

PROJET DE FIN D'ETUDES
3ème Année en développement Informatique et Réseaux

Conception et réalisation d'une application mobile de consultation juridique

Réalisé par :

KHEIREDDINE Nacer-eddine

Encadré par:

M. EL BAKKALI Mohammed

Membres du jury:

M. EL BAKKALI Mohammed

Mme Rkia Fajr

M. Mohamed Rachdi

Au sein de 6SOLUTIONS :



Dédicaces

Je dédie ce travail à:

Ma mère, qui m'a non seulement donné la vie, mais aussi inculqué les valeurs de travail acharné, de persévérance et de compassion. Ta tendresse, tes sacrifices et ta foi en moi m'ont permis de surmonter les moments les plus difficiles. Sans ton soutien indéfectible, ce chemin aurait été bien plus ardu. Chaque étape de cette réalisation est aussi la tienne.

Mon père, l'exemple de la force tranquille et de la sagesse. Tu m'as appris à garder la tête haute face aux défis et à aborder chaque obstacle avec détermination. Tes conseils avisés, ta patience et ta confiance en moi ont été un guide inestimable. Ce travail est aussi le fruit de tout ce que tu m'as transmis.

Mon petit frère, dont la joie de vivre et l'enthousiasme sont une source constante d'inspiration. Même si tu es plus jeune, ta présence m'a souvent rappelé l'importance de rester positif et motivé. Merci pour ta compréhension, ton soutien discret et les moments de légèreté qui m'ont aidé à relâcher la pression.

Mes amis, pour votre soutien indéfectible tout au long de ce parcours. Vos encouragements, vos conseils et votre capacité à me remonter le moral dans les moments de doute ont été précieux. Vous avez transformé cette expérience académique en une aventure plus humaine et enrichissante. Je vous remercie pour votre présence et votre amitié sincère, qui ont fait toute la différence.

Remerciements

Je tiens à exprimer ma sincère gratitude à **Monsieur Mohammed EL BAKKALI**, mon encadrant, pour sa supervision bienveillante, ses précieux conseils et sa disponibilité tout au long de la réalisation de ce projet. Son expertise et son soutien constant ont été essentiels à l'accomplissement de ce travail.

Je remercie également l'ensemble du corps enseignant de **SUPÉMIR** pour leur enseignement de qualité, ainsi que pour les ressources et l'encadrement offerts durant toute la durée de mes études. Leur soutien a grandement contribué à la réussite de ce projet.

Un grand merci à **Madame Rkia Fajr**, mon enseignante, pour avoir facilité l'accès à des ressources importantes et pour sa coopération précieuse durant la période de recherche.

Enfin, je souhaite remercier toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce projet, par leurs encouragements, leur soutien ou leurs conseils.

Table des matières

| | |
|---|-----------|
| Introduction générale..... | 1 |
| Chapitre 1 : Présentation du cadre de projet..... | 2 |
| 1. Introduction | 3 |
| 2. Présentation de 6Solutions | 3 |
| 3. Etude de l'existant..... | 4 |
| 3.1. Description de l'existant..... | 4 |
| 3.2. Critique de l'existant | 4 |
| 3.3. Solution proposée | 5 |
| 4. Choix de modèle de développement..... | 5 |
| 5. Planning prévisionnel | 6 |
| 6. Conclusion..... | 7 |
| Chapitre 2 : Spécification des besoins..... | 8 |
| 1. Introduction | 9 |
| 2. Spécification des besoins fonctionnels | 9 |
| 2.1. Besoin fonctionnel global 1 : Gestion des comptes utilisateurs | 9 |
| 2.1.1. Sous-besoin : Création de compte | 9 |
| 2.1.2. Sous-besoin : Connexion et authentification | 9 |
| 2.1.3. Sous-besoin : Mise à jour du profil | 9 |
| 2.2. Besoin fonctionnel global 2 : Recherche et consultation | 9 |
| 2.2.1. Sous-besoin : Recherche d'avocat..... | 9 |
| 2.2.2. Sous-besoin : Consultation des profils | 9 |
| 2.3. Besoin fonctionnel global 3 : Gestion des consultations..... | 10 |
| 2.3.1. Sous-besoin : Prise de rendez-vous | 10 |
| 2.3.2. Sous-besoin : Suivi des consultations..... | 10 |
| 2.3.3. Sous-besoin : Messagerie intégrée | 10 |
| 3. Spécification des besoins non fonctionnels | 10 |
| 4. Présentation des cas d'utilisation..... | 10 |
| 4.1. Présentation des acteurs..... | 10 |
| 4.2. Description des cas d'utilisation..... | 11 |
| 4.3. Diagramme des cas d'utilisation global..... | 12 |
| Chapitre 3 : Conception du système | 13 |
| 1. Introduction | 14 |
| 2. Modélisation dynamique | 14 |
| 2.1. Diagrammes de navigation | 14 |

| | | |
|---|---|-----------|
| 2.2. | Diagrammes de séquences..... | 15 |
| 3. | Modélisation statique..... | 16 |
| 3.1. | Diagramme de classes | 16 |
| 3.2. | Architecture de l'application | 17 |
| 3.2.1. | Architecture logiciel | 17 |
| 3.2.2. | Architecture matériel | 18 |
| 4. | Conclusion..... | 19 |
| Chapitre 4 : Réalisation du système..... | | 20 |
| 1. | Introduction | 21 |
| 2. | Environnement de développement | 21 |
| 2.1. | Environnement matériel | 21 |
| 2.2. | Environnement logiciel | 22 |
| 3. | Principales interfaces graphiques | 27 |
| 4. | Conclusion..... | 34 |
| Conclusion générale | | 35 |
| Webographie | | 36 |
| ANNEXE (A) : Autres interfaces de l'application..... | | 37 |
| ANNEXE (B) : Schéma de l'Architecture Backend de l'Application..... | | 41 |

Liste des figures

| | |
|--|----|
| Figure 1 : Logo de la société | 3 |
| Figure 2 : Adresse de la société | 3 |
| Figure 3 : Diagramme des cas d'utilisation global | 12 |
| Figure 4 : Diagramme de navigation | 14 |
| Figure 5 : Diagramme de séquences | 15 |
| Figure 6 : Diagramme de classe | 16 |
| Figure 7 : Logo IntelliJ Idea..... | 22 |
| Figure 8 : Logo Android Studio..... | 22 |
| Figure 9 : Logo Flutter | 23 |
| Figure 10 : Logo Dart | 23 |
| Figure 11 : Logo SpringBoot Tomcat | 24 |
| Figure 12 : Logo Java | 24 |
| Figure 13 : Logo Redis | 25 |
| Figure 14 : Logo PostgreSQL..... | 25 |
| Figure 15 : Logo pgAdmin 4 | 25 |
| Figure 16 : Logo Git & GitHub | 26 |
| Figure 17 : Logo StarUML | 26 |
| Figure 18 : Interface Welcome Page | 27 |
| Figure 19 : Interface Sign In..... | 28 |
| Figure 20 : Interface Sign Up | 29 |
| Figure 21 : Interface Home Client..... | 30 |
| Figure 22 : Interface Home Avocat | 31 |
| Figure 23 : Interface Consultation(Upcoming) | 32 |
| Figure 24 : Interface Consultation(Completed)..... | 32 |
| Figure 25 : Interface Consultation(Cancelled) | 32 |
| Figure 26 : Interface Messagerie | 33 |
| Figure 27 : Interface Forgot Password | 37 |
| Figure 28 : Interface Profile Avocat | 38 |
| Figure 29 : Interface Profile Client..... | 39 |
| Figure 30 : Interface Recherche Avocats (Search)..... | 40 |
| Figure 31 : Schéma de l'Architecture Backend de l'Application..... | 41 |

Liste des tableaux

| | |
|--|----|
| Tableau 1 : Planning prévisionnel | 6 |
| Tableau 2 : Exemple de cas d'utilisation n°1 : S'inscrire en tant que client | 11 |
| Tableau 3 : Exemple de cas d'utilisation n°2 : Prendre un rendez-vous avec un avocat | 11 |

Résumé

La gestion des consultations juridiques entre avocats et clients est essentielle pour faciliter l'accès à la justice et améliorer l'efficacité des services juridiques. Ce projet de fin d'études vise à développer une application mobile permettant aux clients de rechercher des avocats, de consulter leurs profils, de prendre des rendez-vous pour des consultations, et d'échanger en temps réel via une messagerie intégrée.

L'application propose plusieurs fonctionnalités clés, telles que la création et la gestion des comptes pour les avocats et les clients, la recherche d'avocats basée sur des critères spécifiques (nom, spécialisation, zone de service), la gestion des consultations (prise de rendez-vous, suivi des demandes), ainsi qu'un système de messagerie en temps réel pour faciliter la communication entre les parties. Un tableau de bord est également mis en place pour suivre l'historique des consultations et des interactions.

Les principaux objectifs de ce projet sont de mettre en place une plateforme conviviale et sécurisée pour améliorer l'accès aux services juridiques, de faciliter la communication entre avocats et clients, et de rendre le processus de prise de rendez-vous plus transparent et efficace.

Introduction générale

Dans un contexte juridique en constante évolution, la gestion efficace des consultations et des relations entre avocats et clients est devenue cruciale pour répondre aux besoins des clients tout en optimisant le temps des avocats. Une mauvaise gestion peut entraîner des retards ou des pertes de clients, tandis qu'une organisation rigoureuse améliore l'efficacité et la réactivité face aux demandes juridiques.

Ce projet propose le développement d'une application mobile facilitant la mise en relation entre avocats et clients. L'application permet de créer des comptes personnalisés, de rechercher des avocats par spécialité et localisation, et de gérer les consultations avec un suivi en temps réel (statuts en attente, complétée, annulée). Grâce à une interface intuitive et des notifications automatiques, elle améliore la gestion des consultations et minimise les erreurs de communication.

Le rapport retrace les différentes phases du projet, de l'analyse des besoins à la réalisation technique, en s'appuyant sur une méthodologie rigoureuse.

Le **Chapitre 1** présente le cadre du projet et une étude critique des solutions existantes, mettant en lumière les lacunes actuelles. Une application mobile est proposée comme solution innovante pour répondre à ces besoins.

Le **Chapitre 2** spécifie les besoins fonctionnels et non fonctionnels, couvrant la gestion des comptes utilisateurs, la recherche d'avocats, et la gestion des consultations.

Le **Chapitre 3** détaille la conception du système à travers des modèles dynamiques et statiques, décrivant l'architecture logicielle et matérielle de l'application.

Enfin, le **Chapitre 4** aborde la réalisation du système, illustrant l'environnement de développement et les principales interfaces graphiques utilisées.

En conclusion, cette application représente une solution pratique et sécurisée, optimisant les interactions entre avocats et clients tout en facilitant la gestion des consultations juridiques.

Chapitre

1

**Présentation
du cadre de
projet**

1. Introduction

Ce premier chapitre a pour objectif de définir le cadre global du projet de développement de l'application mobile de gestion des relations entre avocats et clients. Il s'agit de situer le projet dans son contexte en présentant les raisons qui ont motivé sa création, les enjeux actuels du domaine juridique, ainsi que les objectifs spécifiques que cette solution vise à atteindre. Nous aborderons également les principales parties prenantes du projet, les besoins auxquels l'application répond, ainsi que les principales fonctionnalités proposées. Ce chapitre permettra ainsi d'offrir une vue d'ensemble sur le projet avant d'entrer dans les aspects techniques et fonctionnels dans les sections suivantes.

2. Présentation de 6SOLUTIONS



Figure 1 : Logo de la société



Figure 2 : Adresse de la société

6SOLUTIONS c'est une entreprise marocaine spécialisée dans la recherche et le développement de solutions informatiques. Le choix de 6SOLUTIONS comme entreprise de services informatiques, n'est pas dû à un hasard. C'est de part son importance et ses performances professionnelles dans le développement informatique. C'est une entreprise qui a construit une expertise unique auprès des plus grandes entreprises privées comme des institutions publiques, et a professionnalisé le métier de la Transformation pour en faire l'outil de la liberté économique et entrepreneuriale de l'entreprise. Elle est spécialisée dans divers domaines tels que l'agriculture, la télécommunication, l'énergie renouvelable etc. Parmi ses services il y a consulting, développement, installation, formation/assistance, maintenance.

3. Etude de l'existant

3.1. Description de l'existant

Actuellement, au sein de l'entreprise, la gestion des relations entre avocats et clients repose sur des outils disparates et non centralisés. Les avocats gèrent leurs profils, rendez-vous, et consultations de manière manuelle ou via des outils classiques comme des calendriers personnels (Google Calendar, Outlook) ou des systèmes CRM basiques. Les consultations sont planifiées par téléphone ou email, sans synchronisation automatique entre les outils utilisés. Cette approche rend difficile le suivi précis des dossiers, la gestion des créneaux horaires, et la mise à jour des informations sur les consultations.

En ce qui concerne les clients, ils doivent souvent chercher des avocats en utilisant des annuaires en ligne, des recommandations personnelles, ou des recherches locales, ce qui peut prendre du temps et ne garantit pas toujours qu'ils trouveront un avocat correspondant parfaitement à leurs besoins. L'absence d'un outil qui centralise ces opérations complique la gestion des affaires et réduit la fluidité des interactions entre avocats et clients.

3.2. Critique de l'existant

Les insuffisances de l'approche actuelle se manifestent à plusieurs niveaux :

- **Manque de centralisation** : Les outils actuellement utilisés sont fragmentés, et il n'existe pas de plateforme unique pour gérer les profils, les consultations, et la communication entre avocats et clients.
- **Gestion inefficace des consultations** : La planification des rendez-vous se fait de manière manuelle, entraînant des erreurs ou des oublis. Il n'y a pas de suivi en temps réel de l'état des consultations (en attente, complétée, annulée), ce qui complique la gestion du temps pour les avocats.
- **Communication dispersée** : Les échanges avec les clients passent par des canaux multiples (téléphone, email, etc.), sans aucune traçabilité claire des discussions ou des documents échangés.
- **Absence de suivi client** : Les clients n'ont pas accès à des informations en temps réel sur le statut de leur dossier ou la disponibilité de leur avocat, ce qui entraîne une frustration et une perte de temps des deux côtés.

Ces insuffisances montrent qu'il est nécessaire de mettre en place une solution plus structurée et automatisée, qui permettrait de centraliser les processus et d'améliorer la gestion des consultations.

3.3. Solution proposée

Face aux lacunes identifiées, la solution proposée consiste à développer une **application mobile** dédiée, centralisant la gestion des relations avocat-client. Cette solution vise à remédier aux problèmes actuels en introduisant plusieurs fonctionnalités clés :

- **Centralisation des profils d'avocats et de clients** : Les avocats pourront créer et gérer leurs profils, tandis que les clients pourront consulter ces profils pour choisir un avocat selon leurs besoins spécifiques.
- **Gestion des consultations** : L'application offrira la possibilité de planifier et de suivre les consultations en temps réel, avec un statut clair (en attente, complétée, annulée), et des rappels automatiques pour les rendez-vous.
- **Communication centralisée** : Un système de messagerie interne permettra aux avocats et aux clients d'échanger facilement et de manière sécurisée, avec une traçabilité de tous les échanges.

Les avantages de cette solution incluent une meilleure organisation des affaires juridiques, une réduction des erreurs humaines et des oublis, et une amélioration de la satisfaction client grâce à une communication plus fluide. Un inconvénient potentiel est le besoin de formation des utilisateurs pour s'adapter à cette nouvelle plateforme, mais cela sera compensé par les gains d'efficacité à long terme.

Cette solution numérique permettrait donc de moderniser le processus actuel et d'automatiser plusieurs tâches, garantissant une meilleure gestion des consultations et des interactions entre les avocats et leurs clients.

4. Choix de modèle de développement

Pour le développement de notre application mobile, le **modèle en cascade** a été sélectionné en raison de sa structure linéaire et séquentielle, qui s'est avérée adaptée à notre approche. Le modèle en cascade repose sur une progression étape par étape, où chaque phase du développement doit être achevée avant de passer à la suivante. Cette méthodologie convient particulièrement aux projets où les exigences sont bien définies dès le départ, avec peu de modifications prévues en cours de développement.

Dans le cadre de ce projet, chaque fonctionnalité de l'application a été développée de manière isolée. Nous avons d'abord terminé la conception de chaque module avant de passer à l'implémentation du suivant. Ce processus a permis une progression méthodique, facilitant ainsi une gestion claire des tâches et une validation rigoureuse à chaque étape. Le modèle en cascade offre également l'avantage de bien documenter chaque phase du projet, assurant une traçabilité claire des exigences et des résultats. Il est particulièrement adapté pour des projets où la planification initiale est stable, comme dans le cas de notre application, et où les différentes parties du système peuvent être développées et testées de manière indépendante.

Cette approche séquentielle nous a permis de garantir que chaque fonctionnalité livrée soit bien conçue et testée avant d'entamer le développement de la suivante, assurant ainsi une qualité continue tout au long du processus.

5. Planning prévisionnel

Le planning prévisionnel détaille les différentes étapes du développement de l'application tout au long de la période de stage. Les étapes suivantes sont prévues :

| Semaine | Mai | Juin | Juillet | Aout | Septembre |
|--------------------|-----|------|---------|------|-----------|
| Etude préalable | X | | | | |
| Conception | | X | | | |
| Réalisation | | | X | X | |
| Test et validation | | | | X | X |

Tableau 1 : Planning prévisionnel

Description des étapes :

- **Étude préalable** : Cette étape consiste à analyser les besoins des utilisateurs, à comprendre les spécificités du domaine juridique, et à identifier les fonctionnalités nécessaires pour l'application. Cela inclut des recherches sur les exigences réglementaires et les attentes des avocats et des clients.
- **Conception** : Durant cette phase, l'architecture de l'application sera définie, ainsi que les technologies à utiliser (Angular pour le frontend, Spring Boot pour le backend, PostgreSQL pour la base de données). La conception comprendra également la définition des interfaces utilisateur, des schémas de base de données, ainsi que la modélisation à l'aide d'UML (diagrammes de classes, de séquence, et de cas d'utilisation). Cela permettra de visualiser la structure et les interactions du système de manière détaillée.
- **Réalisation** : La phase de réalisation implique le développement de l'application en suivant les spécifications et l'architecture définies. Les fonctionnalités clés, telles que la gestion des consultations et des profils, seront développées, testées en interne et intégrées progressivement.
- **Test et validation** : Enfin, des tests unitaires et d'intégration seront effectués pour garantir que l'application répond aux exigences initiales. Cette phase comprendra également la validation auprès des utilisateurs finaux, avec des corrections apportées en fonction des résultats des tests. Ce planning permet de structurer le travail de manière efficace et de garantir l'atteinte des objectifs du projet dans les délais impartis.

6. Conclusion

Ce chapitre a marqué le début de l'élaboration de mon projet en présentant d'abord l'organisme d'accueil, 6SOLUTIONS, qui a joué un rôle clé dans le développement de l'application. Nous avons ensuite réalisé une étude approfondie de l'existant, en identifiant les lacunes et les problématiques actuelles dans la gestion des relations entre avocats et clients. Cette analyse a permis de cerner les besoins et de définir les objectifs spécifiques du projet.

Nous avons également décrit le déroulement du stage, en exposant les différentes tâches accomplies et les étapes de développement prévues. Cette présentation a permis de structurer le travail de manière claire et de préparer le terrain pour les phases suivantes.

Dans le chapitre suivant, nous nous pencherons sur l'étude conceptuelle de l'application, en détaillant les aspects techniques et fonctionnels qui guideront le développement ultérieur du projet.

Chapitre

2

**Spécification
des besoins**

1. Introduction

Ce chapitre a pour but de présenter les spécifications fonctionnelles et non fonctionnelles de l'application mobile, ainsi que les différents cas d'utilisation qui en découlent. Cette spécification permet de s'assurer que l'application réponde efficacement aux besoins des utilisateurs, tout en garantissant la sécurité, la performance et l'accessibilité.

2. Spécification des besoins fonctionnels

Les besoins fonctionnels décrivent les fonctionnalités principales que l'application doit offrir aux utilisateurs.

2.1. Besoin fonctionnel global 1 : Gestion des comptes utilisateurs

L'application doit permettre la gestion des comptes pour les deux types d'utilisateurs : avocats et clients.

2.1.1. Sous-besoin : Création de compte

- Chaque utilisateur doit pouvoir créer un compte avec des informations spécifiques : nom, prénom, adresse e-mail, mot de passe, rôle (client ou avocat).
- Les avocats doivent aussi pouvoir indiquer leur spécialité et leur localisation.

2.1.2. Sous-besoin : Connexion et authentification

- Les utilisateurs doivent pouvoir se connecter à l'application en utilisant leur adresse e-mail et leur mot de passe.
- Le système doit assurer une authentification sécurisée.

2.1.3. Sous-besoin : Mise à jour du profil

- Les utilisateurs doivent pouvoir modifier leurs informations personnelles et professionnelles à tout moment

2.2. Besoin fonctionnel global 2 : Recherche et consultation

L'application doit permettre aux clients de rechercher et de consulter les profils des avocats disponibles.

2.2.1. Sous-besoin : Recherche d'avocat

- Les clients doivent pouvoir rechercher des avocats par spécialité, localisation ou nom. Un système de filtrage et de tri doit être proposé.

2.2.2. Sous-besoin : Consultation des profils

- Les clients peuvent consulter les profils détaillés des avocats, y compris leurs informations de contact, leur expérience, et leurs avis clients.

2.3. Besoin fonctionnel global 3 : Gestion des consultations

L'application doit faciliter la gestion des consultations entre les clients et les avocats.

2.3.1. Sous-besoin : Prise de rendez-vous

- Les clients doivent pouvoir prendre un rendez-vous avec l'avocat de leur choix, en fonction des disponibilités affichées par l'avocat.

2.3.2. Sous-besoin : Suivi des consultations

- Les consultations doivent être suivies en temps réel (en attente, en cours, complétée, annulée). Les utilisateurs doivent être notifiés de tout changement.

2.3.3. Sous-besoin : Messagerie intégrée

- Une fonctionnalité de messagerie doit être disponible pour permettre la communication entre avocats et clients.

3. Spécification des besoins non fonctionnels

Les besoins non fonctionnels garantissent la qualité du service rendu par l'application.

- **Sécurité**

- L'application doit garantir la sécurité des données personnelles via des méthodes de chiffrement et d'authentification.

- **Performance**

- L'application doit assurer des temps de réponse rapides et être capable de gérer un grand nombre d'utilisateurs simultanés sans dégradation des performances.

- **Ergonomie et accessibilité**

- L'interface utilisateur doit être simple à utiliser, intuitive et accessible à tous, quel que soit le type de terminal mobile utilisé.

- **Disponibilité**

- L'application doit être disponible en permanence, avec une tolérance minimale pour les interruptions de service.

4. Présentation des cas d'utilisation

4.1. Présentation des acteurs

Les deux principaux acteurs de l'application mobile de mise en relation entre avocats et clients sont les suivants :

- **Client** : Une personne à la recherche de services juridiques. Elle peut s'inscrire, rechercher des avocats, consulter leurs profils, prendre rendez-vous, et suivre ses consultations.
- **Avocat** : Un professionnel fournissant des services juridiques. Il peut créer un profil, gérer ses disponibilités, accepter des rendez-vous, et consulter les demandes de consultation des clients.

4.2. Description des cas d'utilisation

A ce niveau, les cas d'utilisation peuvent être présentés par deux façons que ce soit par acteur, soit par fonctionnalité. Généralement, si les fonctions des acteurs sont complètement indépendantes, c'est la première solution qui est adoptée. Si en revanche, une fonctionnalité du système fait intervenir plusieurs acteurs, c'est la deuxième possibilité qui est adoptée.

Les cas d'utilisation présentant certaines ambiguïtés doivent être complétés par une description textuelle en présentant les points suivants :

| | |
|----------------------------|--|
| Cas n° | <i>1</i> |
| Acteur(s) : | <i>Client</i> |
| Objectif : | <i>C'est le but du cas d'utilisation.</i> |
| Pré-condition(s) : | <i>Permettre à un utilisateur de s'inscrire sur la plateforme en tant que client</i> |
| Post-condition(s) : | <i>L'utilisateur ne doit pas déjà avoir un compte</i> |
| Scénario nominal : | <i>- Le client accède au profil de l'avocat.</i> |

Tableau 2 : Cas d'utilisation n°1 : S'inscrire en tant que client

- Il saisit ses informations personnelles (nom, prénom, adresse e-mail, mot de passe).
- Il choisit le rôle "Client".
- L'utilisateur clique sur "S'inscrire".
- Le système enregistre les informations et crée un compte.
- Un message de confirmation s'affiche. || **Scénario alternatif** | - Si l'adresse e-mail est déjà utilisée, un message d'erreur s'affiche. |

| | |
|----------------------------|--|
| Cas n° | <i>2</i> |
| Acteur(s) : | <i>Client, Avocat</i> |
| Objectif : | <i>Permettre à un client de prendre un rendez-vous avec un avocat.</i> |
| Pré-condition(s) : | <i>Le client doit être connecté à son compte, et l'avocat doit avoir des créneaux disponibles.</i> |
| Post-condition(s) : | <i>Le rendez-vous est enregistré et visible par le client et l'avocat.</i> |
| Scénario nominal : | <i>- Le client accède au profil de l'avocat</i> |

Tableau 3 : Cas d'utilisation n°2 : Prendre un rendez-vous avec un avocat

- Il consulte les disponibilités de l'avocat.
- Le client sélectionne un créneau horaire, puis confirme le rendez-vous.
- Le système envoie la demande de rendez-vous à l'avocat.
- L'avocat valide le rendez-vous.
- Le client reçoit une confirmation du rendez-vous. || **Scénario alternatif** | - Si l'avocat n'a pas de disponibilités, un message d'erreur est affiché. |

4.3. Diagramme des cas d'utilisation global

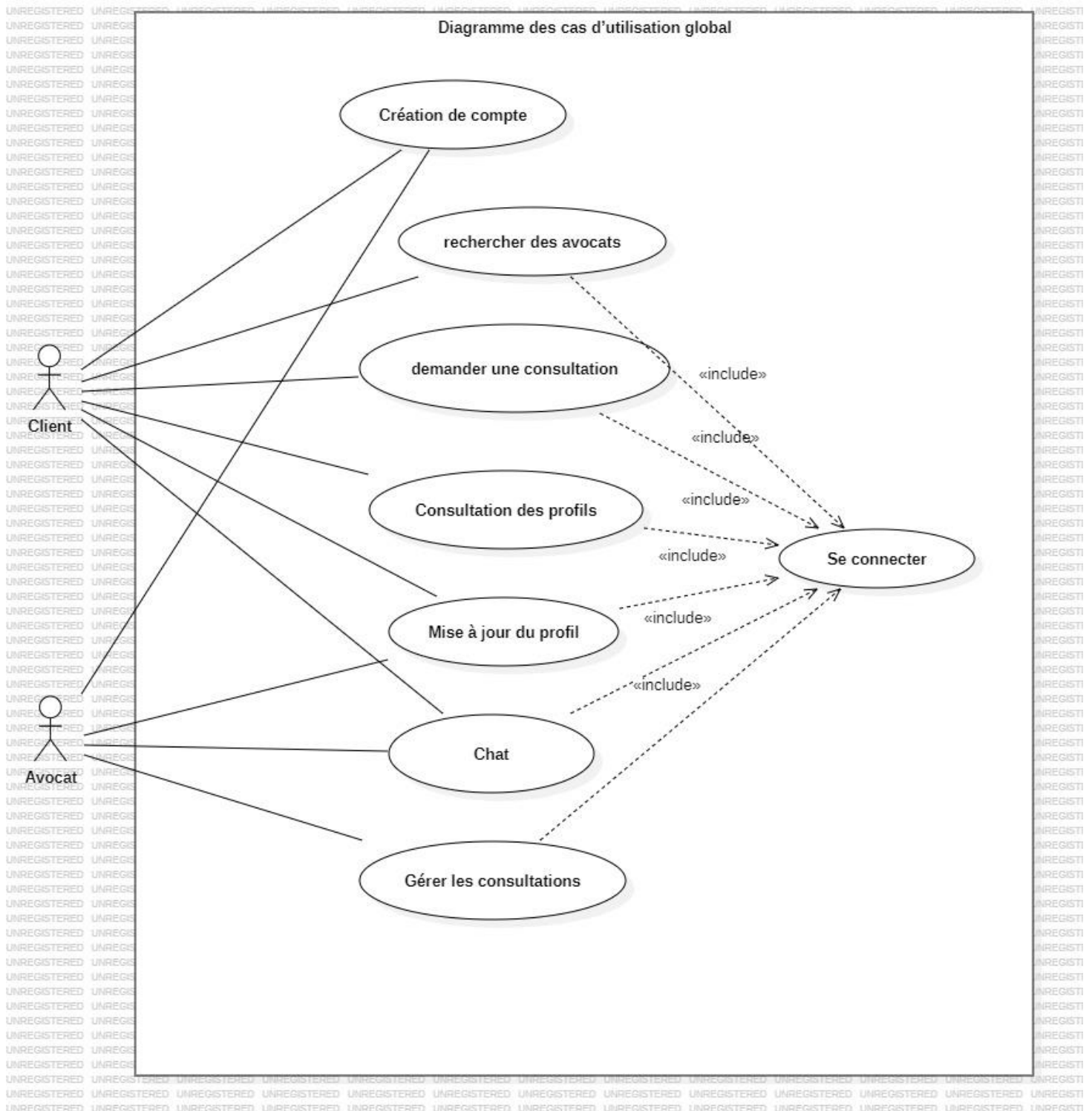


Figure 3 : Diagramme des cas d'utilisation global

Le diagramme global des cas d'utilisation illustre les interactions entre les deux acteurs principaux (client et avocat) et les fonctionnalités essentielles de l'application, telles que l'inscription, la prise de rendez-vous, la gestion des consultations, et la mise à jour des profils.

Chapitre

3

**Conception
du système**

1. Introduction

Ce chapitre a pour objectif de détailler la conception du système de l'application mobile dédiée à la gestion des relations entre avocats et clients. Il s'agit de modéliser l'application à la fois d'un point de vue statique et dynamique, afin de fournir une représentation claire de l'architecture du système et de ses interactions internes.

D'une part, la modélisation dynamique permettra de visualiser le comportement du système à travers des diagrammes tels que les diagrammes de séquences, de collaboration et d'états. Ces diagrammes mettront en lumière les échanges entre les différents composants du système, ainsi que les transitions d'états au cours du cycle de vie de l'application.

D'autre part, la modélisation statique présentera la structure du système via le diagramme de classes, qui décrira les relations entre les différentes entités du projet, et le diagramme de déploiement, qui illustrera la répartition des composants sur l'infrastructure matérielle.

Cette étape de conception est cruciale, car elle constitue la base de la réalisation technique du projet et permet de garantir la cohérence et l'efficacité du système tout au long du développement.

2. Modélisation dynamique

2.1. Diagrammes de navigation

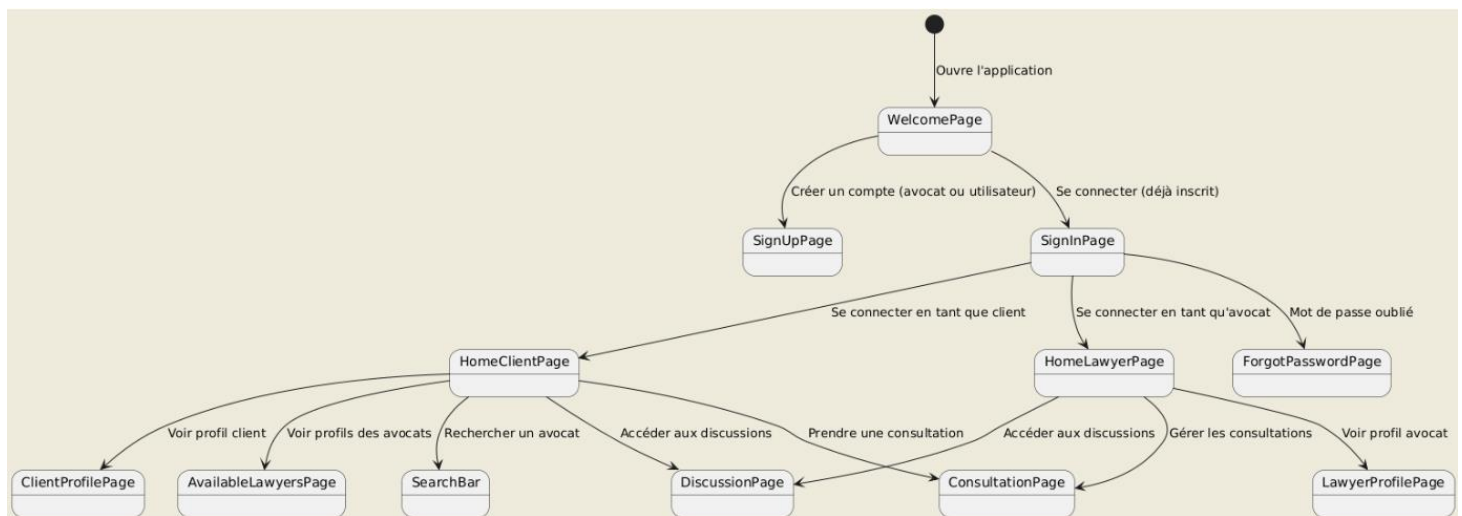


Figure 4 : Diagramme de navigation

2.2. Diagrammes de séquence

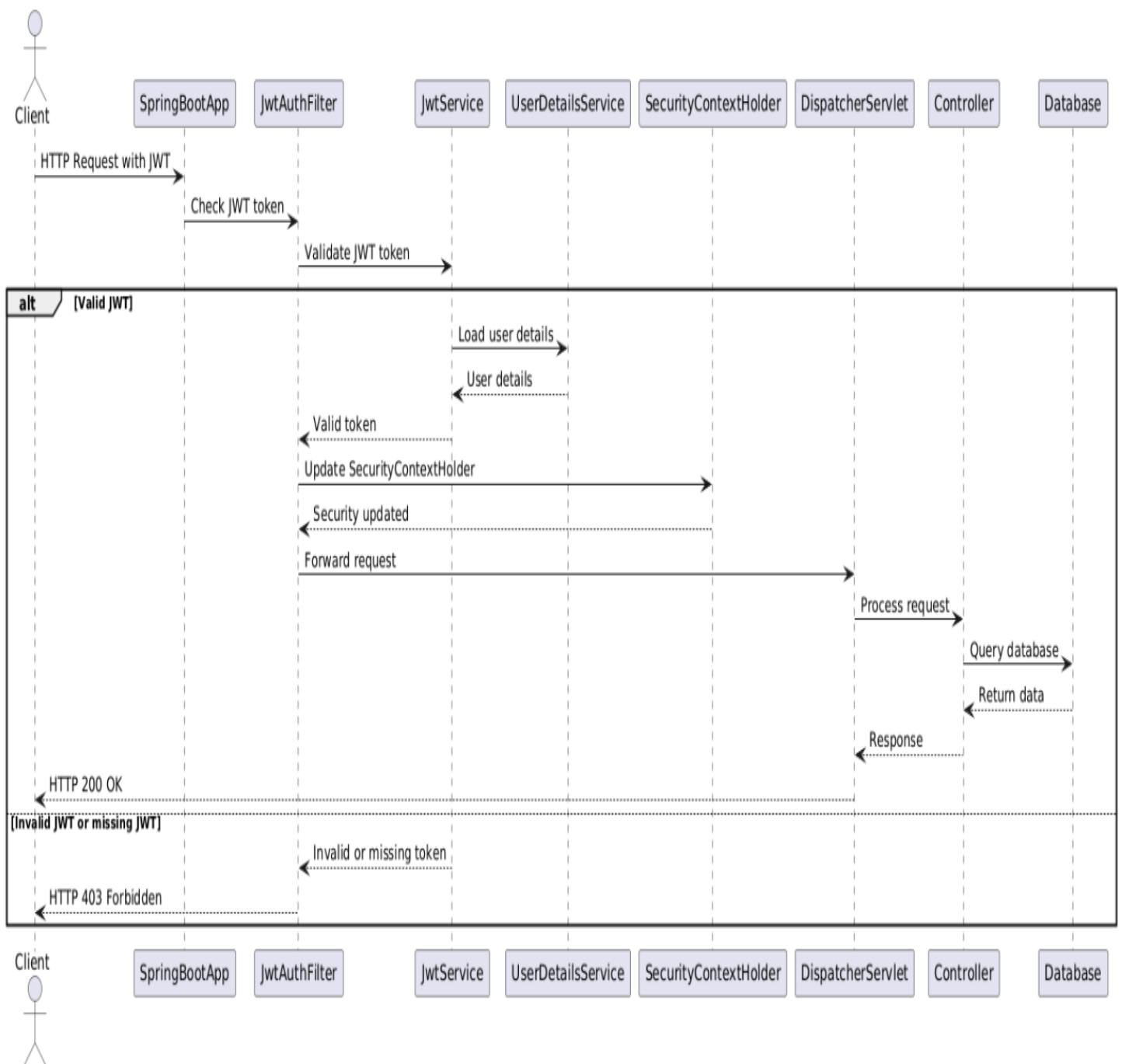


Figure 5 : Diagramme de séquence

3. Modélisation statique

3.1. Diagramme de classes

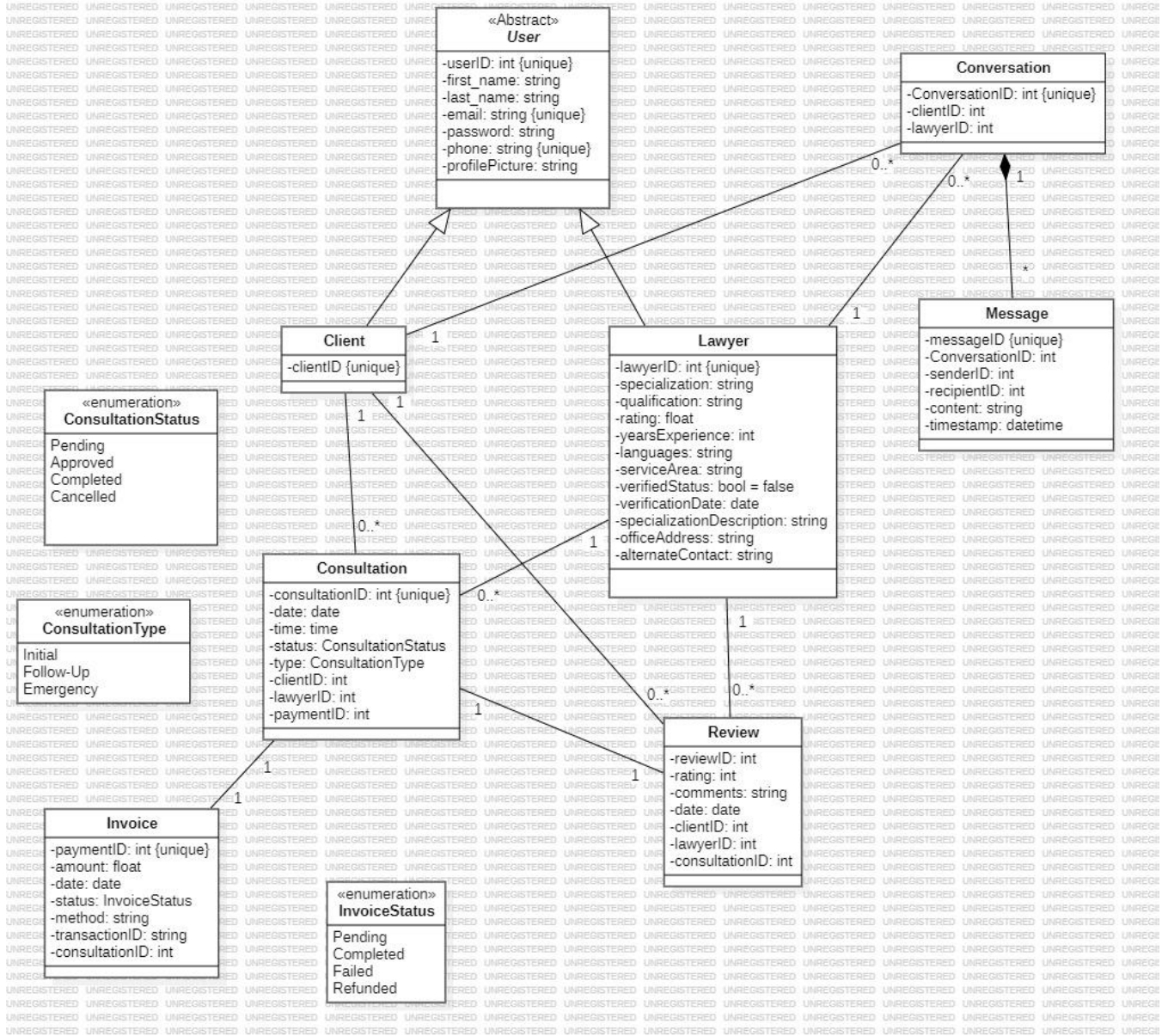


Figure 6 : Diagramme de classe

3.2. Architecture de l'application

3.2.1. Architecture logiciel

L'architecture logicielle de l'application mobile pour les avocats et les clients, développée avec **Flutter** pour l'interface utilisateur et utilisant **Spring Boot** pour le backend, est pensée pour être scalable, modulaire et maintenable. Voici les principales composantes de cette architecture :

1. Couche de présentation (Front-end) :

- **Technologies** : Flutter, Dart.
- **Responsabilités** : Cette couche gère l'interface utilisateur (UI) et permet aux utilisateurs (clients et avocats) d'interagir avec l'application. Elle est responsable de la navigation entre les écrans et de la gestion des interactions utilisateur via des widgets Flutter.
- **Exemples** : Interfaces pour la création de comptes, la gestion des profils d'avocats, la recherche d'avocats par les clients, et la gestion des consultations.

2. Couche de service (Back-end) :

- **Technologies** : Spring Boot, REST API, WebSocket.
- **Responsabilités** : Cette couche contient la logique métier de l'application. Elle traite les requêtes provenant du front-end, applique les règles métiers, et interagit avec la base de données pour stocker et récupérer des informations telles que les profils d'avocats, les consultations, et les interactions entre clients et avocats. En plus des requêtes classiques via l'API REST, **WebSocket** permet une communication bidirectionnelle en temps réel pour certaines fonctionnalités, comme la messagerie instantanée.
- **Exemples** : Services pour l'enregistrement et la gestion des consultations, l'authentification, la gestion des profils, ainsi que l'intégration de **WebSocket** pour la messagerie en temps réel entre avocats et clients.

3. Couche de persistance (Base de données) :

- **Technologies** : PostgreSQL.
- **Responsabilités** : Cette couche est responsable de la gestion des données persistantes. Elle stocke les informations relatives aux utilisateurs (clients et avocats), aux consultations, et aux transactions, en utilisant un modèle relationnel.

- **Exemples** : Structures de tables pour les utilisateurs, les consultations, les statuts de consultations (en attente, complétée, annulée).

4. Couche de sécurité :

- **Technologies** : Spring Security, JWT (JSON Web Token).
- **Responsabilités** : La couche de sécurité assure l'authentification et la protection des données sensibles. Elle s'assure que seules les personnes autorisées peuvent accéder aux informations ou fonctionnalités réservées.
- **Exemples** : Authentification par JWT, gestion des rôles (avocat, client), et contrôle d'accès aux services REST.

3.2.2. Architecture matériel

L'architecture matérielle de l'application mobile pour les avocats et les clients est conçue pour garantir la disponibilité, la fiabilité, et la performance de l'application. Voici les principaux composants matériels impliqués :

1. Appareils clients :

- **Responsabilités** : Les smartphones et tablettes des utilisateurs (avocats et clients) exécutent l'application mobile développée avec Flutter. Ces appareils permettent l'interaction avec le système et l'accès aux fonctionnalités telles que la gestion des consultations et la communication entre avocats et clients.
- **Spécifications** : Smartphones ou tablettes compatibles avec les systèmes iOS et Android, capables de supporter les versions minimales de Flutter et disposant d'une connexion internet stable.

2. Serveur backend :

- **Responsabilités** : Héberge l'API REST développée avec Spring Boot, gérant les requêtes des utilisateurs et la logique métier de l'application. Le serveur est aussi responsable du stockage des données des utilisateurs, des consultations et des transactions.
- **Spécifications** : Serveur cloud avec des capacités de haute disponibilité, capable de supporter des bases de données relationnelles (PostgreSQL) et d'exécuter l'API de manière fluide et évolutive.

3. Réseau :

- **Composants** : Connexions internet fiables pour les utilisateurs finaux.
- **Responsabilités** : Assure la communication fluide entre l'application front-end sur les appareils des utilisateurs et le serveur backend, tout en maintenant la sécurité et la stabilité des connexions. La qualité du réseau est essentielle pour l'accès aux services en ligne, aux données utilisateur, et à la gestion des consultations en temps réel.
- **Exemples** : Connexions via Wi-Fi ou données mobiles (4G/5G) pour une communication rapide et sécurisée avec les services backend.

4. Infrastructure cloud :

- **Services** : PostgreSQL (pour la gestion des bases de données), Spring Boot (API), Redis (pour la gestion du cache et des sessions), et un service cloud pour l'hébergement.
- **Responsabilités** : Héberge les composants backend et gère le traitement des données, la persistance, et l'authentification des utilisateurs. L'infrastructure cloud est dimensionnée pour une montée en charge automatique, offrant une disponibilité continue et des performances optimisées.

4. Conclusion

Dans ce chapitre, nous avons présenté le cadre du projet de développement de l'application mobile pour avocats et clients. Après avoir décrit l'entreprise d'accueil et son fonctionnement, nous avons analysé l'existant et mis en lumière les lacunes des systèmes actuels. Cette analyse a permis de justifier la nécessité d'une nouvelle solution technologique pour améliorer la gestion des consultations et la communication entre avocats et clients. Enfin, la solution proposée a été détaillée, en mettant l'accent sur les avantages qu'elle offre en termes d'efficacité, d'accessibilité et de flexibilité pour les utilisateurs.

Chapitre

4

**Réalisation
du système**

1. Introduction

Ce chapitre détaille la réalisation de notre application mobile pour avocats et clients, en se concentrant sur le processus de développement et la mise en œuvre des fonctionnalités clés. Nous aborderons les différentes étapes techniques, de l'intégration des interfaces utilisateur à la gestion des consultations et des profils, tout en assurant que le système réponde aux exigences de performance et de sécurité. L'objectif est de démontrer comment les concepts de conception ont été traduits en une solution fonctionnelle et opérationnelle, prête à être utilisée par les utilisateurs finaux.

2. Environnement de développement

2.1. Environnement matériel

L'application mobile pour avocats et clients a été développée dans un environnement matériel composé d'un ordinateur portable performant. Voici les spécifications matérielles utilisées durant le développement :

- **Nom de l'appareil** : DESKTOP-4INPTBQ
- **Modèle** : LENOVO ideapad 330
- **Processeur** : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
- **Mémoire vive (RAM)** : 12 Go DDR4
- **Disque dur** : SSD de 1 T pour une vitesse accrue de traitement des données
- **Système d'exploitation** : Windows 10 Professionnel
- **Moniteur** : Écran de 15,6 pouces Full HD
- **Appareil mobile de test** : Smartphone Android (OPPO Reno2 F) pour les tests d'application en condition réelle

Ces caractéristiques ont permis un développement fluide de l'application, une exécution rapide des environnements de développement, et un test efficace de l'application sur mobile.

2.2. Environnement logiciel

Le développement de l'application a été réalisé à l'aide de plusieurs outils logiciels, essentiels pour la programmation, la gestion de bases de données, et la modélisation. Les principaux logiciels utilisés sont :

- **IDE (Environnement de Développement Intégré) :**

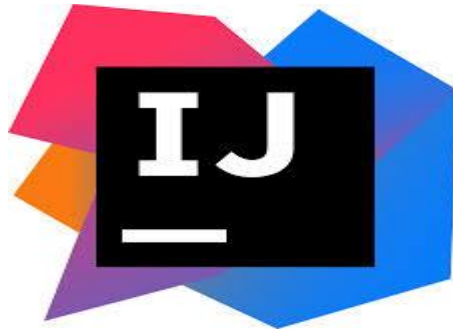


Figure 7 : Logo IntelliJ Idea

- **IntelliJ IDEA '1'** est un environnement de développement intégré (IDE) avancé, principalement utilisé pour le développement en Java, offrant des outils intelligents et des fonctionnalités pour un codage efficace.



Figure 8 : Logo Android Studio

- **Android Studio '2'** pour le développement de l'application mobile en utilisant Flutter (Dart).

- **Frontend :**



Figure 9 : Logo Flutter

- **Flutter ‘3’** est un framework open source de Google qui permet de créer des applications multiplateformes (Android, IOS, web, desktop) à partir d'un seul code source en utilisant le langage Dart.



Figure 10 : Logo Dart

- **Dart ‘4’** pour le développement du front-end de l'application mobile avec Flutter.

- **Backend :**



Figure 11 : Logo SpringBoot Tomcat

- **Spring Boot ‘5’** pour le développement des services backend, notamment pour la gestion des requêtes côté serveur.
- **Tomcat** est un serveur web open source développé par Apache qui permet d'exécuter des applications Java servlets et JSP.



Figure 12 : Logo Java

- **Java ‘6’** est un langage de programmation orienté objet, polyvalent et indépendant de la plateforme, utilisé pour développer des applications web, mobiles, et de bureau.

- **Base de données :**



Figure 13 : Logo Redis

- **Redis ‘10’** est une base de données NoSQL en mémoire, open-source, utilisée pour le stockage clé-valeur, les caches, et la gestion de sessions avec des performances très rapides.

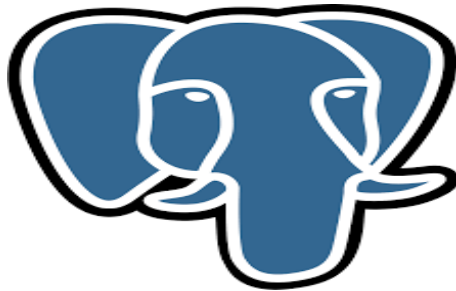


Figure 14 : Logo PostgreSQL

- **PostgreSQL ‘7’** pour la gestion des données des utilisateurs et des consultations.



Figure 15 : Logo pgAdmin 4

- **pgAdmin 4** est un outil open-source de gestion et d'administration pour les bases de données PostgreSQL, offrant une interface web intuitive pour la gestion des données et des requêtes.

- **Systèmes de gestion des versions :**

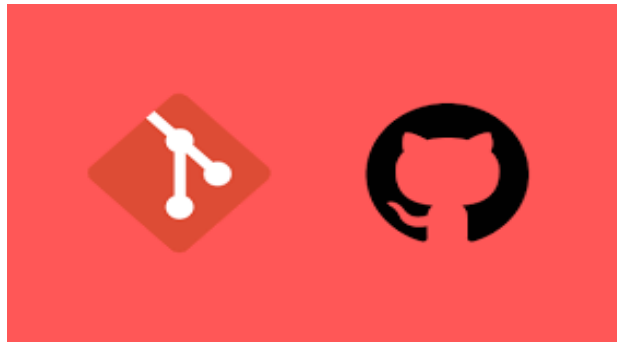


Figure 16 : Logo Git & GitHub

- **Git ‘8’** est un système de contrôle de version distribué qui permet de suivre les modifications du code et de collaborer efficacement sur des projets
- **GitHub ‘9’** est une plateforme en ligne qui s'appuie sur Git, offrant des fonctionnalités pour héberger des projets, collaborer à distance et gérer le code source à travers des dépôts publics ou privés.

- **Outils de modélisation :**



Figure 17 : Logo StarUML

- **StarUML** pour la modélisation des différents diagrammes UML (cas d'utilisation, classes, séquence).

Ces outils logiciels ont permis d'assurer une conception et un développement structuré de l'application, en garantissant la qualité du code, la gestion des versions, et la création des différents diagrammes de conception pour visualiser l'architecture de l'application.

3. Principales interfaces graphiques

- **Interface Welcome Page :**

La première interface de l'application qui accueille l'utilisateur. Elle propose deux options : se connecter (SignIn) ou créer un compte (Sign Up). C'est le point de départ pour accéder aux autres fonctionnalités de l'application.

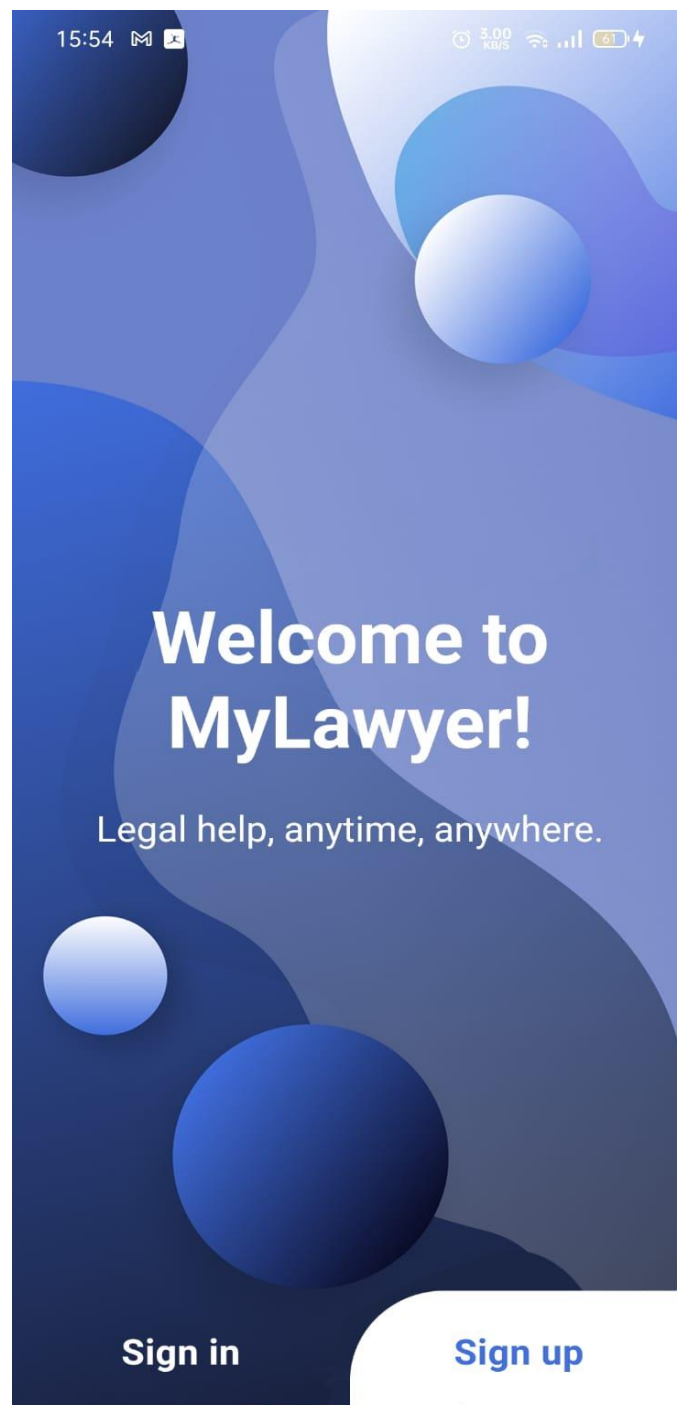
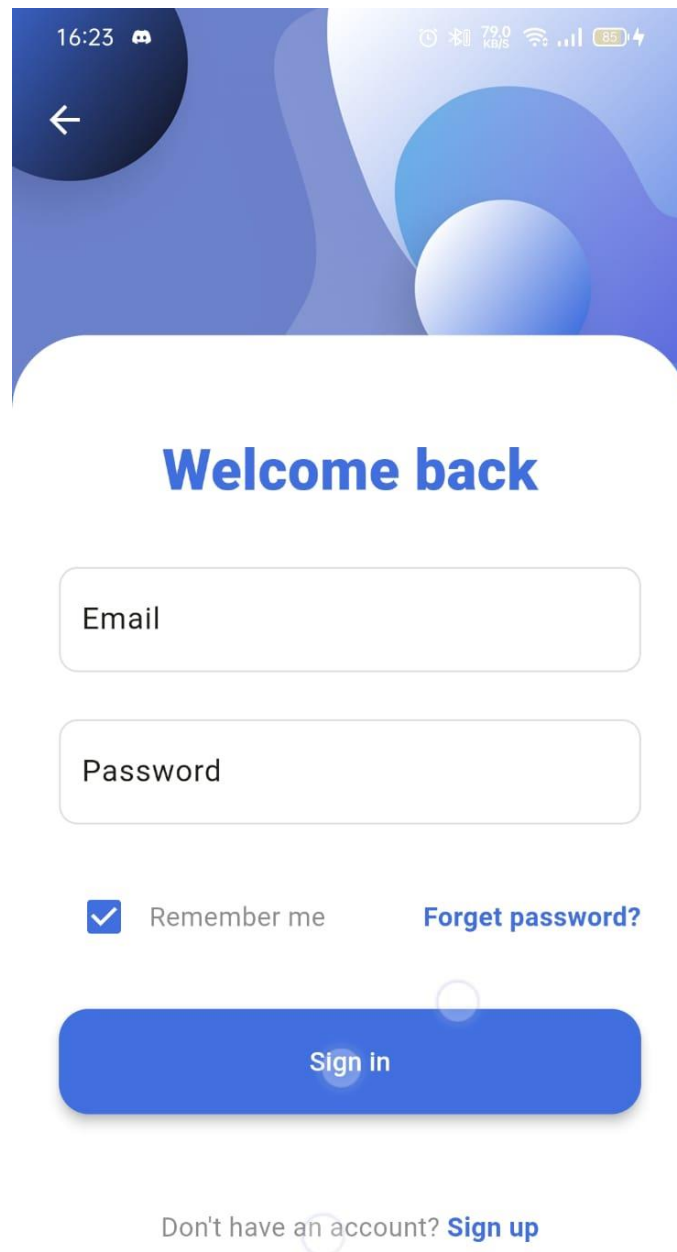


Figure 18 : Interface Welcome Page

- **Interface Sign In :**

Cette interface permet aux utilisateurs, qu'ils soient avocats ou clients, de se connecter en saisissant leurs identifiants (email et mot de passe). Elle inclut également un lien vers l'interface "Forgot Password" pour réinitialiser le mot de passe en cas d'oubli.

A mobile application interface for signing in. The background is a blue abstract design with overlapping circles. At the top, there's a status bar with the time 16:23, a back arrow, and various system icons. Below the header, the text "Welcome back" is displayed in a bold, blue font. Underneath, there are two white input fields with rounded corners, labeled "Email" and "Password". Below the "Email" field, there is a checked checkbox labeled "Remember me" and a blue link labeled "Forgot password?". At the bottom, there is a large blue button with rounded corners labeled "Sign in". Below the button, there is a link that says "Don't have an account? Sign up".

16:23

←

79.0 KB/s

85%

Welcome back

Email

Password

☒ Remember me [Forgot password?](#)

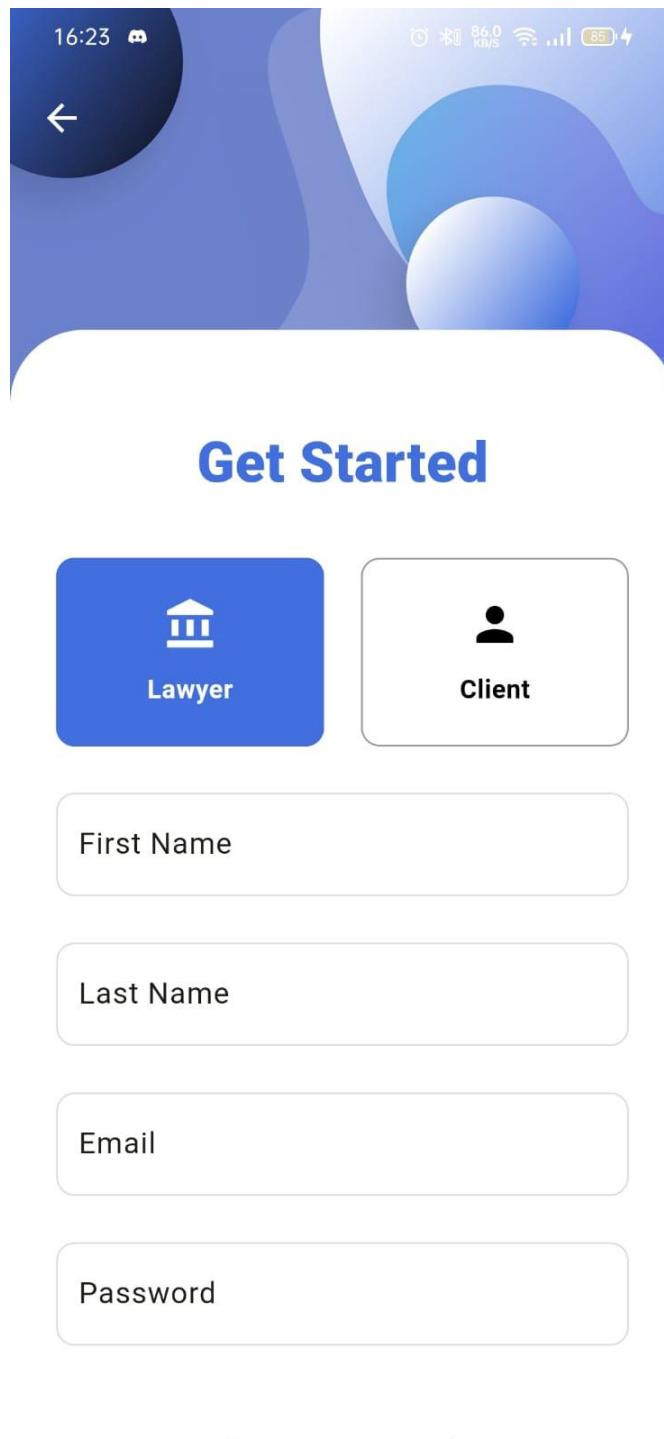
Sign in

Don't have an account? [Sign up](#)

Figure 19 : Interface Sign In

- **Interface Sign Up :**

L'écran d'inscription où les utilisateurs peuvent créer un compte. Les champs varient légèrement en fonction de leur rôle (avocat ou client). Pour les avocats, des informations professionnelles comme la spécialisation et les qualifications sont demandées.




16:23

86.0 KB/s

85%

←

Get Started


Lawyer



Client

Figure 20 : Interface Sign Up

- **Interface Home Client :**

L'interface principale des clients après la connexion. Elle leur permet de consulter les profils d'avocats, de planifier des consultations, d'utiliser la barre de recherche pour trouver des avocats, et d'accéder à la messagerie pour discuter avec eux.

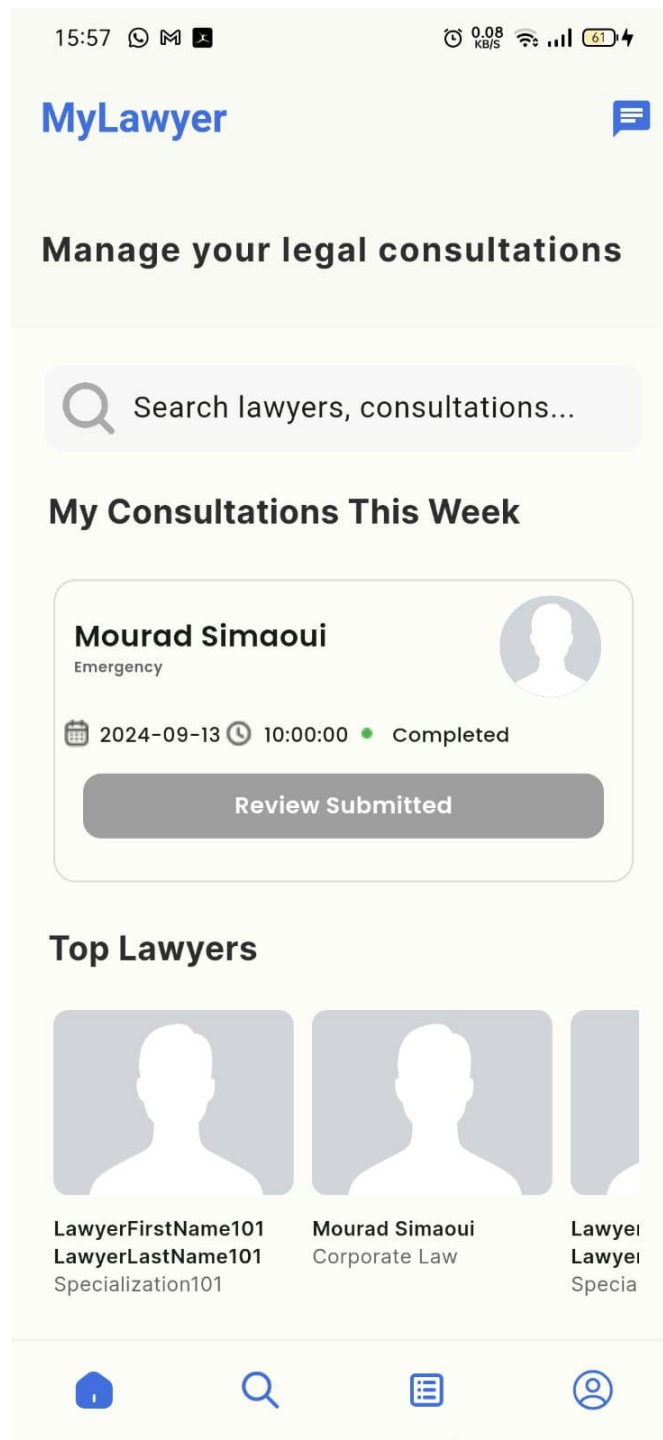


Figure 21 : Interface Home Client

- **Interface Home Avocat :**

L'interface principale des avocats, leur permettant de gérer leurs consultations, de répondre aux messages des clients, et de consulter ou modifier leur profil professionnel.

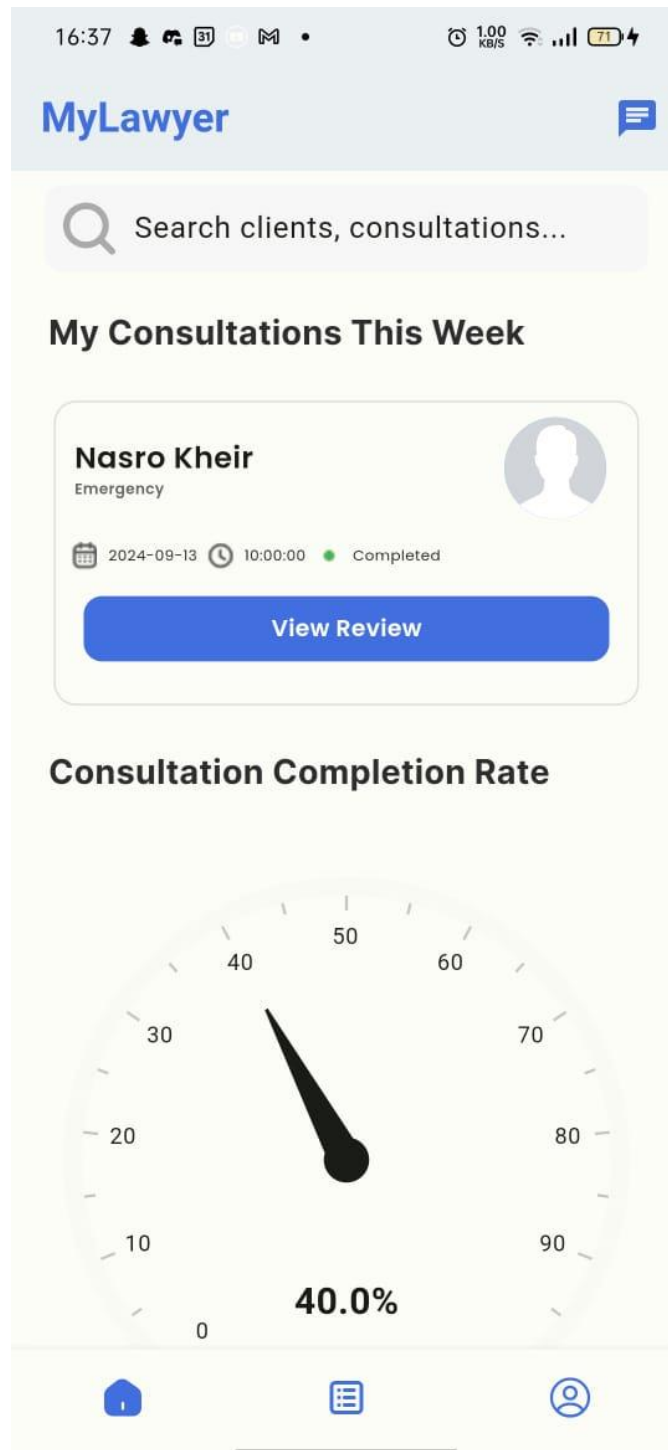
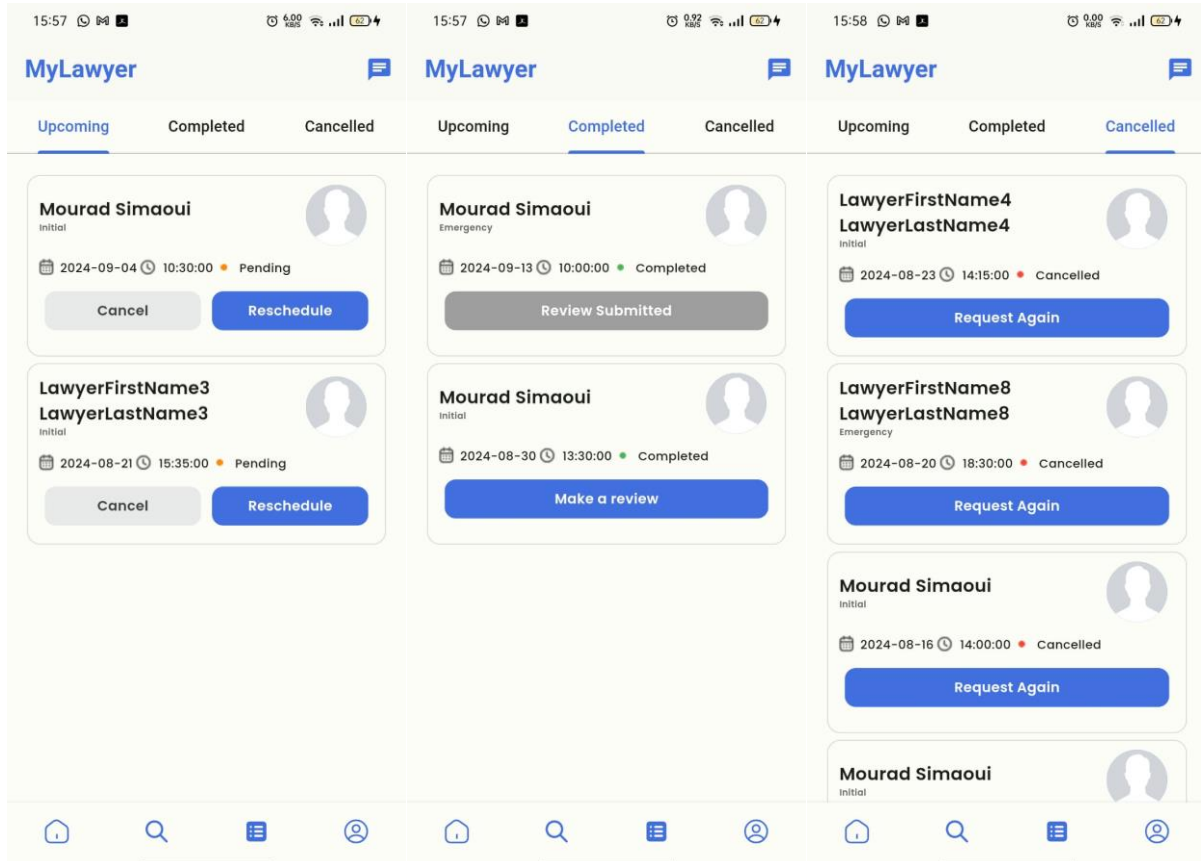


Figure 22 : Interface Home Avocat

- **Interfaces Consultation :**

Un écran où les clients peuvent planifier, consulter l'historique, ou annuler des consultations.



**Figure 23 : Interface Consultation
(Upcoming)**

**Figure 25 : Interfaces Consultation
(Cancelled)**

**Figure 24 : Interfaces Consultation
(Completed)**

- **Interface Messagerie :**

Cette interface permet aux avocats et aux clients de discuter en temps réel. Les utilisateurs peuvent envoyer et recevoir des messages liés à leurs consultations ou à d'autres questions juridiques.

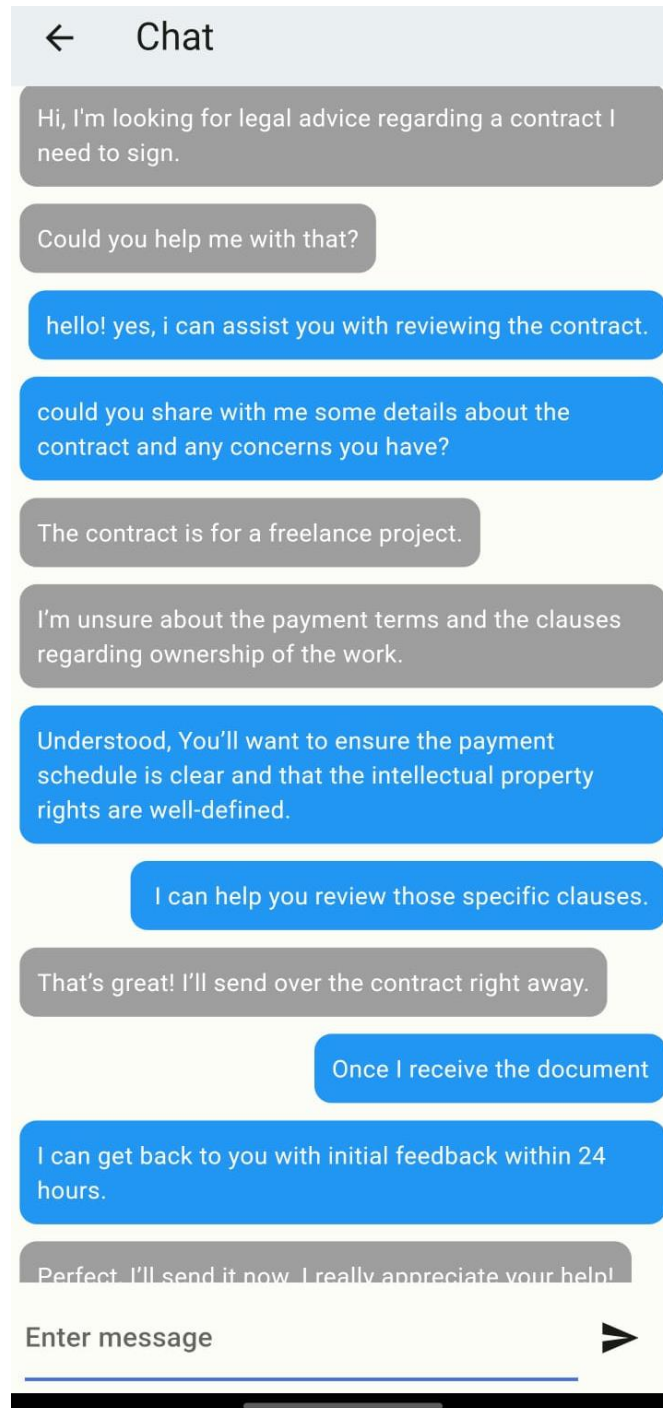


Figure 26 : Interface Messagerie

4. Conclusion

Ce chapitre a présenté en détail l'environnement de développement utilisé pour la création de l'application mobile pour avocats et clients. Nous avons décrit les spécifications matérielles qui ont permis d'assurer des performances optimales durant le processus de développement, ainsi que les différents outils logiciels essentiels à la conception, au codage, et à la gestion des données de l'application. Cet environnement, à la fois matériel et logiciel, a fourni une base solide pour développer une solution performante et répondant aux exigences du projet.

Conclusion générale

Ce rapport a présenté le développement complet de l'application mobile pour avocats et clients, de la phase de conception à la livraison. Le choix des technologies modernes a été essentiel pour répondre aux exigences du projet. Flutter a permis de créer une interface utilisateur fluide et intuitive, assurant une expérience agréable pour les utilisateurs. Spring Boot a fourni une infrastructure backend robuste et évolutive, capable de gérer efficacement les connexions et les données. PostgreSQL a garanti une gestion sécurisée et fiable des informations des utilisateurs et des consultations.

L'application a été conçue pour résoudre les problèmes de communication et de gestion entre avocats et clients en offrant des fonctionnalités telles que la gestion des profils, la recherche d'avocats, et le suivi des consultations. Ces fonctionnalités ont été intégrées pour améliorer la visibilité et le contrôle des interactions entre les utilisateurs.

Les défis techniques, tels que l'optimisation des performances et la sécurité des données, ont été abordés avec succès grâce à l'utilisation des technologies choisies et à une gestion efficace du projet. Ce projet a non seulement permis d'acquérir des compétences en développement mobile et en gestion de projet, mais a aussi apporté une solution innovante au domaine juridique. Bien que l'application soit un produit abouti, elle offre des opportunités d'amélioration continue pour mieux répondre aux besoins futurs des utilisateurs.

Webographie

1. IntelliJ Idea

- Site web : jetbrains.com/idea

2. Android Studio

- Site web : developer.android.com/studio

3. Flutter

- Site web : flutter.dev

4. Dart

- Site web : dart.dev

5. Spring Boot

- Site web : spring.io/projects/spring-boot

6. Java

- Site web : docs.oracle.com/en/java

7. PostgreSQL

- Site web : postgresql.org

8. Git

- Site web : git-scm.com

9. GitHub

- Site web : github.com

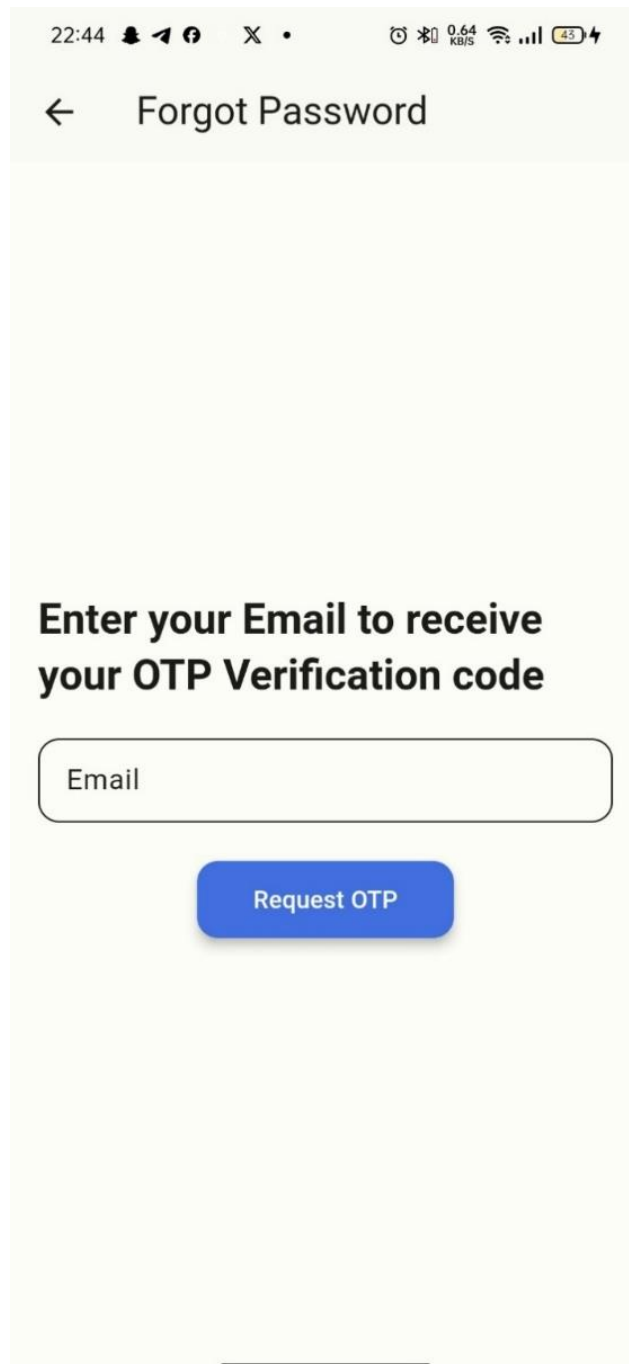
10. Redis

- Site web : <https://redis.io>.

Annexe (A) : Autres Interfaces De l'Application

Interface Forgot Password :

Si l'utilisateur oublie son mot de passe, il peut accéder à cette interface pour le réinitialiser. L'interface demande l'adresse email associée au compte et envoie un lien de réinitialisation.



The screenshot shows a mobile application interface for the 'Forgot Password' screen. At the top, there is a status bar with the time 22:44, various notification icons, and network/battery indicators. Below the status bar is a header bar with a back arrow icon and the text 'Forgot Password'. The main content area has a light yellow background. It features the text 'Enter your Email to receive your OTP Verification code' in bold black font. Below this text is a white rounded rectangular input field with the placeholder text 'Email'. Underneath the input field is a blue rounded rectangular button with the text 'Request OTP' in white. At the very bottom of the screen, there is a thin horizontal line representing the home indicator bar.

Figure 27 : Interface Forgot Password

Interface Profil Avocat :

Les avocats peuvent accéder à leur propre profil via cette interface. Ils y trouvent leurs informations professionnelles et peuvent les modifier, comme leur spécialité, leurs qualifications, leurs tarifs, et leurs horaires de consultation.

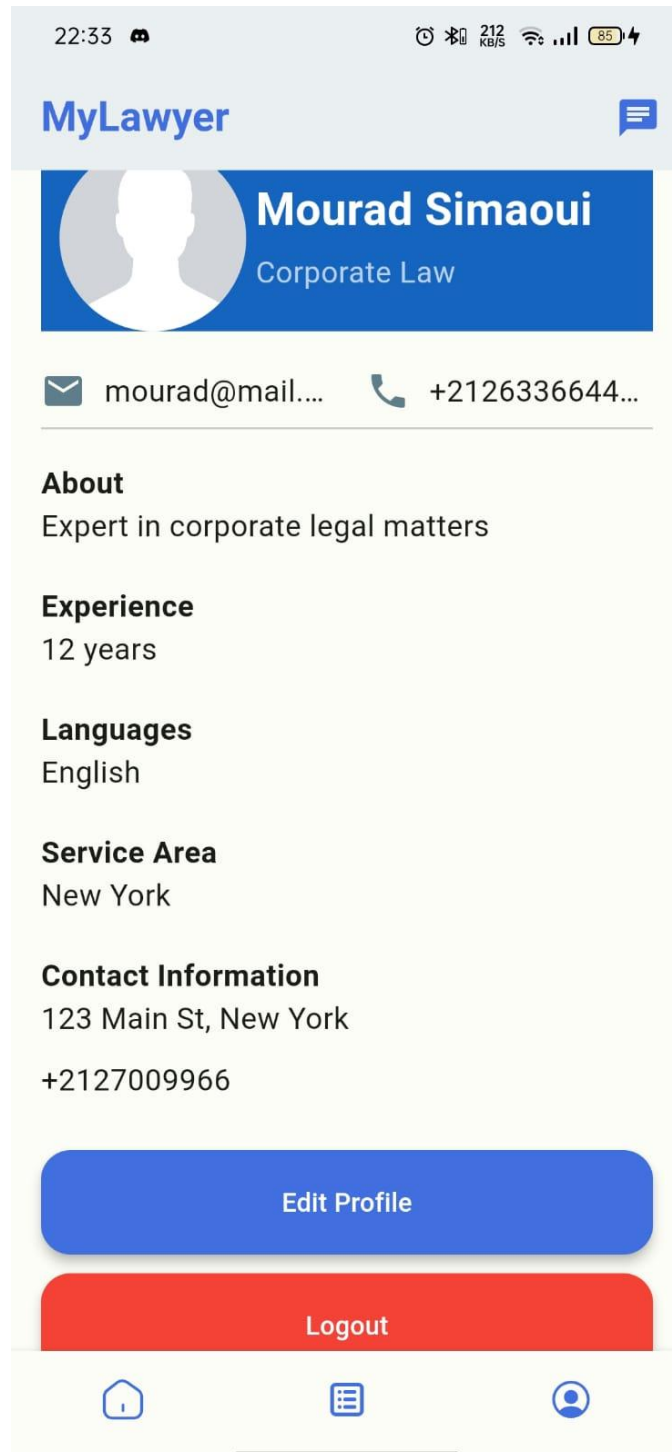


Figure 28 : Interface Profil Avocat

Interface Profil Client :

Les clients peuvent accéder à leur profil personnel, où ils peuvent consulter ou modifier leurs informations personnelles (nom, email, etc.). Ils peuvent également voir l'historique de leurs consultations et gérer leurs préférences.

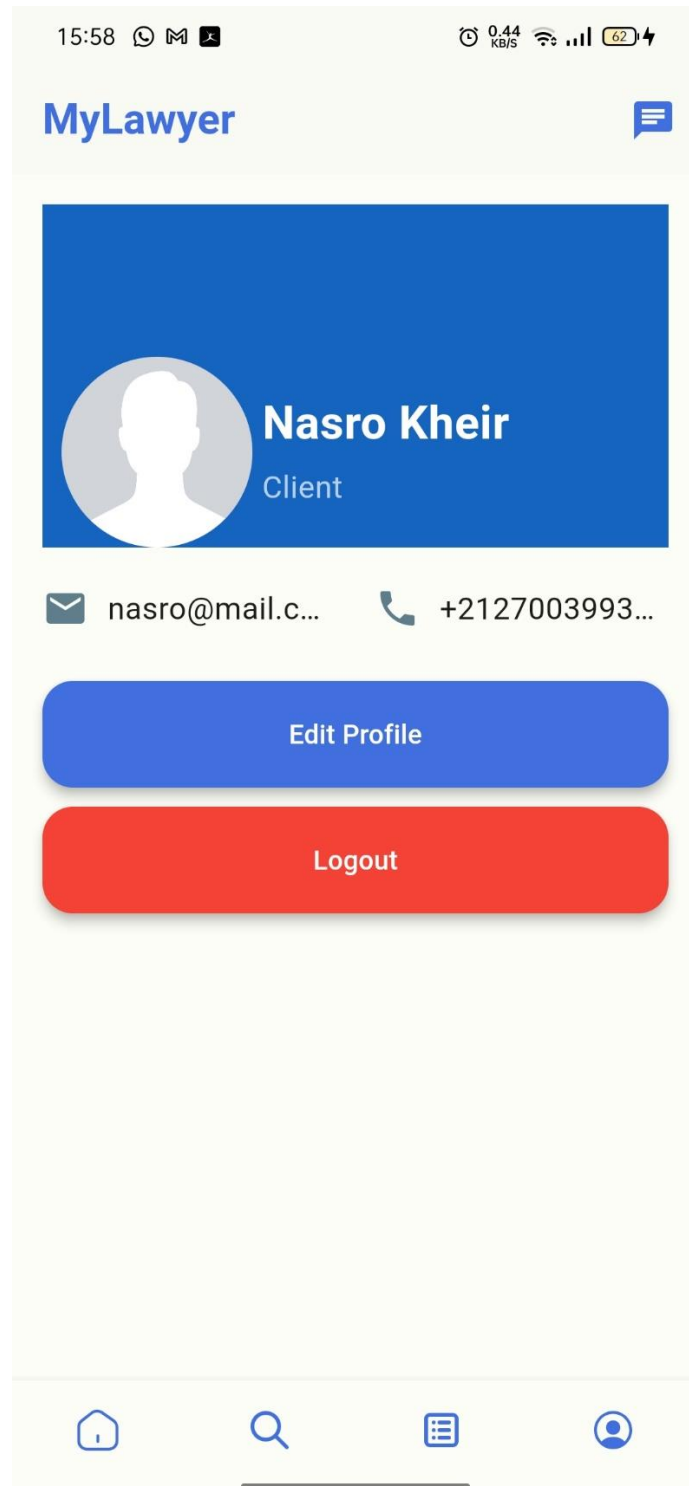


Figure 29 : Interface Profil Client

Interface Recherche Avocats (Search) :

Les clients peuvent utiliser cette interface pour rechercher des avocats en fonction de critères comme la spécialité, la localisation ou les avis clients. Une barre de recherche facilite la navigation et l'affichage des résultats.

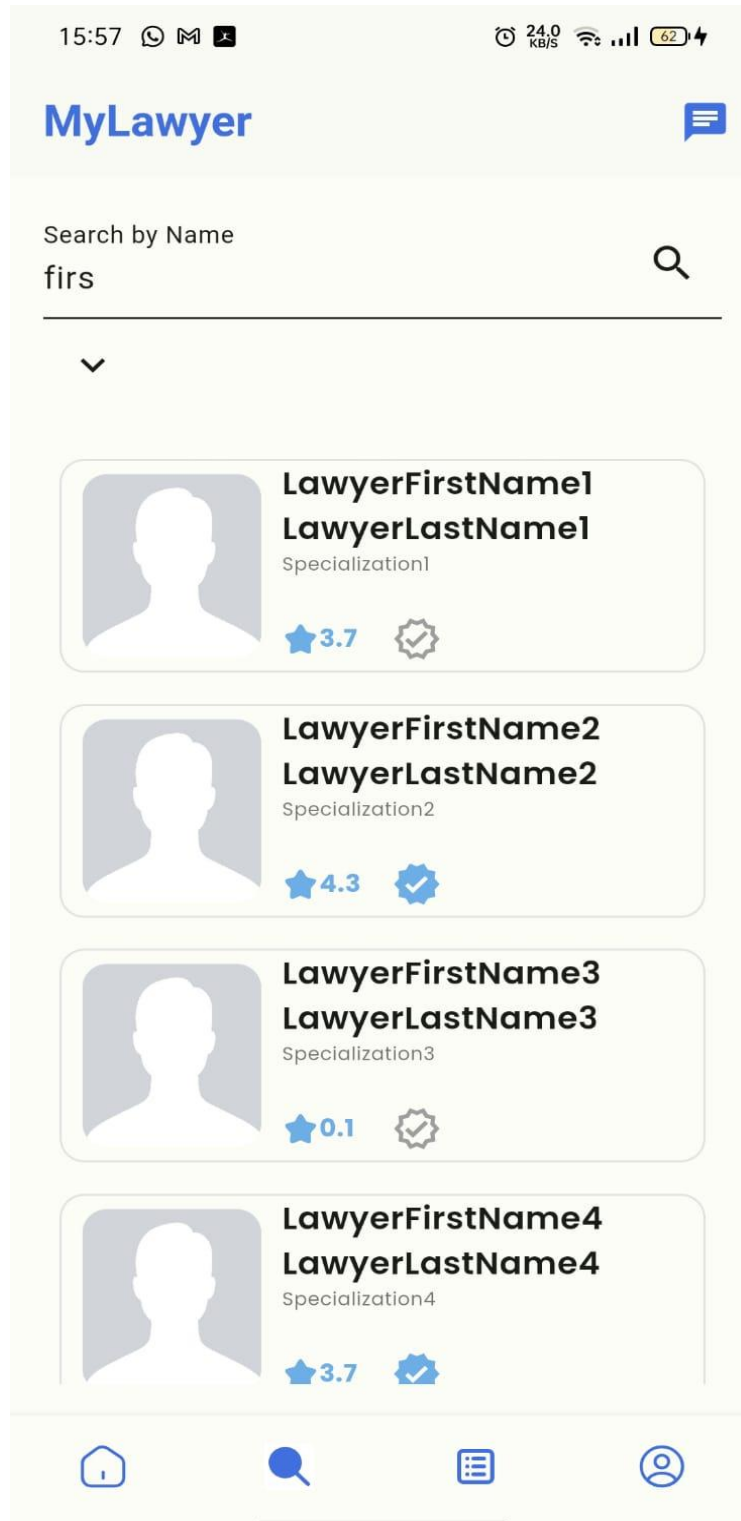


Figure 30 : Interface Recherche Avocats (Search)

Annexe (B) : Schéma de l'Architecture Backend de l'Application

Ce schéma illustre l'architecture backend de l'application, montrant l'intégration des différents composants pour la gestion des requêtes et la sécurisation des échanges. Le schéma inclut des éléments clés tels qu'Apache Tomcat, Spring Boot, le contrôleur principal, le Dispatcher Servlet, ainsi que les mécanismes de sécurité avec le JWT (JSON Web Token). Il montre également le flux de validation des jetons JWT, l'interaction avec le service de sécurité, la vérification des requêtes HTTP, et la gestion des données utilisateurs avec PostgreSQL.

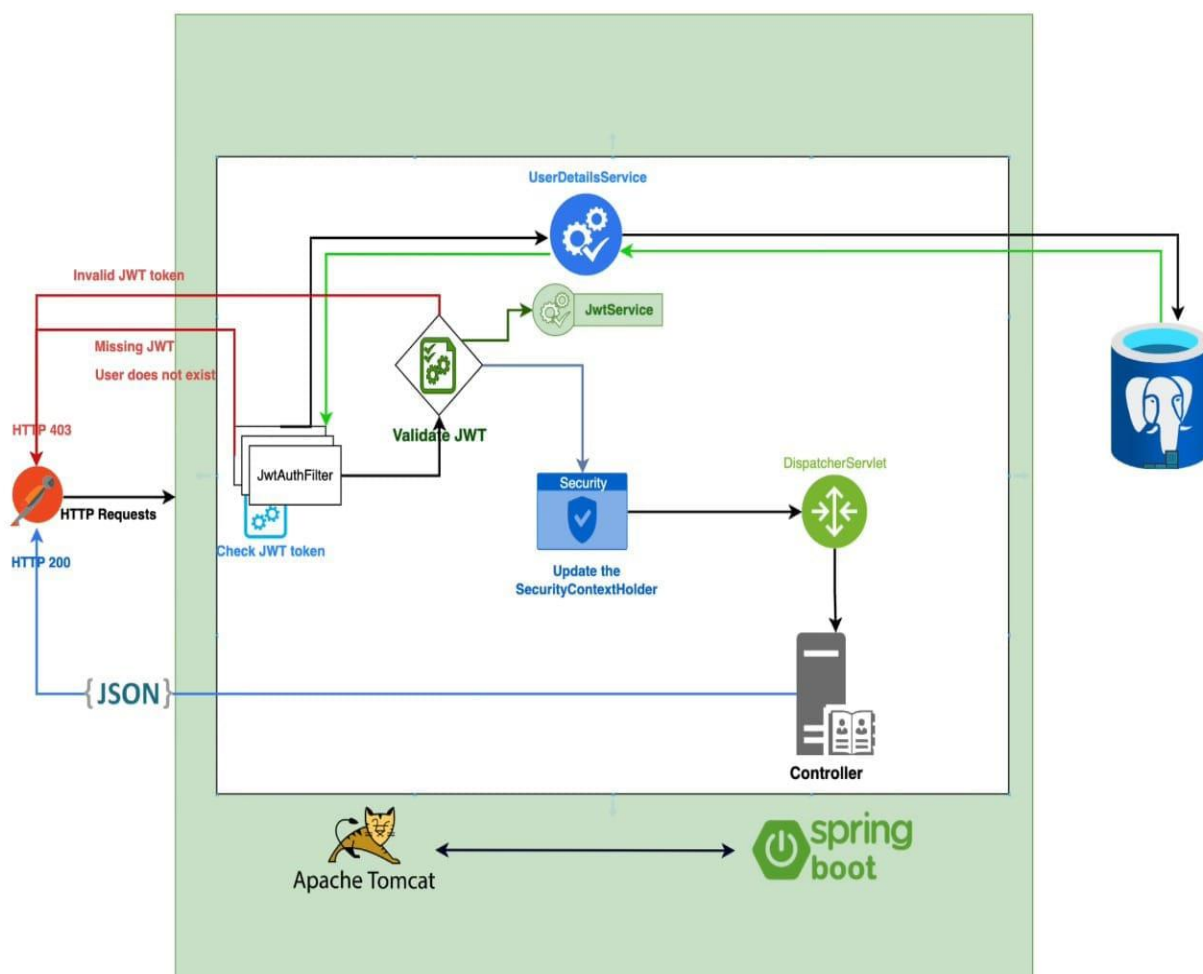


Figure 31 : Schéma de l'Architecture Backend de l'Application