



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة محمد البشير الإبراهيمي برج بوعريريج
Université Mohammed El Bachir El Ibrahimi B.B.A
كلية
Faculté des Mathématiques et d'Informatique
قسم
Département Informatique



Mémoire

En vue de l'obtention du diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Option : Technologie de l'Information et de la Communication

Intitulé

Réalisation application web de gestion de cabinet

D'avocat

Présenté par : -Benmahbous abid

- Rouabah eleulmi

Soutenu le __/ __/2020 devant le Jury :

Président

Univ. M. El-Bachir El-Ibrahimi – BBA

Encadrant *Allou lotfi*

Univ. M. El-Bachir El-Ibrahimi – BBA

Examineur

Univ. M. El-Bachir El-Ibrahimi – BBA

Dédicace

A mes très chers parents, je dédie ce travail

*Aucun mot n'est assez fort pour leur exprimer la reconnaissance
sincère que je leur porte pour la richesse de leurs enseignements.*

A ma très chère grande mère

A mon très cher frère

Et à toute ma famille de Rouabah et Khettal

Dédicace

A mes très chers parents, je dédie ce travail

*Aucun mot n'est assez fort pour leur exprimer la reconnaissance
sincère que je leur porte pour la richesse de leurs enseignements.*

A ma très chère grande mère

A mon très cher frère et soeur

*Et à toute ma famille de **Benmahbous***

Remerciements

Mes remerciements les plus sincères à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de cette mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Je remercie piétrement Allah le tout puissant de m'avoir donné le courage et la volonté de mener à terme ce présent travail.

J'adresse mes vifs remerciements :

*A mon encadreur **Mr. ALLOU Lotfi** pour son encadrement, son soutien sans failles et sa disponibilité. Ses conseils, ses suggestions de lecture, ses commentaires, ses corrections et ses qualités scientifiques ont été très précieux pour mener à bien ce travail.*

Je tiens également à remercier et exprimer mon profond respect aux membres de jury d'avoir accepté de juger ce travail.

Sommaire

Contents

| | |
|---|-----------|
| Introduction Générale | 1 |
| Présentation du thème de projet : | 2 |
| Chapitre 1 : Les technologies du web | 3 |
| I. Introduction | 4 |
| II. Le World Wide Web (WWW) | 4 |
| III. Les Bases Du Web..... | 4 |
| III.1. Architecture Client-Serveur : | 4 |
| III.2. URL..... | 6 |
| III.3. Hypertexte et Topologie du Web : | 6 |
| IV. Technologies du web..... | 7 |
| IV.1. HTML : | 7 |
| IV.2. CSS : | 7 |
| IV.3. HTTP : | 8 |
| IV.4. Javascript : | 8 |
| IV.4.1. Principal Framework de java script : | 8 |
| IV.5. Les api web : | 8 |
| IV.6. Web component : | 9 |
| IV.7. Les technologie média web : | 9 |
| IV.8. Les Api de capteur et de diffusion multimédia | 10 |
| V. Conclusion | 10 |
| Chapitre 2 : Spécifications des besoins | |
| I. Présentation du métier..... | 12 |
| II. Problématique | 13 |
| III. Contexte / Existant : | 13 |
| III.1. Contexte | 13 |
| III.2. Analyse de l'existant..... | 13 |
| III.3. Critique de l'existant | 14 |

| | |
|---|-----------|
| IV. Spécifications fonctionnelles | 14 |
| IV.1. Les besoins fonctionnels | 14 |
| IV.2. Le domaine étudié..... | 14 |
| IV.3. Les acteurs du système | 15 |
| IV.4. Les besoins non fonctionnels | 15 |
| V. Présentation de projet :..... | 16 |
| VI. Conclusion | 16 |
| Chapitre 3 : Conception | |
| I. Introduction | 18 |
| II.UML : | 18 |
| II.1. Définition de UML : | 18 |
| II.2 Définition d'un cas d'utilisation :..... | 18 |
| II.2.1. Représentation de cas d'utilisation :..... | 18 |
| II.2.2. Représentation de cas d'utilisation de gestion de cabinet avocate (Cipher) :..... | 19 |
| II.3 Définition de diagramme de séquence : | 20 |
| II.3.1 représentations d'un diagramme de séquence : | 20 |
| II.4 Définition du diagramme d'activité :..... | 20 |
| II.4.1 représentation d'un diagramme d'activité : | 21 |
| III. Description UML :..... | 22 |
| III.1 Description des cas d'utilisation :..... | 22 |
| III.1.1 Cas d'utilisation « authentification » :..... | 22 |
| III.1.2. Cas d'utilisation « gestion d'affaire GA » : | 23 |
| III.1.3. Cas d'utilisation « gestion de contact GC » :..... | 27 |
| III.1.4. Cas d'utilisation « gestion d'utilisateur (employé) GU » :..... | 32 |
| III.1.5. Cas d'utilisation « gestion de taches (tasks) GT » : | 35 |
| III.1.6 Cas d'utilisation « gestion des finances GF » :..... | 37 |
| IV. Description des diagrammes de classes :..... | 40 |
| IV.1. Définition de diagramme de classes : | 40 |
| IV.2. Le diagramme de classes..... | 41 |
| IV.3. Les attributs des classes : | 42 |
| IV.4. Les méthodes des classes..... | 43 |
| V. Conclusion | 43 |

| | |
|---|-----------|
| Chapitre 4 : Réalisation | 44 |
| I. Introduction | 45 |
| II. Environnement du travail | 45 |
| II.1. Environnement matériel | 45 |
| II.2. Environnement logiciel | 45 |
| II.2.1. Système d'exploitation | 45 |
| II.2.2. Outil de modélisation UML | 46 |
| II.2.3. Environnement de développement intégré | 46 |
| II.2.4. Serveur d'application : « Node.js» | 46 |
| II.2.5 Outil d'administration de la base de données | 47 |
| II.2.6. Langage de programmation | 47 |
| III. L'architecture de l'application | 48 |
| III.1. Architecture 3-tiers | 48 |
| IV. Développement | 49 |
| V. Choix de développement | 49 |
| V.1. Le modèle | 50 |
| V.2. La vue | 50 |
| V.3. Le contrôleur | 51 |
| VI. Les technologies utilisées : | 51 |
| VII. Les interfaces graphiques | 51 |
| VII.1. Interface Authentification | 52 |
| VII.2. Interface de tableau de bord (Dashboard) | 53 |
| VII.3. Interface du menu d'application | 54 |
| VII.4. Interface de gestion d'affaires | 55 |
| VII.5. Interface de détails d'affaire | 56 |
| VII.6. Interface d'ajoute une affaire | 57 |
| VII.7. Interface de gestion de client (contact) | 58 |
| VII.8. Interface de détails de client (contact) | 58 |
| VII.9. Interface de gestion de taches (tasks) | 59 |
| VIII. Conclusion | 59 |
| Conclusion Générale et perspectives | 60 |

Liste de figure

| | |
|---|----|
| Figure I.1 : Architecture Client-Serveur 1-tier | 5 |
| Figure I.2 : Architecture Client-Serveur N-tier | 5 |
| Figure I.3 : URL..... | 6 |
| Figure I.4 : Topologie du Web | 7 |
| Figure III.1 : Représentation simple de cas d'utilisation. | 18 |
| Figure III.2 : Diagramme de cas d'utilisation général de système. | 19 |
| Figure III.3 : représentation d'un diagramme de séquence..... | 20 |
| Figure III.4 : représentation d'un diagramme d'activité. | 21 |
| Figure III.5 : Diagramme de cas d'utilisation « authentification ». | 22 |
| Figure III.6 : Diagramme de séquence du scénario « authentification ». | 23 |
| Figure III.7 : Diagramme de cas d'utilisation « gestion d'affaire ». | 23 |
| Figure III.8 : Diagramme d'activité « Gestion d'affaire ». | 25 |
| Figure III.9 : Diagramme de séquence du scénario « créer un affaire ». | 26 |
| Figure III.10 : Diagramme des cas d'utilisation « gestion des contact ». | 27 |
| Figure III.11 : Diagramme d'activité « Gestion de contact ». | 29 |
| Figure III.12 : Diagramme de séquence du scénario « ajoute un contact ». | 30 |
| Figure III.13 : Diagramme des cas d'utilisation « gestion d'utilisateur (employé) GU». | 32 |
| Figure III.14 : Diagramme de séquence du scénario « ajoute un utilisateur ». | 33 |
| Figure III.15 : Diagramme de séquence du scénario « modifier un utilisateur ». | 34 |
| Figure III.16 : Diagramme des cas d'utilisation « gestion des Taches GT ». | 35 |
| Figure III.17 : Diagramme de séquence du scénario « ajoute une tache ». | 36 |
| Figure III.18 : Diagramme des cas d'utilisation « gestion des finances GF ». | 37 |
| Figure III.19 : Diagramme de séquence du scénario « ajoute une transaction ». | 39 |
| Figure III.20 : Diagramme de classe. | 41 |
| Figure IV.1 : Architecture 3 – tiers | 48 |
| Figure IV.2 : Schéma du modèle MVC. | 50 |
| Figure IV.3 : Interface Authentification..... | 52 |
| Figure IV.4 : Interface de tableau de bord. | 53 |
| Figure IV.5 : Interface du menu d'application..... | 54 |
| Figure IV.6 : Interface de gestion d'affaires. | 55 |
| Figure IV.7 : Interface de détails d'affaire | 56 |
| Figure IV.8 : Interface d'ajoute une affaire. | 57 |
| Figure IV.9 : Interface de gestion de client(contact)..... | 58 |
| Figure IV.10 : Interface de détails de client. | 58 |
| Figure IV.11 : Interface de gestion de taches..... | 59 |

Liste des tableaux

| | |
|--|----|
| Tableau II.1 : Description des acteurs | 15 |
| Tableau III.1: Description des méthodes des classes d'objets et d'association..... | 42 |
| Tableau III.2 : Description des méthodes des classes. | 43 |

Résumé

Dans le premier chapitre nous avons parlé des technologies Web et expliqué certaines des techniques que nous utiliserons dans notre application, grâce à quoi nous pouvons produire ces applications. Et dans dixième chapitre présente une phase indispensable pour l'étude et l'analyse de notre application. Nous avons défini les différents besoins fonctionnels et non fonctionnels. Nous entamerons dans le chapitre suivant la conception de cette application qui comporte les diagrammes de cas d'utilisation détaillés, les différents diagrammes de séquence et le diagramme de classe. Et Dans troisième chapitre, nous avons présenté le langage UML. Aussi nous avons fait la description des diagrammes des cas d'utilisation, classe et de séquence et d'activité, et de contexte afin de délimiter le cadre de notre travail et de préparer un terrain favorable pour la prochaine étape. La dernière partie de ce projet était dédiée à la navigation dans notre application. Cette partie constitue le dernier volet de ce rapport, elle a pour objet de présenter l'environnement logicielle matériel de réalisation. Elle a été clôturée par l'exposition des imprimés écrans décrivant quelques interfaces de notre outil.

Résumé

In the first chapter we talked about web technologies and explained some of the techniques that we will use in our application, thanks to which we can produce these applications. And in the tenth chapter presents an essential phase for the study and analysis of our application. We have defined the different functional and non-functional needs. In the next chapter, we will start designing this application which includes the detailed use case diagrams, the various sequence diagrams and the class diagram. And In third chapter, we presented the UML language. Also we have described the diagrams of the use cases, class and sequence and activity, and context in order to delimit the scope of our work and prepare a favorable ground for the next step. The last part of this project was dedicated to navigation in our application. This part constitutes the last part of this report, its purpose is to present the hardware software environment. It ended with the exhibition of screen prints describing some interfaces of our tool.

Résume

تحدثنا في الفصل الأول عن تقنيات الويب وشرحنا بعض التقنيات التي سنستخدمها في تطبيقنا ، والتي بفضلها يمكننا إنتاج هذه التطبيقات. وفي الفصل العاشر يقدم مرحلة أساسية لدراسة وتحليل تطبيقنا. لقد حددنا الاحتياجات الوظيفية وغير الوظيفية المختلفة. سنبدأ في الفصل التالي بتصميم هذا التطبيق الذي يتضمن مخططات حالة الاستخدام التفصيلية ومخططات التسلسل المختلفة ومخطط الفصل. وفي الفصل الثالث قدمنا لغة UML كما وصفنا المخططات لحالات الاستخدام ، والفئة والتسلسل والنشاط ، والسياق من أجل تحديد نطاق عملنا وإعداد أرضية مواتية للخطوة التالية. تم تخصيص الجزء الأخير من هذا المشروع للملاحظة في تطبيقنا. يشكل هذا الجزء الأخير من هذا التقرير ، والغرض منه هو تقديم بيئة برامج الأجهزة لتحقيقها. انتهى بمعرض مطبوعات الشاشة التي تصف بعض واجهات أدواتنا.

Introduction Générale

Actuellement, le monde connaît une avancée technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science étudiant les techniques du traitement automatique de l'information.

Elle joue un rôle important dans le développement de l'entreprise et d'autres établissements. Avant l'invention de l'ordinateur, nous enregistrons toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de gestion de données des entreprises, ce qui est la partie essentielle dans leur développement aujourd'hui

En droit, l'avocat est un juriste dont la fonction traditionnelle est conseiller et de défendre ses clients, personnes physiques ou morales, en justice, en plaidant pour faire valoir leurs intérêts et, plus généralement, pour les représenter. L'« avocat » acquitte d'une fonction de conseil et de rédacteur d'actes. Le ministère d'avocat est parfois rendu obligatoire par le droit national, notamment afin d'assurer les droits de la défense devant certaines juridictions.

L'activité professionnelle des avocats se divise en deux domaines : les activités pour lesquelles les avocats ont un monopole, et celles qu'ils peuvent exercer concurremment avec d'autres professions.

L'avocat a notamment pour fonction de défendre les droits des justiciables devant les juridictions, ou toutes instances disciplinaires en assistant et représentant ses clients. Ils disposent du monopole de plaidoirie et de la postulation, même si ce monopole connaît des exceptions.

En effet, devant le juge de proximité, la présentation par un avocat n'est pas obligatoire.

L'avocat a également des tâches qu'il partage avec d'autres professions juridiques, de délivrer des avis ou des consultations, de rédiger des contrats, actes ou transaction sous seing privé .il peut aussi représenter les tiers et les assister devant toutes les administrations publiques.

Présentation du thème de projet :

Notre mission dans le cadre de ce projet est de créer une application permettant de gérer un cabinet d'avocat, il s'agit de définir les responsabilités de la gestion, mettre à jour les données, organiser des données collectées après du secrétariat afin de concevoir des fichiers de bases pour l'Avocat, de renforcer le contrôle et la confrontation, assurer une meilleure gestion et une cohérence de l'information.

Notre application aura comme principales fonctionnalités :

- Gestion et suivi des dossiers .
- Gestion des tâches .
- Gestion des factures ;
- Gestion de clientèle.
- Gestion d'employé

Chapitre 1

Les technologies du web

I. Introduction

Aujourd'hui Internet est devenu un nouvel outil d'information et de communication en pleine évolution offrant des perspectives de croissance exceptionnelles. C'est devenu un formidable moyen de communication, échange, travail, rencontre, et même de commerce. Je peux dire que le web est le moyen le plus utilisé pour la diffusion des informations, en informatique, une application Web (aussi appelée Web App) est un logiciel applicatif manipulable grâce à un navigateur Web. De la même manière que les sites Web, une application Web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur Web.

L'objectif de ce travail est de réaliser un site web dynamique pour la gestion d'un cabinet d'avocat

II. Le World Wide Web (WWW)

Est un ensemble de protocoles (ex : HTTP) d'outils (ex : HTTPD, navigateurs...) et de normes (URL) permettant de créer, formater, rechercher échanger...bref partager de manière interactive des informations hétérogènes à travers Internet sur le principe du Client/serveur.

III. Les Bases Du Web

Le Web fait appel à un certain nombre de concepts de base. Nous verrons les trois plus importants dans cette première partie.

- 1. Architecture Client-Serveur
- 2. URL
- 3. Hypertexte et Topologie du Web

III.1. Architecture Client-Serveur :

- Le Web s'appuie sur la notion d'architecture client-serveur.
- Un serveur est une machine en général assez puissante qui fournit un ou plusieurs services (accès à des sources de données, applications...).
- Pour fournir ces services elle fait tourner en permanence des programmes que l'on appelle aussi des serveurs en l'occurrence ce sont des serveurs Web ou serveurs HTTP.
- De l'autre côté les utilisateurs font tourner sur leur machine (machine cliente) un programme client qui, comme son nom l'indique va être

Demandeur de services, en l'occurrence ce client est un navigateur Web qui va demander des pages Web à un serveur Web.

- Le dialogue entre le client et le serveur se compose donc de requêtes émises par le client et de réponses données par le serveur.

- On appelle démon un programme qui tourne en tâche de fond sur une machine et le cas échéant répond à des requêtes qui lui sont adressées ou déclenche des actions en réponse à des événements ou un planning.

- Sur le Web les documents s'échangent selon le protocole HTTP (HyperText Transfer Protocol) et le démon qui se charge de répondre aux requêtes des autres machines se nomme HTTPD (HyperText Transfer Protocol Daemon).

- De la même façon un serveur offrant des fichiers via FTP est une machine sur laquelle tourne un serveur FTP encore appelé démon FTPD (File Transfer Protocol Daemon), de même pour l'e-mail, etc...

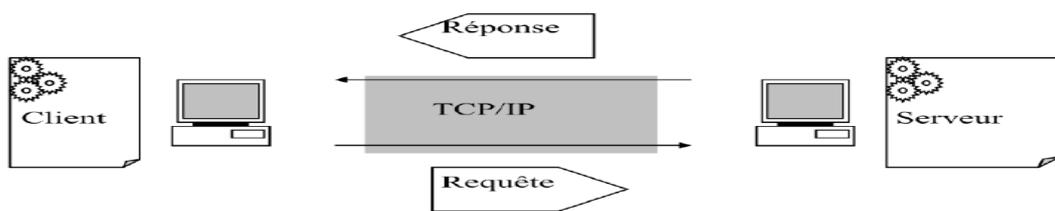


Figure I.1 : Architecture Client-Serveur 1-tier

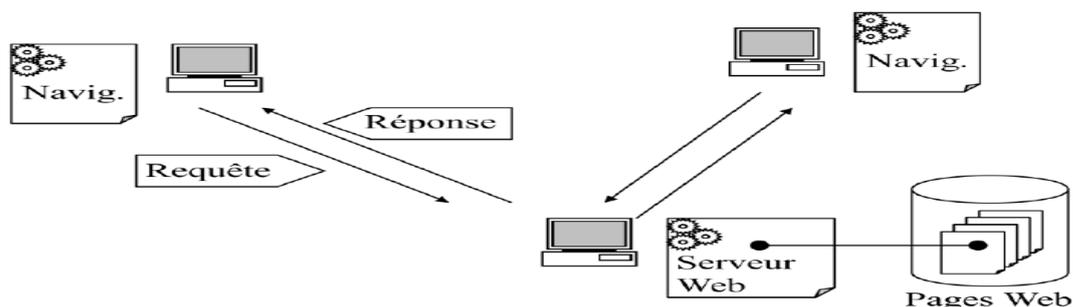


Figure I.2 : Architecture Client-Serveur N-tier

III.2. URL

- Pour accéder à une page web il faut d'abord pouvoir décrire où elle se trouve.
- Pour repérer un document, un fichier, une source de données ... on a développé la notation URL (Universal/Uniform Resource Locator).
- Un URL peut désigner : un serveur ftp, un fichier sur votre disque, un serveur gopher, une image, une adresse courrier, un serveur de News, un serveur telnet et bien sûr une page Web publiée par un serveur http, c'est-à-dire un serveur de Web.
- En particulier, dans ce dernier cas l'URL contient : le nom du protocole d'accès au fichier (HTTP), le nom du serveur (adresse IP ou nom symbolique), le chemin d'accès au fichier et bien sûr le nom du fichier :

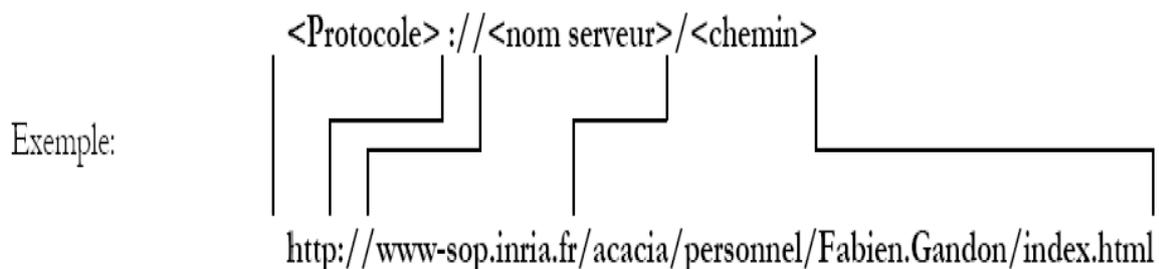


Figure 1.3 : URL

III.3. Hypertexte et Topologie du Web :

- On qualifie d'Hypertexte (terme et notion inventés par Ted Nelson en 1960) un document essentiellement textuel, dynamique, capable de changer et de réagir en fonction de certains événements comme par exemple un clic à la souris.
- Un tel document offre une très grande convivialité et la tendance étant à élargir cette technique à d'autres médias (image, vidéo, animations...) on parle maintenant d'hypermédia.
- Le langage permettant de décrire les pages Web est le HTML (Hyper Text Markup Language).
- Ce langage à balise permet de doter certains mots, ou images d'une propriété d'hyperlien ou plus simplement de lien qui est constitué d'une adresse URL que vous atteindrez en cliquant dessus.
- L'information disponible sur le Web a la caractéristique d'être distribuée sur une zone géographique très grande, et au sein d'une même page web, on peut conjuguer des ressources placées aux quatre coins de la planète.[1]

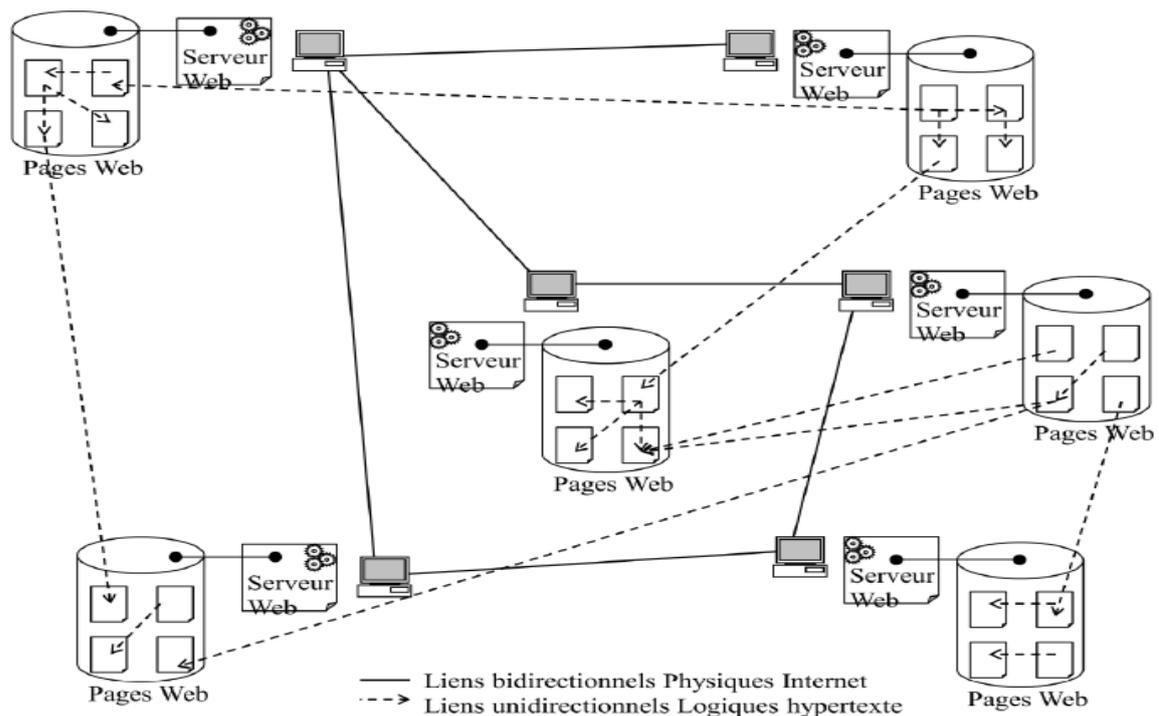


Figure 1.4 : Topologie du Web.

IV. Technologies du web

IV.1. HTML :

L'**HTML** est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie *HyperText Markup Language*, ce qui signifie en français "*langage de balisage d'hypertexte*". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.[2]

IV.2. CSS :

Le terme **CSS** est l'acronyme anglais de *Cascading Style Sheets* qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers **HTML** ou **XML**. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en **HTML**. [2]

IV.3. HTTP :

HTTP est un protocole d'application qui s'exécute au-dessus de la suite de protocoles TCP/IP.

L'un des concepts du protocole HTTP inclut l'idée que les fichiers peuvent contenir des références à d'autres fichiers (d'où la notion d'hypertexte) dont la sélection va solliciter d'autres demandes de transfert.

Tous les serveurs Web contiennent, en plus des fichiers de pages Web qu'ils servent, un daemon HTTP, c'est-à-dire un programme conçu pour attendre les demandes HTTP et les traiter à leur arrivée. [3]

IV.4. Javascript :

JavaScript désigne un langage de développement informatique, et plus précisément un **langage de [script](#) orienté objet**. On le retrouve principalement dans les pages Internet. Il permet, entre autres, d'introduire sur une page web ou HTML des petites animations ou des effets.

Créé en 1995 par Brendan Eich, en même temps que la technologie Java, le langage JavaScript se distingue des langages [serveurs](#) par le fait que l'exécution des tâches est opérée par le navigateur lui-même, sur l'ordinateur de l'utilisateur, et non sur le serveur web. Il s'active donc généralement sur le poste client plutôt que côté serveur.

IV.4.1. Principal Framework de java script :

Il existe de nombreux frameworks JavaScript orientés vers les interfaces web (ou "orientés client"). Les trois plus connus sont **Jquery**, [AngularJS](#) (qui a été initialement développé par Google) et **React** (qui, lui, est né chez [Facebook](#)). Il existe néanmoins quelques infrastructures JavaScript open source orientées serveur, même si ce langage n'avait pas été conçu dans cette optique au départ. La plus célèbre d'entre elles n'est autre que **NodeJS**. [4]

IV.5. Les api web :

Une API (*Application Programming Interface* ou interface de programmation) est un ensemble de services offert par un logiciel à d'autres logiciels. Grâce aux API, les programmes informatiques peuvent interagir entre eux selon des conditions déterminées.

Le DOM est une API: il définit une manière de représenter une page web et fournit un certain nombre de services qui permettent à nos programmes JavaScript d'interagir avec la page.

On appelle API web une API accessible via les technologies du Web, notamment le protocole HTTP ou sa version sécurisée HTTPS. Un nombre croissant de sites et de services en ligne proposent des API web destinées aux développeurs. Elles offrent à ces développeurs le moyen d'exploiter leurs données et leurs services.[5]

IV.6. Web component :

Les Web Components sont des blocs de code englobant la structure interne des éléments HTML incluant le CSS et le JavaScript et permettant d'utiliser le code concerné à l'endroit voulu dans les sites Internet et les applications Web. Le concept a été développé par un groupe de travail du World Wide Web Consortium (W3C) fondé en 1994 par l'inventeur du Web Tim Berners-Lee qui s'est depuis engagé activement pour une standardisation de toutes les technologies Web élémentaires. Le modèle de composants Web publié en 2012 prévoit à la base les quatre spécifications suivantes pour la création des composants HTML utiles :

- **Custom Elements** : ensemble d'API JavaScript pour la définition des éléments définis par l'utilisateur
- **Shadow DOM** : ensemble d'API JavaScript permettant d'ajouter des éléments DOM
- **ES Modules** : des modules permettant l'intégration et la réutilisation des documents JavaScript
- **HTML Templates** : des modèles de balisage qui ne sont pas présentés sur la page affichée et peuvent être utilisés pour servir de base aux éléments définis par l'utilisateur

Ce standard de composants Web est aujourd'hui pris en charge par tous les navigateurs courants. Pour travailler avec les codes HTML contenus dans ces éléments, on peut avoir recours à l'ensemble des framework ou des bibliothèques JavaScript utilisant le HTML.[6]

IV.7. Les technologie média web :

Au fil des ans, la capacité du Web à présenter, créer et gérer l'audio, la vidéo et d'autres médias s'est développée à un rythme croissant. Aujourd'hui, il existe un grand nombre d'API

disponibles, ainsi que des éléments HTML, des interfaces DOM et d'autres fonctionnalités qui permettent non seulement d'effectuer ces tâches, mais aussi d'utiliser des médias en tandem avec d'autres technologies pour faire des choses vraiment remarquables.[7]

IV.8. Les Api de capteur et de diffusion multimédia

Souvent appelée API Media Streams ou simplement API MediaStream, est une API liée à WebRTC qui prend en charge le streaming de données audio et vidéo. Il fournit les interfaces et les méthodes pour travailler avec les flux et leurs pistes constitutives, les contraintes associées aux formats de données, les rappels de réussite et d'erreur lors de l'utilisation des données de manière asynchrone et les événements déclenchés pendant le processus.

V. Conclusion

Dans ce chapitre nous avons parlé des technologies Web et expliqué certaines des techniques que nous utiliserons dans notre application, grâce à quoi nous pouvons produire ces applications.

Chapitre 2

Spécifications des besoins

I. Présentation du métier

Les avocats sont de plus en plus spécialisés. Les avocats de proximité, qui règlent les problèmes de la vie quotidienne (droit de la famille, droit du travail...), restent de loin les plus nombreux. Mais ce sont le droit des affaires et le conseil aux entreprises qui sont aujourd'hui les créneaux les plus porteurs.

Les sociétés qui n'ont pas de service juridique font régulièrement appel à des avocats extérieurs pour régler les affaires juridiques courantes, commerciales ou financières. Même si elles disposent de juristes en interne, elles peuvent aussi solliciter un avocat pour régler des problèmes plus complexes.

L'avocat exerce souvent plusieurs fonctions (conseil juridique et défense).

Le conseil juridique s'exerce surtout auprès des entreprises (multinationales, banques, assurances).

L'objectif est souvent d'éviter les risques de procès préjudiciables à l'entreprise. En cas de litige, l'avocat assure le rôle de médiateur, cherchant à concilier les parties adverses.

L'avocat pénaliste est spécialisé dans le traitement des crimes et délits. Il peut se mettre aussi bien au service de la victime que du côté de l'auteur. Au service de l'auteur présumé, il représente son client cours des comparutions immédiates, des gardes à vues, des détentions provisoires, de l'instruction puis du procès. Il rencontre plusieurs fois son client avant le procès, constitue son dossier et rédige les pièces nécessaires à la défense. Il assiste les inculpés chez le juge d'instruction pour veiller au respect de leurs droits, ainsi qu'à l'audition des témoins et aux expertises. Lors du procès, il soutient la cause de son client par ses interventions et une plaidoirie finale. Dans le cadre de la défense de la victime, son rôle sera d'obtenir un dédommagement pour le préjudice causé.

Pour exercer, **l'avocat** doit être inscrit à un barreau (ensemble des avocats rattachés à un tribunal de grande instance).

Quel que soit sa spécialisation, l'avocat est un homme de dossier. Doté d'une grande capacité de travail, il dispose d'un esprit de synthèse et d'analyse et d'une capacité de raisonnement logique. Des qualités oratoires et rédactionnelles sont des atouts indispensables pour exercer ce métier.

L'avocat est soumis aux règles de la déontologie de sa profession.

II. Problématique

D'après les visites effectuées au bureau d'avocat et les investigations sur la méthode de travail, j'ai résumé que le processus suivi dans le bureau d'avocat est trop classique et fatigant, soit coté administration ou coté client << utilisation de trop grand nombre de papiers et de l'exploitation d'un grand espace, En outre, les conditions appropriées doivent être fournies pour préserver sa qualité. Il est également difficile de rechercher dans les archives>>.

Pour cela on a réalisé une application web qui simplifie et facilite toutes les procédures de fonctionnement.

III. Contexte / Existant :

Avant de spécifier les besoins il est nécessaire de réaliser une étude de l'existant. L'étude de l'existant est une étape clé dans la réalisation de n'importe quelle application informatique, quel que soit le domaine concerné. Il s'agit d'une étude permettant de comprendre la problématique du projet et de détecter les avantages et inconvénients des solutions proposées actuellement sur le marché afin d'en profiter pour la réalisation de notre projet.

III.1. Contexte

Conception et réalisation d'une application Web pour une Facturation, Gestion et Suivi des dossiers.

Les activités internes d'application

- Gestion d'affaire
- Gestion clients(contact)
- Gestion des taches
- Gestion des finances
- Gestion d'utilisateur(employer)

III.2. Analyse de l'existant

Le cabinet d'avocat repose principalement sur trois grandes parties qui assurent son fonctionnement normal qui sont : le service de finance, la gestion et enfin le suivis des dossiers.

En effet Le coût des transferts d'argent des clients est basé sur le prix du dossier déterminé par l'avocat Puis le département de gestion des dossiers, avec tous ses mérites, le procès, le verdict judiciaire et le dossier.

III.3. Critique de l'existant

L'ensemble des activités du cabinet est géré par des traitements manuels ou bien des applications diversifiées tout en utilisant des supports papiers.

Les besoins de du cabinet s'articulent sur les modules fonctionnels suivants : la gestion des factures, la gestion des employées, la gestion des clients et la gestion d'affaire et de taches.

IV. Spécifications fonctionnelles

La spécification des besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application.

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques les plus critiques et d'identifier les cas d'utilisation initiaux.

IV.1. Les besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système.

Ces besoins se regroupent dans le diagramme de cas d'utilisation.

- Elimination de la redondance de saisie des informations dans une base de données.
- Gestion de tous services relatifs à la gestion des dossiers.
- Espace de facturation et situation financier.
- Gestion d'une base de données clientèle et employés.
- Organisation de taches.

IV.2. Le domaine étudié

Conception et réalisation d'une application Web pour une Facturation, Gestion et Suivi des dossiers.

Les activités internes d'application

- Gestion affaire
- Gestion clients (contact)
- Gestion des taches

- Gestion des finances
- Gestion d'utilisateur (employer)

IV.3. Les acteurs du système

Après l'étude de l'existant, on constate que ce système est en interaction avec deux acteurs principaux sont les suivants

| Acteur | Description |
|------------------|--|
| L'avocat | <p>Superviseur et administrateur système.</p> <p>Peut créer une affaire pour ces clients comme il peut le mettre à jour, toute procédure contient un ou plusieurs pièces joint, il a le droit de crée et visualiser l'ensemble des clients du cabinet.</p> <p>Assure le suivi d'état d'affaire.</p> <p>Créer, gère, édite, consulte et imprime les factures.</p> <p>Créer des taches pour chaque affaire, et visualiser l'emploi de tempe.</p> |
| Client (contact) | <p>Il a le droit de suivre l'avancement de son affaire.</p> <p>Connaitre à tout moment sa situation financière.</p> <p>Ainsi d'imprimer sa facture.</p> <p>Consulter la date de jugement,</p> |
| Employer (user) | <p>Le sous-administrateur créé par l'avocat donne l'autorisation d'utiliser l'application en fonction des tâches qui lui sont assignées</p> <p>Ses tâches sont précisées en fonction des autorisations</p> |

Tableau II.1 : Description des acteurs.

IV.4. Les besoins non fonctionnels

A part les besoins fondamentaux, notre système doit répondre aux critères suivants : La rapidité de traitement :

En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.

La performance : Une application doit être avant tout performante c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale.

V. Présentation de projet :

Le cabinet d'avocat est un bureau d'avocat, de conseil juridique, dirigé par Maître Mebarki Hamza, avocat à la cour depuis l'année 2011.

Le Cabinet veille à offrir des services de qualité, en adoptant les nouvelles technologies, permettant faciliter la communication avec ses clients. A travers le portail électronique de bureau << CIPHER>>, notre avocat n'a qu'à faire entrer son nom d'utilisation et son mot de passe, ainsi il pourra voir son tableau de bord.

Ce service lui permet de prendre connaissance de ces prochains rendez-vous ainsi, les audiences à venir en outre ces affaires actives ainsi les tâches les plus récentes.

L'application <<CIPHER>> permet à l'avocat de lire et modifier la situation financière et les gains ainsi il peut ajouter des nouvelles affaires et des tâches à suivre en outre il peut activer l'alarme pour lui rappeler de ces tâches.

En ce qui concerne sa secrétariat l'avocat peut leur créer un compte et entrer ces informations ainsi il peut spécifier leurs pouvoirs et leurs rôles dans l'application.

A propos des clients, ils peuvent avoir un compte dans l'application à partir de son compte ils peuvent prendre connaissance de ces dossiers et audiences à venir ainsi que les jugements prononcés.

Afin d'assurer la transparence des opérations financières, le client peut, à tout moment connaître la situation financière auprès du cabinet et ce via l'espace facturation du client, dans le souci d'offrir un service de qualité.

VI. Conclusion

Ce chapitre présente une phase indispensable pour l'étude et l'analyse de notre application. Nous avons défini les différents besoins fonctionnels et non fonctionnels. Nous entamerons dans le chapitre suivant la conception de cette application qui comporte les diagrammes de cas d'utilisation détaillés, les différents diagrammes de séquence et le diagramme de classe.

Chapitre 3

Conception

I. Introduction

La phase de la conception est la phase initiale de la création et de la mise en œuvre de notre projet. En fait, elle représente une étape importante de réflexion dans le cycle de développement logiciel après la phase de l'analyse et de spécification.

Dans ce chapitre, nous allons présenter en détails la conception du projet à travers les diagrammes UML suivants : les diagrammes des cas d'utilisation, les diagrammes de séquence, le diagramme de classes et les diagrammes d'activités.

II.UML :

II.1. Définition de UML :

UML, c'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un **langage visuel** constitué d'un ensemble de schémas, appelés des **diagrammes**, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour **représenter** le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. [8]

II.2 Définition d'un cas d'utilisation :

Le diagramme des cas d'utilisation apporte une vision utilisateur et absolument pas une vision informatique. Il ne nécessite aucune connaissance informatique et l'idéal serait qu'il soit réalisé par le client.

Le diagramme des cas d'utilisation n'est pas un inventaire exhaustif de toutes les fonctions du système. Il ne liste que des fonctions générales essentielles et principales sans rentrer dans les détails. [9]

II.2.1. Représentation de cas d'utilisation :

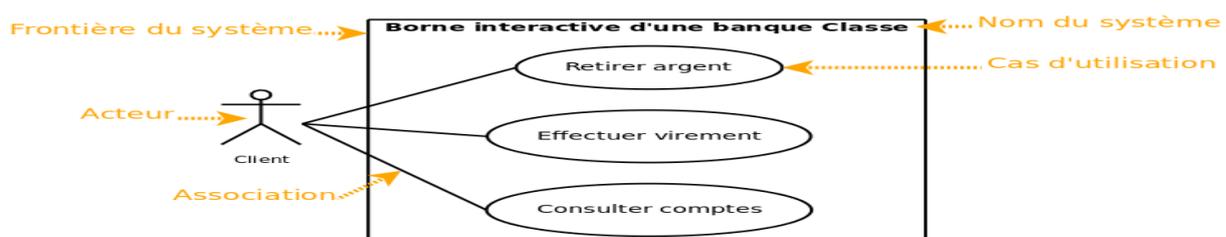


Figure III.1 : Représentation simple de cas d'utilisation.

II.2.2. Représentation de cas d'utilisation de gestion de cabinet avocate (Cipher) :

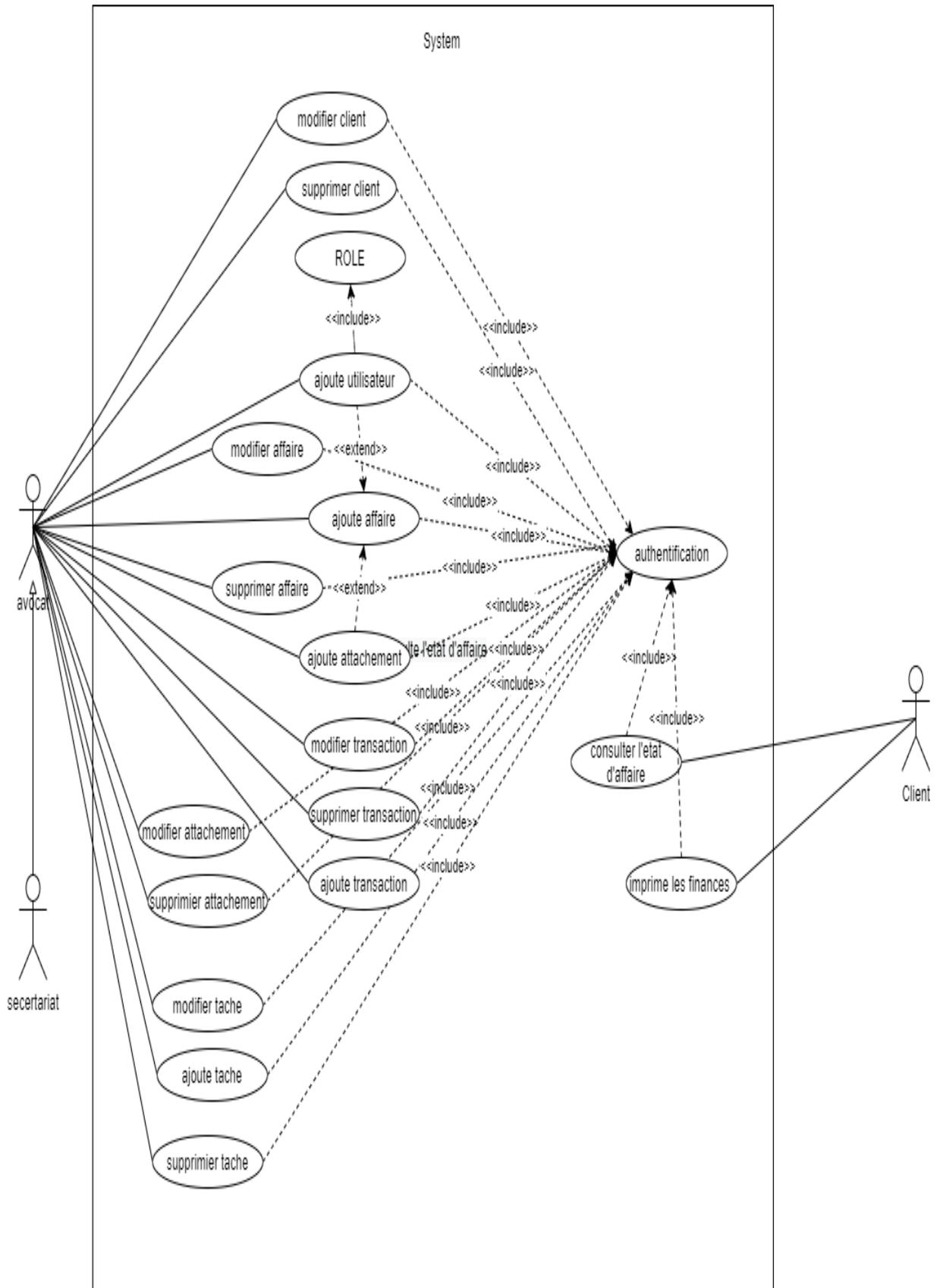


Figure III.2 : Diagramme de cas d'utilisation général de système.

II.3 Définition de diagramme de séquence :

Un diagramme de séquence est un diagramme UML (Unified Modeling Language) qui représente la séquence de messages entre les objets au cours d'une interaction. Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.

Les diagrammes de séquence représentent la séquence de messages transmis entre des objets. Ils peuvent également représenter les structures de contrôle entre des objets. Par exemple, les lignes de vie dans un diagramme de séquence pour un scénario de banque peuvent représenter un client, un guichetier ou un responsable d'agence. Les communications entre le client, le guichetier et le responsable sont représentés par les messages entre ces derniers. Le diagramme de séquence représente les objets et les messages entre ces objets.[10]

II.3.1 représentations d'un diagramme de séquence :

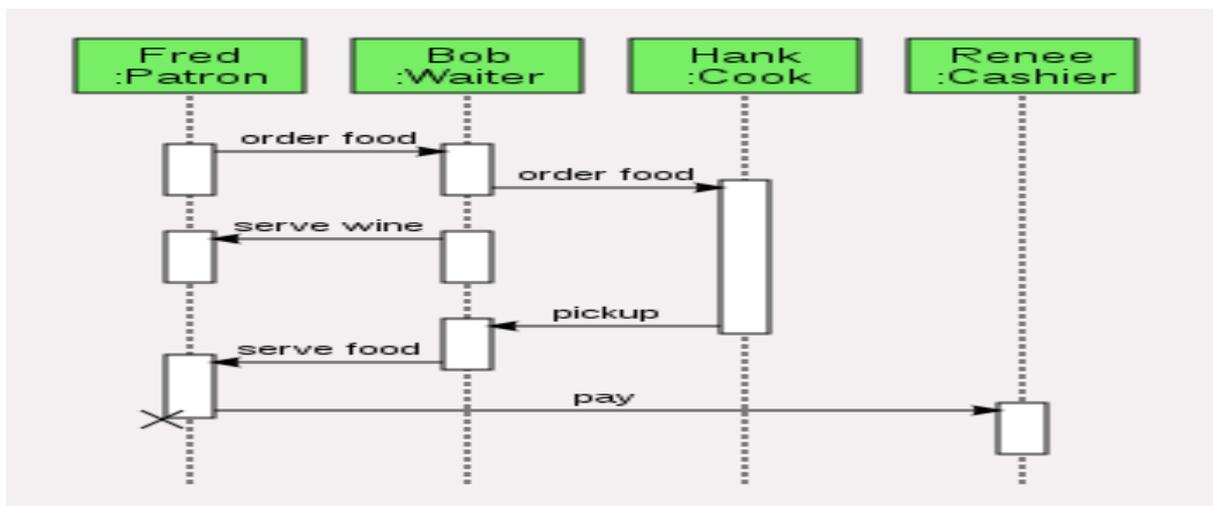
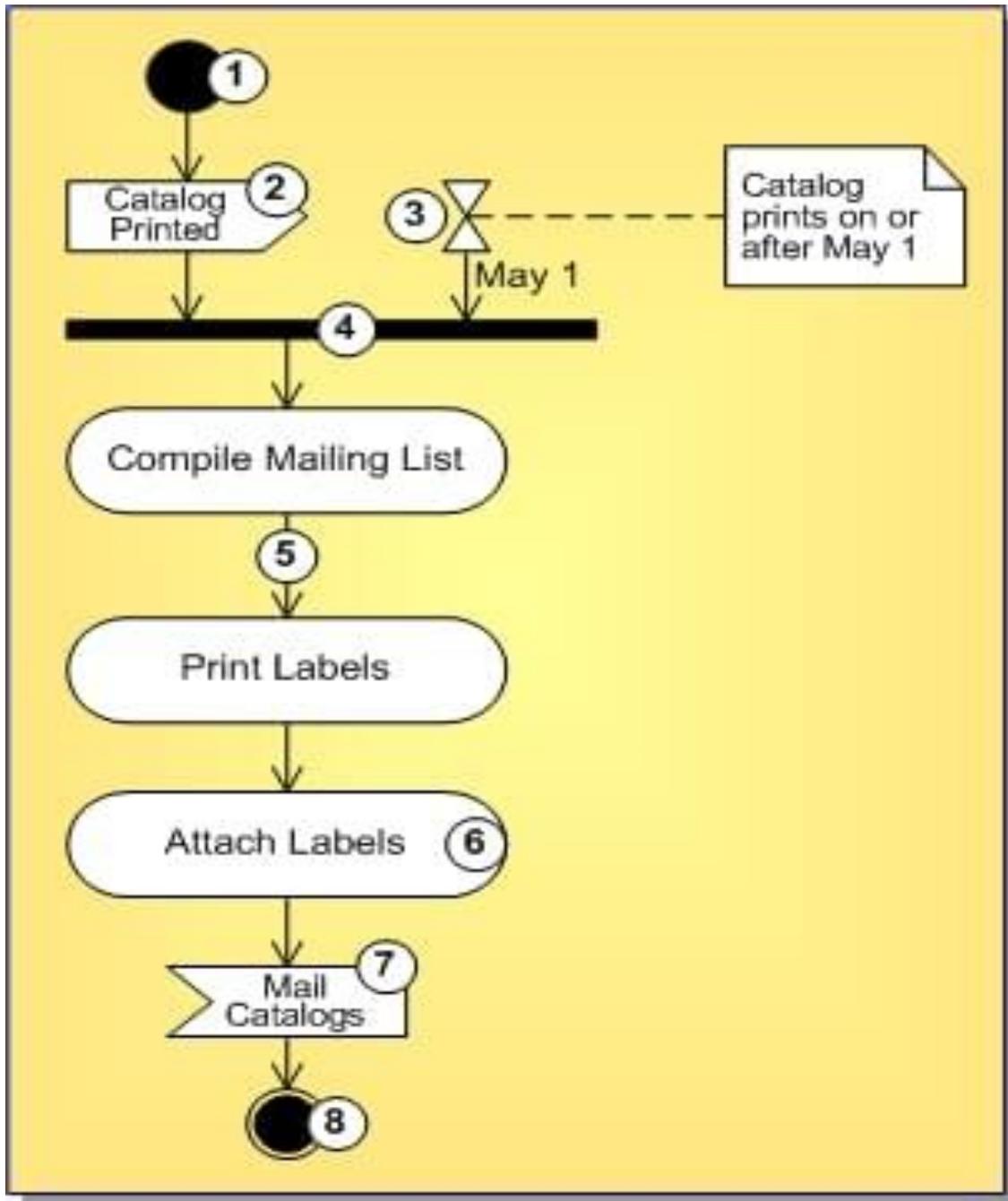


Figure III.3 : représentation d'un diagramme de séquence.

II.4 Définition du diagramme d'activité :

Un diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution. Les actions sont les unités de base du comportement du système. Un diagramme d'activités permet de grouper et de dissocier des actions. Si une action peut être divisée en plusieurs actions en séquence, vous pouvez créer une activité les représentant.[11]

II.4.1 représentation d'un diagramme d'activité :



- | | |
|----------------------------|-----------------------|
| ① Initial State | ⑤ Control Flow |
| ② Send Signal Action | ⑥ Action |
| ③ Accept Time Event Action | ⑦ Accept Event Action |
| ④ Join | ⑧ Final State |

Figure III.4 : représentation d'un diagramme d'activité.

III. Description UML :

III.1 Description des cas d'utilisation :

III.1.1 Cas d'utilisation « authentification » :

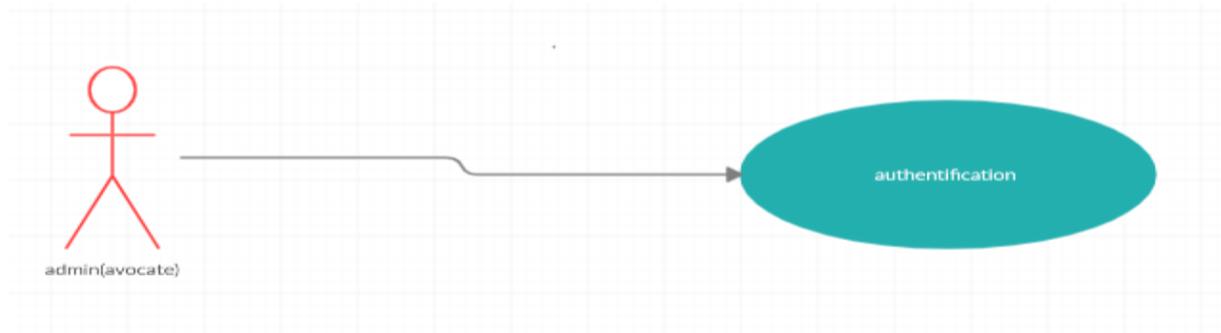


Figure III.5 : Diagramme de cas d'utilisation « authentification ».

- **But :** Ce cas d'utilisation permet au l'avocate de connecté à l'application.
- **Les acteurs :** Admin (avocate).
- **Description textuelle :** pour connecte dans application CIPHER le système demande l'authentification et afficher le formulaire de login, l'avocat entre les informations relatives à authentification (email, mot de passe) et après la confirmation de ces informations, si authentification validé le system afficher la page de tableau de bord, si non le system affiche un message incorrect de email ou mot de passe et affiche le formulaire de login .

➤ **Les scénarios :**

- ✓ Authentification : permet d'authentifie selon l'enchainement suivant :
 1. Le system demande d'authentification et afficher le formulaire d'authentifie.
 2. L'avocate remplit les champs de formulaire (email, mot de passe).
 3. Le system vérifie les champs introduits.
 - 3.1. Si authentification valide le system afficher le tableau de bord.
 - 3.2. Si non le système affiche un message incorrect d'email ou mot de passe et affiche le formulaire de login.

➤ Le scénario « authentification »

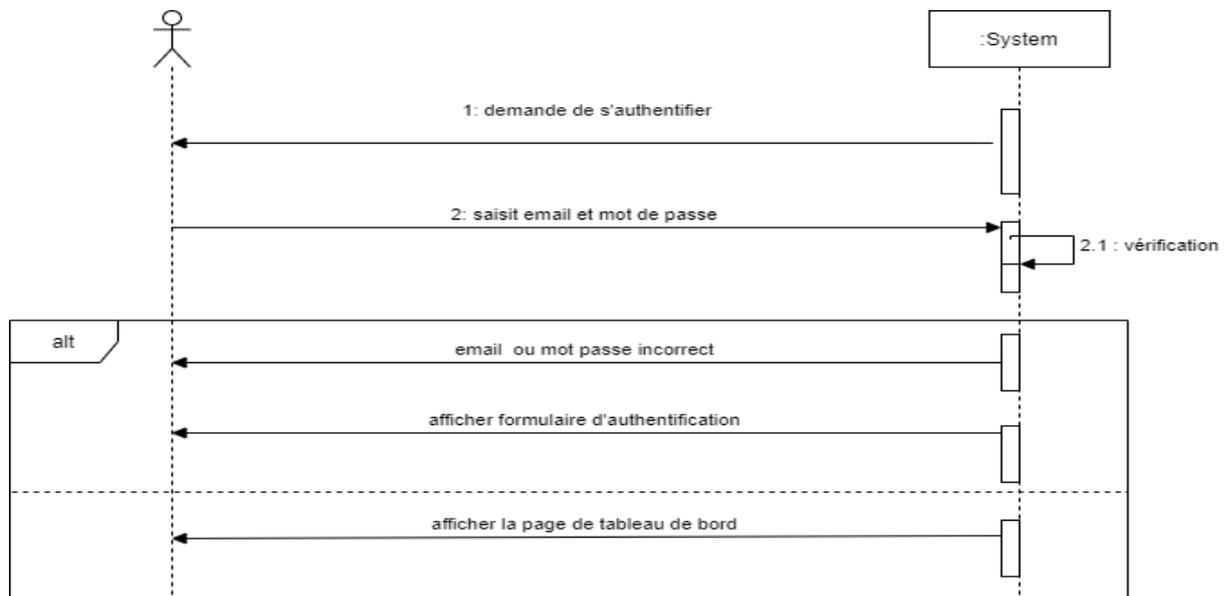


Figure III.6 : Diagramme de séquence du scénario « authentification ».

III.1.2. Cas d'utilisation « gestion d'affaire GA » :

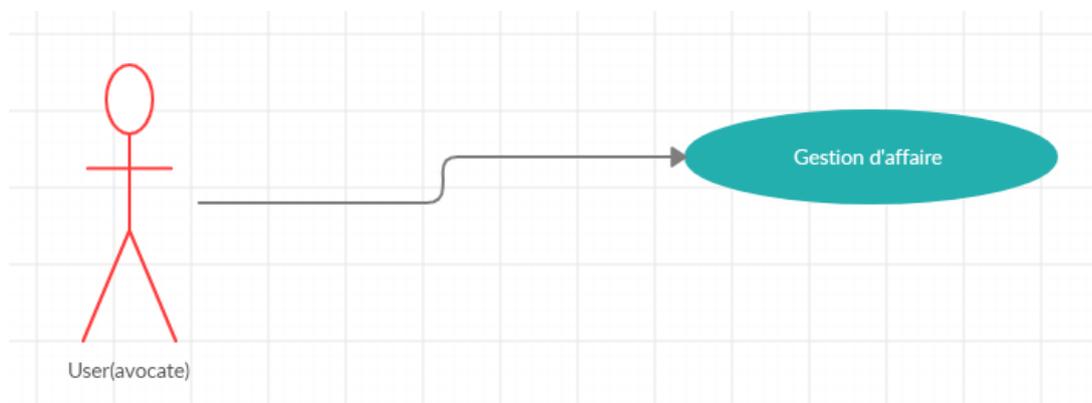


Figure III.7 : Diagramme de cas d'utilisation « gestion d'affaire ».

- **But :** Ce cas d'utilisation permet au technicien de créer des affaires.
- **Les acteurs :** Admin (avocat).
- **Description textuelle :** pour ajouter une nouvelle affaire l'avocat demande au système de l'ajouter. Le système affiche un formulaire ensuite, l'avocat entre les informations relatives à l'affaire (numéro d'affaire, nom d'affaire, prix, nom et prénom de client, nome d'adversaire, les mots clés, emplacement de tribunal, décision judiciaire, la date de décision judiciaire, description d'affaire) et après la confirmation de ces informations, le système enregistre l'affaire.

➤ Les scénarios :

1- Création d'une affaire : permet de créer une nouvelle affaire selon l'enchaînement suivant :

4. L'avocat demande de créer une nouvelle affaire
5. Le système affiche un formulaire de création une affaire.
6. L'avocate remplit les champs de formulaire (nom d'affaire, numéro d'affaire, description)
7. Le système vérifie les champs introduits, sauvegarde l'information saisit, et affiche choix pour ajoute contact et attachements.
8. L'avocat choisi et sauvegarde les informations
9. Le système affiche un message correspondant a la réussite de l'opération.

2- Modification d'une affaire : permet de corriger les erreur et ajoute nouvelle information selon l'enchaînement suivant :

1. L'avocat sélectionne le champ d'archive.
2. Le système affiche la liste de tous les affaires.
3. L'avocate sélectionné une affaire dans la liste.
4. Le système affiche les informations d'affaire correspondant.
5. L'avocat modifie les champs souhaités et valide l'opération.
6. Le système vérifie les champs introduits, mis à jour des informations d'affaire et afficher un message de confirmation de modification.

3- Suppression d'une affaire : permet de supprime une affaire selon l'enchaînement suivant :

1. L'avocat sélectionne le champe d'archive.
2. Le système affiche la liste de tous les affaires.
3. L'avocat sélectionne une affaire dans la liste.
4. L'avocat supprime l'affaire sélectionné.
5. Le système affiche une fenêtre de confirmation de la suppression.
6. L'avocat confirme la suppression.
7. Le système supprime et mis a jour et affiche un message de confirmation de suppression.

➤ Description Diagramme d'activité « Gestion d'affaires »

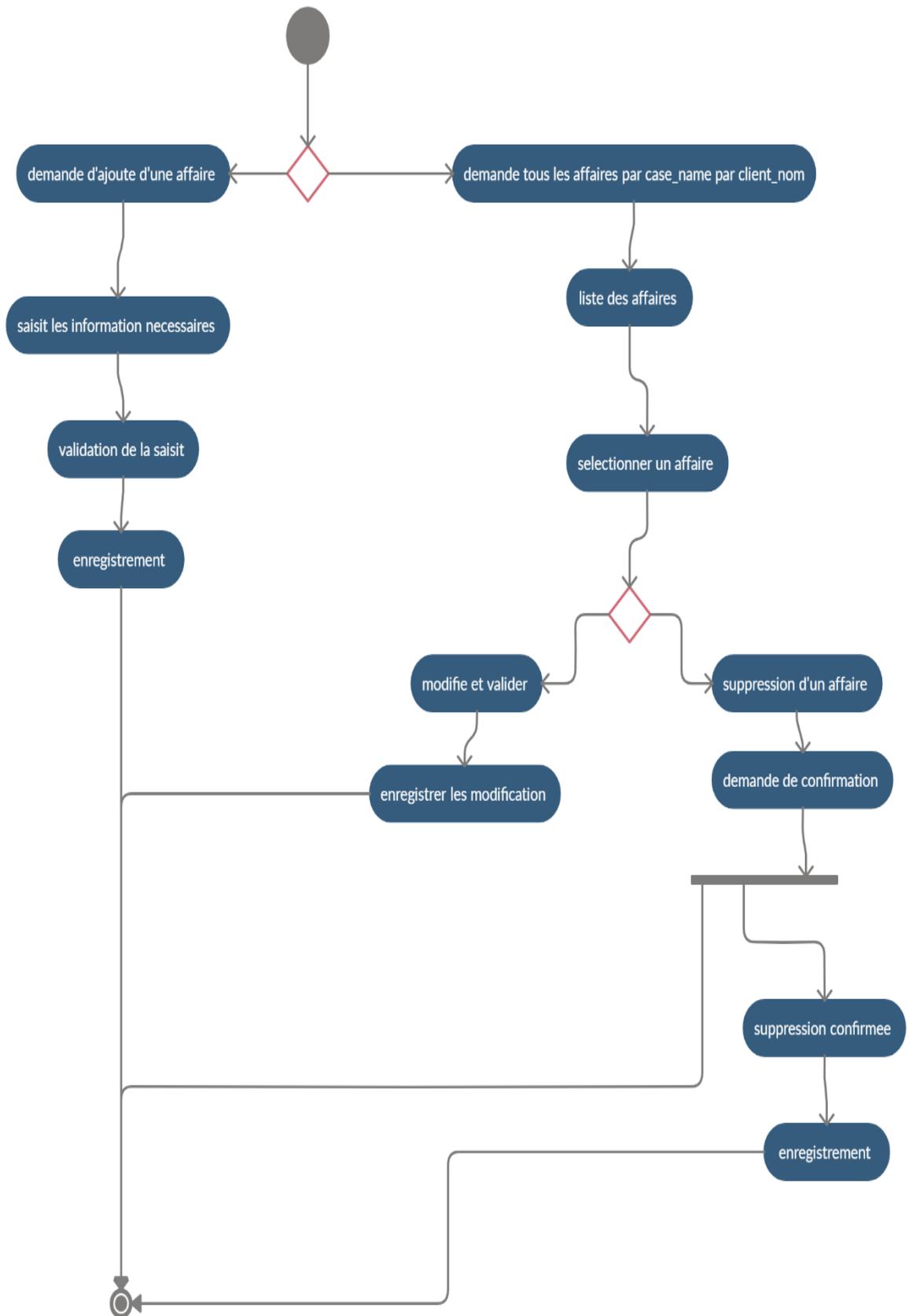


Figure III.8 : Diagramme d'activité « Gestion d'affaire ».

➤ Le scénario « création d'un affaire »

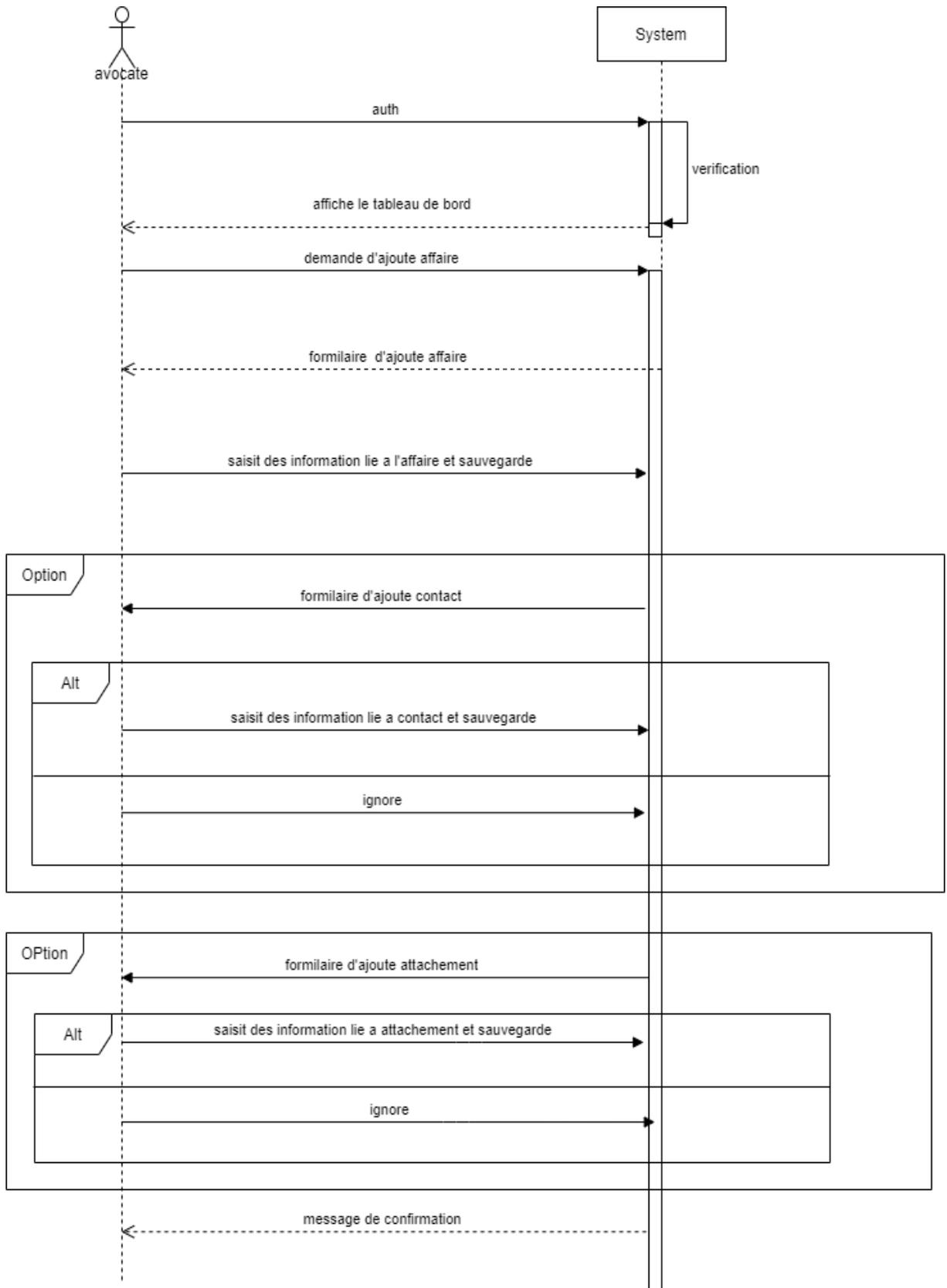


Figure III.9 : Diagramme de séquence du scénario « créer un affaire ».

III.1.3. Cas d'utilisation « gestion de contact GC » :

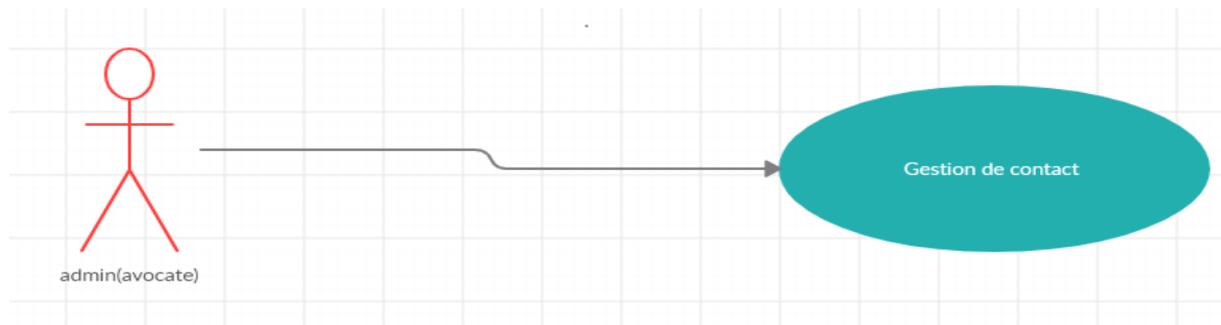


Figure III.10 : Diagramme des cas d'utilisation « gestion des contact ».

- **But :** Ce cas d'utilisation permet à l'avocate d'ajoute des contacts.
- **Les acteurs :** Admin (avocat).
- **Description textuelle :** pour ajouter un nouveau contact l'avocat sélectionné le champ de contact et puis demande au système de l'ajouter. Le système affiche un formulaire ensuite, l'avocat entre les informations relatives à contact (nom et prénom de contact, nom du père, nom et prénom de la mère, numéro de téléphone, la date de naissance, l'adresse, emplacement de tribunal, décision judiciaire, la date de décision judiciaire, description d'affaire) et après la confirmation de ces informations, le système enregistre l'affaire.

➤ **Les scénarios :**

1- Création d'un contact : permet de créer un nouvelle affaire selon l'enchaînement suivant :

1. L'avocat demande de créer un nouveau contact
2. Le system affiche un formulaire de création d'un contact.
3. L'avocat remplit les champs de formulaire (nom et prénom de contact, email, numéro de téléphone, description)
4. Le système vérifie les champs introduits, sauvegarde des informations saisis
5. Le système affiche un message correspond a la réussite de l'opération.

2- Modification d'un contact : permet de corrigé les error et ajoute des nouvelle informations selon l'enchaînement suivant :

1. L'avocat sélectionne le champe de contacts.
2. Le système affiche la liste de tous les contacts.

3. L'avocat sélectionne un contact dans la liste.
4. Le système affiche les informations des contacts correspondant.
5. Le technicien modifie les champs souhaités et valide l'opération.
6. Le système vérifie les champs introduits, mis à jour des informations des contacts et affiche un message de confirmation et de modification.

3- Suppression d'un contact : permet de supprimer une affaire selon l'enchaînement suivant :

1. L'avocat sélectionne le champ de contacts.
2. Le système affiche la liste de tous les contacts.
3. L'avocat sélectionne un contact dans la liste.
4. L'avocate supprime le contact sélectionné.
5. Le système affiche la fenêtre de confirmation de la suppression.
6. L'avocat confirme la suppression.
7. Le système supprime et met à jour et affiche un message de confirmation de suppression.

➤ Description Diagramme d'activité « Gestion de contact »

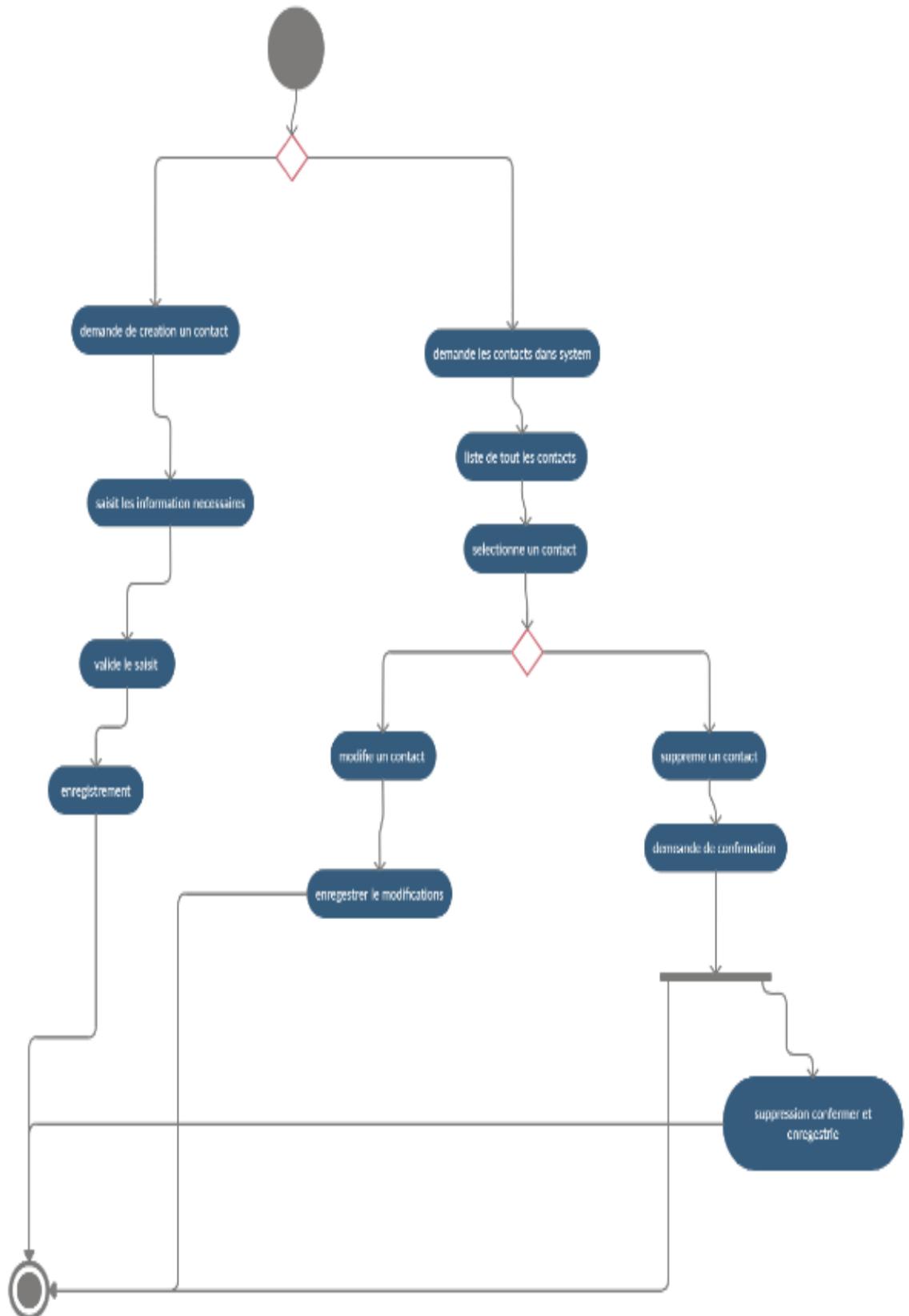


Figure III.11 : Diagramme d'activité « Gestion de contact ».

➤ Le scénario « création d'un contact (client) » :

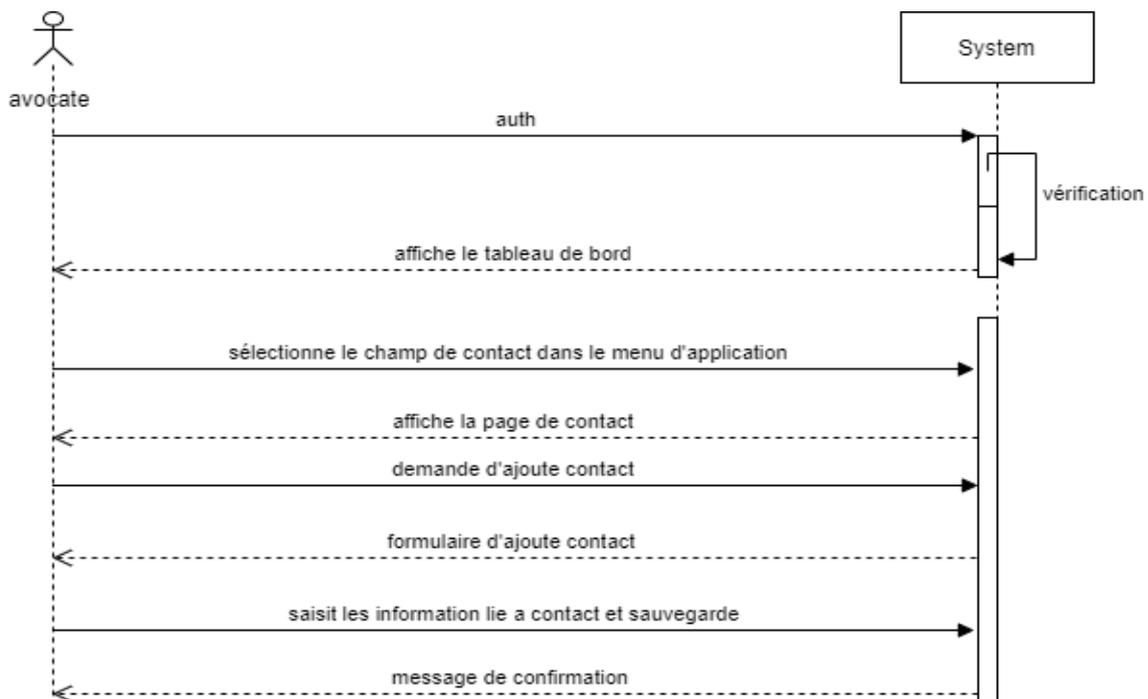


Figure III.12 : Diagramme de séquence du scénario « ajoute un contact ».

➤ Le scénario « modification d'un contact (client) » :

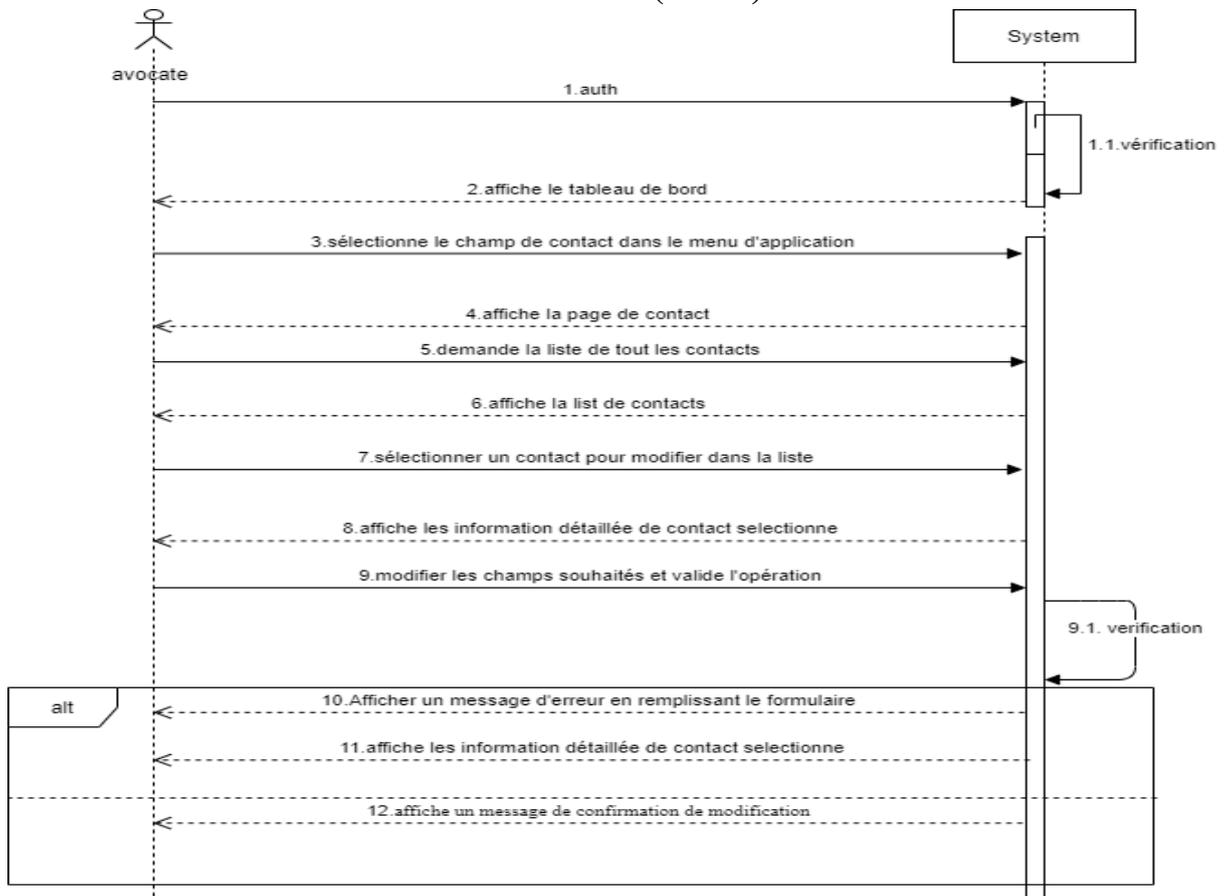


Figure III.13 : Diagramme de séquence du scénario « modifie un contact »

➤ Le scénario « suppression d'un contact (client) » :

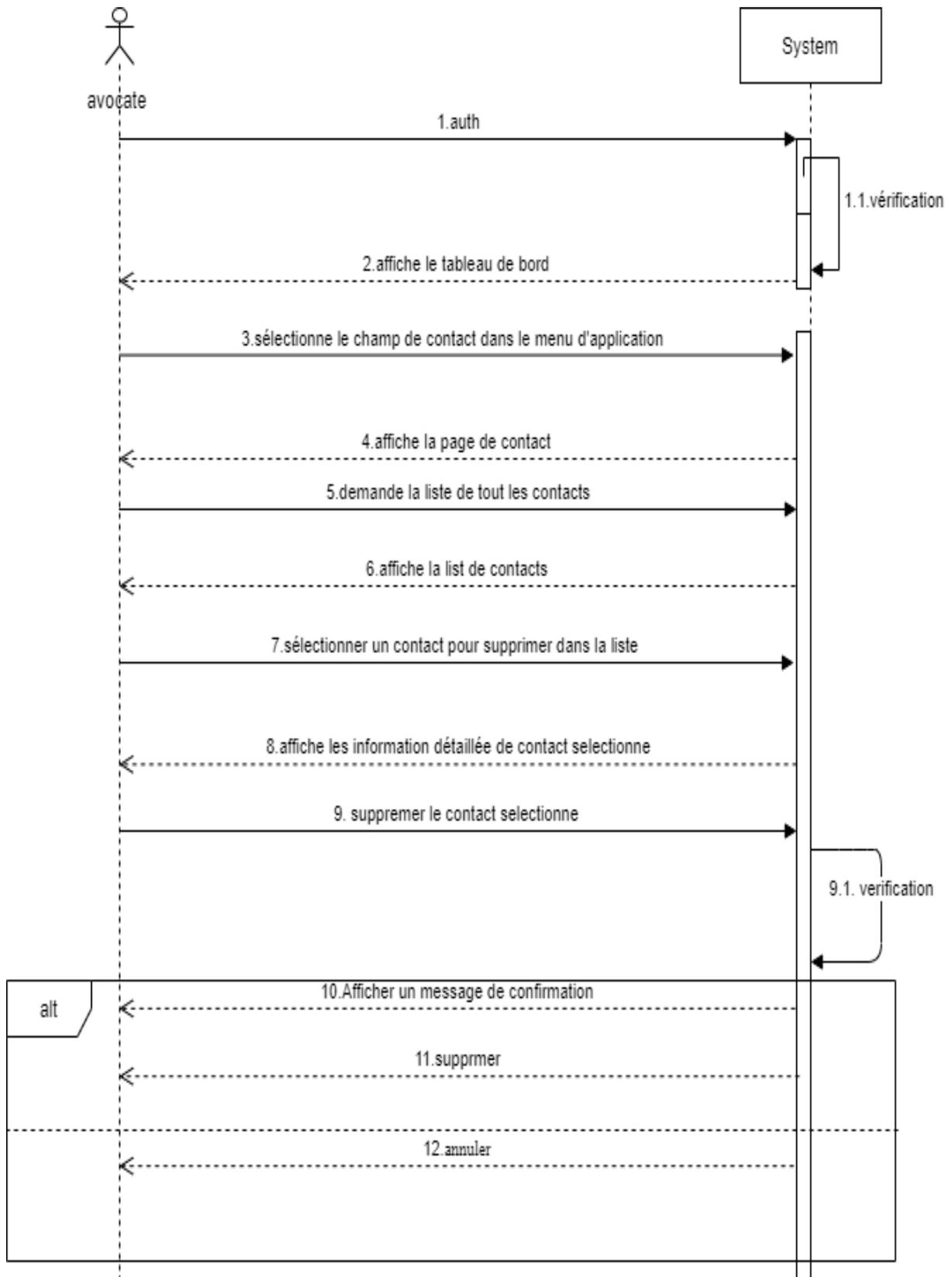


Figure III.13 : Diagramme de séquence du scénario « supprimer un contact »

III.1.4. Cas d'utilisation « gestion d'utilisateur (employé) GU » :

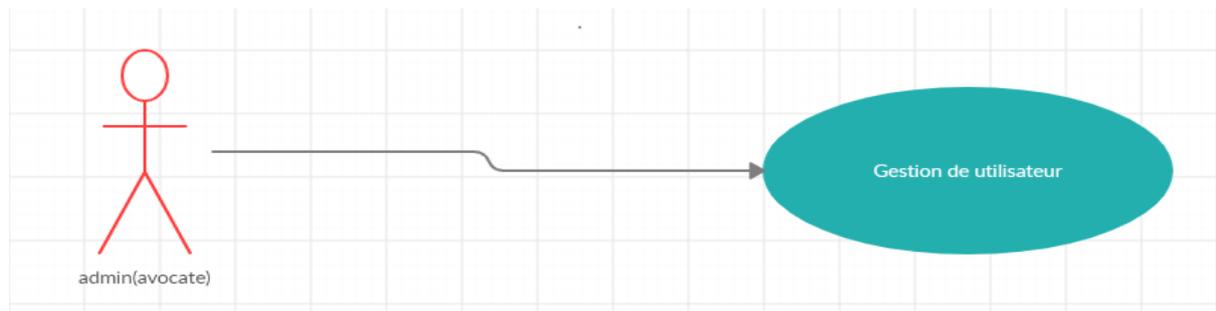


Figure III.13 : Diagramme des cas d'utilisation « gestion d'utilisateur (employé) GU ».

- **But :** Ce cas d'utilisation permet à l'avocat d'ajoute des utilisateurs (employés).
- **Les acteurs :** Admin (avocat).
- **Description textuelle :** pour ajouter un nouvel utilisateur l'avocat sélectionne le champ d'utilisateur (ROLE) et puis demande au système d'ajouter. Le système affiche un formulaire ensuite, l'avocat entre les informations relatives au contact (nom et prénom d'utilisateur, numéro de téléphone, email) et après la confirmation de ces informations, le système enregistre l'utilisateur.

➤ **Les scénarios :**

1- Création d'un utilisateur : permet de créer un nouveau utilisateur selon l'enchaînement suivant :

1. L'avocat demande de créer un nouveau utilisateur.
2. Le système affiche un formulaire de création d'un utilisateur.
3. L'avocate remplit les champs de formulaire (nom et prénom d'utilisateur, numéro de téléphone, email, description).
4. L'avocate sélectionne les rôles d'utilisateur et valide.
5. Le système vérifie les champs introduits, sauvegarde l'information saisit,
6. Le système affiche un message correspondre à la réussite de l'opération.

2- Modification d'un utilisateur : permet de corriger les erreurs et ajoute des nouvelles informations selon l'enchaînement suivant :

1. L'avocate sélectionne le champ utilisateurs.
2. Le système affiche la liste de tous les utilisateurs.
3. L'avocate sélectionne un utilisateur dans la liste.
4. Le système affiche les informations d'utilisateur correspondant.
5. L'avocat modifie les champs souhaités et valide l'opération.

6. Le système vérifie les champs introduits, mise à jour des informations d'affaire et affiche un message de confirmation de modification.

3- Suppression d'un utilisateur : permet de supprimer un utilisateur selon l'enchaînement suivant :

1. L'avocat sélectionne le champ de utilisateurs.
2. Le système affiche la liste de tous les utilisateurs.
3. L'avocat sélectionne un utilisateur dans la liste.
4. L'avocat supprime l'utilisateur sélectionné.
5. Le système affiche fenêtre de confirme la suppression.
6. L'avocate confirme la suppression.
7. Le système supprime et mise à jour et affiche un message de confirmation de suppression.

➤ **Le scénario « création d'un utilisateur (employé) » :**

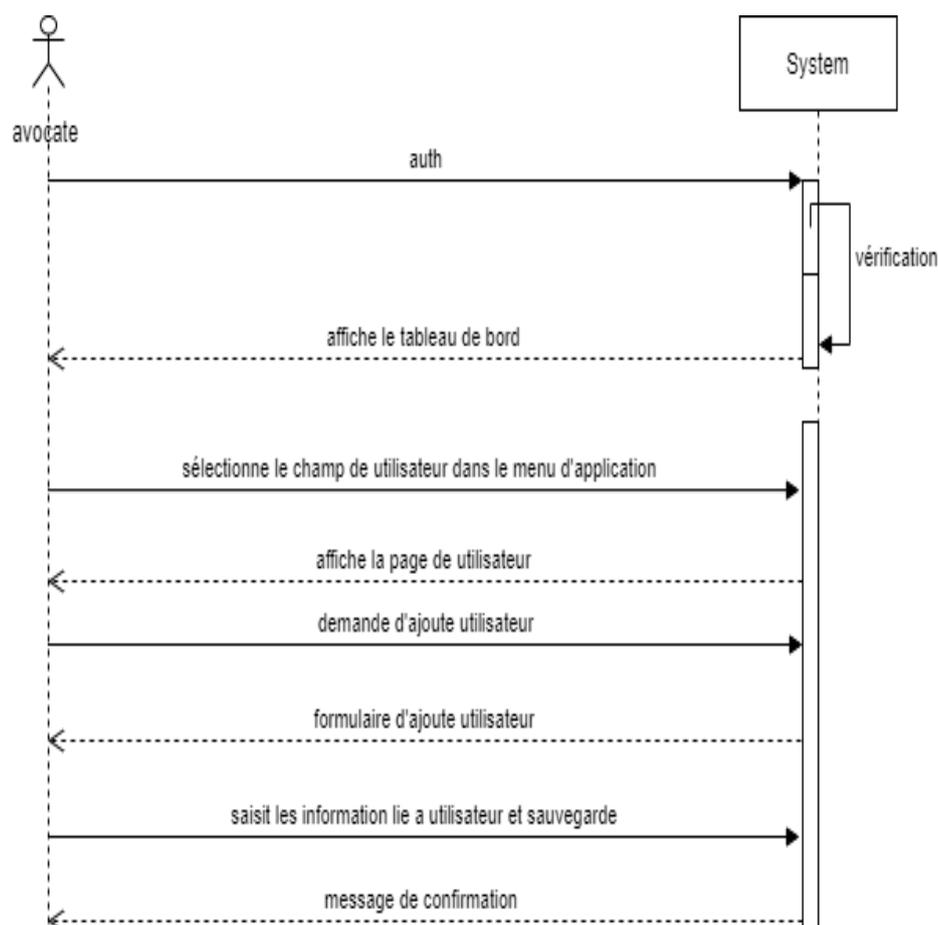


Figure III.14 : Diagramme de séquence du scénario « ajoute un utilisateur ».

➤ Le scénario « modification d'un utilisateur (employé) » :

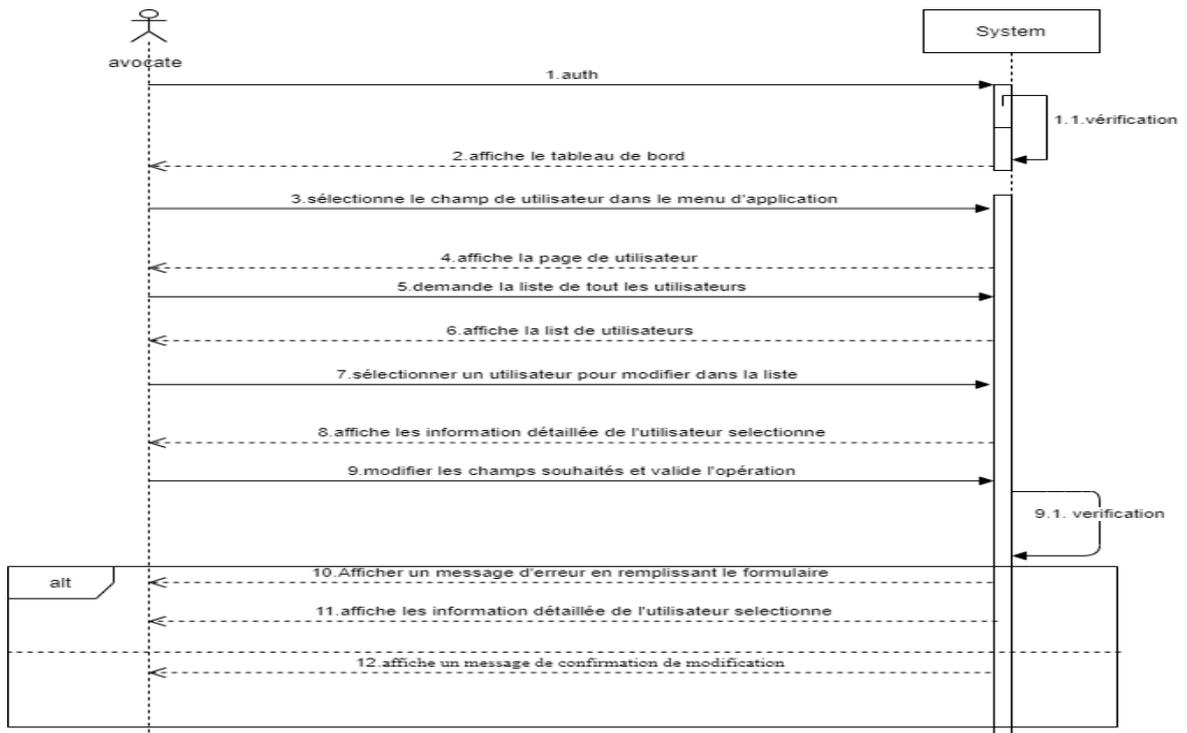


Figure III.15 : Diagramme de séquence du scénario « modifier un utilisateur ».

➤ Le scénario « suppression d'un utilisateur (employé) » :

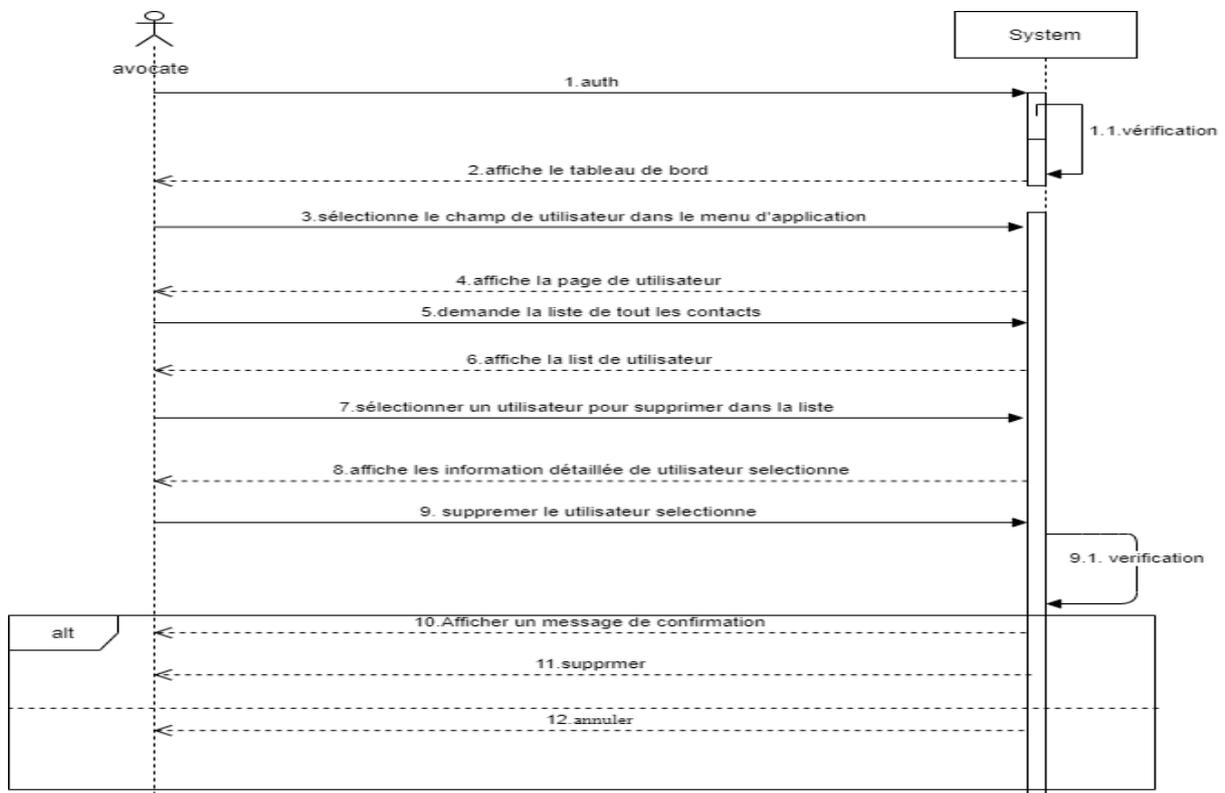


Figure III.15 : Diagramme de séquence du scénario « supprimer un utilisateur ».

III.1.5. Cas d'utilisation « gestion des tâches GT » :

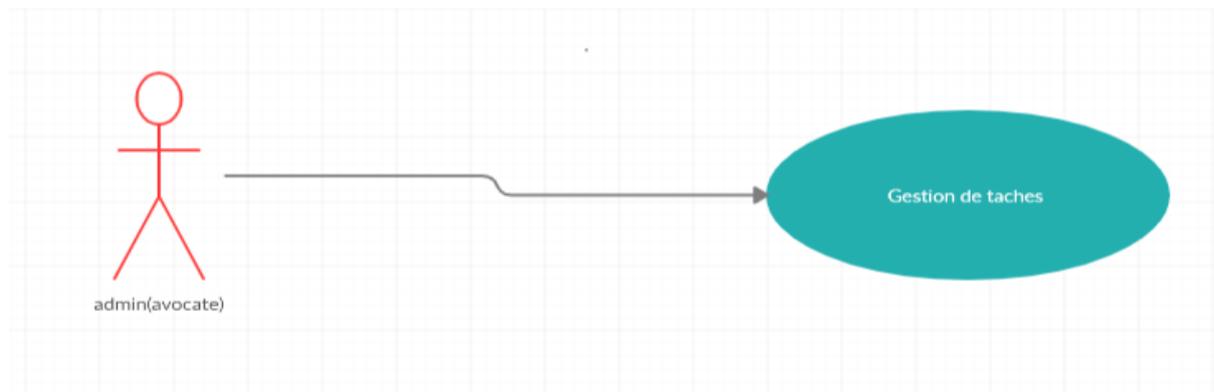


Figure III.16 : Diagramme des cas d'utilisation « gestion des Taches GT ».

- **But :** Ce cas d'utilisation permet à l'avocat d'ajouter une tâche liée à l'affaire.
- **Les acteurs :** Admin (avocat).
- **Description textuelle :** pour ajouter une nouvelle tâche l'avocat sélectionne le champ de tâches et puis demande au système de l'ajouter. Le système affiche un formulaire ensuite, l'avocat entre les informations relatives à tâche (titre et description, la date et l'alarme) et après la confirmation de ces informations, le système enregistre la tâche.

➤ **Les scénarios :**

1- Création d'une tâche : permet de créer une nouvelle tâche selon l'enchaînement suivant :

1. L'avocat demande de créer une nouvelle tâche.
2. Le système affiche un formulaire de création d'une tâche.
3. L'avocat remplit les champs de formulaire ((titre et description, la date et l'alarme)).
4. Le système vérifie les champs introduits, sauvegarde l'information saisie,
5. Le système affiche un message correspondant à la réussite de l'opération.

2- Modification d'une tâche : permet de corriger les erreurs et ajouter de nouvelles informations selon l'enchaînement suivant :

1. L'avocat sélectionne le champ de tâche.
2. Le système affiche la liste de toutes les tâches.
3. L'avocat sélectionne une tâche dans la liste.
4. Le système affiche les informations de la tâche correspondante.
5. L'avocat modifie les champs souhaités et valide l'opération.

6. Le système vérifie les champs introduits, mise à jour des informations d'affaire et affiche un message de confirmation de modification.

3- Suppression d'une tache : permet de supprimer un utilisateur selon l'enchaînement suivant :

1. L'avocat sélectionne le champ de taches.
2. Le système affiche la liste de tous les taches.
3. L'avocat sélectionne une tache dans la liste.
4. L'avocat supprime la tache sélectionnée.
5. Le système affiche fenêtre de confirmation de la suppression.
6. L'avocat confirme la suppression.
7. Le système supprime et mise à jour et affiche un message de confirmation de suppression.

➤ **Le scénario « ajoute une tache » :**

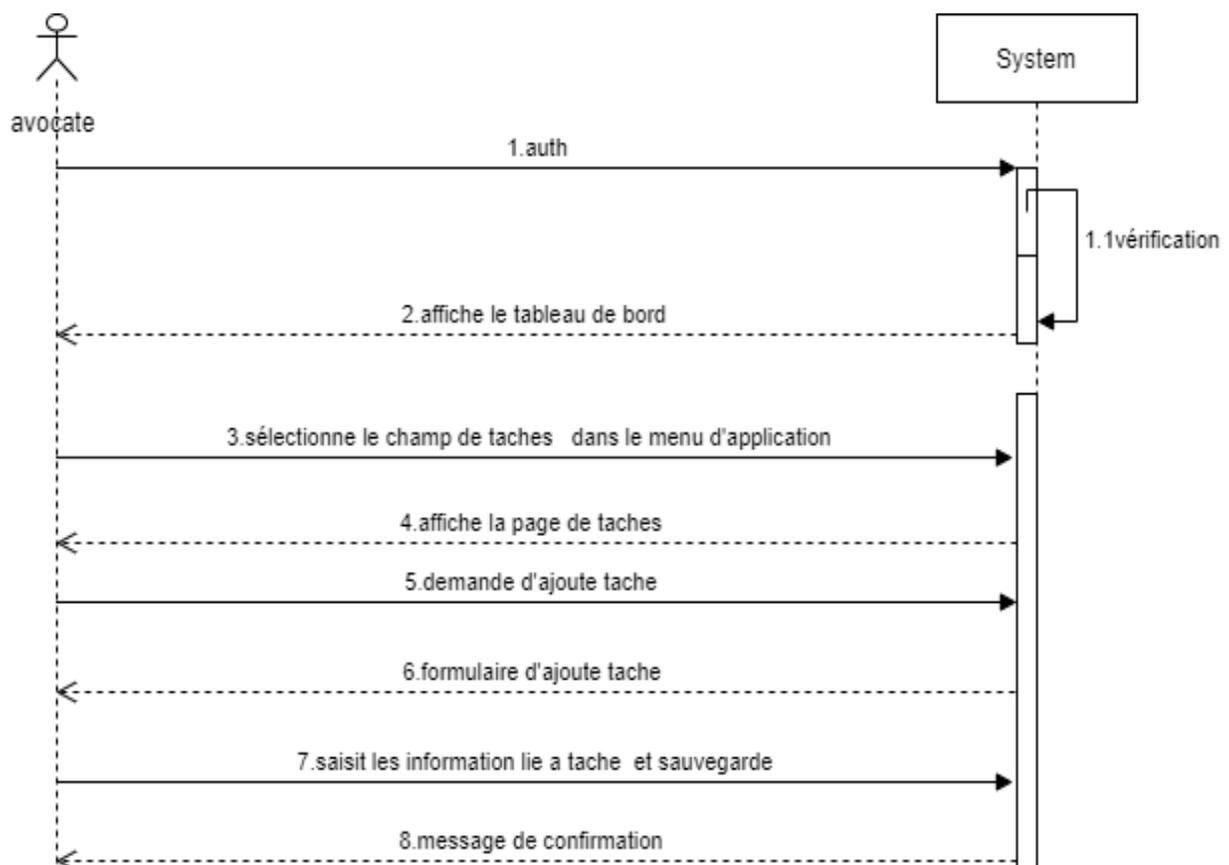


Figure III.17 : Diagramme de séquence du scénario « ajoute une tache ».

➤ Le scénario « supprimer une tâche » :

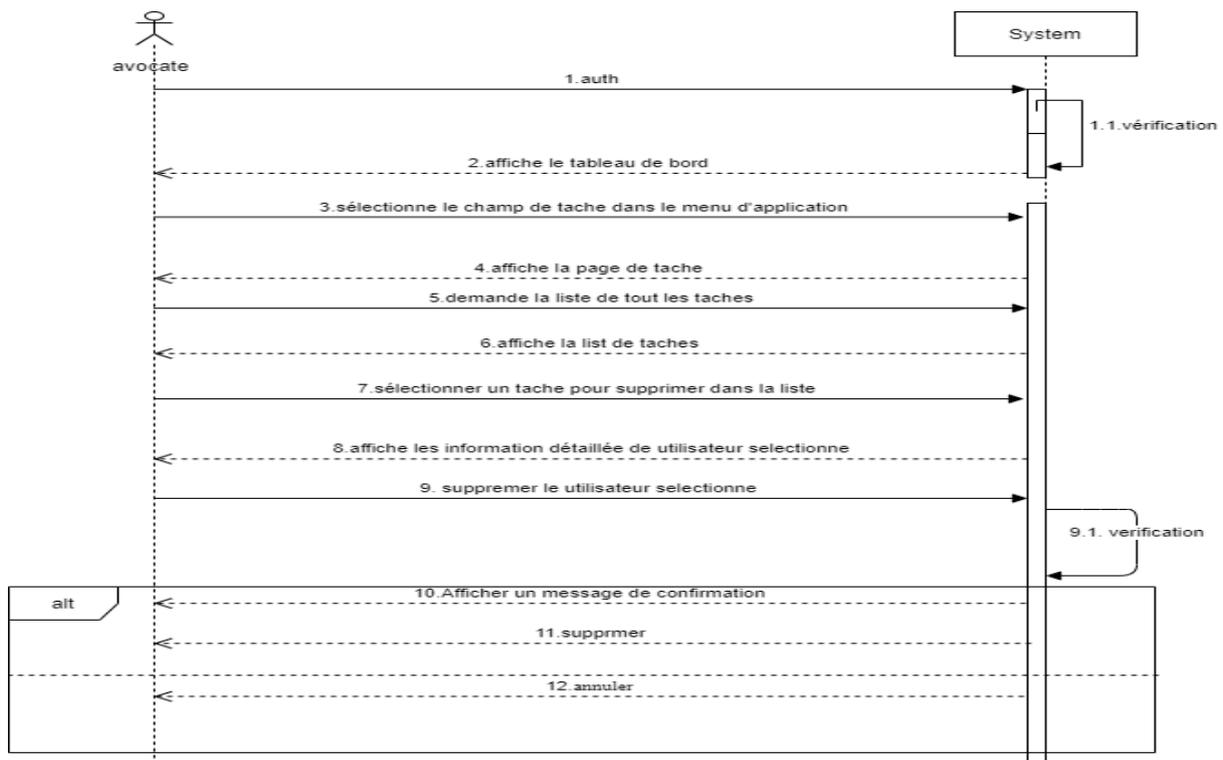


Figure III.17 : Diagramme de séquence du scénario « supprimer une tâche ».

III.1.6 Cas d'utilisation « gestion des finances GF » :

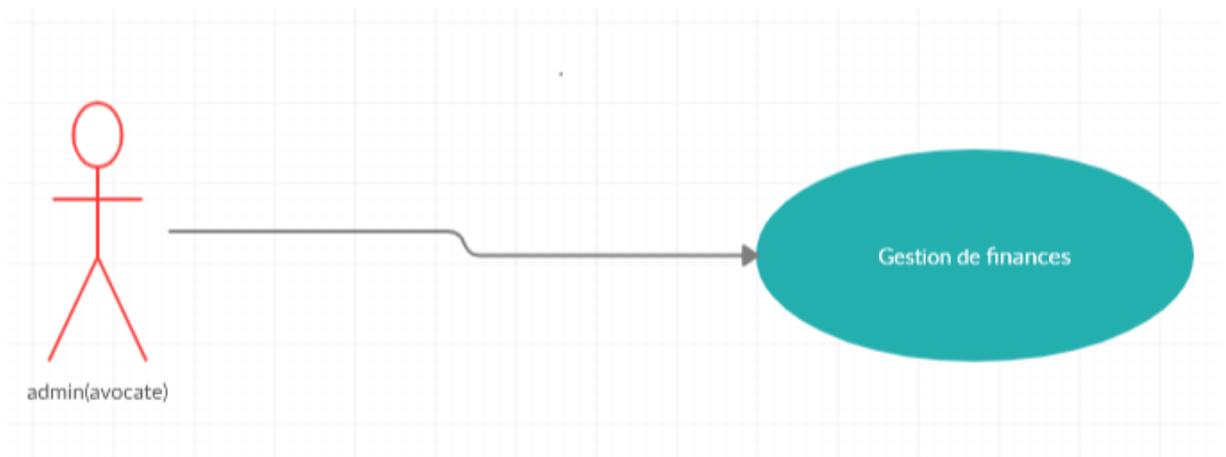


Figure III.18 : Diagramme des cas d'utilisation « gestion des finances GF ».

- **But :** Ce cas d'utilisation permet à l'avocat d'ajouter une transaction liée à l'affaire.
- **Les acteurs :** Admin (avocat).
- **Description textuelle :** pour ajouter une nouvelle transaction l'avocat sélectionne le champ de finances et puis demande au système de l'ajouter. Le système affiche un formulaire ensuite, l'avocat entre les informations relatives à finances (montante, est

reçu, date de création) et après la confirmation de ces informations, le système enregistre la transaction.

➤ **Les scénarios :**

1- Création d'une transaction : permet de créer une nouvelle transaction l'avocat selon l'enchaînement suivant :

1. L'avocat sélectionne une affaire pour ajouter une transaction.
2. L'avocat demande de créer une nouvelle transaction
3. Le système affiche un formulaire de création d'utilisateur.
4. L'avocat remplit les champs de formulaire (montante)
5. Le système vérifie les champs introduits, sauvegarde l'information saisie,
6. Le système affiche un message correspondant à la réussite de l'opération.

2- Modification d'une transaction : permet de corriger les erreurs et ajouter des nouvelles informations selon l'enchaînement suivant :

1. L'avocat sélectionne une affaire pour modifier une transaction.
2. Le système affiche la liste de toutes les transactions.
3. L'avocat sélectionne la transaction cible pour modifier
4. Le système affiche les informations d'une transaction correspondante.
5. L'avocat modifie les champs souhaités et valide l'opération.
6. Le système vérifie les champs introduits, met à jour des informations d'affaire et affiche un message de confirmation de modification.

3- Suppression d'une transaction : permet de supprimer un utilisateur selon l'enchaînement suivant :

1. L'avocat sélectionne une affaire pour modifier une transaction.
2. Le système affiche la liste de toutes les transactions.
3. L'avocat sélectionne la transaction cible pour supprimer.
4. L'avocat supprime la transaction sélectionnée.
5. Le système affiche une fenêtre de confirmation de la suppression.
6. L'avocat confirme la suppression.
7. Le système supprime et met à jour et affiche un message de confirmation de suppression.

➤ Le scénario « ajoute d'une transaction » :

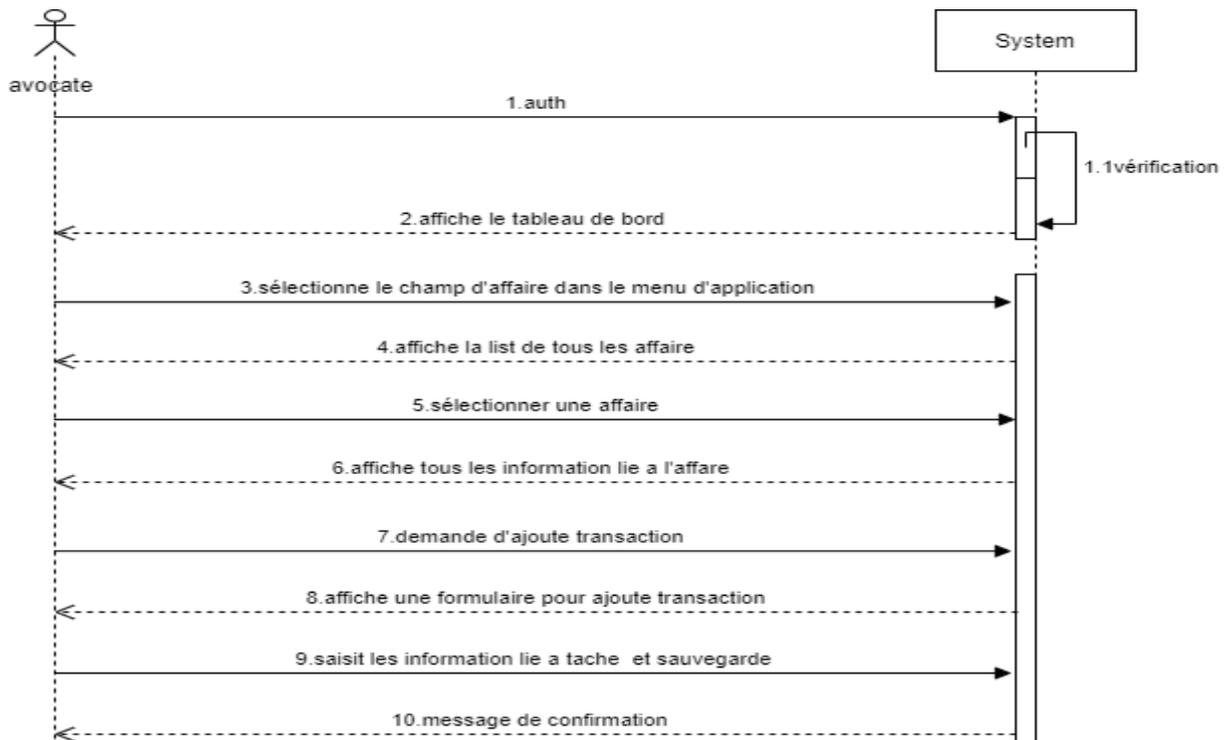


Figure III.19 : Diagramme de séquence du scénario « ajoute une transaction ».

➤ Le scénario « supprimer d'une transaction » :

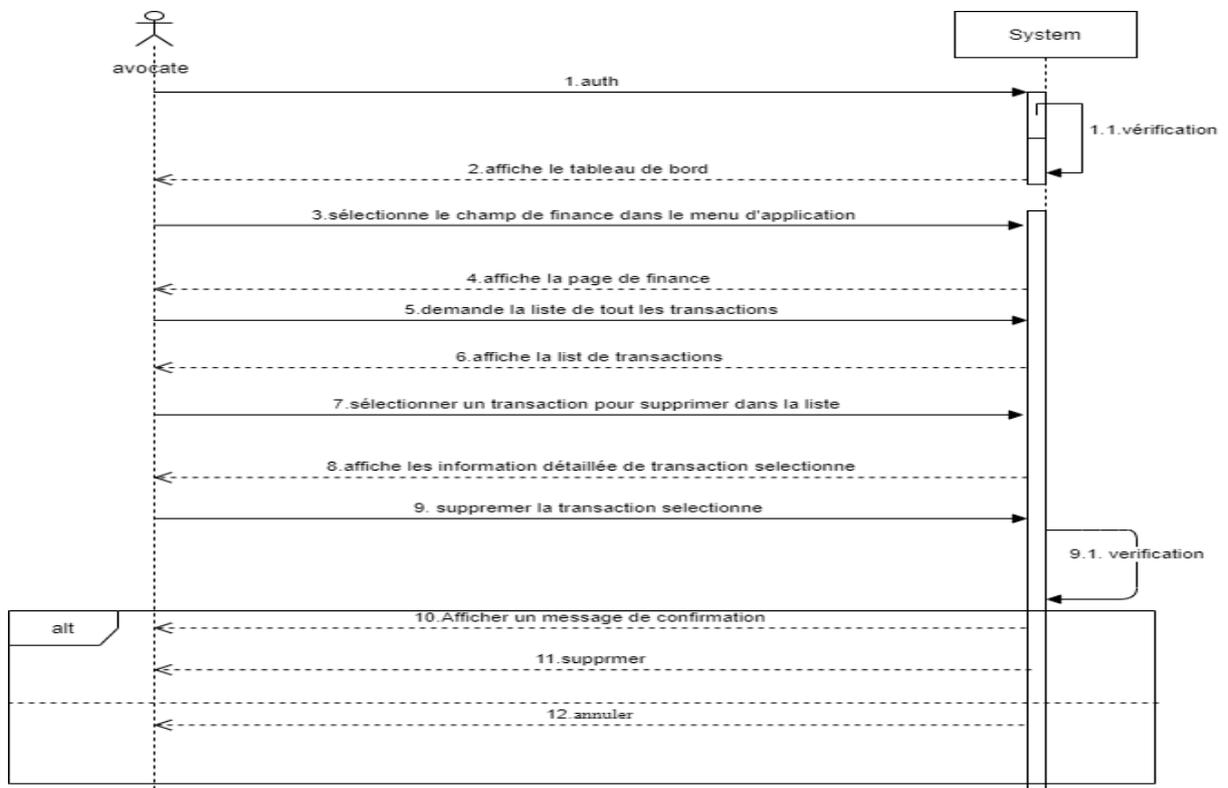


Figure III.19 : Diagramme de séquence du scénario « supprimer une transaction ».

IV. Description des diagrammes de classes :

IV.1. Définition de diagramme de classes :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classes montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il s'agit d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation. Une classe-association possède les caractéristiques des associations et des classes elle se connecte à deux ou plusieurs classes et possède également des attributs et des opérations. Une classe-association est caractérisée par un trait discontinu entre la classe et l'association.

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations ou méthodes) et de relations avec d'autres objets (les associations et les agrégations).

Une classe de conception est composée par :

Attribut chaque attribut d'une classe est le même pour chaque instance de cette classe.

Méthodes elle définit le comportement d'une classe elle-même, et non le comportement de ses instances qui peut être différent.[12]

IV.2. Le diagramme de classes

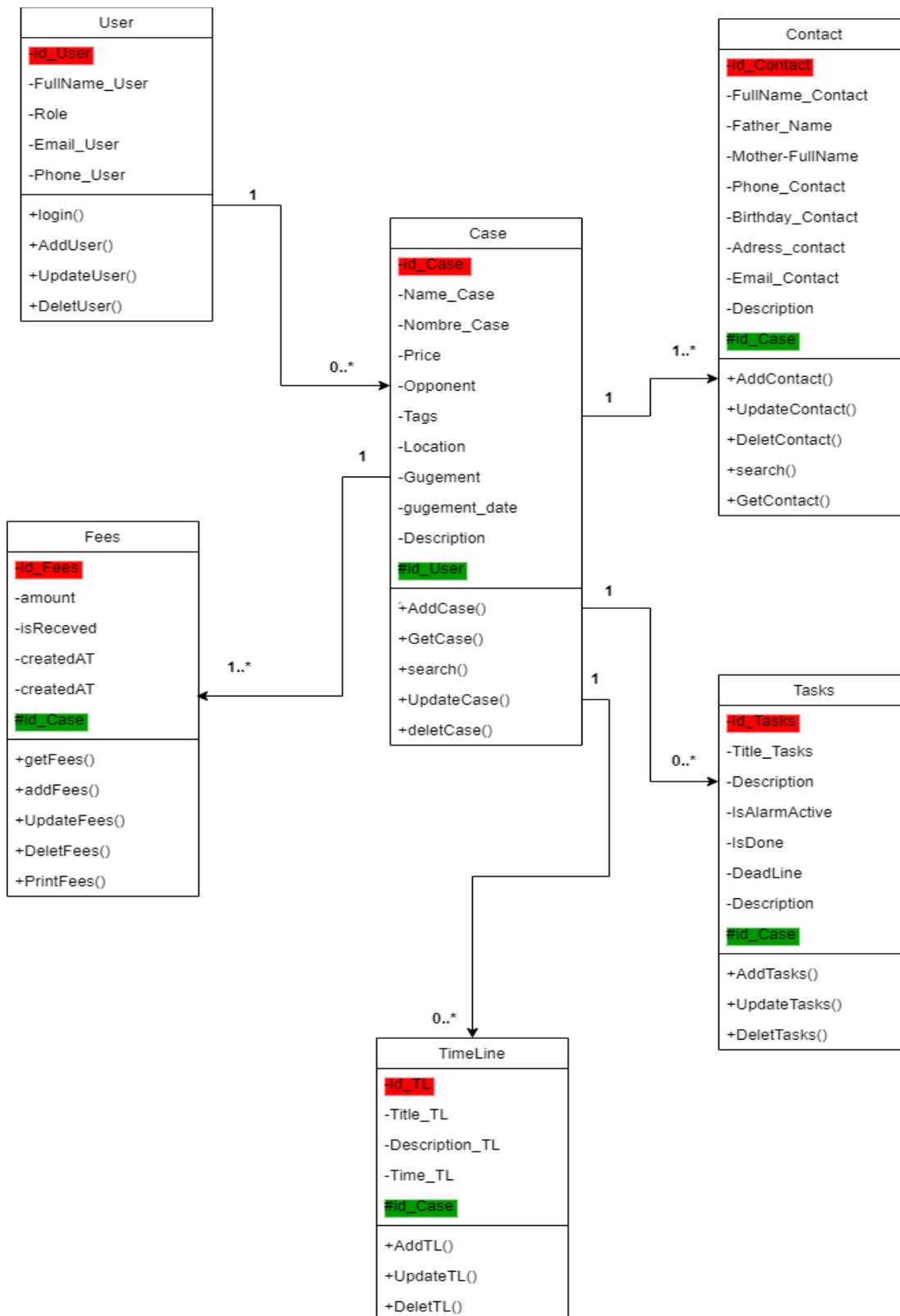


Figure III.20 : Diagramme de classe.

IV.3. Les attributs des classes :

| Class | Attributs | Désignation |
|--------------------------|--|---|
| Case (affaire) | -id_case -Name_Case -Nombre_Case -Price -Opponent -Tags -Location -Gugement -Gugement_date -Description | code d'affaire le nom d'affaire numéro d'affaire prix d'affaire adversaire les mots clés Lieu du procès Jugement Jugement date |
| User(utilisateur) | -id_user -FullName_user -Role -Email_User -Phone_User | code utilisateur nom et prénom de utilisateur autorisation email d'utilisateur numéro de téléphone |
| Fees(finance) | -id_Fees -amount -isReceved -createdAt | Code de transaction Montante est reçu le paiement la date de création |
| TimeLine(état d'affaire) | -id_TL -Title_TL -Description_TL -Time_TL | code d'état d'affaire titre la description la date d'état |
| Tasks(taches) | -id_Tasks -Title_Tasks -Description_Tasks -IsAlarmActive -IsDone -DeadLine | Code de tache Titre de tache La description L'alarme est-elle active La tâche est fait La date de tache |
| Contact(client) | -id_Contact -FullName_Contact -Father_Name -Mother-FullName -Phone_Contact -Birthday_Contact -Adress_Contact -Email_Contact -Description_Contact | Code de client Nom et prénom de client Nom de père Nom et prénom de mère Numéro de téléphone de client Date de naissance de client Adresse de client Email de client La description |

Tableau III.1: Description des méthodes des classes d'objets et d'association.

IV.4. Les méthodes des classes

| Class | Methods | Description |
|--------------------------|---|---|
| Case (affaire) | AddCase() GetCase() UpdateCase() DeletCase() Search() | Permet de créer une affaire Permet de consulte une affaire Permet de modifie une affaire Permet de supprime une affaire Permet de recherche une affaire |
| User(utilisateur) | login() AddUser() UpdateUser() DeletUser() | Permet d'authentifie Permet d'ajouter un utilisateur Permet de modifier un utilisateur Permet de supprimer un utilisateur |
| Fees(finance) | GetFees() addFees() UpdateFees() DeletFees() PrintFees() | Permet de consulter une facture Permet d'ajouter une transaction Permet de modifier une transaction Permet de supprimer une transaction Permet d'imprimer une facture |
| TimeLine(état d'affaire) | AddTL() UpdateTL() DeletTL() | Permet d'ajouter un état d'affaire Permet de modifier un état d'affaire Permet de supprimer un état d'affaire |
| Tasks(taches) | AddTasks() UpdateTasks() DeletTasks() | Permet d'ajouter une tache Permet de modifier une tache Permet de supprimer une tache |
| Contact(client) | AddContact() UpdateContact() DeletContact() Search() GetContact() | Permet d'ajouter un client Permet de modifier un client Permet de supprimer un client Permet de rechercher un client Permet de consulter un client |

Tableau III.2 : Description des méthodes des classes.

V. Conclusion

Dans ce chapitre, nous avons présenté le langage UML. Aussi nous avons fait la description des diagrammes des cas d'utilisation, classe et de séquence et d'activité, et de contexte afin de délimiter le cadre de notre travail et de préparer un terrain favorable pour la prochaine étape.

Maintenant, notre application est prête à être codée. Dans le chapitre suivant, nous allons nous intéresser à l'implémentation de notre système en se basant sur la conception détaillée de ce chapitre.

Chapitre 4

Réalisation

I. Introduction

Dans chapitre précédent nous avons présenté les étapes de conception de l'application, ainsi que les différents diagrammes des cas d'utilisation, par la suite nous avons détaillé cette phase en établissant les diagrammes des classes ainsi que les diagrammes de séquences et d'activités.

Dans ce chapitre nous traitons les différentes étapes d'implémentation de l'application, et au fur et à mesure nous avons établi un ensemble de tests : ce sont les étapes de la phase de réalisation.

Nous allons commencer par la description de l'environnement matériel et logiciel tout en donnant par la suite un aperçu sur le travail accompli au cours de la période de développement.

II. Environnement du travail

Dans cette partie, nous avons étudié le choix des outils matériels et surtout les outils logiciels du développement web.

II.1. Environnement matériel

Nous mentionnons les caractéristiques de l'ordinateur sur lesquelles nous avons développé l'application parce qu'elles peuvent donner une idée sur les conditions du travail. L'application a été développée sur un ordinateur portable lenovo thinkpad qui se caractérise par :

- **Processus** : Intel® Core™ i7-4600U CPU @ 2.10 GHz 2.69 GHz.
- **Mémoire installé (RAM)** : 12.00 GB.
- **Type de système** : système d'exploitation 64 bits.
- **Disque dur** : 500 GB HDD 16 GB SSD.
- **Ecran** : 14 pouces.

II.2. Environnement logiciel

Nous avons énuméré au cours de cette partie les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre application.

II.2.1. Système d'exploitation

Nous avons utilisé comme système d'exploitation :

- Microsoft Windows 10 Edition professional.

II.2.2. Outil de modélisation UML

- **Creately :**

Creately est un logiciel de diagramme et d'organigramme facile à utiliser conçu pour la collaboration en équipe. Prend en charge plus de 40 types de diagrammes et contient des milliers de modèles dessinés par des professionnels.

- **Draw.io :**

Draw.io est un logiciel propriétaire pour créer des diagrammes et des graphiques. Le logiciel vous permet de choisir parmi une fonction de mise en page automatique ou de créer une mise en page personnalisée. Ils ont un grand choix de formes et des centaines d'éléments visuels pour rendre votre diagramme ou graphique unique en son genre. La fonction glisser-déposer facilite la création d'un superbe diagramme ou graphique.

II.2.3. Environnement de développement intégré

- **Visual Studio Code**

Est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS2.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

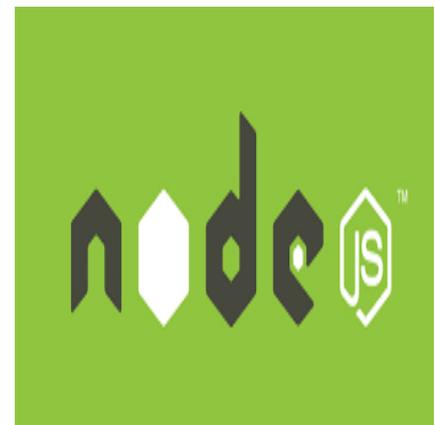


II.2.4. Serveur d'application : « Node.js »

- **Node.js**

Est une plateforme de développement open source pour exécuter du code JavaScript côté serveur. Node est utile pour développer des applications qui nécessitent une connexion persistante du navigateur au serveur et est souvent utilisé pour des applications en temps réel telles que le chat, les flux d'actualités et les notifications push Web.

Node.js est destiné à s'exécuter sur un serveur HTTP dédié et à utiliser un seul thread avec un processus à la fois. Les applications Node.js sont basées sur des événements et s'exécutent de manière



asynchrone. Le code construit sur la plate-forme Node ne suit pas le modèle traditionnel de réception, traitement, envoi, attente, réception. Au lieu de cela, Node traite les demandes entrantes dans une pile d'événements constante et envoie les petites demandes les unes après les autres sans attendre les réponses.

II.2.5 Outil d'administration de la base de données

- **PgAdmin**

Est un outil d'administration graphique pour PostgreSQL distribué selon les termes de la licence PostgreSQL. Il peut être utilisé sur les plateformes Linux, FreeBSD, Solaris, Mac OS X et Windows pour gérer les différentes moutures de PostgreSQL à partir de la version 7.3, mais également les versions commerciales et dérivées de PostgreSQL telles que Postgres Plus Advanced Server et Greenplum Database

- **PostgreSQL**

Est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD.

Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MariaDB et Firebird), ou propriétaires (comme Oracle, MySQL, Sybase, DB2, Informix et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises.

II.2.6. Langage de programmation

- **TypeScript**

Est un langage de programmation libre et open source développée par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScript). Le code TypeScript est transcompilé en JavaScript, et peut ainsi être interprété par n'importe quel navigateur web ou moteur JavaScript.

TypeScript a été cocréé par Anders Hejlsberg, principal inventeur de C#



- **HTML**

HTML (**H**yper **T**ext **M**arkup **L**anguage / langage hypertexte) est le langage dans lequel sont écrites les pages du web. Un site web est constitué d'un ou plusieurs documents HTML, appelées aussi pages.

Pour se déplacer d'une page à l'autre dans nos modules on passe par l'intermédiaire d'hyperliens. Pour ajouter des objets graphiques on utilise le HTML d'autre part pour tester des pages web html en local, il suffit d'ouvrir le fichier dans un navigateur.

Le HTML n'est pas un langage de programmation comme le C++. Les langages dynamiques comme PHP et Javascript vont d'ailleurs générer des pages HTML statiques.

- **CSS**

Les CSS, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus, dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle.



III. L'architecture de l'application

Nous présentons l'architecture de l'application, puis nous détaillerons les différentes infrastructures qui le composent.

III.1. Architecture 3-tiers

Nous avons développé l'application en nous basant sur l'architecture trois-tiers. En adoptant cette architecture, le système sera divisé en trois couches (ou niveaux) différentes présentations des données, logique métier, et accès aux données.

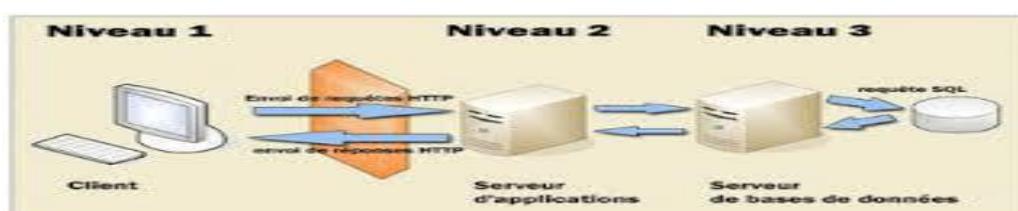


Figure IV.1 : Architecture 3 – tiers

IV. Développement

- **LoopBack**

Est un Framework Node.js, hautement extensible, open-source qui vous permet de créer des APIs REST de bout en bout avec très peu voire pas de code.

Construit pour les développeurs d'API

- Définissez les points d'accès de votre API via le standard OpenAPI.
- Développez vos points d'accès avec du JavaScript moderne en utilisant ES2017, async / await et les modules.
- Utilisez vos points d'accès existants et schémas comme source de vérité, et ce, sans générer de code.

Construit pour les équipes

Vérifiez les modifications de l'API sans avoir à lire du JavaScript.

Maintenez une consistance en automatisant la validation de vos points d'accès ainsi que de vos schémas.

Support de première classe de TypeScript (JavaScript fortement typé).

Construit pour vos plateformes

Utilisez LoopBack comme point de départ pour votre propre framework ou plate-forme.

Construisez des bibliothèques de composants réutilisables de façon standardisée.

Intégrez-vous à des bases de données, des web services et d'autres plate-formes via des connecteurs.

V. Choix de développement

Pour le développement, nous avons appliqué le modèle Model-View-Controller (MVC). Ce paradigme divise l'IHM (Interface Homme Machine) en un modèle (M pour modèle de données) une vue (V pour la présentation, l'interface utilisateur) et un contrôleur (C pour la logique de contrôle, et la gestion des événements / synchronisation), chacun ayant un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. L'architecture MVC ne résout pas tous les problèmes. Elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

Ce patron d'architecture impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur.

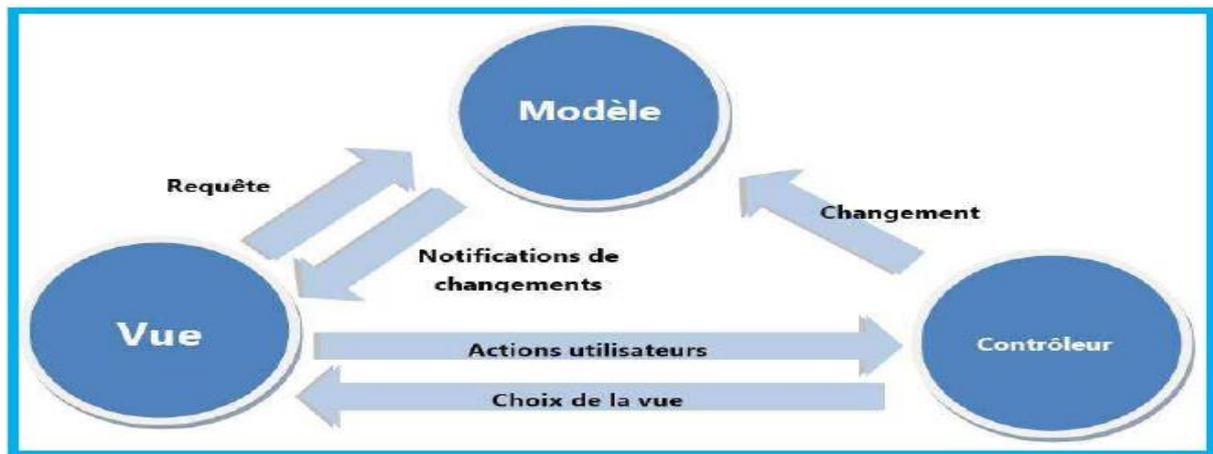


Figure IV.2 : Schéma du modèle MVC.

V.1. Le modèle

Le modèle représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient. Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.

V.2. La vue

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

V.3. Le contrôleur

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle, et ce dernier notifie la vue que les données ont changé pour qu'elle les mette à jour.

VI. Les technologies utilisées :

- **Couche présentation : angularJS**

Est un Framework web frontend open-source basé sur JavaScript principalement maintenu par Google et par une communauté d'individus et d'entreprises pour relever de nombreux défis rencontrés dans le développement d'applications à une seule page. Il vise à simplifier à la fois le développement et le test de ces applications en fournissant un cadre pour les architectures modèle-vue-contrôleur (MVC) et modèle-vue-modèle (MVVM) côté client, ainsi que des composants couramment utilisés dans les applications Internet riches.

AngularJS est utilisé comme interface de la pile MEAN, constituée de la base de données MongoDB, du cadre de serveur d'applications Web Express.js, d'Angular.js lui-même et de l'environnement d'exécution du serveur Node.js. La version 1.8.x est sur le support à long terme jusqu'au 31 décembre 2021. Après cette date, AngularJS ne sera plus mis à jour et Angular (2.0+) est suggéré à la place.

- **Couche métier : REST**

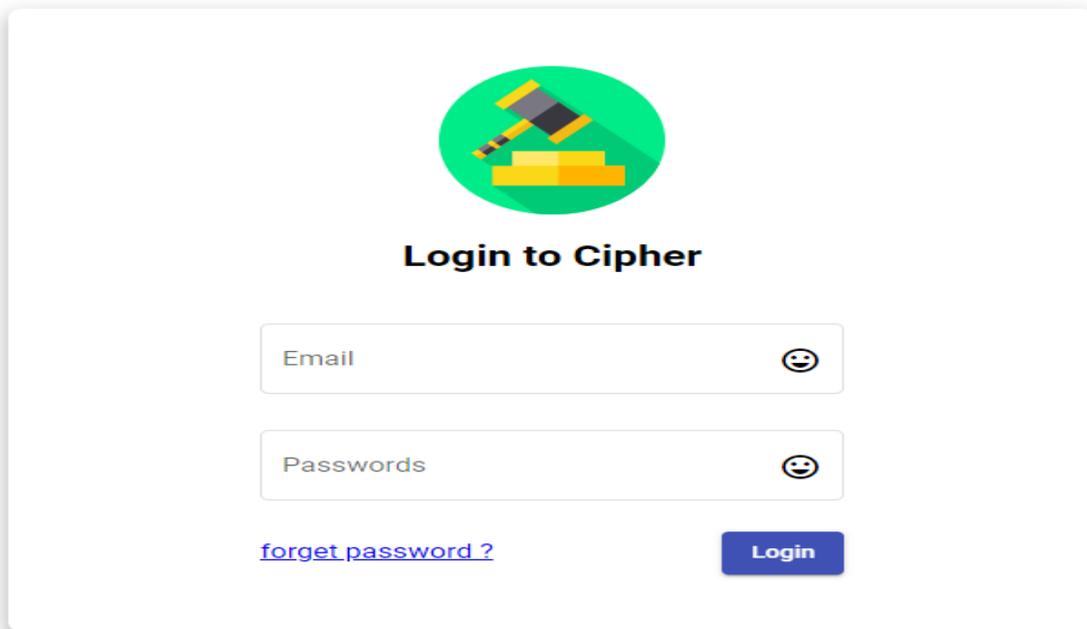
REST est un style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web. Les services web conformes au style d'architecture REST, aussi appelés services web RESTful, établissent une interopérabilité entre les ordinateurs sur Internet.

VII. Les interfaces graphiques

L'interface graphique est une partie très importante pour la réalisation d'une application Web convenable et conviviale offrant un certain plaisir à l'utilisateur lors de sa navigation. Ainsi ce critère peut faire la différence entre une application et un autre bien qu'elles aient les mêmes fonctionnalités.

Voici maintenant un ensemble de captures d'écrans sur les principaux points d'entrées de l'application.

VII.1. Interface Authentification



The screenshot shows a login form titled "Login to Cipher". At the top center is a circular logo with a green background and a yellow and black geometric design. Below the logo, the text "Login to Cipher" is displayed in bold. The form consists of two input fields: "Email" and "Passwords", each with a smiley face icon on the right. Below the "Email" field is a blue link labeled "forget password?". At the bottom right of the form is a blue button labeled "Login".

Figure IV.3 : *Interface Authentification*

Lors du clic sur le bouton « Entrer » de l'interface home, une interface d'authentification est affichée.

L'utilisateur doit introduire son Email et son mot de passe dans les champs correspondants pour pouvoir accéder aux différentes fonctionnalités de l'application.

Une fois que l'avocat a cliqué sur le bouton « Login », le système vérifie les données entrées.

En cas d'échec, il réaffiche la page d'authentification avec un message d'erreur. Si l'Email et le mot de passe sont valides, le système passe au tableau de bord.

VII.2. Interface de tableau de bord (Dashboard)

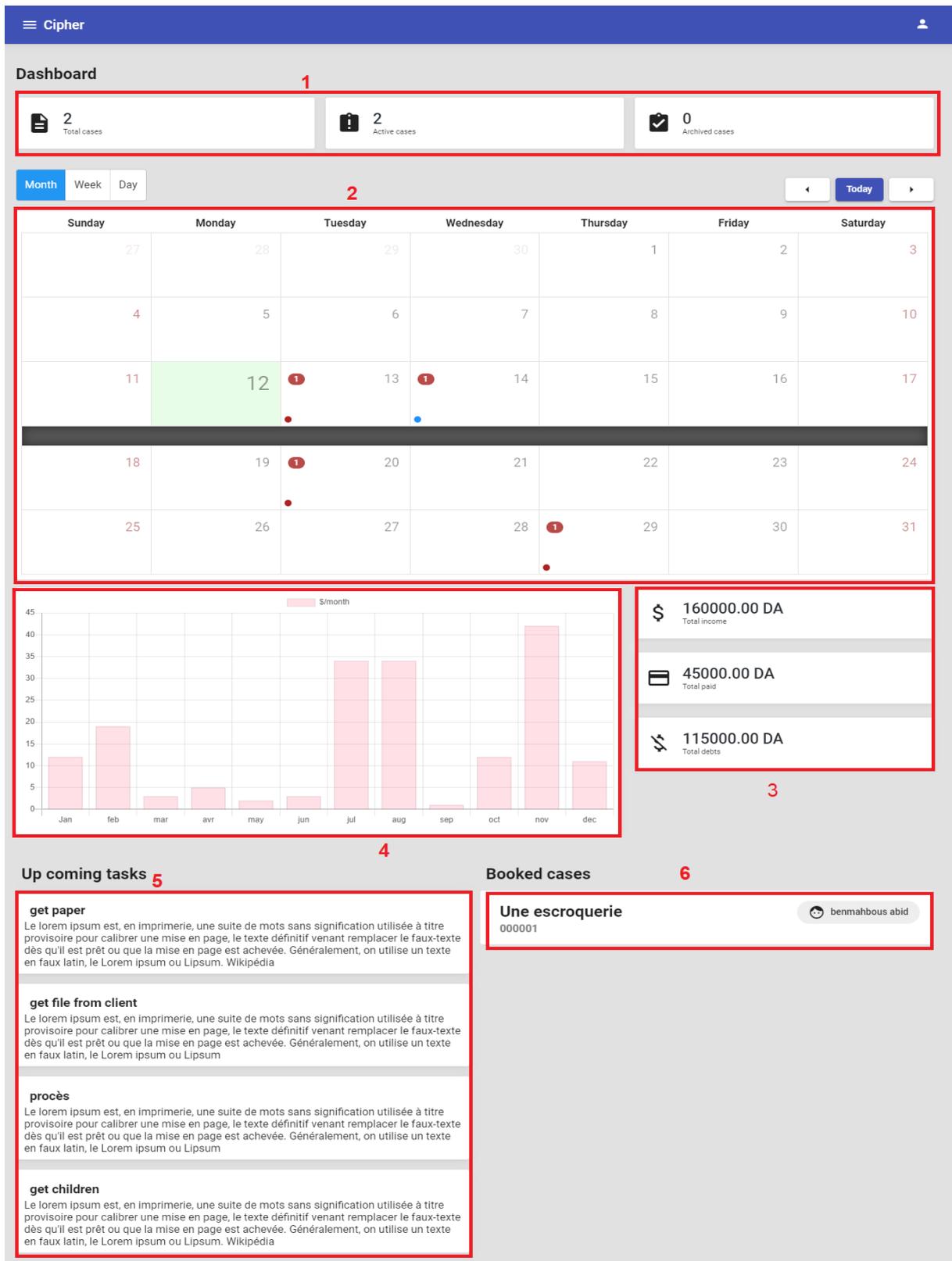


Figure IV.4 : Interface de tableau de bord.

L'interface principale qui contient les éléments importants et les statistiques liées au bureau

1. Représente les statistiques des affaires.
2. Emploi de temps de tâches.
3. Les statistiques financiers.
4. Représentation graphique des statistiques financier.
5. Liste des tâches à proximité
6. Les affaires importantes sur lesquels vous travaillez actuellement.

VII.3. Interface du menu d'application

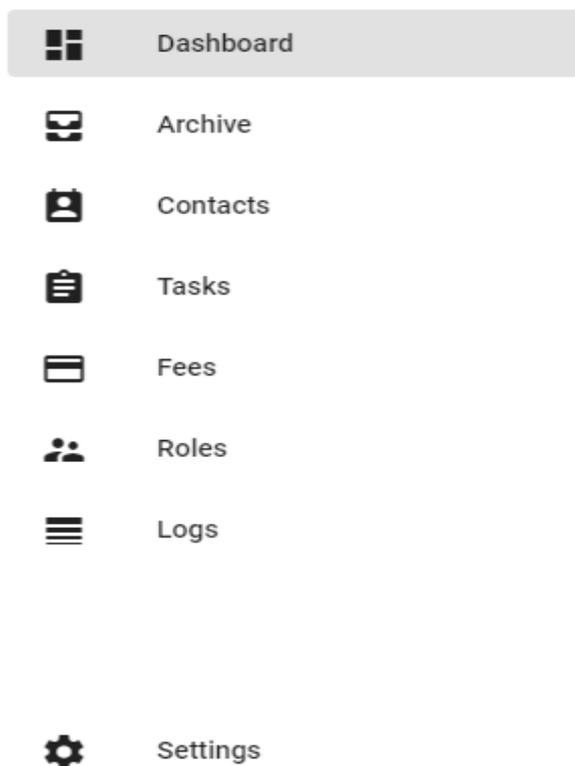


Figure IV.5 : Interface du menu d'application.

VII.4. Interface de gestion d'affaires

The interface displays a list of cases with the following details:

| Case ID | Client | Lawyer | Location | Amount | Status | Categories |
|---------|------------------|-----------------|----------|-----------|--------|-----------------------------------|
| 00002 | mrs austina | austin biber | bba | 100000 DA | Active | family, civile, children, divorce |
| 000001 | rouabeh el eulmi | benmahbous abid | bba | 60000 DA | Active | comercial, family, friend, money |

Figure IV.6 : Interface de gestion d'affaires.

Cette interface représente la liste des affaires, peut rechercher les affaires, peut découvrir l'information pour chaque affaire.

VII.5. Interface de détails d'affaire

CIPHER

Une escroquerie 000001

Le lorem ipsum est, en imprimerie, une suite de mots sans signification utilisée à titre provisoire pour calibrer une mise en page, le texte définitif venant remplacer le faux-texte dès qu'il est prêt ou que la mise en page est achevée. Généralement, on utilise un texte en faux latin, le Lorem ipsum ou Lipsum

1

benmahbous abid

Infos

Opponent: rouabeh el eulmi

Location: bba

Price: 60000

Judgement:

Judgement Date:

Total debt: 15000

Created At: Oct 12, 2020

Update At: Oct 12, 2020

comercial family friend money

Tasks Add task

procès Oct 20, 2020 ✓

get file from cl... Oct 14, 2020 ✓

2

Attachments Add file

Mandat d'arrêt papier 3

Payments 4

| | | |
|----------|--------------|---|
| 40000 DA | Oct 12, 2020 | ✎ |
| 5000 DA | Oct 12, 2020 | ✎ |

Timeline 5

Réponse du tribunal Oct 12, 2020 ✓

reponse d'avocate Oct 13, 2020 ✓

Le lorem ipsum est, en imprimerie, une suite de mots sans signification utilisée à titre provisoire pour calibrer une mise en page, le texte définitif venant remplacer le faux-texte dès qu'il est prêt ou que la mise en page est achevée. Généralement, on utilise un texte en faux latin, le Lorem ipsum ou Lipsum. Wikipédia

Figure IV.7 : Interface de détails d'affaire.

Cette interface représente tous les détails d'affaire

1. Les informations liées à l'affaire.
2. La liste des taches liées à l'affaire.
3. La liste des attachement liées à l'affaire.
4. La liste des transactions liées à l'affaire.
5. Représente l'état d'affaire.

VII.6. Interface d'ajoute une affaire

Add new case

1 Case

case name *
put a significant case name

Case number
put case number

Case price
0
put price here

Client *
put client name

Opponent *
put opponent name

court *
case court

Current state *
Active

Description

Tags
put some case related info like type , act etc..

Create

2 Client

3 Attachments
Optional

4 Done
Optional

Figure IV.8 : Interface d'ajoute une affaire.

VII.7. Interface de gestion de client (contact)

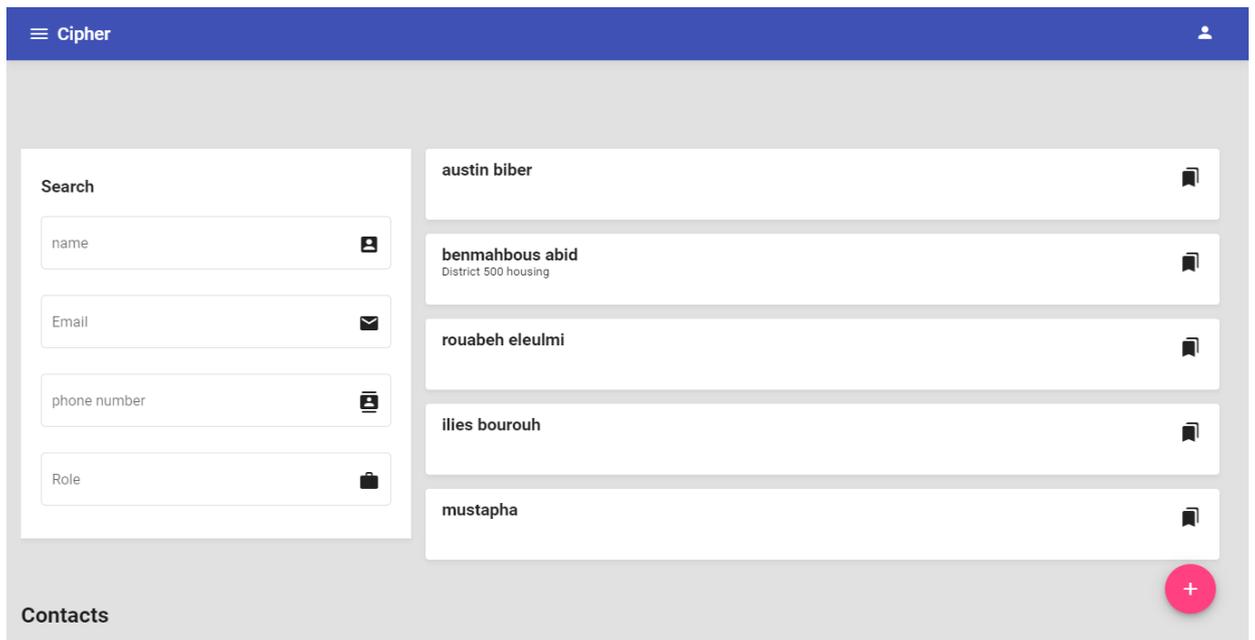


Figure IV.9 : Interface de gestion de client (contact).

VII.8. Interface de détails de client (contact)

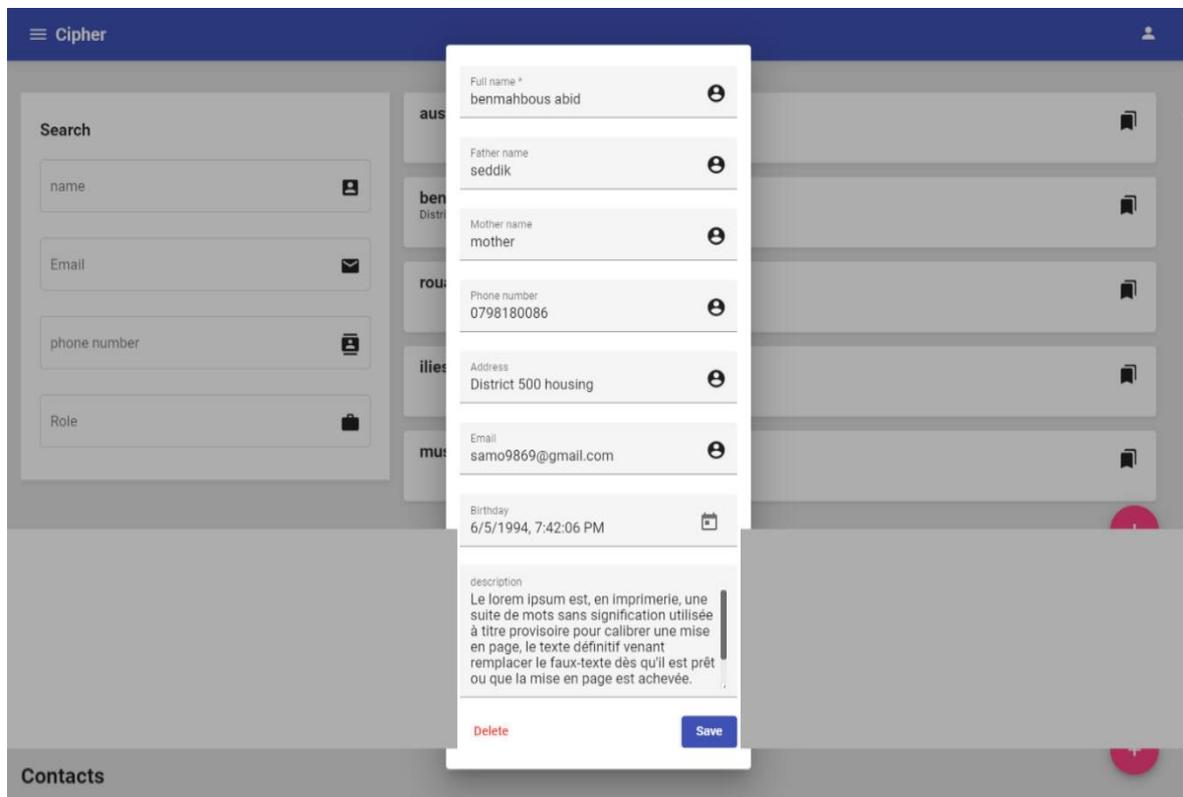


Figure IV.10 : Interface de détails de client (contact).

VII.9. Interface de gestion de tâches (tasks)

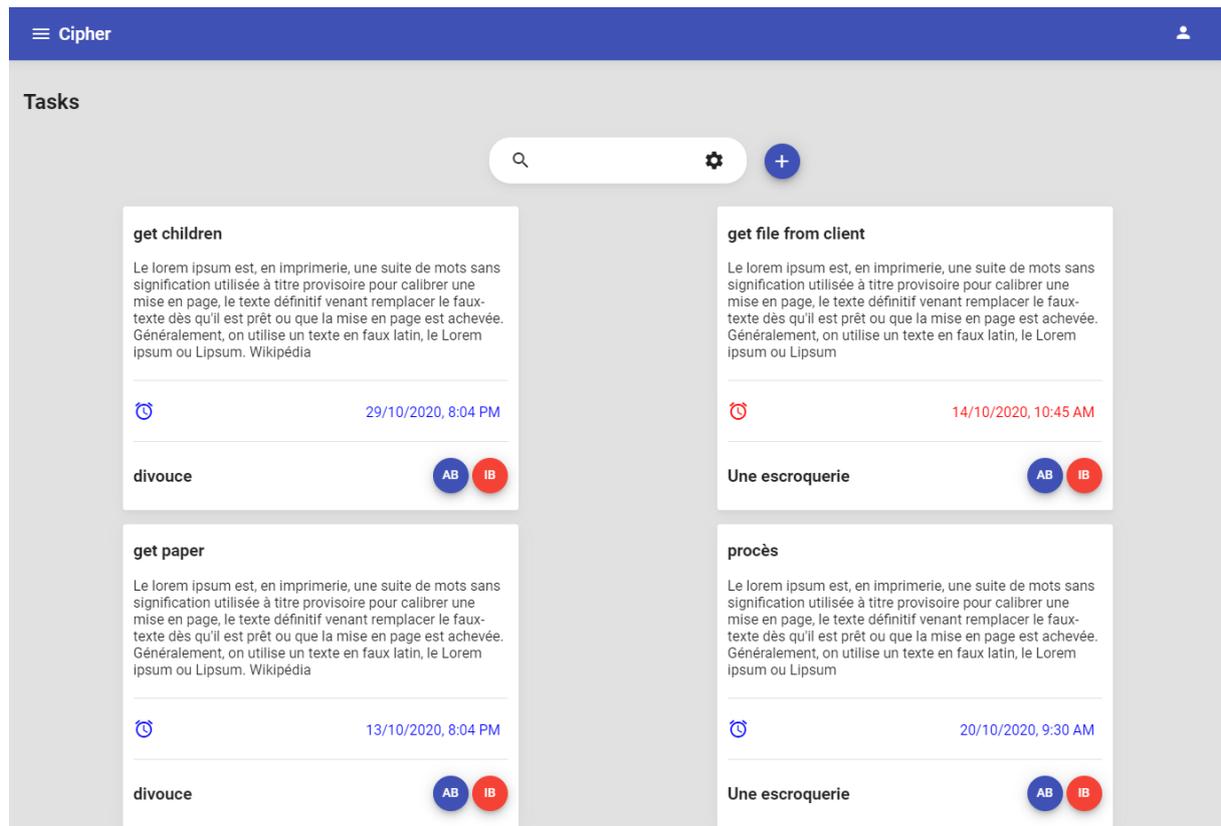


Figure IV.11 : Interface de gestion de tâches.

VIII. Conclusion

La dernière partie de ce projet était dédiée à la navigation dans notre application. Cette partie constitue le dernier volet de ce rapport, elle a pour objet de présenter l'environnement logiciel matériel de réalisation. Elle a été clôturée par l'exposition des imprimés écrans décrivant quelques interfaces de notre outil.

Conclusion Générale et perspectives

Tout au long de ce rapport, nous avons présenté les différentes étapes de réalisation de l'application. Pour le développement de ce projet, la technique UML a été utilisée, ce qui a permis de mener correctement la tâche d'analyse des besoins à l'aide du diagramme de cas d'utilisation et la tâche de conception, ainsi les scénarios sont aussi détaillés afin d'expliquer toutes les tâches faites. Puisque nous travaillons avec la plateforme Node.JS.

Ce projet nous a donné l'opportunité de s'initier à la vie professionnelle dans un milieu réel et avoir un début d'expérience significatif, et il nous a appris comment dès le matin on peut prendre le parti de la gaieté, comment réussir de bonnes relations pour assurer un travail de groupe, comment compter sur soi pour résoudre les problèmes au cas où ils se présentent, comment être méticuleuses dans notre travail, comment être attentives aux indications de nos supérieurs, comment être bien organisées pour accomplir dans les meilleurs délais, et meilleures conditions les tâches qui nous sont confiées.

Au cours de la réalisation de notre projet, nous avons été astreints par quelques limites notamment, la contrainte du temps qui était relativement un obstacle devant l'ajout de certaines autres fonctionnalités. Cependant, il était une occasion pour mettre en évidence et déployer sur le plan pratique nos connaissances en informatique.

Le projet peut être amélioré, et ce, en lui ajoutant quelques modules ou interfaces pour mieux l'adopter aux besoins de l'utilisateur et pour qu'il soit toujours fiable et au niveau des progrès atteints par la société.

Suite aux travaux effectués, plusieurs points restent à développer et à améliorer. Parmi lesquels citons :

- Facilité l'application pour les clients aussi.
- Donner la main à l'utilisateur pour modifier ou ajouter le style qu'il le veut.
- Par la suite, relier les données entre les avocats et les juges et l'administration tribunal.

Webographie

[1] <https://docplayer.fr/23451064-2-url-3-hypertexte-et-topologie-du-web.html> (19/09/2020)

[2] <http://glossaire.infowebmaster.fr/html/> (19/09/2020)

[3] <https://www.lemagit.fr/definition/HTTP> (19/09/2020)

[4] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/> (20/09/2020)

[5] <https://openclassrooms.com/fr/courses/3306901-creez-des-pages-web-interactives-avec-javascript/3722361-utilisez-des-api-web> (20/09/2020)

[6] <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/web-components/> (21/09/2020)

[7] <https://developer.mozilla.org/fr/docs/Web/Media> (24/09/2020)

[8] <https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2035851-uml-c-est-quoi> (24/09/2020)

[9] <https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours04.html> (25/09/2020)

Pierre Gérard, « UML Cours 1 : Diagrammes de cas d'utilisation », IUT de Villetaneuse, DUT informatique, S2 2013.

[10] https://www.ibm.com/support/knowledgecenter/fr/SSRTLW_9.6.1/com.ibm.xtools.sequence.doc/topics/cseqd_v.html (25/09/2020)

[11] http://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition_des_diagrammes_d%27activit%C3%A9s_UML_2.0 (26/09/2020)

[12] <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes> (26/09/2020)