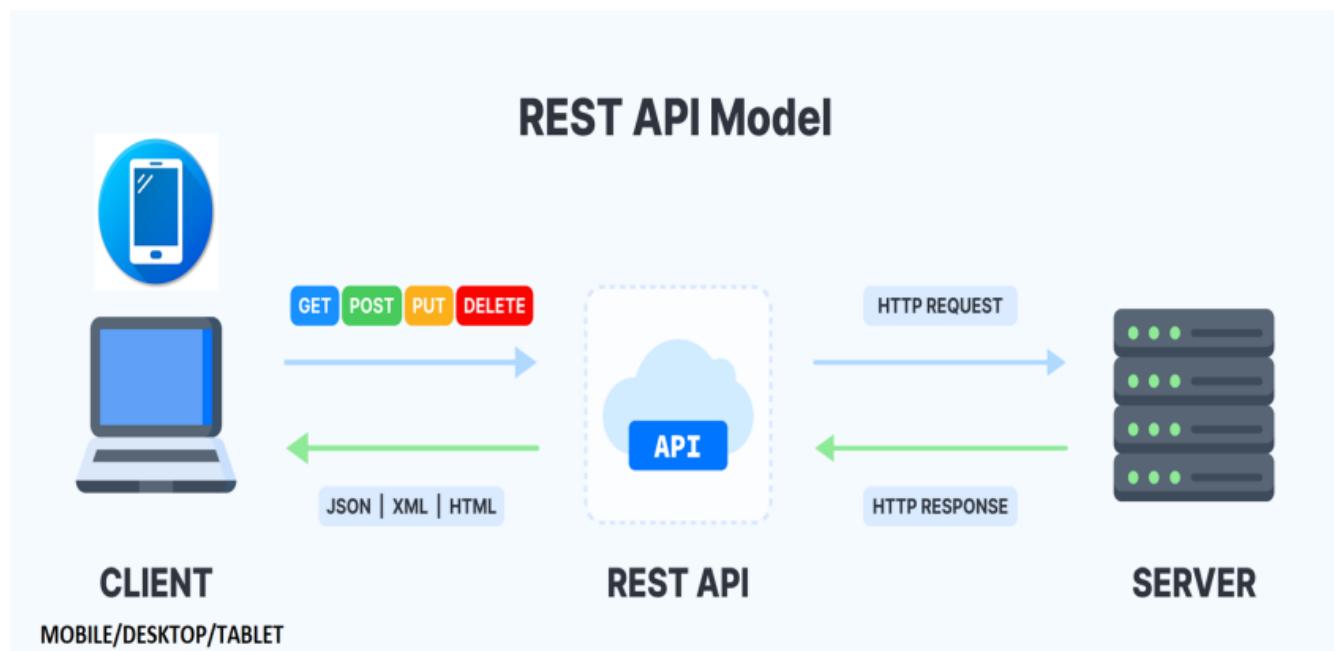


Figure IV.1 : REST API[6]

IV.2.2 Architecture de l'application E-Tafakna

Dans la section suivante, nous donnons un aperçu de l'architecture logicielle choisie pour notre application, ainsi que du schéma architectural choisi pour organiser notre implémentation, comme le montre la **figure IV.2**.

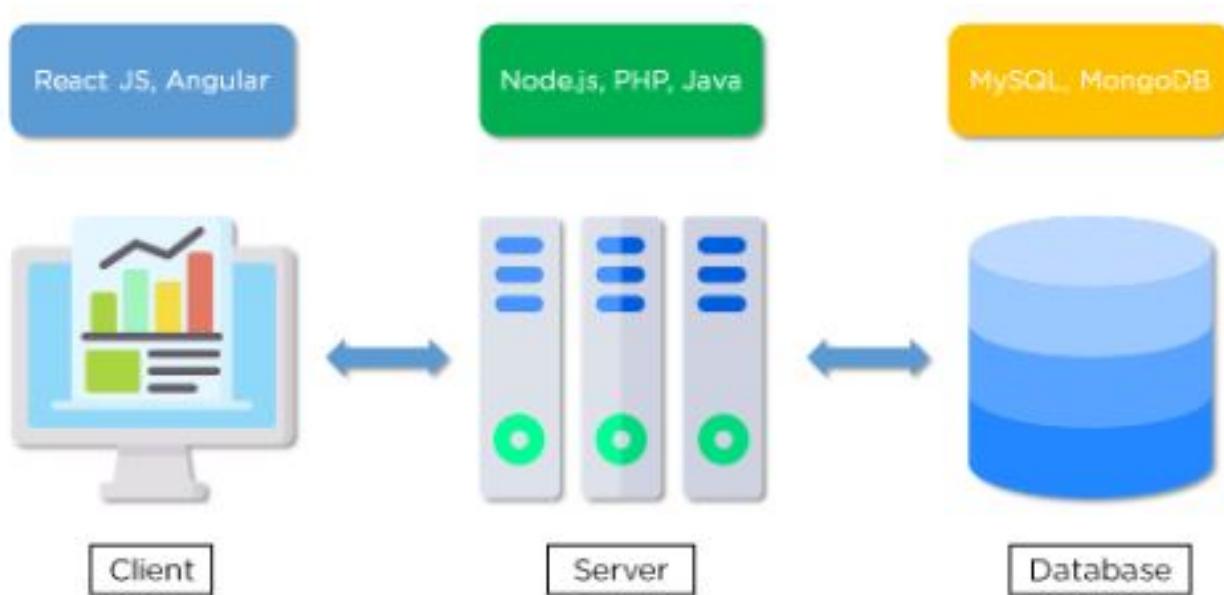


Figure IV.2 : Architecture 3-tiers[7]

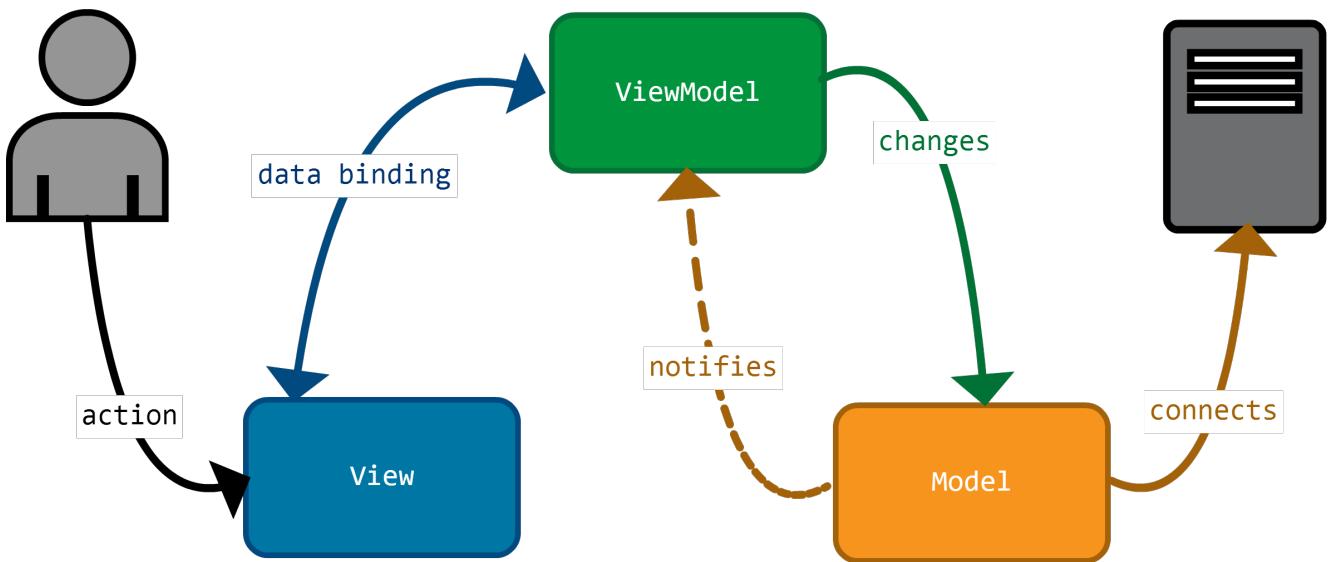
IV.2.2.1 Architecture 3-tiers

Une application Web est un logiciel accessible via un navigateur internet, hébergé sur un serveur et exécuté par ce dernier. Pour fonctionner, elle utilise le réseau Internet afin de récupérer les ressources nécessaires à son affichage et à son interaction avec l'utilisateur. En trois parties, on peut la décomposer ainsi :

- **Client** : L'utilisateur interagit avec l'interface d'une application web, généralement construite avec des langages comme HTML et CSS pour la mise en page, et des frameworks JavaScript populaires comme ReactJS pour faciliter la conception et le développement des applications.
- **Serveur** : Le serveur a pour mission de gérer les requêtes des clients, de réaliser les tâches requises et de fournir les réponses adéquates. Il agit comme un intermédiaire entre l'interface utilisateur et les données stockées, permettant ainsi aux utilisateurs de modifier ces données. Node.js est l'une des technologies les plus fréquemment utilisées pour développer et gérer les serveurs web.
- **Base de données** : La base de données conserve les informations d'une application web, permettant leur ajout, modification ou suppression en fonction des requêtes de l'utilisateur. MySQL figure parmi les systèmes de gestion de bases de données les plus couramment utilisés pour le stockage des données des applications web.

IV.2.2.2 Le design pattern MVVM (Modèle-Vue-VueModèle)

Le modèle MVVM de la **figure IV.3** (Modèle-Vue-VueModèle) est une approche architecturale qui sépare la logique de l'interface utilisateur de la logique métier. Son objectif est de maintenir une interface utilisateur épurée et dépourvue de code d'application, facilitant ainsi sa maintenance et sa mise à jour.

**Figure IV.3 :** Modèle MVVM[8]

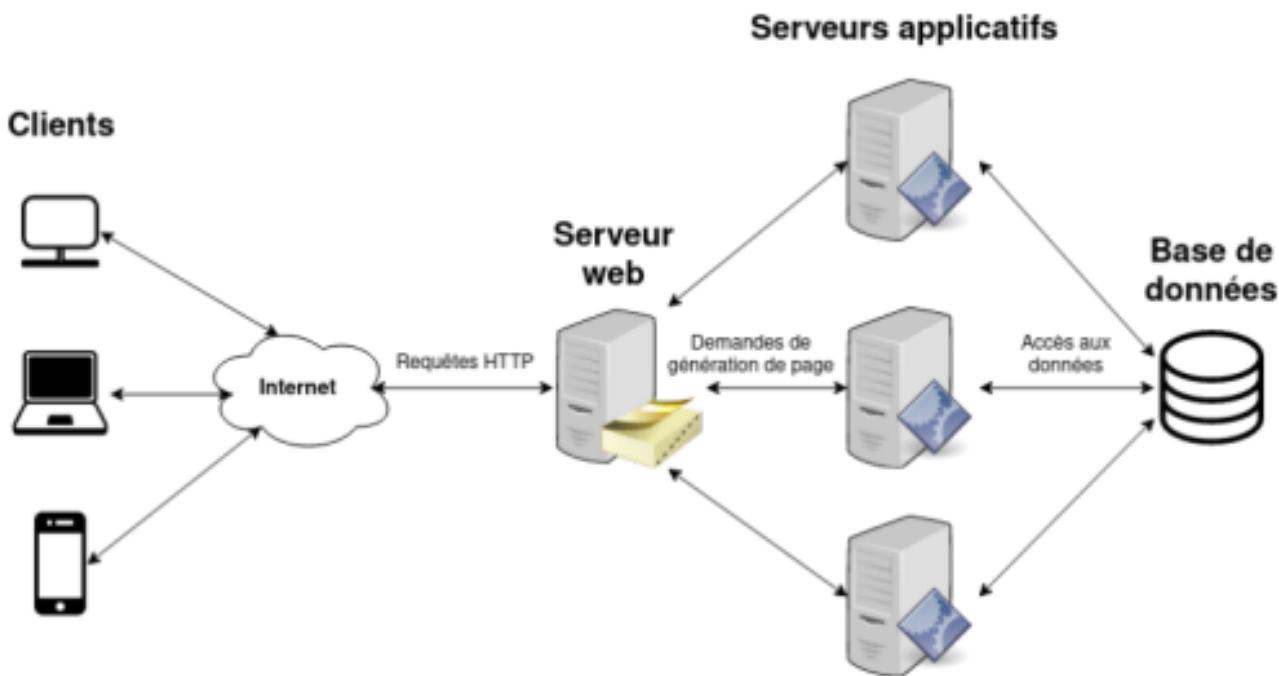
L’architecture MVVM repose sur trois couches distinctes :

- **Modèle (Model)** : Encapsule la logique métier et les données de l’application. Il est responsable de la manipulation et de la gestion des données.
- **Vue-Modèle (ViewModel)** : Agit comme intermédiaire entre le Modèle et la Vue. Il prépare les données observables pour la Vue et interagit avec le Modèle en invoquant ses méthodes.
- **Vue (View)** : Affiche les données à l’utilisateur et met à jour l’interface graphique en se basant sur les données fournies par le Vue-Modèle.

Cette séparation permet de maintenir une interface utilisateur épurée et dépourvue de code applicatif, facilitant ainsi la maintenance et la testabilité du code.

IV.3 Architecture physique

Dans le but de rendre la maintenance et l’évolution de notre application plus aisées, nous avons décidé d’adopter une architecture à trois niveaux. Cette configuration se divise entre les postes clients, les serveurs d’application et les serveurs de bases de données, comme représenté dans le **schéma IV.4**.

**Figure IV.4 : Architecture 3-tiers**

Le serveur prend en charge le traitement des requêtes HTTP reçues sur le port 3000, où est hébergée la partie front-end en mode production. Quant au serveur de gestion de base de données MySQL, il assure la gestion des flux de données entre la base de données et le serveur d'application. Les requêtes HTTP émises par le client depuis un navigateur web sont dirigées vers le serveur d'application pour y être traitées. Ce dernier réceptionne ces requêtes via sa couche web, puis exécute les traitements nécessaires. Parallèlement, le serveur de base de données gère la circulation des données entre la base de données et le serveur d'application.

IV.4 Environnement de travail

Dans le prochain paragraphe, nous examinerons les divers frameworks et technologies employés pour concrétiser l'architecture évoquée antérieurement dans notre environnement logiciel.

IV.4.1 Environnement matériel

Device specifications	
Device name	DESKTOP-03HAM4D
Processor	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
Installed RAM	12.0 GB (11.8 GB usable)
Device ID	50BE0228-A7F3-46D5-928B-2445583BB738
Product ID	00327-30833-98045-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

[Copy](#)

[Rename this PC](#)

Windows specifications

Edition	Windows 10 Home Single Language
Version	22H2
Installed on	23/10/2023
OS build	19045.4291
Experience	Windows Feature Experience Pack 1000.19056.1000.0

[Copy](#)

Figure IV.5 : Environnement de travail

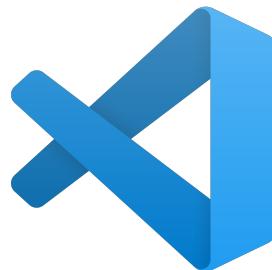
La **figure IV.5** présente ,notre environnement matériel est constitué par un ordinateur portable dont les caractéristiques sont présentées ci-dessous :

- **Modèle : Hp**
- **Système d'exploitation : Windows 10**
- **Processeur : Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz**
- **RAM : 12.0 GB (11.8 GB usable)**
- **Disque dur : 225Go SSD**

IV.4.2 Environnement logiciel

Voici une description de notre environnement logiciel, mettant en lumière les divers logiciels employés durant le processus de développement de notre application.

IV.4.2.1 Outil Visual Studio



VS Code, un éditeur de code réputé développé par Microsoft, offre gratuitement des fonctionnalités avancées pour les programmeurs. Compatible avec divers langages de programmation, il s'impose comme l'outil idéal pour le développement d'applications de tous types, des sites web aux applications mobiles et de bureau. La popularité croissante de VS Code témoigne de son efficacité pour simplifier le processus de développement [9].

IV.4.2.2 Outil Visual Paradigme



Visual Paradigm, un logiciel de modélisation visuelle de premier plan, permet aux professionnels de l'informatique et de l'ingénierie de conceptualiser, documenter et communiquer des idées complexes à l'aide de diagrammes, de modèles et d'architectures. Son interface intuitive et ses fonctionnalités avancées facilitent la création et l'analyse de solutions logicielles et systémiques. Compatible avec diverses notations de modélisation (UML, BPMN, SysML), Visual Paradigm permet de générer du code, de collaborer efficacement et de donner vie aux concepts de manière visuelle. C'est un outil polyvalent qui aide les professionnels à concrétiser leurs idées d'une manière claire et efficace [10].

IV.4.2.3 Thunder Client



Thunder Client, une extension d'API REST pour Visual Studio Code, est conçue pour être légère et facile à utiliser. Développée par Ranga Vadhineni, elle privilégie la simplicité, avec une interface épurée et un stockage local pour une utilisation pratique. Il permet de tester et de déboguer rapidement des API en envoyant des requêtes HTTP. Il affiche les réponses reçues et permet d'organiser les requêtes dans des collections pour une gestion aisée [11].

IV.5 Choix technologiques

Notre environnement technique repose sur une combinaison de frameworks et de technologies soigneusement sélectionnés pour concrétiser l'architecture décrite précédemment. Voici un aperçu détaillé :

IV.5.1 HTML



HTML signifie « HyperText Markup Language », qui peut être traduit par « Hypertext Markup Language ». Il est utilisé pour créer et représenter le contenu des pages Web et leur structure. D'autres technologies sont utilisées avec HTML pour décrire le rendu d'une page (CSS) et/ou ses fonctionnalités interactives (JavaScript) [12].

IV.5.2 CSS



Les Cascading Style Sheets (CSS) sont le langage incontournable pour façonnner l'apparence des pages web. À l'aide d'instructions concises organisées en feuilles de style claires, CSS permet de contrôler avec précision chaque élément de la mise en page, des couleurs aux polices. Crucialement, CSS sépare le style du contenu, préservant ainsi l'intégrité sémantique et structurelle des documents. Introduit au milieu des années 1990, CSS est devenu le langage standard pour styliser les pages web, garantissant une présentation cohérente et attrayante sur tous les navigateurs modernes [13].

IV.5.3 JavaScript



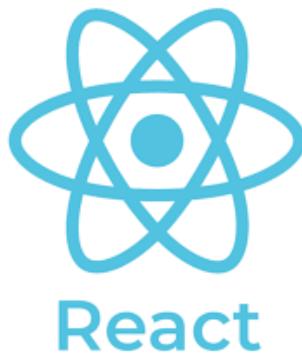
JavaScript est un langage de programmation qui vous permet d'implémenter des éléments complexes sur des pages Web (pas seulement des pages Web contenant des informations statiques). Chaque page affichera des mises à jour de contenu en temps réel, des cartes interactives, des graphiques animés 2D/3D ou des juke-box vidéo défilants, et bien plus encore [14].

IV.5.4 Bootstrap



Bootstrap, un framework front-end open-source réputé, est le choix privilégié pour créer des sites web et des applications web réactifs. Son éventail de composants pré-définis accélère la conception d'interfaces utilisateur attrayantes et efficaces. Sa simplicité d'utilisation, ses options de personnalisation et sa compatibilité multi-navigateurs et multi-résolutions en font un allié précieux pour les développeurs web. La popularité de Bootstrap témoigne de sa capacité à simplifier le développement web tout en offrant une flexibilité maximale [15].

IV.5.5 ReactJS



React peut être utilisé comme base dans le développement d'applications monopages ou mobiles. Cependant, React ne s'occupe que de la gestion de l'état et du rendu de cet état dans le DOM (Document Object Model), de sorte que la création d'applications React nécessite généralement l'utilisation de bibliothèques supplémentaires pour le routage, ainsi que certaines fonctionnalités côté client [16].

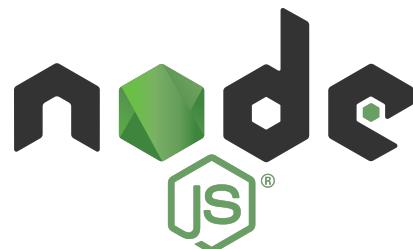
IV.5.6 Vitest



Vitest, un framework de test JavaScript populaire, ainsi que React Testing Library, ont été utilisés pour les tests unitaires et les tests d'intégration des composants frontend. Vitest offre une large gamme d'outils de test, y compris le mock et les assertions,

ce qui le rend adapté aux tests des composants React. React Testing Library, quant à elle, est spécialement conçue pour tester les composants React de manière centrée sur l'utilisateur, en simulant les interactions utilisateur et en vérifiant le comportement des composants.[18].

IV.5.7 NodeJS



Node.js, construit autour du moteur JavaScript V8 de Google Chrome, permet de développer des applications côté serveur à l'aide de JavaScript. Son module "http" intégré fournit des fonctionnalités de serveur HTTP, éliminant le besoin de serveurs web tiers comme Nginx ou Apache. Dans notre projet, Node.js sert d'environnement d'exécution pour le code JavaScript côté serveur, offrant une intégration transparente entre le front-end et le back-end. Cette approche simplifie le développement et garantit la cohérence du langage tout au long de la pile technologique [17].

IV.5.8 OpenAI



La bibliothèque OpenAI Node vous permet d'accéder facilement à l'API REST OpenAI à partir d'applications écrites en JavaScript côté serveur. Il contient des définitions TypeScript pour tous les paramètres de requête et champs de réponse[20].

IV.5.9 Express.js



Express.js, un framework web minimaliste pour Node.js, simplifie le développement d'applications web. Il offre une structure solide pour gérer les requêtes et réponses HTTP, ainsi que les routes. Sa syntaxe concise et ses fonctionnalités intégrées permettent aux développeurs de créer rapidement des applications web performantes et évolutives, tirant parti de la puissance de Node.js. En choisissant Express.js, les développeurs peuvent se concentrer sur la logique métier de leurs applications, laissant le framework gérer les aspects techniques du développement web [19].

IV.5.10 MySQL Workbench



MySQL Workbench, autrefois connu sous le nom de MySQL Administrator, est un outil puissant pour gérer et administrer les bases de données MySQL. Lancé en 2004, il offre une interface graphique intuitive qui permet aux utilisateurs de créer, modifier et supprimer des tables et des comptes d'utilisateurs en toute simplicité. MySQL Workbench centralise également les tâches essentielles de gestion de base de données, telles que l'exécution de requêtes SQL, la sauvegarde et la restauration. Nécessitant une connexion à un serveur MySQL, cet outil offre un contrôle complet sur les bases de données, permettant aux administrateurs et aux développeurs de garantir leur intégrité et leurs performances optimales [21].

IV.6 Conclusion

En guise de synthèse, ce chapitre a posé les fondations de l'architecture de l'application, clarifiant les choix technologiques et décrivant les différentes strates et composants qui la composent. Ce cadre servira de guide pour la mise en œuvre et le développement futurs de l'application, garantissant une approche structurée et cohérente.

Chapitre V

Interface et Architecture

Sommaire

V.1	Introduction	69
V.2	Interface d'authentification	69
V.3	Interface d'accueil	70
V.4	Interface de la carte géographique	73
V.5	Interface gérer clients	73
V.6	Interface de réunion	74
V.7	Interface de calendrier	76
V.8	Interface de profile	77
V.9	Interface de fichier	78
V.10	Interface de note	82
V.11	Interface de sélection le type de chat	83
V.12	Interface de chatbot	84
V.13	Interface de chatroom	84
V.14	Conclusion	86

V.1 Introduction

D'après les exigences exprimées dans la section des besoins, il est crucial que les interfaces utilisateur soient conviviales, simples et intuitives. Cette partie du projet se concentre sur sa réalisation pratique en présentant une série de captures d'écran illustrant les différentes interfaces graphiques. Ces visuels permettent de mieux appréhender l'aspect et les fonctionnalités des interfaces de l'application. Cette approche visuelle vise à faciliter la compréhension de l'expérience utilisateur et à évaluer si les interfaces répondent aux normes de convivialité définies.

V.2 Interface d'authentification



Figure V.1 : Capture d'écran de l'interface login

Tous les utilisateurs, qu'ils soient avocats ou comptables, doivent s'authentifier avant d'accéder aux fonctionnalités. L'interface de connexion, illustrée à la **figure V.1**, permet à l'utilisateur de saisir son adresse email et son mot de passe.

V.3 Interface d'accueil

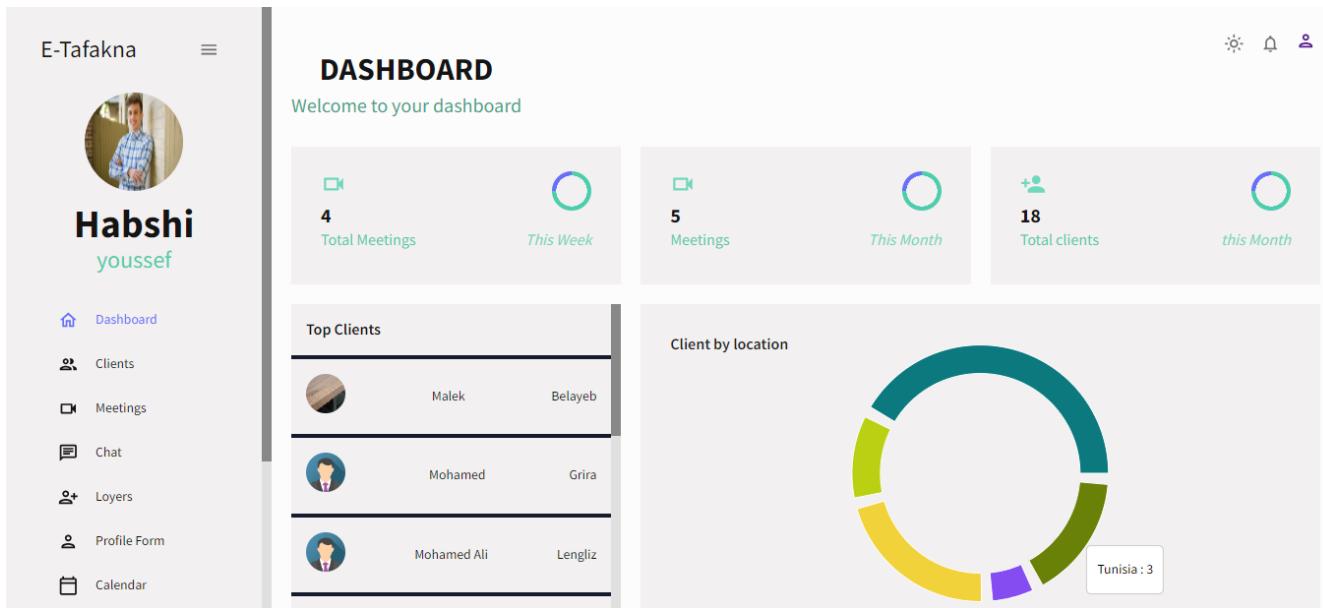


Figure V.2 : Capture d'écran de l'interface d'accueil

La figure V.2 présente la page d'accueil de l'application, qui affiche diverses informations telles que le nombre de réunions effectuées au cours de la semaine et le total des réunions du mois. Elle indique également le nombre total de clients. Un graphique en secteurs (pie chart) permet de visualiser la répartition des clients par pays. De plus, les meilleurs clients, ceux qui participent le plus régulièrement aux réunions, sont mis en avant. Des liens cliquables permettent aux utilisateurs d'accéder à leurs statistiques sous forme de graphiques, montrant le nombre de clients par jour ou par mois, ainsi que le total des clients représenté par un diagramme en barres.



Figure V.3 : Capture d'écran de l'interface graphique en secteurs

L'interface de la **figure V.3** présente un diagramme circulaire illustrant la répartition des clients selon leur pays d'origine.

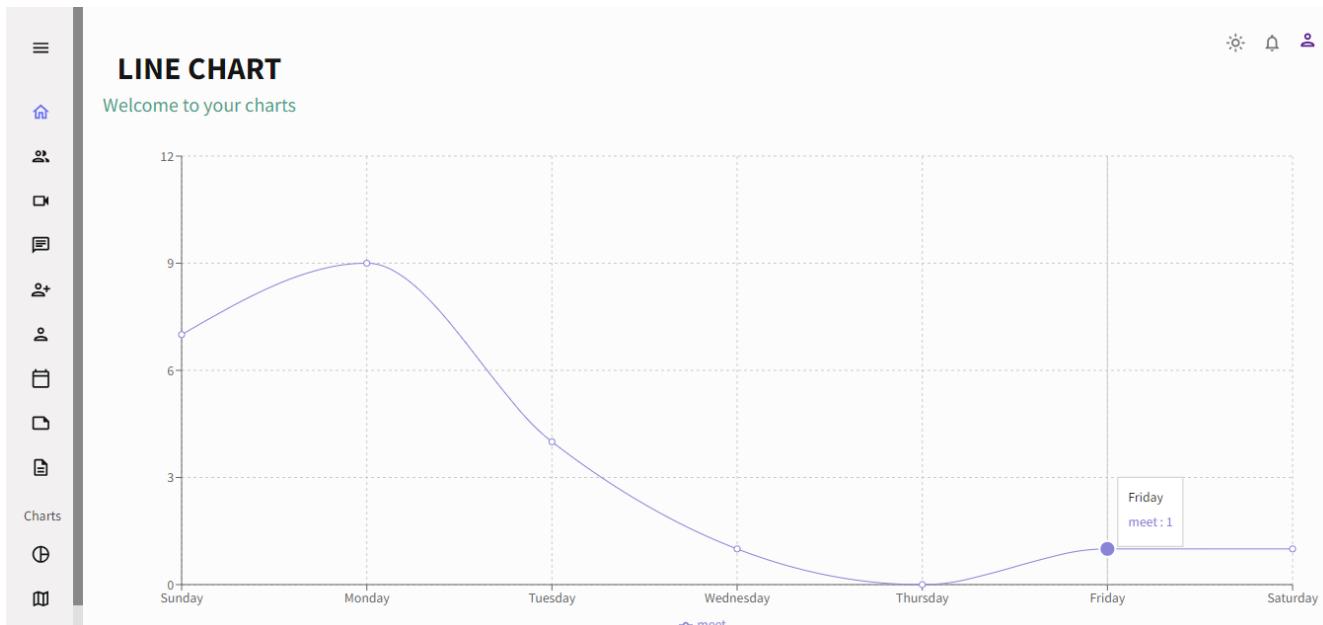


Figure V.4 : Capture d'écran de l'interface graphique 0

La **figure V.4** présente un graphique qui met en évidence le nombre de réunions planifiées chaque semaine. Ce graphique permet de suivre l'activité et de visualiser l'évolution de la charge de travail en matière de réunions au fil du temps.

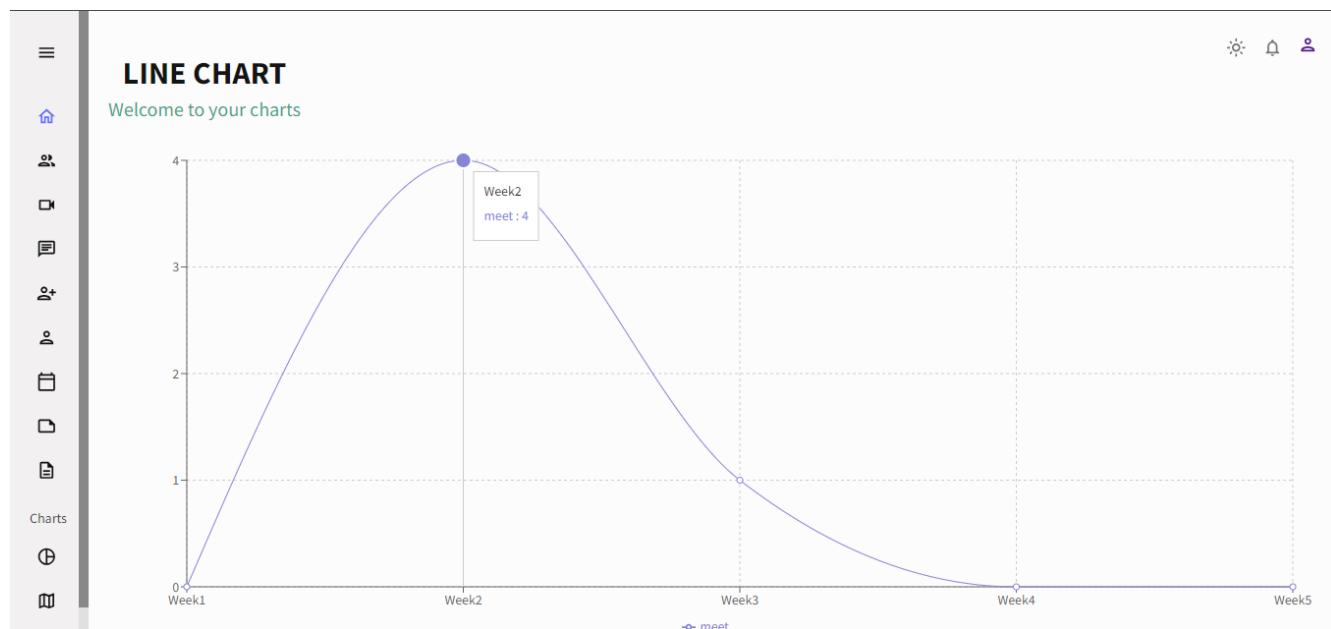


Figure V.5 : Capture d'écran de l'interface graphique 1

La **Figure V.5** présente une courbe illustrant l'évolution du nombre de réunions organisées chaque mois sur une période donnée. Cette courbe permet d'observer les tendances et les variations du nombre de réunions au fil des mois.

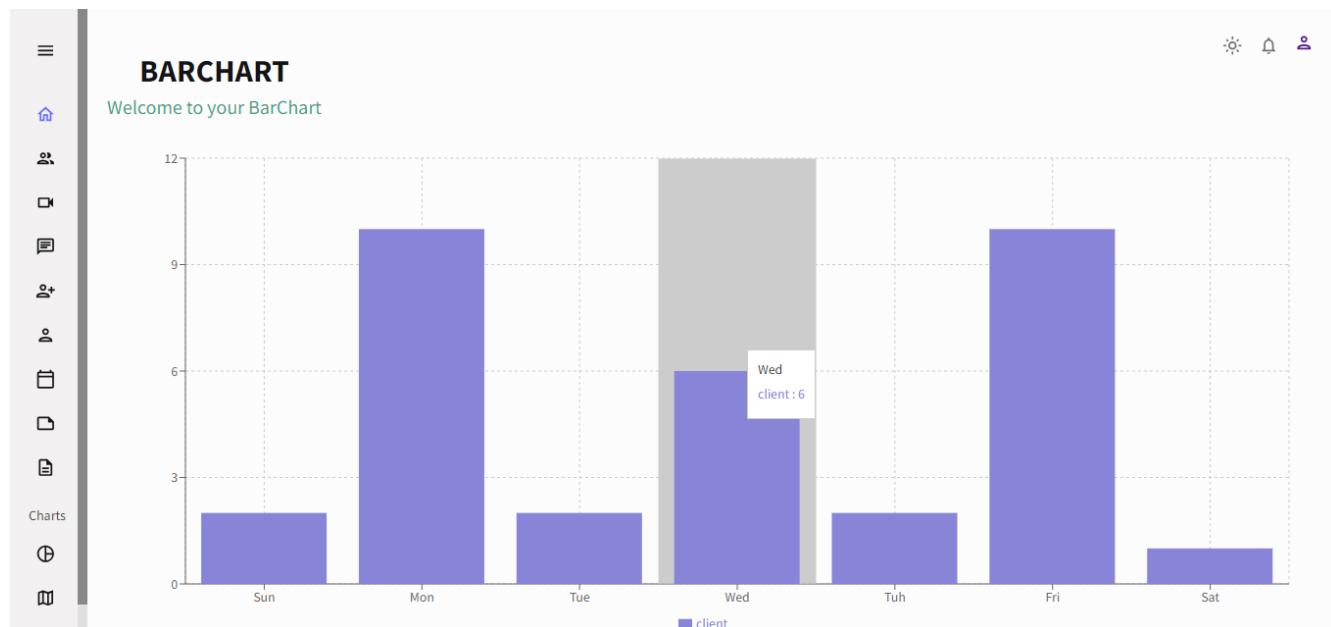


Figure V.6 : Capture d'écran de l'interface graphique 2

La **Figure V.6** présente un diagramme en barres illustrant la progression du nombre total de clients au fil du temps. Ce graphique permet de visualiser l'augmentation du nombre de clients et de suivre leur évolution.

V.4 Interface de la carte géographique

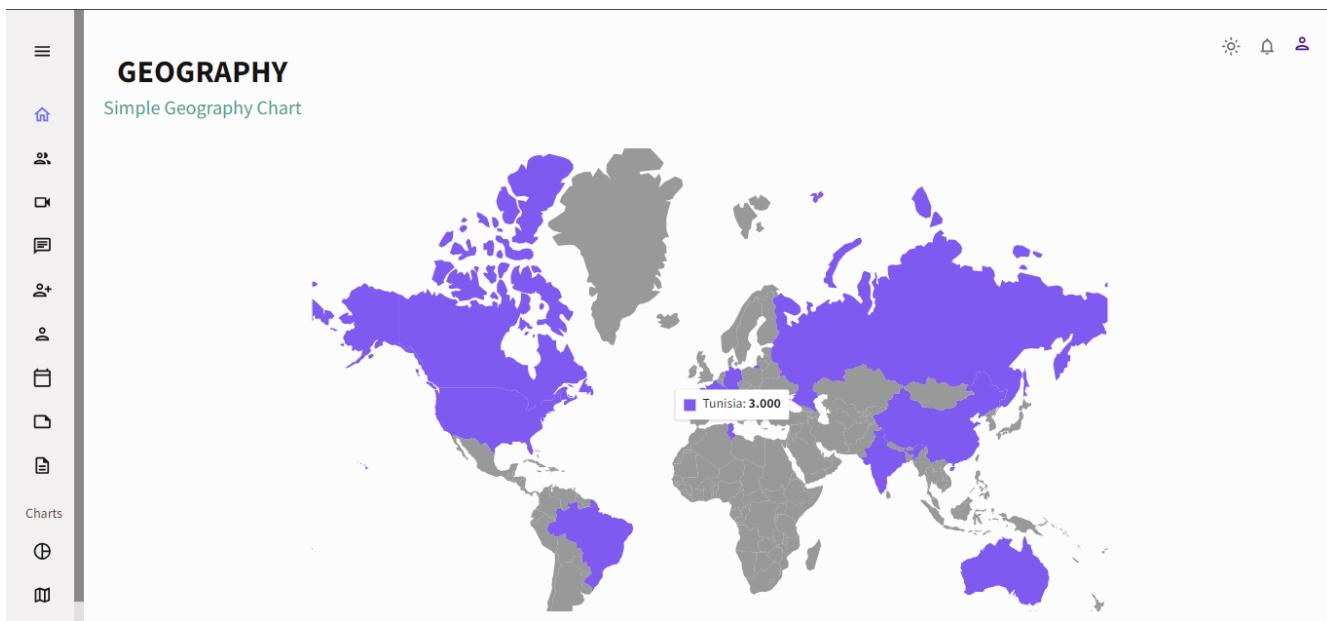


Figure V.7 : Capture d'écran de l'interface de la carte géographique

La **Figure V.7** présente une interface interactive avec une carte du monde permettant d'analyser la distribution des clients à l'échelle internationale. En naviguant sur cette carte, vous pouvez identifier les régions où la présence de clients est la plus importante ainsi que celles où elle est moins significative.

V.5 Interface gérer clients

La **figure V.8** est dédiée à la présentation de l'interface permettant aux avocats et comptables de consulter leur liste de clients. Cette interface offre également la possibilité de supprimer un client. Si un utilisateur souhaite supprimer un client, il doit cliquer sur le bouton de suppression, ce qui déclenche l'affichage d'une alerte pour confirmer ou annuler l'action.

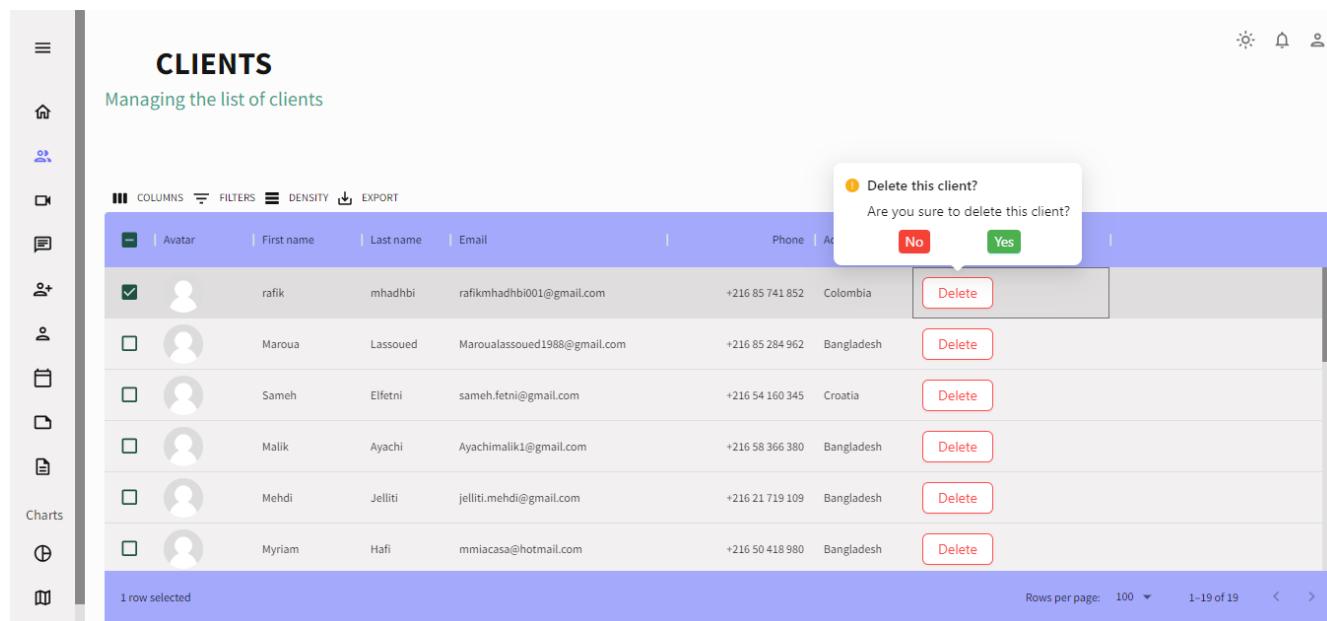


Figure V.8 : Capture d'écran de l'interface de gestion des clients

V.6 Interface de réunion

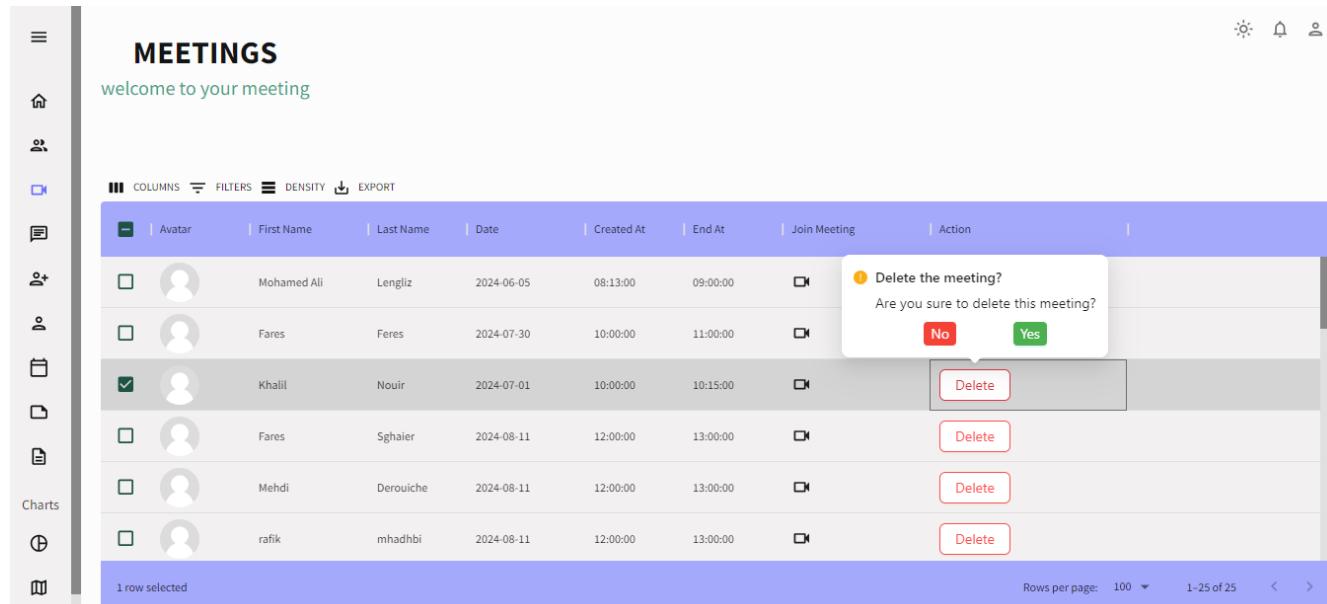


Figure V.9 : Capture d'écran de l'interface des réunions

La figure V.9 présente l'interface permettant aux avocats et comptables de consulter leur liste de réunions. Un bouton "Join Meeting" est disponible pour rejoindre une réunion à l'heure prévue. En cliquant sur ce bouton, l'interface de Google Meet s'ouvre, permettant ainsi la participation à la réunion. De plus, les utilisateurs ont la possibi-

lité de supprimer une réunion si celle-ci est déjà passée. En cliquant sur le bouton de suppression, une alerte s'affiche pour confirmer ou annuler la suppression.

The screenshot shows a user interface titled "MEETINGS" with a sidebar containing icons for home, users, calendar, and charts. The main area displays a table of meetings with columns: Avatar, First Name, Last Name, Date, Created At, End At, Join Meeting, and Action. The "Action" column contains a "Delete" button for each row. A red notification box in the top right corner states: "Link can only be deleted after the meeting or date has passed". The table data is as follows:

Avatar	First Name	Last Name	Date	Created At	End At	Join Meeting	Action
	Mohamed Ali	Lengliz	2024-06-05	08:13:00	09:00:00		<button>Delete</button>
	Fares	Feres	2024-07-30	10:00:00	11:00:00		<button>Delete</button>
	Khalil	Nouir	2024-07-01	10:00:00	10:15:00		<button>Delete</button>
	Fares	Sghaier	2024-08-11	12:00:00	13:00:00		<button>Delete</button>
	Mehdi	Derouiche	2024-08-11	12:00:00	13:00:00		<button>Delete</button>
	rafik	mihadhibi	2024-08-11	12:00:00	13:00:00		<button>Delete</button>

Rows per page: 100 1-25 of 25

Figure V.10 : Capture d'écran de l'interface des réunions

The screenshot shows the same "MEETINGS" interface after a deletion. The red notification box now says: "Link deleted successfully". The table data is as follows:

Avatar	First Name	Last Name	Date	Created At	End At	Join Meeting	Action
	Fares	Feres	2024-07-30	10:00:00	11:00:00		<button>Delete</button>
	Khalil	Nouir	2024-07-01	10:00:00	10:15:00		<button>Delete</button>
	Fares	Sghaier	2024-08-11	12:00:00	13:00:00		<button>Delete</button>
	Mehdi	Derouiche	2024-08-11	12:00:00	13:00:00		<button>Delete</button>
	rafik	mihadhibi	2024-08-11	12:00:00	13:00:00		<button>Delete</button>
	Maroua	Lassoud	2024-08-11	12:00:00	13:00:00		<button>Delete</button>

Rows per page: 100 1-24 of 24

Figure V.11 : Capture d'écran de l'interface des réunions

Les **interfaces V.10 et V.11** montrent que lorsque l'utilisateur clique sur le bouton "Delete", une notification apparaît pour informer l'utilisateur du statut de cette action. Cette notification précise si la suppression de la réunion a réussi ou si une erreur s'est produite, permettant à l'utilisateur de confirmer le résultat de sa demande.

V.7 Interface de calendrier

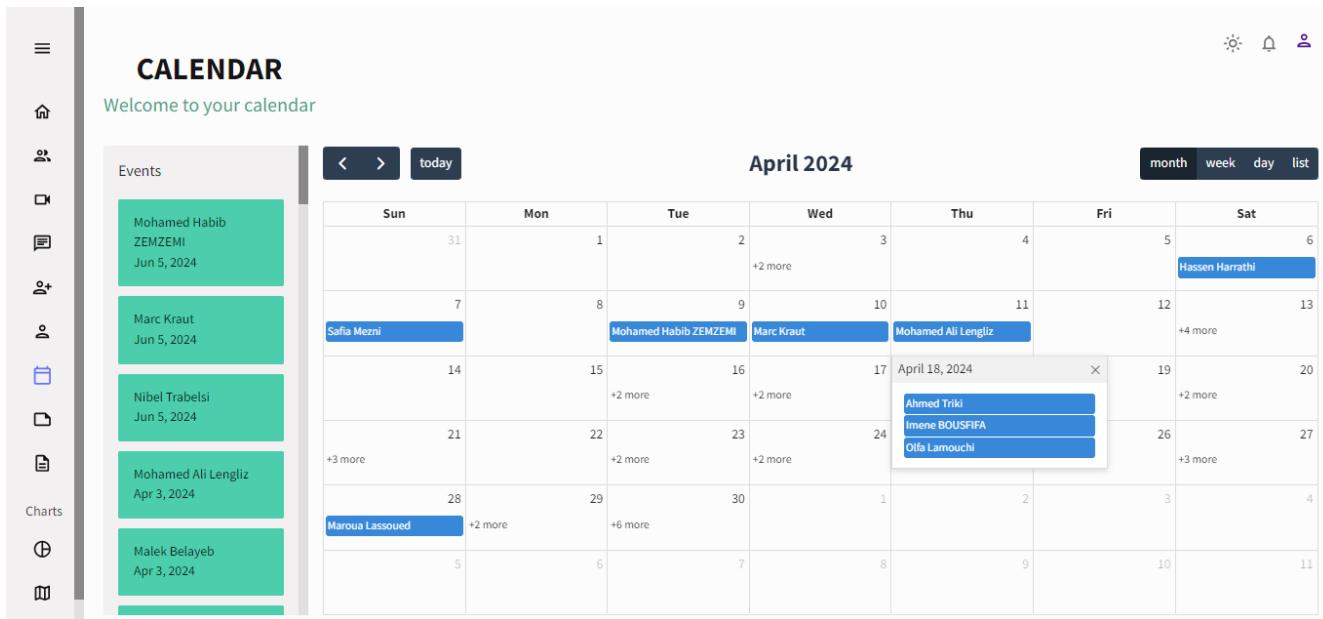


Figure V.12 : Capture d'écran de l'interface calendrier par mois

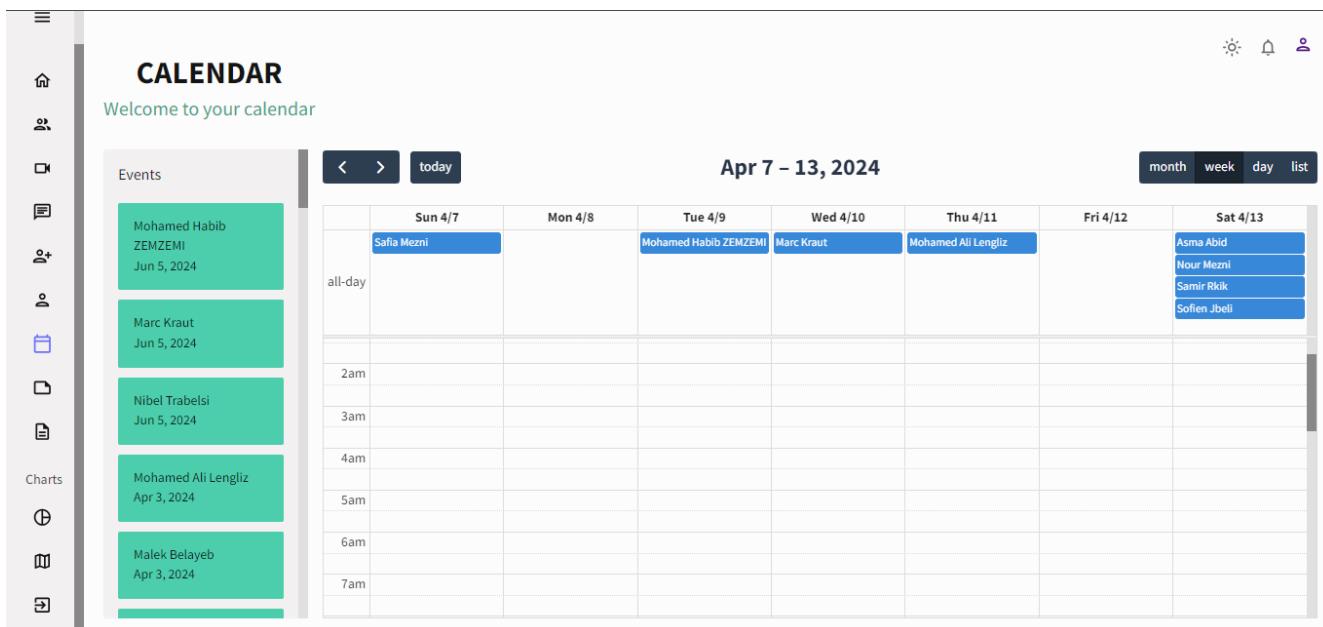


Figure V.13 : Capture d'écran de l'interface calendrier par semaine

Les **figures V.12 et V.13** présentent l'interface du calendrier personnel dans notre système, accessible à chaque utilisateur, qu'il soit avocat ou comptable. Cette fonctionnalité permet aux utilisateurs de visualiser et de gérer efficacement leur planning professionnel. En accédant au calendrier, ils peuvent voir tous les événements programmés,

y compris les réunions avec les clients, ce qui leur offre une vue d'ensemble de leurs activités professionnelles et leur permet de mieux planifier leurs journées. De plus, les dates des rendez-vous clients sont clairement affichées, facilitant ainsi la coordination des rendez-vous et des réunions d'affaires.

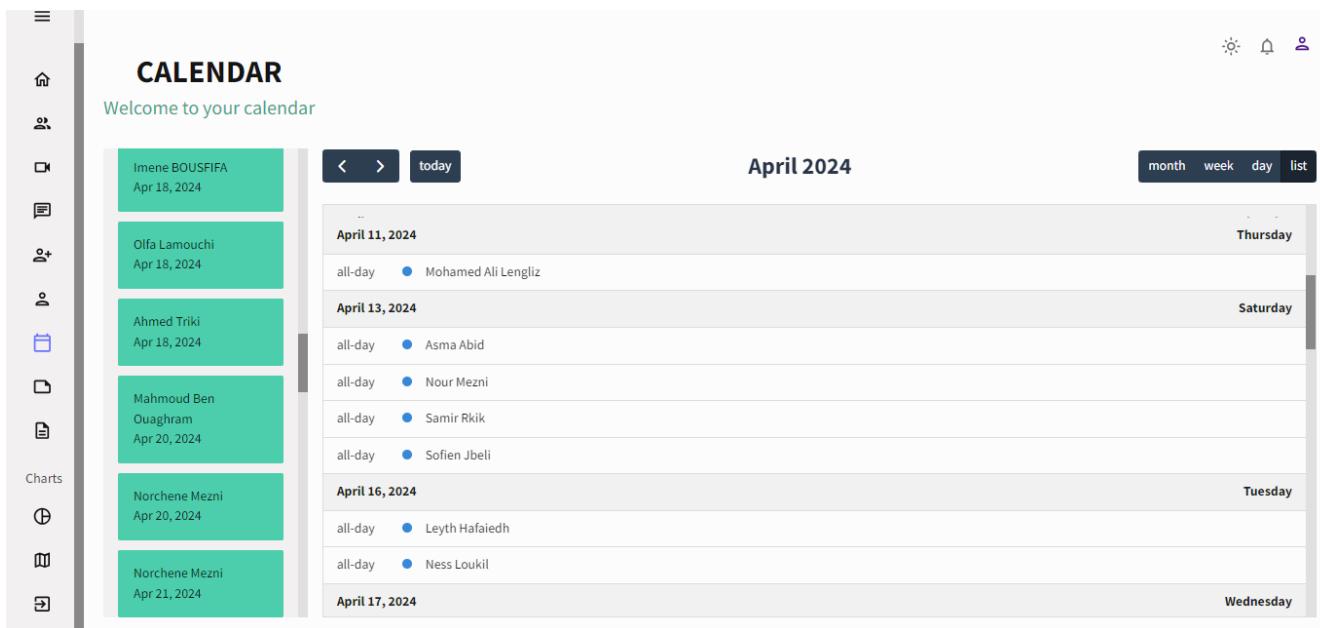


Figure V.14 : Capture d'écran de l'interface calendrier par mois

L'interface de la **figure V.14** présente de manière claire et structurée toutes les réunions prévues pour le mois d'avril. Chaque réunion est répertoriée avec des détails précis tels que la date et la liste des participants. Cela permet aux utilisateurs d'avoir une vue complète et organisée de leurs réunions planifiées pour ce mois spécifique.

V.8 Interface de profile

La **figure V.15** illustre l'espace de profil dédié où les utilisateurs peuvent consulter et modifier leurs informations personnelles. L'interface permet aux utilisateurs de visualiser des détails tels que leur nom, leur adresse, leur numéro de téléphone, et d'autres informations pertinentes. Lorsque l'utilisateur clique sur le bouton "Update" (Mettre à jour), une fenêtre modale s'affiche, offrant la possibilité de modifier les informations nécessaires, comme le nom, l'adresse, le numéro de téléphone, et d'autres champs essentiels. Cela permet aux utilisateurs de maintenir leurs informations personnelles à jour et pertinentes dans le système.

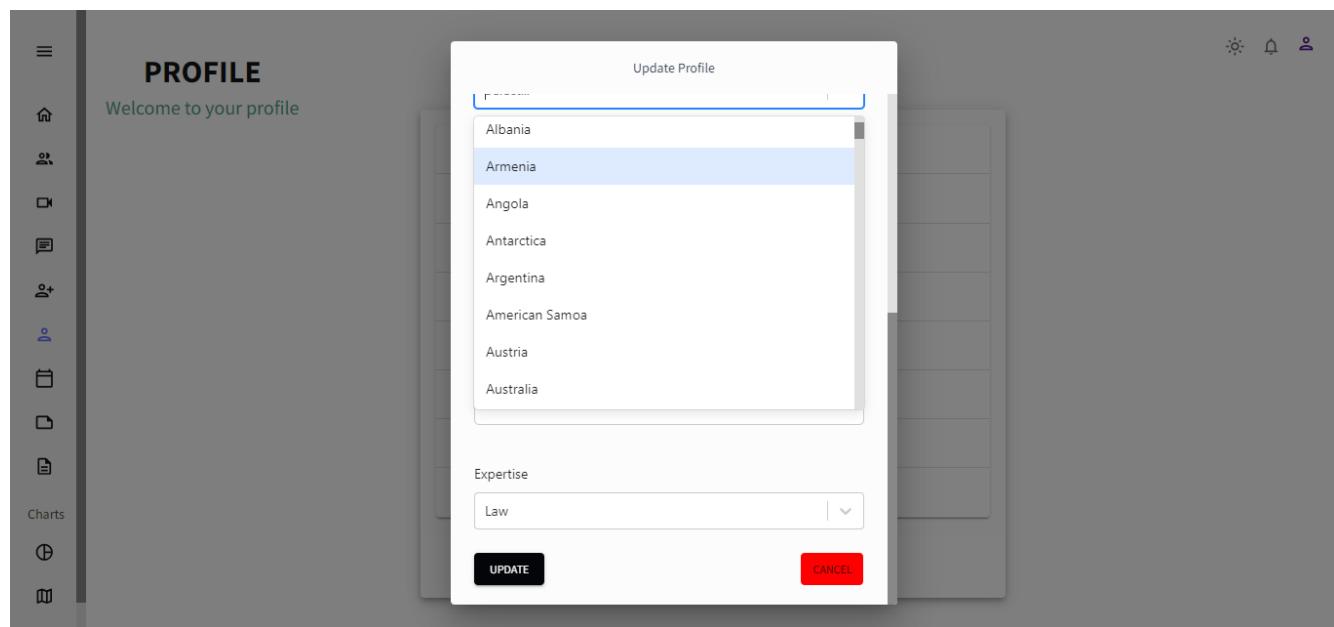


Figure V.15 : Capture d'écran de l'interface profile

V.9 Interface de fichier

Dans cette **interface V.16**, les utilisateurs peuvent organiser leur travail en utilisant des dossiers, avec la possibilité de créer et d'ajouter des fichiers pour maintenir une structure cohérente dans leurs interactions avec leurs clients. De plus, ils ont la capacité de modifier et de supprimer ces fichiers selon leurs besoins.

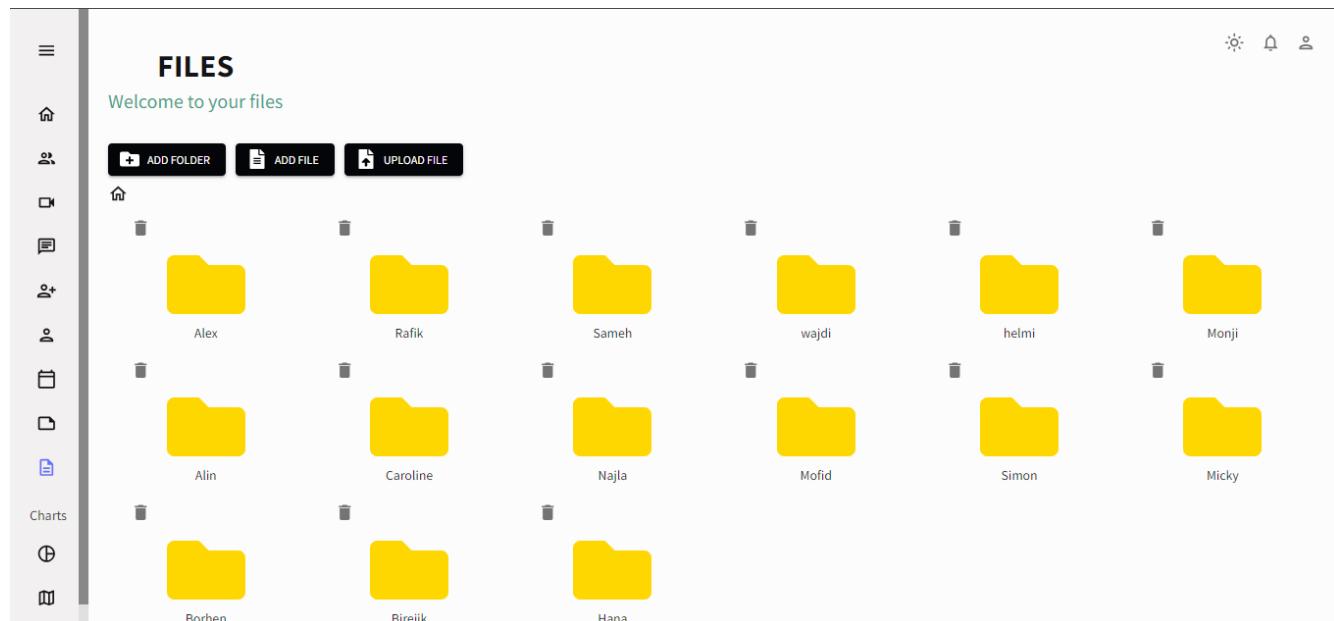


Figure V.16 : Capture d'écran de l'interface gestion de fichiers

Pour ajouter un nouveau dossier, l'utilisateur clique sur le bouton «ADD FOLDER»

comme le montre la **figure V.17**. Une fenêtre apparaîtra alors permettant à l'utilisateur de saisir un nom pour le dossier. L'utilisateur peut alors cliquer sur le bouton «SAVE» pour vérifier la création du dossier, ou sur «CANCEL» s'il souhaite annuler l'opération.

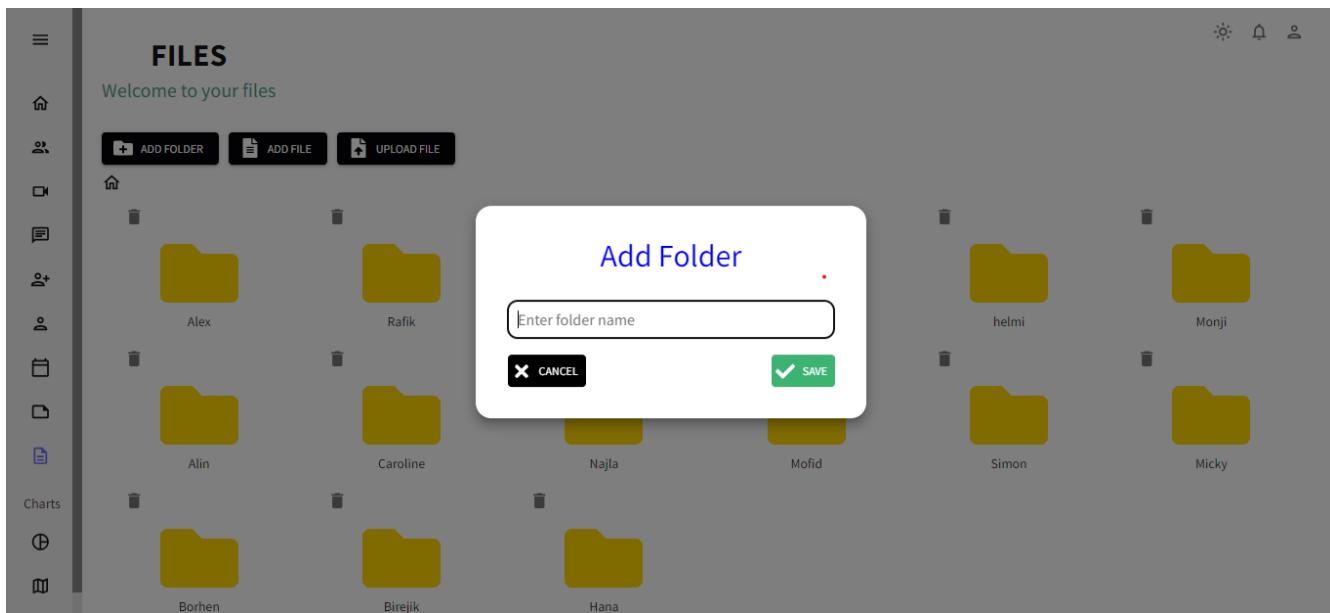


Figure V.17 : Capture d'écran de l'interface ajouter dossier

Pour ajouter un nouveau fichier, l'utilisateur clique sur le bouton "ADD FILE", comme indiqué par la **figure V.18**. Un éditeur de texte s'ouvre alors, permettant à l'utilisateur de saisir le nom du fichier ainsi que son contenu. Ensuite, il peut cliquer sur le bouton "SAVE" pour valider la création du fichier, ou sur "CANCEL" s'il souhaite annuler l'opération.

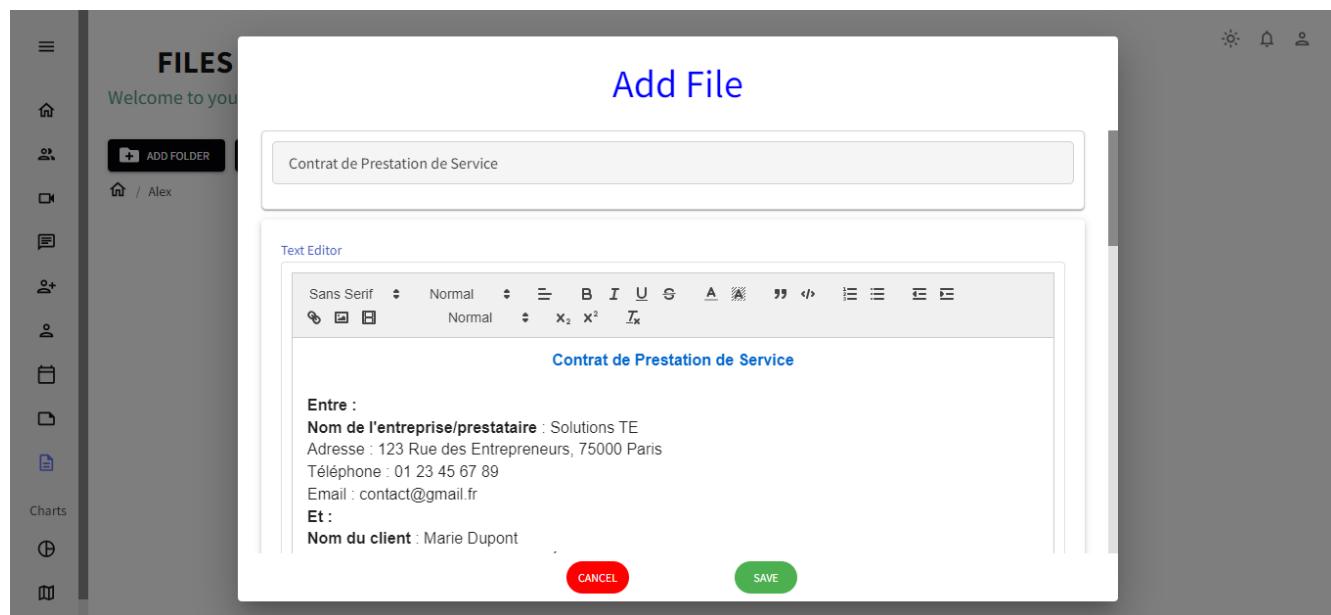


Figure V.18 : Capture d'écran de l'interface créer fichier

L'utilisateur a également la possibilité de modifier le contenu du fichier en rouvrant l'éditeur de texte, comme illustré dans la **figure V.19**. Après avoir apporté les modifications nécessaires, l'utilisateur peut cliquer sur le bouton "EDIT" pour enregistrer les modifications, ou sur "CANCEL" pour annuler la modification.

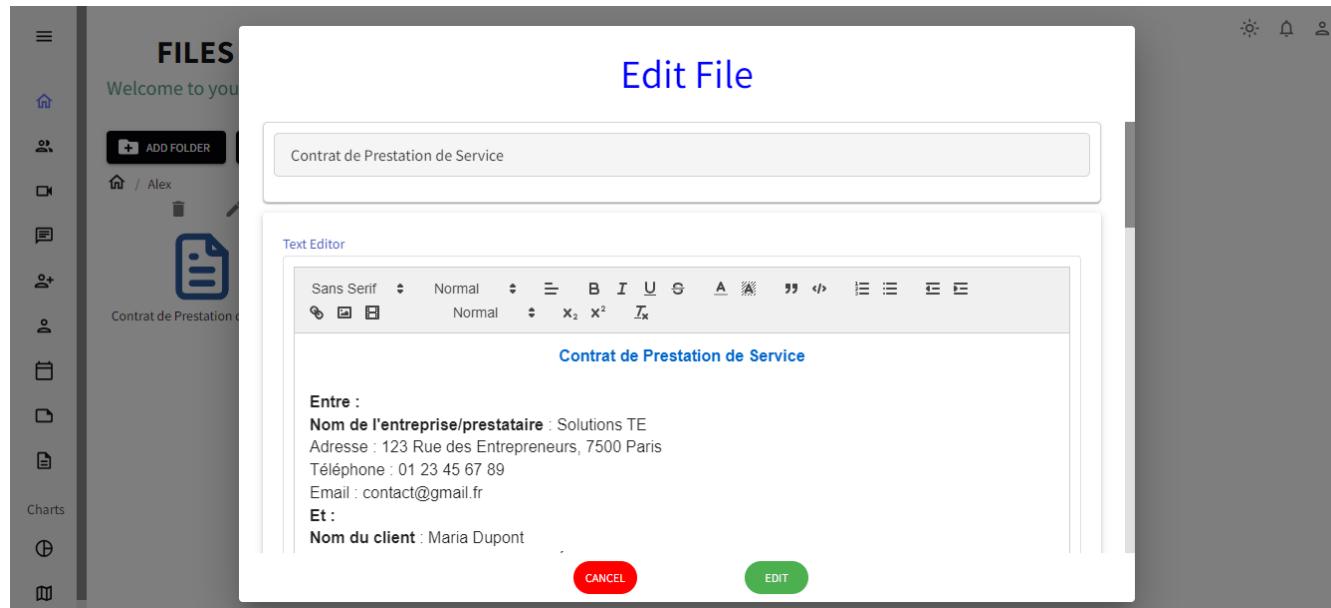


Figure V.19 : Capture d'écran de l'interface de modification

De plus, l'utilisateur peut supprimer un fichier selon ses besoins en cliquant sur l'icône de suppression, comme illustré dans la **figure V.20**. Une fenêtre de confirmation s'affiche alors, demandant à l'utilisateur de confirmer la suppression. Pour procéder,

l'utilisateur clique sur "Yes" pour confirmer la suppression du fichier, ou sur "No" pour annuler l'opération.

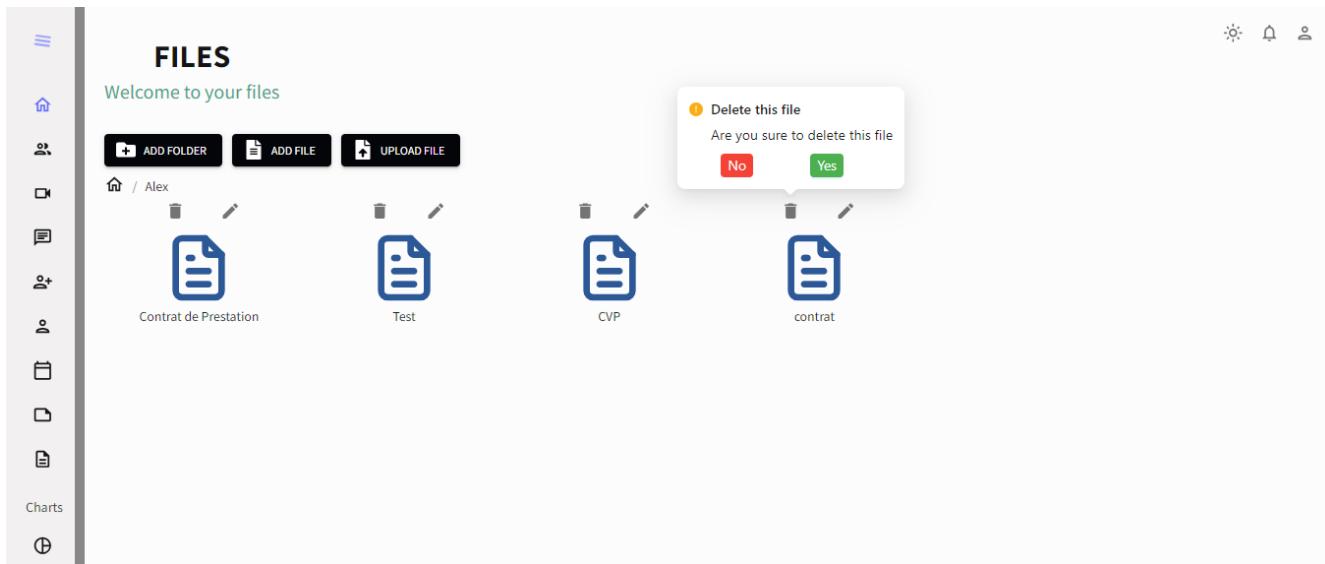


Figure V.20 : Capture d'écran de l'interface de suppression

Les utilisateurs peuvent également ajouter des fichiers en cliquant sur le bouton «UPLOAD FILE», comme le montre la **figure V.21**. Une fenêtre apparaîtra alors où il pourra cliquer sur «CHOOSE FILE» pour sélectionner le fichier à télécharger. Il clique ensuite sur le bouton "SAVE" pour vérifier l'ajout du fichier, ou sur "CANCEL" pour annuler l'opération.

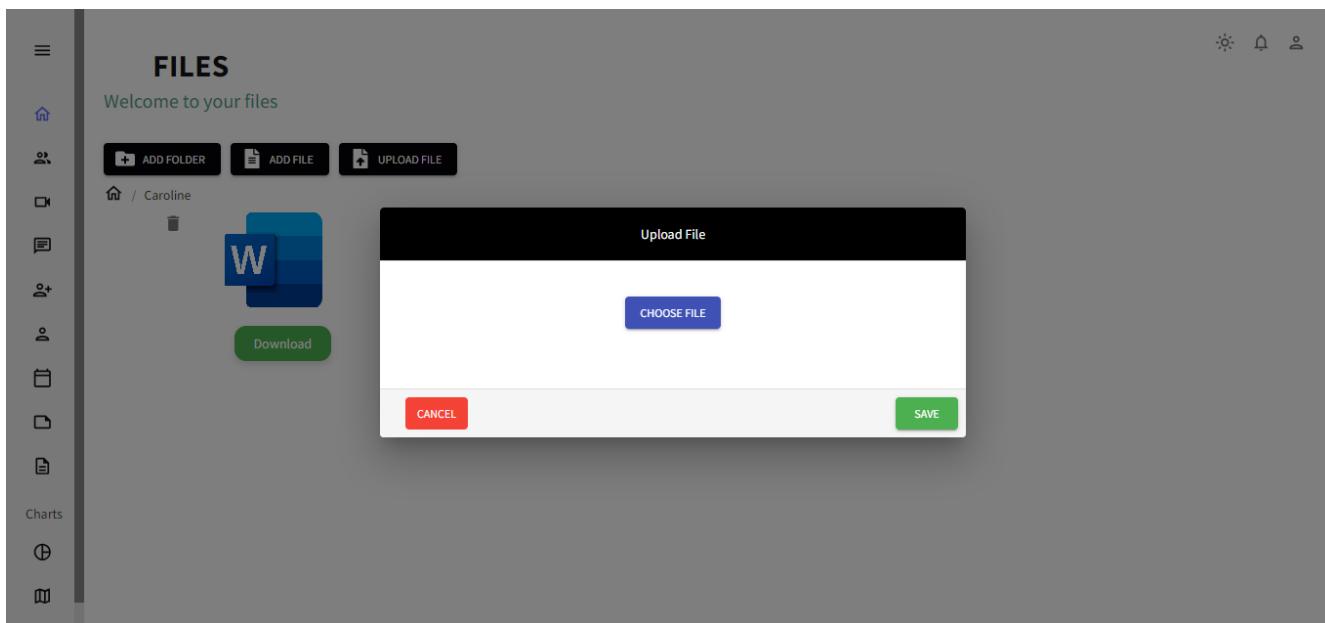


Figure V.21 : Capture d'écran de l'interface d'ajout d'une fichier

Lorsque l'utilisateur clique sur le bouton "SAVE", comme indiqué dans la **figure V.22**, le fichier est alors sauvegardé et intégré au système.

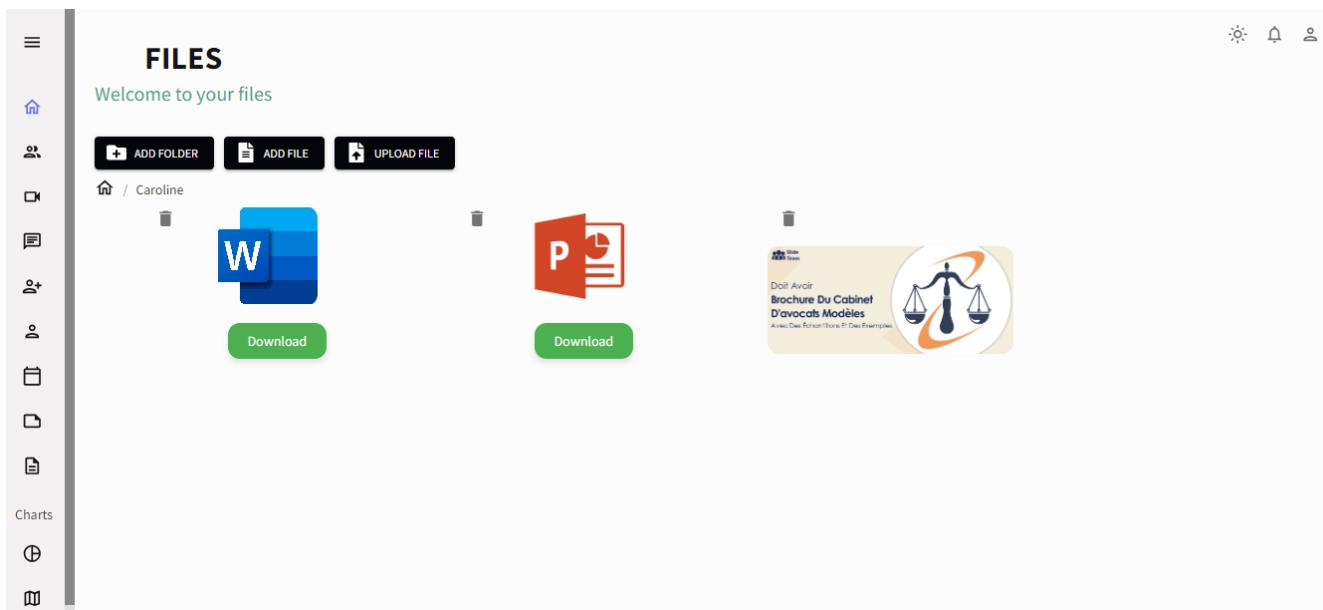


Figure V.22 : Capture d'écran de l'interface d'ajout des fichiers

V.10 Interface de note

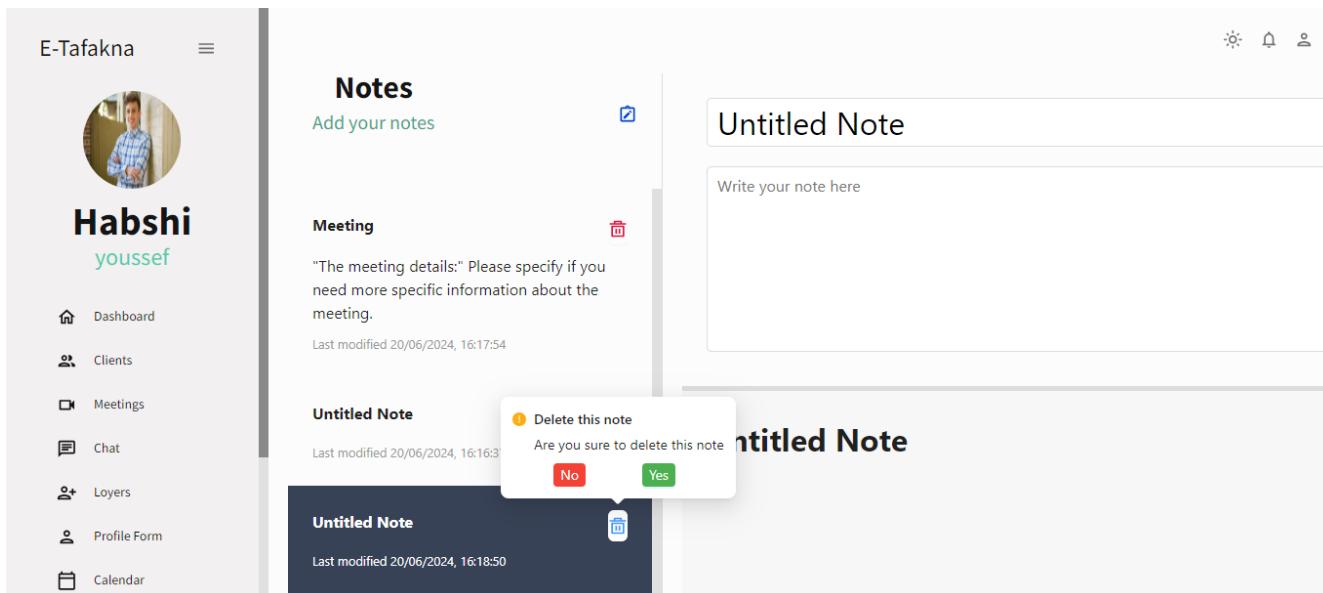


Figure V.23 : Capture d'écran de l'interface note

L'interface de la **figure V.23** permet aux utilisateurs de gérer facilement les notes clients en créant, modifiant et supprimant des annotations. Elle assure la sauvegarde

d'informations pertinentes avec la possibilité d'ajouter des détails supplémentaires au besoin. Pour assurer la sécurité des données, une alerte de confirmation est affichée avant toute suppression d'annotation. Cela garantit une gestion flexible et sécurisée des informations client, répondant ainsi aux besoins variés des utilisateurs.

V.11 Interface de sélection le type de chat

Dans cette interface, l'utilisateur doit sélectionner un type de chat en cliquant sur l'un des deux boutons disponibles, comme le montre la **figure V.24**. Chaque bouton représente une option de discussion disponible, permettant aux utilisateurs de sélectionner l'option qui correspond le mieux à leurs besoins en un seul clic.

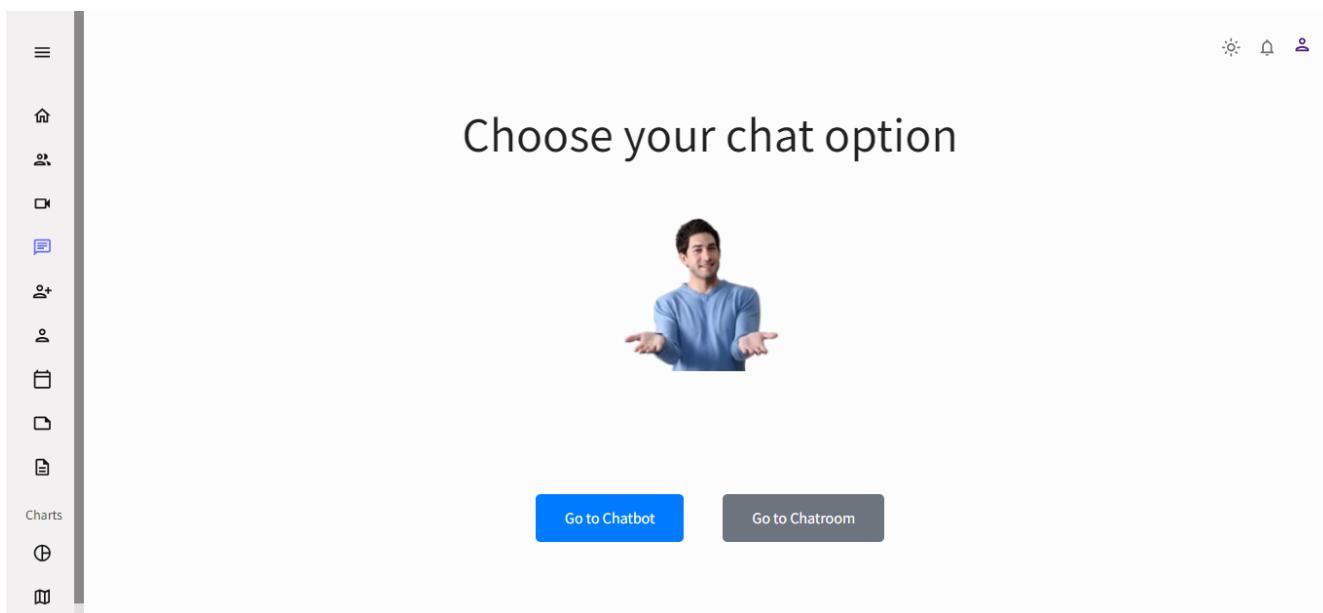


Figure V.24 : Capture d'écran de sélection

V.12 Interface de chatbot

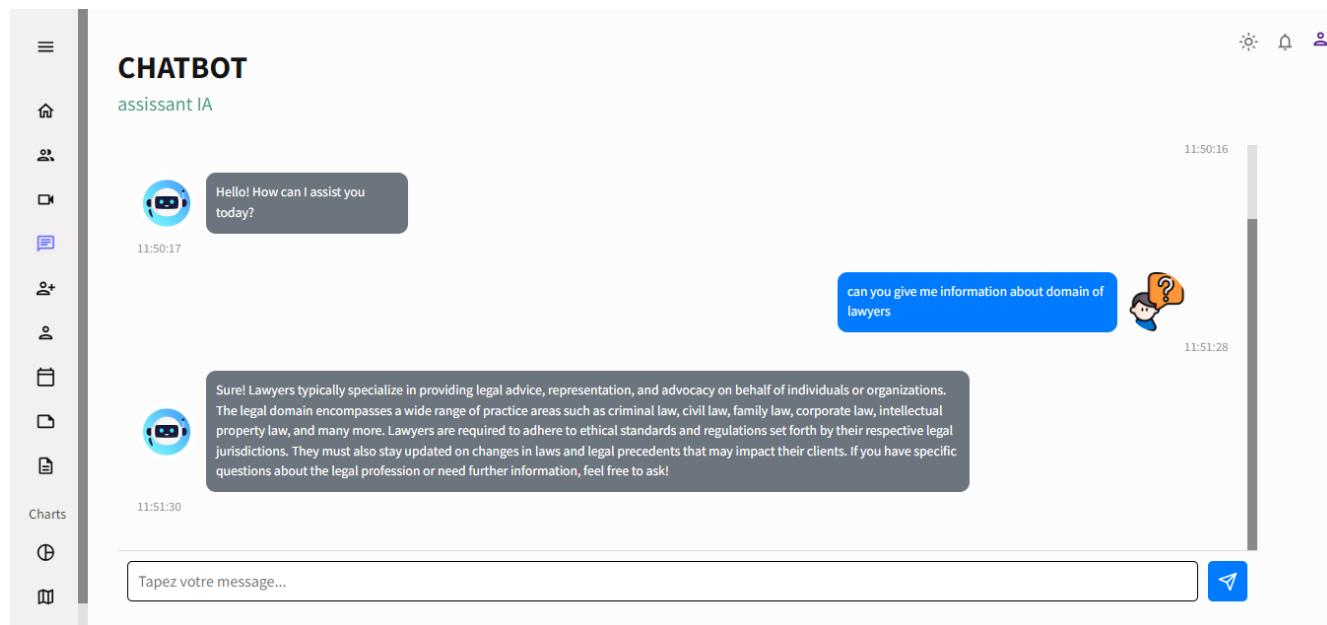


Figure V.25 : Capture d'écran de l'interface chatbot

Au sein de cette interface, les utilisateurs ont la possibilité d'engager une conversation par chat avec un expert en IA, comme illustré dans la **figure V.25**. Ce chatbot est conçu pour fournir divers services et répondre à différents besoins, notamment dans des domaines tels que le service client, le support technique, et la gestion des questions fréquentes. Les utilisateurs peuvent interagir avec le chatbot en posant des questions ou en demandant de l'aide dans leur domaine d'expertise.

V.13 Interface de chatroom

Dans cette section de la **figure V.26**, l'utilisateur commence par envoyer une invitation à un autre utilisateur pour engager une conversation par chat. Une fois que cette invitation est reçue, l'autre utilisateur a la possibilité d'accepter cette invitation, ce qui permet de démarrer la conversation et d'échanger des informations. Dans cette interface spécifique, l'utilisateur nommé "Hamdi Hamza" envoie une invitation à communiquer à l'utilisateur nommé "Habshi Youssef".

First name	Last name	Email	Phone number	Address	Region	Country	Action
Jihed	Boussarsar	jb@as-finance.tn	50011005	Laouina	Tnis	Tunisia	CHAT
youssef	Habshi	youssef@gmail.com	27128539	corba centre	tunis	French	Invitation pending
Achref	Saada	achref@gmail.com	56789321	centre ville	Nabel	Italia	Invitation pending
Salime	hamdi	salim@gmail.com	56789321	centre ville	sfax	Tunis	CHAT

Figure V.26 : Capture d'écran de l'interface d'invitation 1

Habshi Youssef peut alors accepter ou refuser l'invitation, comme le montre la **figure V.27**.

First name	Last name	Email	Phone number	Address	Region	Country	Action
Jihed	Boussarsar	jb@as-finance.tn	50011005	Laouina	Tnis	Tunisia	CHAT
Hamza	Hamdi	hamza@gmail.com	27128539	tunis	tozer	Tunisia	ACCEPT REFUSE
Achref	Saada	achref@gmail.com	56789321	centre ville	Nabel	Italia	Invitation pending
Salime	hamdi	salim@gmail.com	56789321	centre ville	sfax	Tunis	SEND INVITATION

Figure V.27 : Capture d'écran de l'interface d'invitation 2

Maintenant, les deux utilisateurs peuvent initier une conversation par chat et échanger des informations entre eux, comme illustré dans les **figures V.28** et **V.29** ci-dessous. Ils ont également la possibilité d'obtenir des détails supplémentaires sur divers sujets, facilitant ainsi une communication approfondie et efficace.

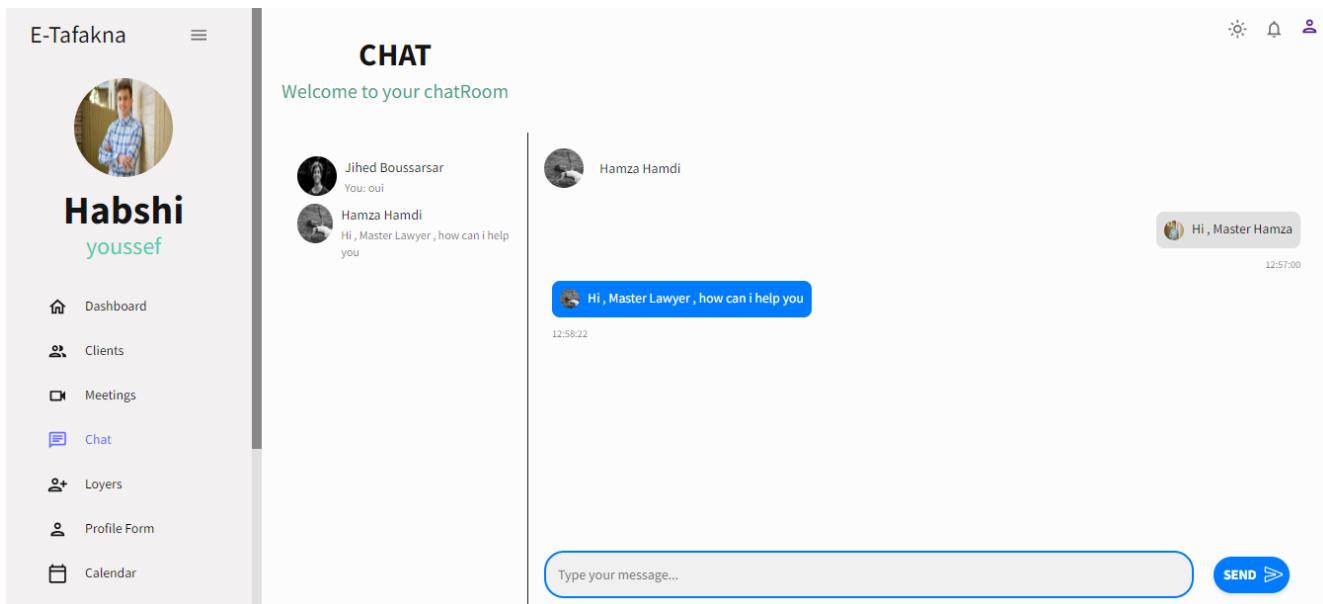


Figure V.28 : Capture d'écran de l'interface de chat 1

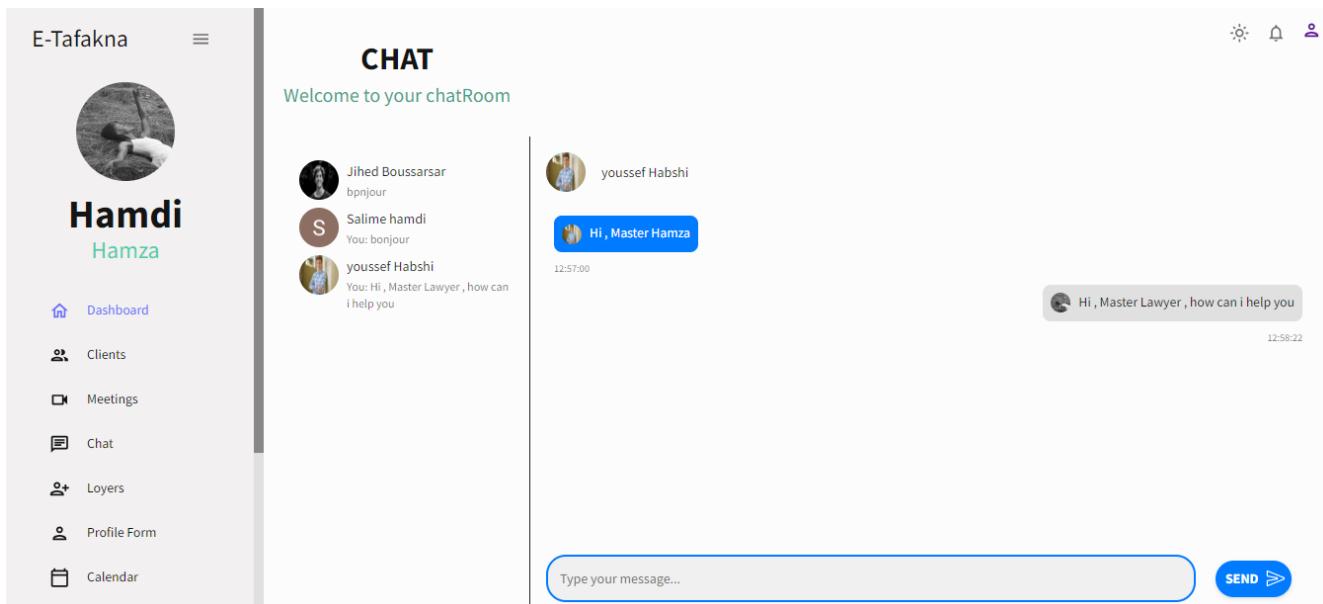


Figure V.29 : Capture d'écran de l'interface de chat 2

V.14 Conclusion

Ce chapitre a mis en lumière les diverses interfaces qui enrichissent notre site web, en soulignant leur conception méticuleuse. Nous avons examiné en détail les fonctionnalités essentielles de chaque interface, en insistant sur leur importance dans l'expérience utilisateur globale. Ces interfaces soigneusement conçues constituent le fondement d'une navigation fluide, intuitive et conviviale.

Chapitre VI

Tests et Résultats

Sommaire

VI.1	Introduction	88
VI.2	Les types de tests	88
VI.3	Test utilisé	88
VI.4	Configuration et Structure des Répertoires	89
VI.5	Cas de test	89
VI.5.1	Détails sur les cas de test	90
VI.6	Résultats	91
VI.7	conclusion	92
	Conclusion Générale	93

VI.1 Introduction

Cette section vise à fournir un aperçu des activités de tests menées pour ce projet. Le but des tests est de garantir la qualité, la fonctionnalité et la fiabilité du projet et de s'assurer qu'il est conforme aux spécifications architecturales et de conception définies. Les activités de tests sont réalisées sur la base du plan de tests et des cas définis à cet effet.

VI.2 Les types de tests

Tests fonctionnels : vérifier la fonctionnalité des applications Web et mobiles en fonction des exigences définies. Cela comprend le test de diverses fonctionnalités telles que l'enregistrement des utilisateurs, la connexion, la navigation, la soumission de formulaires, la validation des données et la gestion des erreurs.

Tests de performances : évaluez les performances des applications Web et mobiles dans différentes conditions de charge. Cela inclut le test des temps de réponse, des temps de chargement des pages et de l'utilisation des ressources pour garantir que les performances des applications répondent aux normes requises.

Tests de sécurité : identifiez et corrigez les vulnérabilités de sécurité potentielles dans les applications. Cela inclut des tests pour détecter les risques de sécurité potentiels tels que l'injection SQL, les scripts intersites , la falsification de requêtes intersites et d'autres menaces de sécurité.

Tests d'utilisabilité : évaluez la convivialité et l'expérience utilisateur de deux applications. Cela comprend le test de l'interface, de la navigation, de la présentation du contenu et de la convivialité globale de l'application.

Tests de régression : garantissent que les modifications apportées à l'application au cours du cycle de développement n'introduisent pas de nouveaux défauts ou n'impactent pas les fonctionnalités existantes. Cela inclut la revalidation des fonctionnalités et des cas de test précédemment testés.

VI.3 Test utilisé

Dans ce projet, une attention particulière sera portée aux tests de performances pour la partie frontend. Ces tests visent à évaluer la réactivité et la robustesse de l'interface utilisateur dans des conditions de charge variées. Plus spécifiquement, nous allons mesurer la vitesse de chargement des pages, la réactivité des interactions utilisateur telles que la navigation et les opérations de formulaire, ainsi que l'utilisation des ressources

du navigateur sous différents scénarios d'utilisation intensive. Ces tests permettront de garantir une expérience utilisateur fluide et sans interruption, même lors de pics de demande élevés.

VI.4 Configuration et Structure des Répertoires

Nous nous sommes focalisés sur les tests frontend en ajoutant Vitest et React Testing Library aux dépendances du projet, comme illustré dans la **figure VI.1**.

```
108 |   "@testing-library/jest-dom": "^4.4.6",
109 |   "@testing-library/react": "^16.0.0",
110 |   "axios-mock-adapter": "^1.22.0",
111 |   "babel-jest": "^29.7.0",
112 |   "jsdom": "^24.1.0",
113 |   "vitest": "^1.6.0"
```

Figure VI.1 : Liste de dépendances des outils pour les tests

Bien qu'il existe plusieurs façons de structurer les suites de tests dans le projet, nous avons choisi de créer un dossier séparé contenant des sous-dossiers, chacun dédié aux tests unitaires associés à chaque fichier. Cette **figure VI.2** illustre la structure des tests :

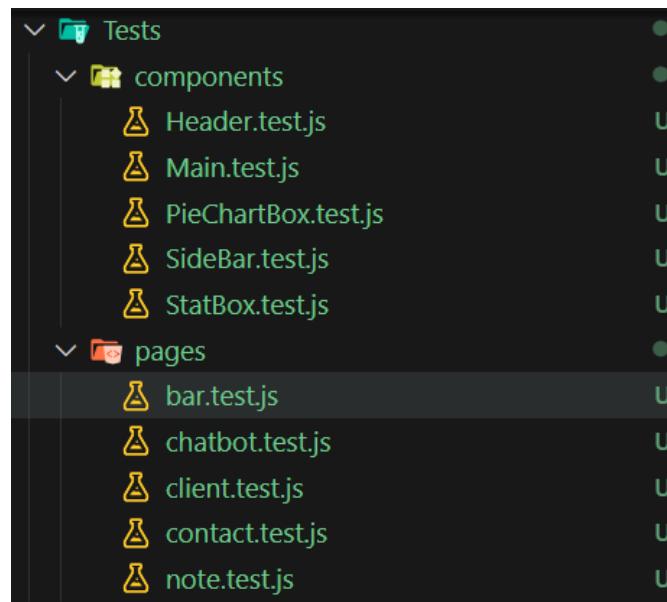


Figure VI.2 : Structure du dossier de test

VI.5 Cas de test

Dans cette section, nous allons détailler les différents cas de test pour un fichier particulier. La figure présente le code du test du fichier ‘Main.test.js’ :

```
15 describe('Main Component', () => {
16   // Définition d'une note active fictive pour les tests
17   const activeNote = [
18     title: 'Test Title',
19     body: 'Test Body',
20     lastModified: Date.now(),
21   ];
22
23   // Création d'une fonction mock pour onUpdateNote
24   const onUpdateNote = jest.fn();
25
26   // Teste l'affichage du message "No note selected" lorsque aucune note active n'est présente
27   test('displays "No note selected" when no active note is present', () => {
28     // Rendu du composant Main avec activeNote défini à null
29     render(<Main activeNote={null} onUpdateNote={onUpdateNote} />);
30     // Vérification que le texte "No note selected" est bien présent dans le document
31     expect(screen.getByText(/no note selected/i)).toBeInTheDocument();
32   });
33
34   // Teste que le titre et le corps de la note sont rendus correctement
35   test('renders the note title and body correctly', () => {
36     // Rendu du composant Main avec une note active
37     render(<Main activeNote={activeNote} onUpdateNote={onUpdateNote} />);
38     // Vérification que le titre de la note est bien présent dans le document
39     expect(screen.getByDisplayValue(activeNote.title)).toBeInTheDocument();
40     // Vérification que le corps de la note est bien présent dans le document
41     expect(screen.getByDisplayValue(activeNote.body)).toBeInTheDocument();
42   });
43
44   // Teste l'appel à onUpdateNote avec le titre mis à jour lors du changement de titre
45   test('calls onUpdateNote with updated title on title change', () => {
46     // Rendu du composant Main avec une note active
47     render(<Main activeNote={activeNote} onUpdateNote={onUpdateNote} />);
48     const newTitle = 'Updated Title';
49     // Simulation d'un changement de titre
50     fireEvent.change(screen.getByDisplayValue(activeNote.title), {
51       target: { value: newTitle },
52     });
53   });
54 }
```

Figure VI.3 : Exemple du cas de test

VI.5.1 Détails sur les cas de test

- Test "displays 'No note selected' when no active note is present" :
 - **But :** Vérifier que le composant affiche "No note selected" lorsqu'aucune note active n'est sélectionnée.
 - **Méthode :** Le composant Main est rendu avec activeNote défini à null. Le texte "No note selected" est recherché dans le DOM et on s'assure qu'il est bien présent.
- Test "renders the note title and body correctly" :
 - **But :** Vérifier que le titre et le corps de la note active sont affichés correctement.
 - **Méthode :** Le composant Main est rendu avec une activeNote définie. On vérifie que les champs de titre et de corps contiennent bien les valeurs de la note active.
- Test "calls onUpdateNote with updated title on title change" :
 - **But :** Vérifier que la fonction onUpdateNote est appelée avec le nouveau titre lorsque le titre est modifié.
 - **Méthode :** Le composant Main est rendu avec une activeNote définie. Lors d'un changement de titre, on vérifie que la fonction onUpdateNote est effectivement appellée avec le nouveau titre.

- **But :** Vérifier que la fonction onUpdateNote est appelée avec le nouveau titre lorsque le titre est modifié.
 - **Méthode :** Le composant Main est rendu avec une activeNote définie. Le champ de titre est modifié avec une nouvelle valeur, et on vérifie que onUpdateNote est appelé avec les nouvelles valeurs de la note.
- Test "calls onUpdateNote with updated body on body change" :
 - **But :** Vérifier que la fonction onUpdateNote est appelée avec le nouveau corps lorsque le corps est modifié.
 - **Méthode :** Le composant Main est rendu avec une activeNote définie. Le champ de corps est modifié avec une nouvelle valeur, et on vérifie que onUpdateNote est appelé avec les nouvelles valeurs de la note.

Ces tests permettent de s'assurer que le composant Main fonctionne correctement en affichant les bonnes informations et en appelant les bonnes fonctions lors des modifications.

VI.6 Résultats

Dans cette section, nous allons présenter en détail les résultats du test de performance, comme illustré dans la **figure VI.4**

```
PASS  src/Tests/components/Main.test.js
Main Component
  ✓ displays "No note selected" when no active note is present (33 ms)
  ✓ renders the note title and body correctly (23 ms)
  ✓ calls onUpdateNote with updated title on title change (16 ms)
  ✓ calls onUpdateNote with updated body on body change (11 ms)

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        2.203 s
Ran all test suites matching /vitest|.\Main.test.js/i.
```

Figure VI.4 : Résultat de test

Les résultats des tests pour le fichier ‘Main.test.js‘ montrent que tous les tests ont réussi. La section "Main Component" regroupe les quatre tests individuels, chacun validé avec succès. Le premier test vérifie que le composant affiche "No note selected" lorsqu'aucune note active n'est présente, avec une durée de 33 ms. Le second test s'assure que le titre et le corps de la note active sont affichés correctement, prenant 23 ms.

Le troisième test confirme que la fonction ‘onUpdateNote‘ est appelée avec le nouveau titre lorsque celui-ci est modifié, exécuté en 16 ms. Enfin, le quatrième test vérifie que ‘onUpdateNote‘ est appelé avec le nouveau corps lorsque celui-ci est modifié, en 11 ms. En résumé, une suite de tests a été exécutée et tous les quatre tests ont réussi en 2,203 secondes au total, sans utiliser de captures instantanées (snapshots). Ces résultats démontrent la stabilité et l’efficacité du composant ‘Main.test.js‘

VI.7 conclusion

Cette section conclut les tests effectués pour garantir la qualité, la fonctionnalité et la fiabilité du projet selon les spécifications architecturales et de conception définies. Nous avons examiné divers types de tests et utilisé plusieurs bibliothèques pour assurer une couverture complète. Un cas de test spécifique a été détaillé pour illustrer sa mise en œuvre et ses résultats obtenus.

Conclusion Générale et Perspectives

Le projet de fin d'études a représenté une étape cruciale dans le développement d'une application web innovante, mettant en œuvre des techniques modernes pour répondre aux besoins émergents des utilisateurs. Tout au long de ce parcours, notre approche a été caractérisée par une méthodologie rigoureuse, assurant ainsi la qualité et la pertinence de chaque étape, de la conception à la réalisation.

Le rapport débute en exposant en détail le contexte dans lequel s'inscrit notre projet, ainsi que la problématique à laquelle il cherche à répondre. Une attention particulière est portée à la présentation de l'organisme d'accueil, E-Tafakna, afin de situer notre travail dans son environnement.

Une analyse minutieuse de l'existant a été menée pour identifier les lacunes et les opportunités d'amélioration. Cette phase a permis d'orienter notre réflexion vers une solution adaptée aux besoins spécifiques des utilisateurs.

Ensuite, nous avons présenté en détail notre solution envisagée, en mettant en lumière les choix stratégiques et les orientations prises pour son développement. La méthodologie de travail adoptée ainsi que la planification précise ont été détaillées pour assurer une exécution efficace du projet.

L'analyse et la spécification des besoins ont été abordées de manière exhaustive, couvrant les démarches à suivre, l'identification des acteurs impliqués, et l'analyse détaillée des exigences fonctionnelles et non fonctionnelles. Le diagramme de cas d'utilisation global a été présenté comme un outil central pour comprendre l'ensemble du système.

Chaque fonctionnalité a été ensuite détaillée à travers des diagrammes de cas d'utilisation spécifiques, des diagrammes de classe et des diagrammes de séquence, offrant ainsi une vue détaillée de l'architecture et du fonctionnement de l'application.

Une discussion approfondie sur l'architecture logique et physique de l'application E-Tafakna a été menée, mettant en évidence les choix technologiques et les considérations d'implémentation.

En conclusion, ce rapport est enrichi par des captures d'écran illustrant les interfaces

développées, fournissant une vision concrète du résultat de notre travail et mettant en lumière l'impact positif de notre solution sur l'expérience utilisateur. Nous avons également ajouté des tests unitaires pour assurer l'efficacité et l'optimisation de l'application. À ce stade, le développement de l'application est en cours, avec des améliorations continues pour répondre aux besoins des utilisateurs et aux exigences du projet.

Webographie

- [1] Présentation de la société. Disponible sur <https://www.e-tafakna.com/> (consulté le 12 Février 2024).
- [2] Présentation de l'application Clio https://pdf.wondershare.fr/business/legal-practice-management-software.html?source=5&third_register=1 (consulté le 13 Février 2024).
- [3] Présentation de l'application Lexicata https://pdf.wondershare.fr/business/legal-practice-management-software.html?source=5&third_register=1 (consulté le 13 Février 2024).
- [4] Présentation de l'application Jarvis Legal https://pdf.wondershare.fr/business/legal-practice-management-software.html?source=5&third_register=1 (consulté le 13 Février 2024).
- [5] Le modèle en cascade. Disponible sur https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade (consulté le 14 Février 2024).
- [6] Présentation du modèle REST API. Disponible sur <https://www.c-sharpcorner.com/article/restful-api-in-net-core-using-ef-core-and-postgres/> (consulté le 14 Février 2024).
- [7] Disponible sur <https://www.bezkoder.com/react-node-express-mysql/> (consulté le 25 Février 2024).
- [8] Présentation du modèle MVVM. Disponible sur <https://www.docdoku.com/blog/2015/02/17/architecturer-ses-applications-js-pattern-mvvm/> (consulté le 2 Mars 2024).
- [9] Présentation de Visual studio code. Disponible sur <https://code.visualstudio.com/> (consulté le 10 Mars 2024).
- [10] Présentation de Visual-paradigme. Disponible sur <https://online.visual-paradigm.com/> (consulté le 21 Mars 2024).
- [11] Présentation de Thunder Client. Disponible sur <https://docs.thunderclient.com/> (consulté le 29 Mars 2024).

- [12] Présentation de HTML. Disponible sur <https://www.pngegg.com/fr/search?q=html+5> (consulté le 1 Mai 2024).
- [13] Présentation de CSS. Disponible sur https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade(consulté le 3 Mai 2024).
- [14] Présentation de JavaScript. Disponible sur <https://quintagroup.com/cms/js>(consulté le 5 Mai 2024).
- [15] Présentation de Bootstrap. Disponible sur <https://getbootstrap.com/>(consulté le 7 Mai 2024).
- [16] Présentation de react. Disponible sur <https://legacy.reactjs.org/> (consulté le 10 Mai 2024).
- [17]
- [18] Présentation de l'outil de test. Disponible sur <https://medium.com/@krithi.muthuraj/lets-try-vitest-for-a-change-349b58fffb39> (consulté le 23 Juin 2024). Présentation de nodeJS. Disponible sur <https://nodejs.org/> (consulté le 14 Mai 2024).
- [19] Présentation de l'expressJS. Disponible sur <https://kinsta.com/fr/base-de-connaissances/qu-est-express-js/> (consulté le 21 Mai 2024).
- [20] Présentation de la bibliothéque OpenAi. Disponible sur <https://www.npmjs.com/package/openai/v/4.0.0-beta.4>(consulté le 24 Mai 2024).
- [21] Présentation de mysqlWorkbench. Disponible sur https://fr.wikipedia.org/wiki/MySQL_Workbench(consulté le 10 Juin 2024).
- [22] Présentation des charts.Disponible sur <https://recharts.org/en-US/> (consulté le 11 Juin 2024).
- [23] Présentation de la bibliothéque géographie. Disponible sur <https://www.react-google-charts.com/>(consulté le 13 Juin 2024).
- [24] Présentation des packages express. Disponible sur <https://www.npmjs.com/package/express>(consulté le 17 Juin 2024).
- [25] Présentation de l'architecture du backend. Disponible sur <https://www.simplilearn.com/understanding-node-js-architecture-article>(consulté le 21 Juin 2024).

RÉSUMÉ

Ce travail a été réalisé dans le cadre de notre stage de fin d'études chez E-TAFAKNA, en vue de l'obtention du Diplôme national d'Ingénieur en informatique à la Faculté des Sciences de Tunis pour l'année universitaire 2023/2024. Ce projet, d'une durée de quatre mois, avait pour but de développer une application web destinée aux avocats et comptables. L'application permettait de planifier des réunions, gérer les clients, visualiser des graphiques, faciliter les échanges de messages, ainsi que d'ajouter des fichiers spécifiques et des notes lors des réunions. Pour ce faire, nous avons utilisé plusieurs technologies du web ainsi qu'une API de génération d'intelligence artificielle et test unitaire.

Mots clés : Développement Web, IA Générative, Test Unitaire

ABSTRACT

This work was carried out as part of our end-of-studies internship at E-TAFAKNA, towards obtaining the National Diploma in Computer Engineering from the Faculty of Sciences of Tunis for the academic year 2023/2024. This four-month project aimed to develop a web application for lawyers and accountants. The application facilitated scheduling meetings, managing clients, viewing graphs, facilitating message exchanges, as well as adding specific files and notes during meetings. To achieve this, we utilized various web technologies , an artificial intelligence generative API and unit test.

Key words : Web Development, Generative AI, Unit Test

ملخص

تم تنفيذ هذا العمل في شركة E-TAFAKNA ، كجزء من مشروع ختم الدروس للحصول على الشهادة الوطنية للمهندسة الإعلامية بكلية العلوم بتونس بعنوان العام الدراسي 2023/2024. كان هذا المشروع الذي استمر لمدة أربعة أشهر يهدف إلى تطوير تطبيق ويب مخصص للمحامين والمحاسبين. ساعد التطبيق في جدولة الاجتماعات، وإدارة العملاء، وعرض الرسوم البيانية، وتسهيل تبادل الرسائل، بالإضافة إلى إمكانية إضافة ملفات محددة وملاحظات خلال الاجتماعات. لتحقيق هذه الأهداف، تم استخدام مجموعة متنوعة من التقنيات المتقدمة مثل تطوير الويب، الذكاء الإصطناعي التوليدى، واختبار الوحدة

كلمات مفاتيح : تطوير الويب، الذكاء الإصطناعي التوليدى، واختبار الوحدة

norchenmezni1@gmail.com
Tél : 52 979 979

البريد الإلكتروني : Fax : 52 979 979 : 52 979 979 : 52 979 979 :
Email : norchenmezni1@gmail.com