

Chapitre 3 : Conception

1. Introduction

Ce chapitre détaille la conception de l'application de gestion de cabinet d'avocat, en présentant la structure du système et ses interactions.

2. Architecture générale de l'application

L'architecture générale de notre application de gestion de cabinet d'avocat repose sur une structure en trois tiers, garantissant une séparation claire des responsabilités entre les composants de l'application. Cette approche améliore la flexibilité, la sécurité et la maintenabilité du système.

2.1. Architecture deux tiers / Architecture trois tiers

Lors de la conception de notre application de gestion de cabinet d'avocat, nous avons étudié deux principales architectures : l'architecture deux tiers et l'architecture trois tiers. Ces modèles définissent la manière dont les différentes parties de l'application communiquent et interagissent.

2.2. Architecture trois tiers

L'architecture trois tiers est une architecture modulaire divisée en trois couches distinctes :

- **Client (Frontend) :** L'interface utilisateur développée avec Angular, permettant une interaction fluide et dynamique.
- **Serveur (Backend/API) :** Géré par Node.js avec Express, il traite les requêtes du client et exécute la logique métier.
- **Base de données :** MongoDB, qui stocke les informations de manière persistante (clients, dossiers, documents, rendez-vous).

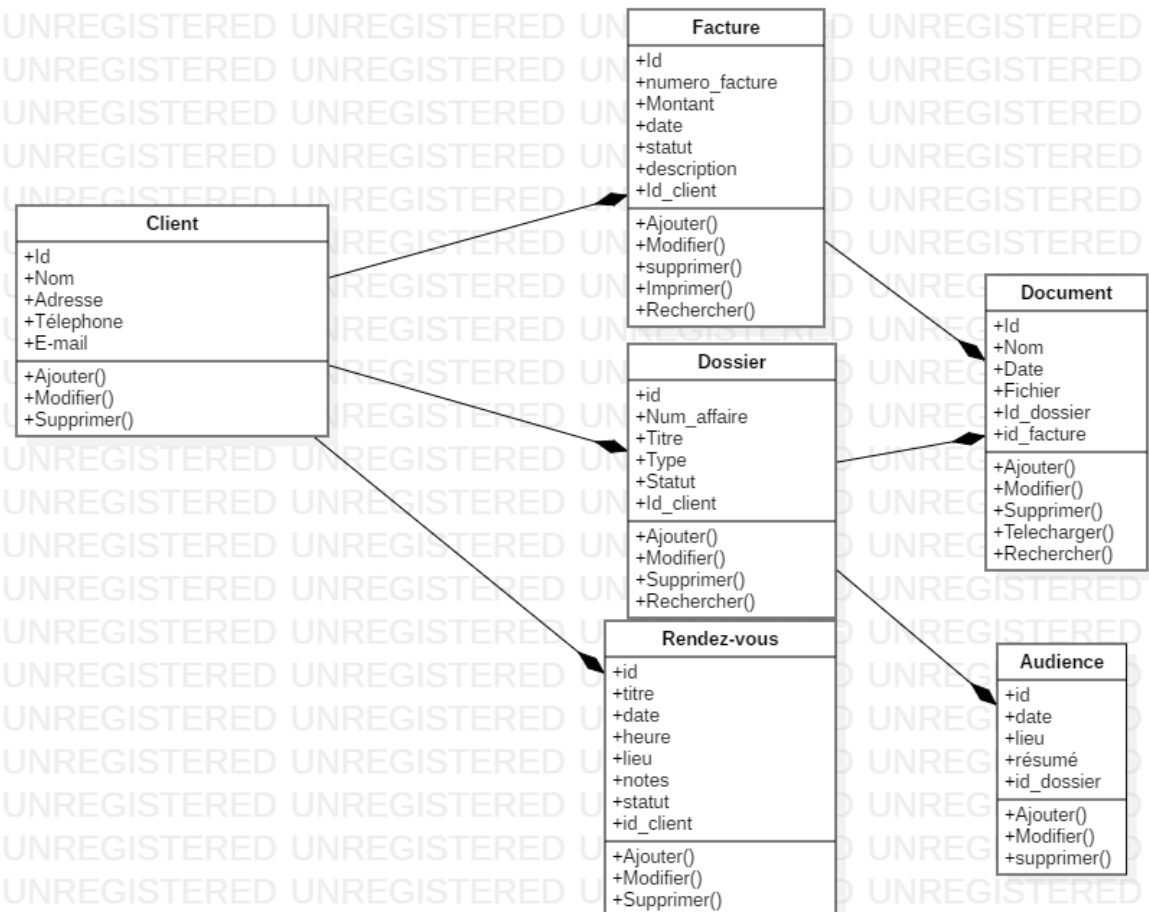
2.3. Justification du choix

- Une séparation claire des couches facilite la maintenance.
- La sécurité est renforcée avec la gestion des accès via une API sécurisée.

- L'évolutivité est garantie par la possibilité d'améliorer chaque composant indépendamment.

3. Conception détaillée

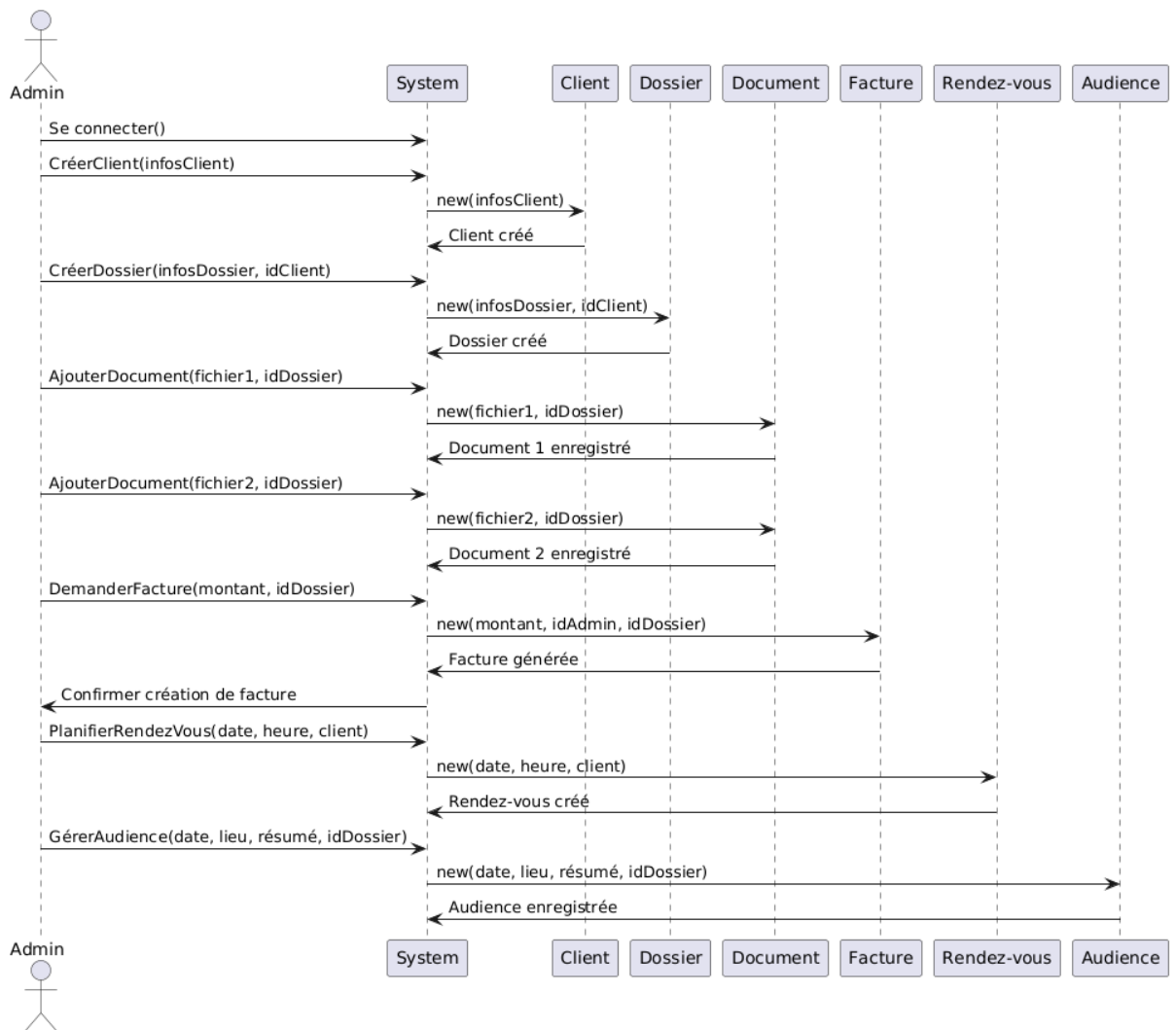
3.2. Vue Statique : Diagramme de classe



3.2. Vue dynamique

3.2.1. Diagramme de séquences

Pour Shématiser la vue dynamique de notre appliacion, nous faisons recours au diagramme de séquence d'UML



4.Conclusion:

Ce chapitre a permis de modéliser les aspects essentiels du système à travers deux représentation fondamentales:

Le diagramme des classes et le diagramme de séquence.

Le diagramme de classes a structuré les entités principales

De l'appliacion (client,Dossiers,Document,Facture,etc..) ainsi que leurs relations, mettant en évidence l'organisation logique des données .

Le diagramme de séquence, quant a lui , a illustré dynamiquement les interactions entre l'acteur principal (l'avocat) et le système , lors de la création d'un dossier et de l'émission d'une facture. Ces

modèles servent de base solide pour la phase de conception détaillée et le développement de l'application .

Chapitre 4 : Réalisation

1. Introduction

Dans ce chapitre, nous présenterons la réalisation de notre application de gestion de cabinet d'avocat. Cette phase consiste à transformer les concepts théoriques définis dans les chapitres précédents en une solution logicielle complète et fonctionnelle. Nous aborderons l'environnement de développement, les outils utilisés, ainsi que les interfaces de l'application.

2. Environnement et outils de développement

2.1. Environnement matériel

- **Ordinateur** : PC avec 4 Go de RAM.
- **Processeur** : Intel Core i5.
- **Disque dur** : 256 Go SSD.
- **Connexion Internet** : Haut débit.

2.2. Environnement logiciel et langages

- **Système d'exploitation** : Windows 10.
- **IDE** : Visual Studio Code pour le développement front-end (Angular) et back-end (Node.js).
- **Navigateurs** : Google Chrome, Mozilla Firefox.
- **Langages de programmation** : TypeScript (Angular), JavaScript (Node.js).
- **Base de données** : MongoDB.
- **Frameworks** : Angular 17 (Front-end), Node.js/Express (Back-end).
- **Gestion de versions** : Git (GitHub).

3. Interface de l'application

3.1. Interface "Accueil"

L'interface "Accueil" présente une vue d'ensemble de l'application, avec un design clair et une navigation intuitive. Elle permet à l'utilisateur d'accéder rapidement aux différentes sections de l'application (Clients, Dossiers, Documents, Rendez-vous, Factures).

4. Conclusion

Dans ce chapitre, nous avons détaillé l'ensemble des étapes de réalisation de notre application de gestion de cabinet d'avocat. Nous avons commencé par présenter l'environnement de développement, en décrivant à la fois les outils matériels et logiciels utilisés. Ensuite, nous avons décrit les interfaces principales de l'application, en mettant l'accent sur leur conception et leur fonctionnalité.

L'interface "Accueil" offre une vue d'ensemble claire et intuitive pour l'utilisateur, tandis que l'interface "Connexion" garantit une authentification sécurisée pour accéder au système. Nous avons également expliqué les choix techniques et les bonnes pratiques adoptées pour garantir une application stable, sécurisée et performante.

Cette phase de réalisation constitue une étape cruciale dans le développement de notre projet, car elle permet de concrétiser les concepts théoriques définis dans les chapitres précédents en une solution logicielle fonctionnelle. Dans le prochain chapitre, nous aborderons les tests et la validation de l'application, afin de garantir son bon fonctionnement et sa conformité aux besoins définis.

Conclusion générale et perspectives

En conclusion, ce projet de gestion de cabinet d'avocat a permis de mettre en pratique nos compétences en développement web, en utilisant Angular pour le front-end, Node.js pour le back-end, et MongoDB pour la gestion

des données. Cette application offre une solution complète pour la gestion des clients, dossiers, documents, rendez-vous et factures.

Les perspectives d'amélioration incluent l'intégration de nouvelles fonctionnalités comme la gestion des notifications par e-mail, la mise en place d'un calendrier interactif pour les rendez-vous, et une sécurisation avancée des données des utilisateurs. Ces évolutions permettront d'optimiser encore davantage l'efficacité et l'ergonomie de l'application.