

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abdelhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité: Sciences et Technologies de l'Information et de la Communication
(STIC)

Conception et réalisation d'une application mobile pour la gestion d'un Cabinet d'avocat

Préparé par :Nora Rezaiki

Soutenue devant le jury:

Président:	Meriem Boumessata	MAA	C.U.Abd Elhafid Boussouf
Examineur:	Faiza Afri	MAA	C.U.Abd Elhafid Boussouf
Encadreur :	Asma Bouchakouf	MAA	C.U.Abd Elhafid Boussouf

Année Universitaire : 2018/2019

Remerciement

Je remercie d'abord et avant tout Allah qui m'a donné le courage et la patience pour réaliser ce travail. Un remerciement particulier à mon encadreur madame «**Bouhekouf Asma**» pour sa présence, ses précieux conseils et son aide durant toute la période du travail

Mon vifs remerciements va également aux membres de jury pour l'intérêt qu'ils ont porté à ma recherche en acceptant

d'examiner mon travail et de l'enrichir par leurs propositions Je tiens à remercier l'ensemble de mes professeurs de Master 2 STIC Toute personne ayant de près ou de loin contribué à l'élaboration de ce travail un grand merci

Nora Rezaiki.

Dédicaces

Je dédie ce modeste travail :

A mes parents **Bachir et Salima** , qui ont toujours été là pour moi

A mon frère **Nabil**

A mon mari **Zakaria**

A ma sœur **Loubna** et a son mari **Fouad** et son petit **Alla**

A ma sœur **Assia** et a son mari **Ismail** et ses petites Fille **Silin et**

Wiam

A toute ma famille **Rezaiki**.

A toute ma famille **Stambouli**.

A mes très chères amies **Chaima , Asma , Boutheyne**

A mon encadreur **Bouhekouf Asma** et à mes enseignants

A mes cousine **Nihad et Houssame**

A tous ceux qui m' ont soutenus de mon travail notamment ma prof

Zekiouk Mounira

Table des matières

Remerciement	2
Dédicaces	3
Table des figures	7
Liste des tableaux	12
Résumé	16
Introduction générale	19
1 Application mobile	21
1.1 Introduction	21
Introduction	21
1.2 L'Informatique mobile	21
1.3 Les applications mobiles	23
1.4 Les systèmes d'exploitation mobiles	23
1.4.1 Définition :	23
1.4.2 Types de systèmes d'exploitation mobiles	23
1.5 Découverte de la plateforme android	27
1.5.1 Présentation d'Android	27
1.5.2 Les versions d'Android	27
1.6 Pourquoi le choix d'Android?	28
1.7 Conclusion	29

Conclusion	29
2 Présentation du domaine d'étude : système juridique	30
2.1 Introduction	30
Introduction	30
2.2 Terminologies	30
2.2.1 Tribunal	30
2.2.2 Avocat	31
2.2.3 Affaire juridique	31
2.2.4 Affaire civile	31
2.2.5 Affaire pénale	31
2.3 La justice en Algérie	31
2.3.1 Le secrétaire général	32
2.3.2 Le chef de cabinet	32
2.3.3 L'inspection générale :	32
2.3.4 Les sous structures suivantes	32
2.4 Le rôle et les missions de l'avocat	33
2.4.1 Le premier juge	33
2.4.2 Le conseil	33
2.4.3 Le mandataire	33
2.4.4 Le défenseur	34
2.4.5 Le conciliateur :	34
2.4.6 Le négociateur	34
2.4.7 Le plaideur	34
2.5 Avocat visité	35
2.6 Conclusion	35
3 l'étude préliminaire	36
3.1 Introduction	36
Introduction	36
3.2 Situation de l'étude préliminaire dans 2TUP	36
3.3 Élaboration du cahier des charges	37
3.3.1 La présentation du projet	37

3.3.2	Problématique	38
3.3.3	L'Objectifs	38
3.3.4	Les grands choix techniques	38
3.3.5	Spécification des besoins fonctionnels	39
3.3.6	Spécification des besoins non fonctionnels	40
3.3.7	Identification des acteurs	41
3.3.8	Identification des messages	41
3.3.9	Le diagramme de contexte dynamique	42
	Conclusion	42
3.4	Conclusion	43
4	Capture des besoins	44
4.1	Introduction	44
4.2	Situation de la capture des besoins dans 2TUP	44
4.3	Capture des besoins fonctionnels	45
4.3.1	Identification des cas d'utilisation	46
4.3.2	Description des cas d'utilisation	48
	4.3.2.1 Description textuelle des cas d'utilisation :	48
	4.3.2.2 Description graphique des cas d'utilisation :	66
4.3.3	Identification des classes candidates :	91
4.3.4	Diagramme de classe participantes :	92
4.4	Capture des besoins technique :	113
	4.4.1 La spécification technique du point de vue matériel :	114
	4.4.2 La capture des spécifications logicielles :	115
4.5	Conclusion	118
	Conclusion	118
5	Analyse et Conception	119
5.1	Introduction	119
	Introduction	119
5.2	Analyse :	119
	5.2.1 Découpage en catégories :	120
	5.2.1.1 La répartition des classes candidates en catégories : . . .	120

5.2.1.2	Élaboration des diagrammes de classes préliminaires par catégorie :	121
5.2.1.3	Dépendance entre catégories :	122
5.2.2	Développement du modèle statique :	122
5.2.2.1	Diagramme de classe de la catégorie « Client » :	123
5.2.2.2	Diagramme de classe de la catégorie « Affaire » :	124
5.2.3	Développement du modèle dynamique :	125
5.2.3.1	Diagrammes de séquence détaillés :	125
5.3	Conception :	137
5.3.1	Dictionnaire de données :	138
5.3.2	Diagramme de classe détaillé :	140
5.3.3	Le passage de modèle objet au modèle relationnel :	141
5.3.3.1	Les règles de passage :	141
5.3.3.2	Les relations de la base de données :	142
5.4	Conclusion :	143
	Conclusion	143
6	Réalisation	144
	Introduction	144
6.1	Environnement de développement	144
6.1.1	Le langage java :	144
6.1.2	Le langage XML	145
6.1.3	Le SDK(Software Development Kit)	145
6.1.4	L'IDE Android studio	145
6.2	Les bases de données	146
6.2.1	Wamp server :	146
6.2.2	Langage de Manipulation de Bases de Données MySQL	147
6.2.3	PHP :	147
6.2.4	Notepad++ :	147
6.3	Présentation de quelques interfaces de notre application :	148
	Conclusion	155

Conclusion générale

157

Bibliographie 158

Table des figures

1.1	un Smartphone.	22
1.2	application mobile.	23
1.3	Exemple système d'exploitation mobile.	24
1.4	Logo de Symbian.	24
1.5	Logo D'iOS.	25
1.6	Logo de Blackberry.	25
1.7	Logo de Windows phone.	26
1.8	Logo d'Android.	26
3.1	Situation de l'étude préliminaire dans 2TUP.	37
3.2	Diagramme du contexte.	42
4.1	la situation de la capture des besoins dans 2TUP.	45
4.2	Diagramme de cas d'utilisation générale	47
4.3	Diagramme de séquence du cas «S'inscrire»	66
4.4	Diagramme de séquence du cas consulter demande	67
4.5	Diagramme de séquence du cas confirmer demande	68
4.6	Diagramme de séquence du cas consulter procuration	69
4.7	Diagramme de séquence du cas gérer client	70
4.8	Diagramme de séquence du cas ajouter client	71
4.9	Diagramme de séquence du cas Modifier Client	72
4.10	Diagramme de séquence du cas supprimer client	73
4.11	Diagramme de séquence du cas gérer séance	74
4.12	Diagramme de séquence du cas ajouter séance tribunal	75

4.13	Diagramme de séquence du cas modifier séance tribunal	76
4.14	Diagramme de séquence du cas supprimer séance tribunal	77
4.15	Diagramme de séquence du cas gérer affaire	78
4.16	Diagramme de séquence du cas ajouter affaire	79
4.17	Diagramme de séquence du cas modifier affaire	80
4.18	Diagramme de séquence du cas supprimer affaire	81
4.19	Diagramme de séquence du cas gérer facture	82
4.20	Diagramme de séquence du cas ajouter facture	83
4.21	Diagramme de séquence du cas modifier facture	84
4.22	Diagramme de séquence du cas supprimer facture	85
4.23	Diagramme de séquence du cas Effectuer Statistiques	86
4.24	Diagramme de séquence du cas demander consultation	87
4.25	Diagramme de séquence du cas Établir Procuration	88
4.26	Diagramme de séquence du cas consulter séance tribunal	89
4.27	Diagramme de séquence du cas consulter affaire	90
4.28	Diagramme de séquence du cas consulter facture	91
4.29	Diagramme de classe participantes pour le cas d'utilisation «s'inscrire»	93
4.30	Diagramme de classe participantes pour le cas d'utilisation «s'authentifier»	94
4.31	Diagramme de classe participantes pour le cas d'utilisation « Consulter demande »	95
4.32	Diagramme de classe participantes pour le cas d'utilisation «Confirmer Demande »	96
4.33	Diagramme de classe participantes pour le cas d'utilisation «Ajouter client »	97
4.34	Diagramme de classe participantes pour le cas d'utilisation «Modifier Client »	98
4.35	Diagramme de classe participantes pour le cas d'utilisation «Supprimer Client»	99
4.36	Diagramme de classe participantes pour le cas d'utilisation «Ajouter Séance Tribunal»	100
4.37	Diagramme de classe participantes pour le cas d'utilisation «Modifier Séance Tribunal»	101

4.38	Diagramme de classe participantes pour le cas d'utilisation «Supprimer Séance Tribunal»	102
4.39	Diagramme de classe participantes pour le cas d'utilisation «Ajouter Affaire»	103
4.40	Diagramme de classe participantes pour le cas d'utilisation «Modifier Affaire»	104
4.41	Diagramme de classe participantes pour le cas d'utilisation «Supprimer Affaire»	105
4.42	Diagramme de classe participantes pour le cas d'utilisation «Ajouter facture»	106
4.43	Diagramme de classe participantes pour le cas d'utilisation «Modifier facture»	107
4.44	Diagramme de classe participantes pour le cas d'utilisation «Supprimer facture»	108
4.45	Diagramme de classe participantes pour le cas d'utilisation «Demander consultation»	109
4.46	Diagramme de classe participantes pour le cas d'utilisation « Etablir procuration»	110
4.47	Diagramme de classe participantes pour le cas d'utilisation « Consulter séance tribunal»	111
4.48	Diagramme de classe participantes pour le cas d'utilisation «Consulter Affaire»	112
4.49	Diagramme de classe participantes pour le cas d'utilisation «Consulter Facture»	113
4.50	Architecture à 2 niveaux	114
4.51	Modèle de spécification logicielle du système.	115
4.52	Diagramme de séquence du cas s'authentifier	117
4.53	Diagramme de séquence de cas d'utilisation « Gérer l'intégrité des données ».	118
5.1	Situation de la phase d'analyse dans 2TUP.	120
5.2	Découpage en catégories	120
5.3	Diagramme des classes préliminaire de la Catégorie « Client »	121

5.4	Diagramme des classes préliminaire de la Catégorie « Affaire »	121
5.5	Dépendance entre catégories	122
5.6	Diagramme de classe de la catégorie « Client »	123
5.7	Diagramme de classe de la catégorie « Affaire »	124
5.8	Diagrammes de séquence détaillés « S’authentifier »	126
5.9	Diagrammes de séquence détaillés « Consulter Demande »	127
5.10	Diagrammes de séquence détaillés « Confirmer Demande »	128
5.11	Diagrammes de séquence détaillés « Ajouter Client »	129
5.12	Diagrammes de séquence détaillés « Modifier Client »	130
5.13	Diagrammes de séquence détaillés « Supprimer Client »	131
5.14	Diagrammes de séquence détaillés « Ajouter affaire »	132
5.15	Diagrammes de séquence détaillés « Modifier affaire »	133
5.16	Diagrammes de séquence détaillés « Supprimer affaire »	134
5.17	Diagrammes de séquence détaillés « Demander Consultation »	135
5.18	Diagrammes de séquence détaillés « Consulter séance tribunal »	136
5.19	Diagramme de classe détaillé	140
6.1	Logo du langage Java	145
6.2	Logo de l’IDE Android Studio	146
6.3	Logo du WampServer	146
6.4	Logo de Langage de Manipulation de Bases de Données MySQL	147
6.5	Logo de Notepad++	148
6.6	Fenêtre d’authentification	149
6.7	Fenêtre d’inscription	150
6.8	Fenêtre d’inscription	150
6.9	Accueil Avocat	151
6.10	Accueil Client	152
6.11	Fenêtre liste des client	153
6.12	Fenêtre Ajouter client	153
6.13	Fenêtre liste des demande client	153
6.14	Fenêtre discription d’affaire	153
6.15	Fenêtre Ajouter séance tribunaale	154
6.16	Fenêtre Modifier séance tribunaale	154

6.17 Fenêtre demander consultation	154
6.18 Fenêtre établir procuration	154

Liste des tableaux

1.1	Les versions d'Android.	28
4.1	Description textuelle du cas d'utilisation « S'inscrire »	48
4.2	Description textuelle du cas d'utilisation « Consulter Demande »	49
4.3	Description textuelle du cas d'utilisation « Confirmer Demande »	49
4.4	Description textuelle du cas d'utilisation « Consulter procuration »	50
4.5	Description textuelle du cas d'utilisation « Gérer Client »	50
4.6	Description textuelle du cas d'utilisation « Ajouter Client ».	51
4.7	Description textuelle du cas d'utilisation « Modifier Client »	52
4.8	Description textuelle du cas d'utilisation « supprimer Client ».	53
4.9	Description textuelle du cas d'utilisation « Gérer séance tribunal »	54
4.10	Description textuelle du cas d'utilisation « Ajouter séance tribunal ».	54
4.11	Description textuelle du cas d'utilisation « Modifier séance tribunal ».	55

4.12	Description textuelle du cas d'utilisation	
	« Supprimer séance tribunal »	56
4.13	Description textuelle du cas d'utilisation	
	« Gérer Affaire »	57
4.14	Description textuelle du cas d'utilisation	
	« Ajouter Affaire »	57
4.15	Description textuelle du cas d'utilisation	
	« Modifier Affaire »	58
4.16	Description textuelle du cas d'utilisation	
	« Supprimer Affaire »	59
4.17	Description textuelle du cas d'utilisation	
	« Gérer facture »	60
4.18	Description textuelle du cas d'utilisation	
	« Ajouter facture »	60
4.19	Description textuelle du cas d'utilisation	
	« Modifier facture »	61
4.20	Description textuelle du cas d'utilisation	
	« Supprimer facture »	62
4.21	Description textuelle du cas d'utilisation	
	« Effectuer statistiques »	63
4.22	Description textuelle du cas d'utilisation	
	« Demander Consultation »	63
4.23	Description textuelle du cas d'utilisation	
	« Etablir procuration »	64
4.24	Description textuelle du cas d'utilisation	
	« Consulter séance tribunal »	64
4.25	Description textuelle du cas d'utilisation	
	« Consulter affaire »	65
4.26	Description textuelle du cas d'utilisation	
	« Consulter facture »	65
4.27	liste des classes candidates	92
4.28	Description textuelle du cas d'utilisation « S'authentifier »	116

4.29	Description textuelle du cas d'utilisation « Gérer l'intégrité des données. »	117
5.1	Dictionnaire de données	139
5.2	« Équivalence entre les concepts objets et relationnels.	141
5.3	Les relations de la base de données de système	142

Résumé

Le thème abordé dans ce projet de fin d'études consiste à la conception et l'implémentation d'une application mobile pour la gestion de cabinet d'avocat. L'objectif de notre projet est pour faciliter la communication entre le client et l'avocat cette application permet au client d'être avisé de la Nouvelle date de jugement. En outre, le client peut directement prendre connaissance des détails de son dossier, le numéro de l'affaire, La date de son enregistrement.

Pour atteindre ces objectifs nous avons choisi de modéliser avec le formalisme UML2.0 avec la démarche 2TUP, notre choix s'est porté par rapport à sa stabilité et son indépendance vis à vis des langages d'implémentation. Afin de réaliser notre application, nous avons utilisé l'outil Android studio et «MySQL» comme SGBD. Quant à la réalisation des interfaces, nous avons utilisé le langage XML ainsi que JAVA.

Mots clés : Application mobile, Cabinet d'avocat, UML2.0, 2TUP, MySQL, JAVA, ANDROID, XML.

Abstract

The theme addressed in this final project is the design and implementation of a mobile application for the management of law firms. The purpose of our project is to facilitate communication between the client and the lawyer this application allows the client to be notified of the new date of judgment. In addition, the client can directly read the details of his file, the number of the case, The date of its registration.

To achieve these objectives we have chosen to model with the UML2.0 formalism with the 2TUP approach, our choice is related to its stability and its independence with respect to the implementation languages. In order to realize our application, we used the Android studio tool and "MySql" as DBMS. As for the realization of the interfaces, we used the XML language as well as JAVA.

Keywords : Mobile Application, Law Firm, UML2.0, 2TUP, MySQL, JAVA, ANDROID, XML.

Introduction générale

Introduction générale

Aujourd'hui, et avec les nouvelles technologies connues, l'homme cherche de plus en plus de faciliter le mode de sa vie dans n'importe quel domaine, en utilisant les différentes sortes de technologies créées, ce qui donne une automatisation de sa vie n'importe où et n'importe quand. Alors que parmi les innovations les plus dominantes de nos jours et les plus attirantes, on trouve les applications android, ce qu'on a connu aussi par les applications mobiles.

Une application android, c'est une application mobile spécifiée pour les Smartphones et utilise le système android comme système d'exploitation, dont la mobilité est actuellement une composante très importante pour mesurer le développement des gens et leurs cultures.

Donc, parmi les domaines qu'on s'intéresse est le domaine des avocats.

En droit, l'avocat est un juriste dont la fonction traditionnelle est de 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 s'acquiesce 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. Il dispose du monopole de la plaidoirie et de la postulation, même si ce monopole connaît des exceptions. En effet, devant le juge de proximité, la représentation 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 transactions sous seing privé. Il peut aussi représenter les tiers et les assister devant toutes les administrations publiques.

Le travail des avocats dans nos jours est devenu une procédure délicate qui nécessite un contact permanent entre l'avocat et ses clients. Tous types d'affaires judiciaires demandent plusieurs séances de discussion entre les clients et les avocats.

Au cœur de ses critiques, la nécessité de développer une application mobile sera une obligation, pour faciliter le travail de l'avocat d'une manière souple et en temps

réel Alors que notre objectif est de :

- Faciliter la communication entre l’avocat et le client.
- réduction du temps de communication entre ces acteurs.
- L’organisation des données.
- Assurer une meilleure gestion et une cohérence de l’information.

Pour réaliser cette application, on a choisi le langage UML2.0 avec la démarche 2TUP pour la modélisation, ainsi on a utilisé JAVA sous plateforme Android Studio pour l’implémentation, avec MySQL pour les bases de données.

Nous avons organisé notre mémoire comme la suite :

- **Chapitre 01** :définit le monde mobile.
- **Chapitre 02** :Un aperçu sur le sujet de notre étude ainsi présentation du cahier des charges qui consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, ensuite, il décrit les acteurs et les cas d’utilisation du futur système.
- **Chapitre 03** :Sera consacré pour le la capture des besoins qui comportent deux étapes : la capture des besoins fonctionnels et celles des besoins techniques.
- **Chapitre 04** :Ce chapitre est organisé en deux grandes parties. La première est consacrée à la phase d’analyse de notre système en le découpant en catégories jusqu’à l’arrivée au modèle dynamique. La deuxième représentera la phase de conception détaillée, qui présente la conception détaillée des classes et leurs associations qui construiront notre système et décrit le passage du modèle à objet au modèle relationnel.
- **Chapitre 05** :Sera dédiée à la phase réalisation. Dans lequel nous définirons les outils de développement que nous avons utilisés et nous illustrerons également quelques interfaces de l’application mis en œuvre.

A la fin le mémoire sera terminé par une conclusion générale qui décrit les résultats obtenus avec les perspectives.

Application mobile

1.1 Introduction

Depuis quelques années, le marché de la téléphonie mobile est en plein essor et une part de plus en plus importante de la population possède un téléphone ou une tablette ayant la capacité de calcul d'un ordinateur. De cela, l'accès à internet a connu un changement majeur, de sorte que les mobiles sont devenus les principaux points d'accès pour l'utilisation d'internet, et grâce à l'évolution de la technologie, les mobiles ne sont pas utilisés juste pour appeler ou jouer à des jeux, mais avec les Smartphones par exemple nous pouvons planifier notre journée complète, consulter nos mails, faire des conférences téléphoniques, se connecter en utilisant un réseau social et effectuer une foule d'autres activités ; tous cela en utilisant ce qu'on appelle "les applications mobile" . Les applications mobiles ont un impact primordial sur notre quotidien, elles nous permettent par exemple de réduire la charge de travail, substituer le format papier en format électronique, diminuer les coûts téléphoniques et nous offre d'innombrables autres services.

Dans ce chapitre, nous commençons par définir l'informatique mobile avec ces différents aspects, suivi des terminaux mobiles (une présentation avec leurs types) et puis nous allons nous concentrer sur les applications mobiles, donner leurs différents types et rajouter quelques exemples sur les Systèmes d'Exploitation mobiles, et nous concluons avec une brève description sur le système android.

1.2 L'Informatique mobile

• Définition de l'informatique mobile

La mobilité est devenue au fil du temps une véritable philosophie de vie. Ainsi, elle incite à la redéfinition des habitudes de chacun et façonne graduellement notre relation

à l'information, aux transports et au temps., ... Le concept d'informatique mobile réfère à la capacité d'accéder à des services ou à des applications indépendamment de la localisation physique des utilisateurs, de leurs comportements ou de leurs mouvements. [1]

- **Les Smartphones**

- **Definition**

Le Smartphone ou « téléphone intelligent » désigne un téléphone mobile doté de fonctionnalités évoluées qui s'apparentent à celles d'un ordinateur : navigation sur Internet, lecture de vidéos, de musique, jeux vidéo, courrier électronique, vidéoconférence, bureautique légère.

Muni d'un processeur puissant, souvent multi cœur, il embarque une série de capteurs (boussole, accéléromètre, gyroscope, GPS) qui lui permettent de faire fonctionner des applications dédiées à l'activité physique, de navigation assistée ainsi que des jeux que l'on peut contrôler d'un simple mouvement. Les Smartphones sont généralement dotés d'un appareil photo-vidéo et d'une caméra frontale dont les performances ne cessent de progresser. [2]



FIGURE 1.1 – un Smartphone.

1.3 Les applications mobiles

• Présentation des applications mobiles

Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation de l'appareil mobile.

Les applications mobiles sont adaptées aux différents environnements techniques des terminaux mobiles et à leurs contraintes et possibilités ergonomiques (écran tactile notamment). Elles permettent généralement un accès plus confortable et plus efficace à des sites ou services accessibles par ailleurs en versions mobile ou web. [3]



FIGURE 1.2 – application mobile.

1.4 Les systèmes d'exploitation mobiles

1.4.1 Définition :

Un système d'exploitation mobile ou OS mobile ("Operating System") est un logiciel conçu pour fonctionner sur un appareil mobile, il s'agit d'un ensemble de programmes permettant, entre autre, la gestion des autres applications et interfaces, la connectivité sans fil ainsi que la communication avec le support physique. [4]

La figure ci-dessous présente quelques exemples d'os mobiles : .

1.4.2 Types de systèmes d'exploitation mobiles

Aujourd'hui il existe plusieurs systèmes d'exploitation qui rivalisent entre eux. Nous allons présenter quelques exemple. [4]

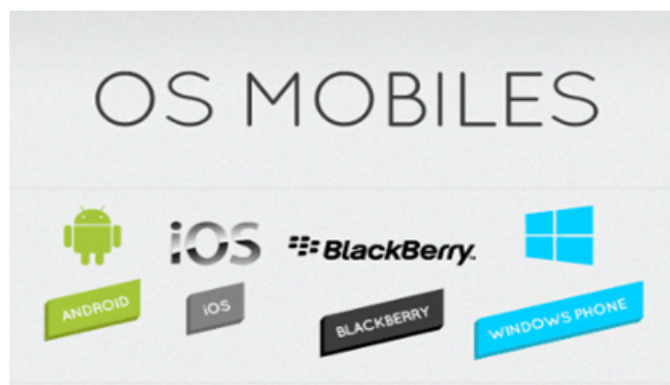


FIGURE 1.3 – Exemple système d’exploitation mobile.

- **Symbian**

Symbian est le système d’exploitation utilisé sur la plupart des Smartphones Nokia. Cet OS permet de télécharger des applications d’Ovi Store, l’application store développé par Nokia. .



FIGURE 1.4 – Logo de Symbian.

- **Apple iOS**

Apple iOS est le système d’exploitation développé par Apple pour l’iPhone. Cependant, il a depuis lors aussi été utilisé pour l’iPod Touch, l’iPad, ainsi que pour l’Apple TV. Apple iOS ne permettant pas que ce système supporte des applications extérieures, un application store a été mis en ligne et offre aujourd’hui plus de 700 000 applications à télécharger sur votre appareil.



FIGURE 1.5 – Logo D'iOS.

- **Blackberry OS**

C'est l'OS développé par Research in Motion (RIM) pour leur ligne de Smartphones BlackBerry. La plateforme BlackBerry était autrefois bien connue pour son support des serveurs e-mails professionnels, ainsi que pour l'envoi et la réception d'e-mails grâce à la technologie push (en temps réel) là où le réseau Internet mobile était disponible. BlackBerry est également très populaire pour ses performances d'assistant personnel digital puisqu'il offre un carnet d'adresses assez poussé, un agenda très complet, la possibilité de faire des listes de tâches, etc.



FIGURE 1.6 – Logo de Blackberry.

- **Windows Phone**

Windows Mobile, développé par Microsoft. Il est actuellement utilisé dans de nombreux Smartphones de marques bien connues, telles que HTC, Samsung, LG et Dell. Tout comme Apple a créé son propre application store, Windows Phone Marketplace est le magasin d'applications en ligne développé par Microsoft dans le but de permettre aux utilisateurs de Windows Mobile Phones de télécharger des applications et des jeux.



FIGURE 1.7 – Logo de Windows phone.

- **Android**

Android est un SE basé sur Linux conçu principalement pour équipements à écran tactile comme les Smartphones et les tablettes. Android est devenu la plateforme plus populaire pour mobile avec plus de 600000 applications sur le marché. Il est prévu que le store d'Android atteigne le chiffre d'un million d'applications bien avant Apple store. Comme c'est la plateforme sur laquelle on va travailler dans ce projet, nous allons revoir cette plateforme plus tard en détails.



FIGURE 1.8 – Logo d'Android.

1.5 Découverte de la plateforme android

1.5.1 Présentation d'Android

Android est une plate-forme logicielle open source destinée aux appareils mobiles qui intègre différents composants : un système d'exploitation, un middleware, une interface graphique et un éventail d'applications. Android n'est donc pas lié à un appareil donné, un constructeur de téléphone mobile, ou un opérateur. Google a fédéré autour d'Android une trentaine de sociétés (dont Samsung, Motorola, Sony Ericsson ou LG) à l'intérieur de l'Open Handset Alliance pour diffuser en masse son système. Android laisse la possibilité aux développeurs de créer leurs propres applications à l'aide du langage de programmation Java, et Google fournit aux développeurs de nombreux outils pour écrire et mettre au point leurs programmes : émulateur pour les tests, SDK, et plugins pour le développement, ainsi qu'une documentation fournie. Android repose sur un noyau Linux qui gère les services système comme la sécurité, la gestion mémoire, la gestion des processus, la pile réseau.[5]

1.5.2 Les versions d'Android

Annoncé officiellement en novembre 2007, le système Android a fait du chemin depuis ses débuts. Depuis la version 1.5, les versions possèdent des noms de desserts (en anglais).[5]

Toutes les versions d'android sont résumées dans le tableau suivant :

Version	Nom de code	Date de sortie	Api
1.0	Aucun	11 novembre 2007	1
1.1	Petit four	22 octobre 2008	2
1.5	Cupcake	30 avril 2009	3
1.6	Donut	15 septembre 2009	4
2.0	Eclair	26 octobre 2009	5
2.2	Froyo	20 mai 2010	8
2.3	Gingerdroid	6 décembre 2010	10
3.0	Honeycomb	22 février 2011	11
4.0	Icecream sandwich	19 octobre 2011	15
4.1	Jelly Bean	9 juillet 2012	16
4.2		13 novembre 2012	17
4.3		24 juillet 2013	18
4.4	Kitkat	31 octobre 2013	19
5.0	Lollipop	3 novembre 2014	21
5.1		9 mars 2015	22
6.0	Marshmallow	5 octobre 2015	23
7.0	Nougat	22 août 2016	24

TABLE 1.1: Les versions d'Android.

1.6 Pourquoi le choix d'Android ?

Dans les sections précédentes, nous avons étudié les connaissances générales sur plusieurs plateformes. Dans la partie pratique, nous avons choisi la plateforme Android pour développer notre application. Nous citons quelques raisons :

- **Le système**

Android est une plateforme en code source ouverte et basée sur linux. elle est puissante, moderne et sûre. Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer, d'agrandir et de remplacer les composants existants. Les utilisateurs peuvent adapter les applications à leur besoins.

- **La gestion des données**

Avec un téléphone Android on copie les fichiers où l'on veut sur la carte SD (ou la mémoire interne) et tout est détecté par le système. Chez Apple la logique est différente. Le seul moyen de mettre des données sur le mobile est de passer par iTunes.

- **Optimisation pour les services Google**

Android a été développé depuis le début avec l'intention d'intégration avec Google (ce qui n'est juste mais bon). Les services Google fonctionnent parfaitement sur Android et grâce à ça Android bénéficiera bien du succès de Google. Les développeurs aussi peuvent intégrer plus facilement des services Google dans leurs applications.

- **Le coût**

Développer une application pour Android ne vous coûtera absolument rien. Tous les outils nécessaires sont fournis par Google et gratuitement ainsi que la documentation.

- **La position du système et le taux de progression**

Android domine le marché mondial des Smartphones et leurs utilisateurs augmentent rapidement.[6]

1.7 Conclusion

Suite à l'étude faite dans ce chapitre concernant les applications mobiles et leur importance dans le monde d'aujourd'hui, cela nous a aidés à comprendre les notions de base.

Aussi on a pu distinguer que la concurrence entre les différents Systèmes d'exploitation mobile est vraiment rude. En revanche, Android a pu récemment se démarquer de ses concurrents grâce à sa plateforme innovante (qui intègre toutes les nouvelles technologies), son accessibilité et son ouverture (licence open source), qui se sont révélées correspondre aux attentes des consommateurs.

Présentation du domaine d'étude : système juridique

2.1 Introduction

Dans nos jours le domaine de justice est devenu un centre d'intérêt pour les individus et les entreprises ; ce domaine assure une protection des droits physique ou morale contre tout type de violation. Plusieurs acteurs et entreprise coopèrent afin d'assurer la réalisation des objectifs et du rôle attendus de ce secteur.

Dans le présent chapitre nous allons présenter quelques concepts de base liées à notre domaine d'étude.

2.2 Terminologies

La justice est un principe philosophique, juridique et moral fondamental en vertu duquel les actions humaines doivent être sanctionnées ou récompensées en fonction de leur mérite au regard du droit, de la morale, de la vertu ou autres sources normatives de comportements.[19]

2.2.1 Tribunal

Le mot "tribunal", est un vocable générique employé pour désigner toute formation juridictionnelle ayant pour fonction d'apporter une solution à un litige soit entre personnes privées, (individus, associations, syndicats, entreprises, groupements jouissant de la capacité juridique etc).

2.2.2 Avocat

Un avocat est une personne dont la profession est de défendre ses clients en faisant prévaloir ses intérêts. Il apporte ses compétences au vue du problème soulevé par son client. De manière générale, il est totalement indépendant dans le choix de ses clients et n'est soumis à aucun régime en particulier. En ce sens, il exerce une fonction libérale.

L'avocat peut également défendre des personnes morales, des entreprises et autres. Il n'est pas un simple serviteur des particuliers. Ainsi, on a vu se développer la profession d'avocat d'affaire se spécialisant dans des litiges entre grandes entreprises.

2.2.3 Affaire juridique

Une affaire judiciaire est un litige opposant des parties opposées, résolu par un tribunal ou par une procédure judiciaire équivalente. Une affaire judiciaire peut être de droit civil ou pénal. Dans chaque cas juridique, il y a un accusateur et un ou plusieurs accusés.

2.2.4 Affaire civile

Une affaire civile est une affaire privée dans laquelle une personne en poursuit une autre. On parle alors d'une poursuite ou d'une action. Une action ou poursuite civile commence lorsque des personnes ou des entreprises se trouvent en désaccord sur une question juridique, comme les modalités d'un contrat ou la propriété d'un bien.

2.2.5 Affaire pénale

Une affaire pénale désigne la prise en charge par l'institution judiciaire des conduites que la société réprime par la condamnation à une peine. Contrairement à la matière civile, qui porte sur les litiges entre personnes privées, la matière pénale fait ainsi référence à la défense des valeurs, normes et comportements essentiels au bon fonctionnement de notre société.

2.3 La justice en Algérie

Le ministère de la Justice est le département ministériel du gouvernement algérien chargé de veiller au bon fonctionnement du système judiciaire. L'administration centrale du Ministère de la Justice, comprend :[20]

2.3.1 Le secrétaire général

auquel sont rattachés le bureau du courrier et de la communication et le bureau ministériel de la sûreté interne d'établissement.

Il est assisté de huit (8) directeurs d'études dont six (6) affectés à des missions dans le cadre des activités du comité d'animation et du suivi de la réforme de la justice.

2.3.2 Le chef de cabinet

assisté de huit (8) chargés d'études et de synthèse, chargés notamment :

- de la préparation et de l'organisation de la participation du ministre aux activités gouvernementales
- de l'information, de la communication et des relations avec les organes d'information .
- du suivi et de l'animation des relations avec les institutions publiques et la société civile.
- du suivi des relations avec les associations et les organisations socioprofessionnelles .
- du suivi de l'élaboration de synthèses concernant le développement du secteur.
- de l'établissement des bilans d'activités du secteur.
- de la préparation, de l'exploitation et du suivi des dossiers de coopération internationale
- du suivi de l'activité du ministre avec les organisations et organismes internationaux.

2.3.3 L'inspection générale :

régie par un texte particulier.

2.3.4 Les sous structures suivantes

- La Direction Générale des Affaires Judiciaires et Juridiques
- La Direction Générale des Ressources Humaines.
- La Direction Générale des Finances et des Moyens .
- La Direction Générale de la Modernisation de la Justice .
- La Direction Générale de l'Administration Pénitentiaire et de la Réinsertion, régie par un texte particulier.

2.4 Le rôle et les missions de l'avocat

Le système juridique en Algérie est géré principalement par la contribution de plusieurs avocats distribués dans le territoire national afin de fournir des services permanents aux citoyens. La profession d'avocat existe en Algérie depuis 1848, plusieurs organisations professionnelles de la profession .[18]

2.4.1 Le premier juge

Lors de la première consultation, l'avocat est le premier juge de l'affaire qui lui est présentée. Il analyse et évalue de manière objective le différend, les droits et intérêts des parties, ainsi que les chances de succès du dossier.

S'il considère que la « cause » qui lui est présentée n'est pas juste, l'avocat doit refuser de conseiller le justiciable ou de défendre.

2.4.2 Le conseil

L'avocat conseille son client avant, pendant et après le recours à un mode alternatif de règlement des conflits (arbitrage, conciliation, droit collaboratif et médiation). Il en va de même lorsque le différend fait l'objet d'éventuelles procédures (judiciaires, administratives et/ou disciplinaires)

En règle générale, l'avocat qualifie les faits qui lui sont exposés (c'est-à-dire qu'il les "traduit" juridiquement), avant de poser un « diagnostic », critique et objectif, de la situation. Il lui appartient de préconiser les mesures à prendre à titre conservatoire, défensif, voire offensif.

L'avocat joue également un rôle essentiel dans le cadre des modes de règlement alternatif des conflits : il concourt au bon déroulement du processus notamment en aidant son client à clarifier ses objectifs et à distinguer sa position et ses intérêts.

2.4.3 Le mandataire

L'avocat pose des actes juridiques au nom et pour le compte de son client. Il est donc un mandataire de son client.

2.4.4 Le défenseur

L'avocat défend les droits et intérêts de son client dans le respect de la loi et des règles déontologiques qui s'imposent à lui. Il présente de manière claire, argumentée et professionnelle la situation et la thèse de son client. En sa qualité d'auxiliaire de la justice, l'avocat veille au respect des règles et principes fondamentaux, tels les droits de la défense, la contradiction des débats, etc.

2.4.5 Le conciliateur :

Tant avant l'introduction d'une procédure judiciaire (administrative ou autre) que durant celle-ci, l'avocat tente autant que faire se peut de favoriser et de concourir à un règlement amiable du litige.

2.4.6 Le négociateur

L'avocat est également un négociateur. Ainsi, il assiste ou représente son client par exemple pour la conclusion de contrat(s) et/ou d'affaire(s) ou pour la résolution amiable d'un conflit.

2.4.7 Le plaideur

La plaidoirie est l'exposé oral de l'argumentation du client, effectuée par l'avocat devant les cours et tribunaux. Lors de sa plaidoirie, l'avocat porte la toge, sauf certaines exceptions. Ainsi, l'avocat ne porte pas de toge dans les justices de Paix dont le siège est établi à l'extérieur du palais de justice proprement dit.

2.5 Avocat visité

Afin de consolider nos informations et pour assurer une bonne compréhension du travail des avocats nous avons réalisé une suite de visites régulièrement au bureau de l'avocate « boudraa Faizî » de la wilaya de Mila , durant nos visites nous avons collecté pas mal d'informations exploitées dans la conception et l'analyse de notre système.



2.6 Conclusion

Dans ce chapitre nous avons résumé les notions de base liées à notre domaine d'étude avec une brève présentation du système juridique en Algérie

Chapitre 3

l'étude préliminaire

3.1 Introduction

L'étude préliminaire est la première phase de notre processus de développement. Elle consiste à effectuer un repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou des diagrammes très simples.

Nous commencerons à déterminer les besoins fonctionnels en considérant le système comme une boîte noire, afin d'étudier sa place dans le système. Après avoir identifié les acteurs qui interagissent avec le système, nous développerons un premier modèle UML au niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système.

3.2 Situation de l'étude préliminaire dans 2TUP

L'étude préliminaire est la première étape dans le processus de développement 2TUP. Elle prépare les étapes les plus formelles de la capture des besoins fonctionnels et de la capture des besoins techniques. [7]

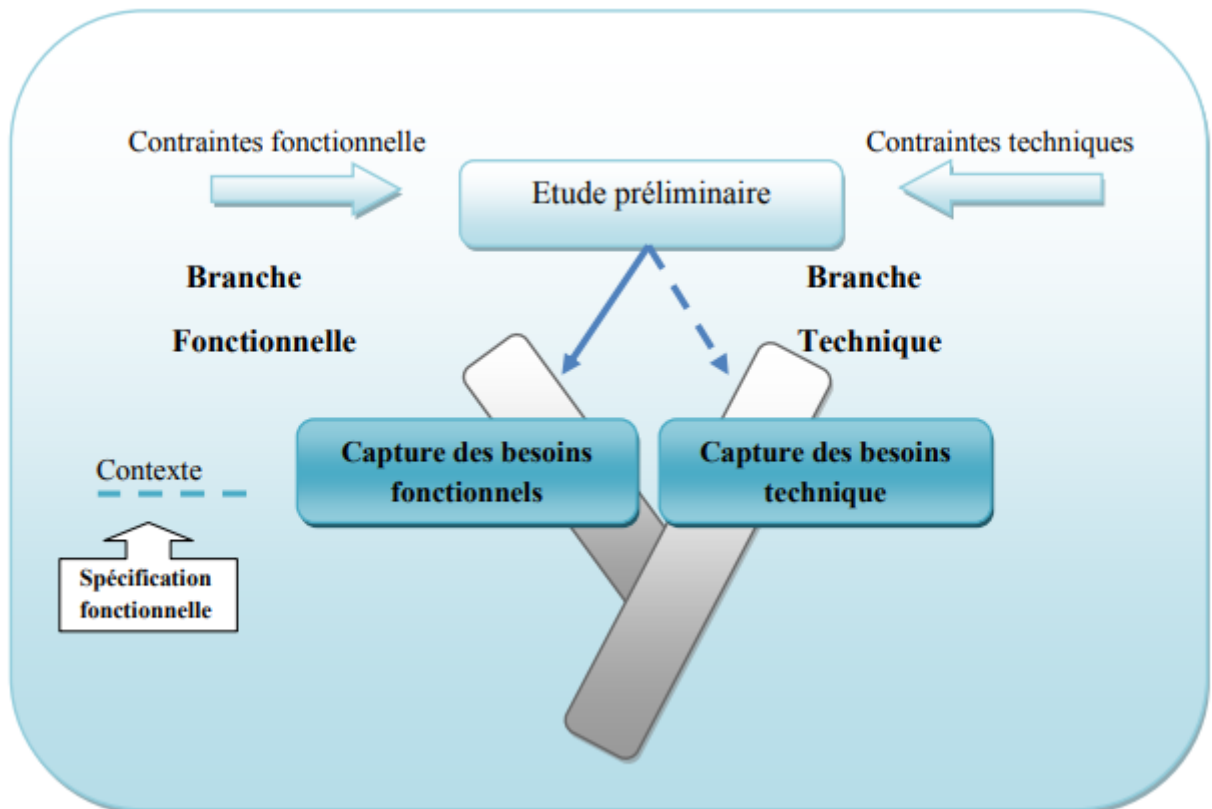


FIGURE 3.1 – Situation de l'étude préliminaire dans 2TUP.

3.3 Élaboration du cahier des charges

Un cahier des charges est un document qui doit être respecté lors de la réalisation d'un projet. Il a pour objectif de définir et de décrire tous les détails et les spécifications d'un projet à réaliser. [7]

L'élaboration du cahier de charges s'effectue sur plusieurs étapes :

- La présentation du projet.
- Les grands choix techniques.
- Le recueil des besoins fonctionnels.
- Le recueil des besoins non fonctionnels.
- Identifications des acteurs.
- Identifier les messages.
- Modélisation du contexte.

3.3.1 La présentation du projet

Notre Mission dans le cadre de ce projet est de créer une application mobile permettant de gérer un cabinet d'avocat, il s'agit de définir la responsabilité de la gestion, mettre à jour les données, organiser des données collectées auprès de l'avocat afin de

concevoir des fichiers de bases , de renforcer le contrôle et la confrontation, assurer une meilleure gestion et une cohérence de l'information. Notre application aura comme principale fonctionnalités :

- Gestion des affaire..
- Gestion des procédures..
- Gestion des factures.
- Gestion de la relation clientèle..

3.3.2 Problématique

Le travail des avocats dans nos jours est devenue une procédure délicate qui nécessite un contact permanent entre l'avocat et ses clients. Tous type d'affaire judiciaire demande plusieurs séances de discussion entre le clients et les avocats. Généralement ; le clients aime toujours consulter dans les plus proches délais les démarches juridiques réalisées dans ses affaires et en même temps les avocat préfère un contacte directe avec les clients et d'une manière souple et en temps réel afin d'assurer la réussite des démarches. Ce type de communication client avocat est réalisé généralement par téléphones par des contacts directs ce qui retard la transmission des informations entre ces deux acteur.

3.3.3 L'Objectifs

L'objectif de notre projet est la conception et la réalisation d'une application mobile pour la gestion d'un cabinet d'avocat de telle sorte qu'elle soit rapide et efficace afin de résoudre les problèmes cités si haut d'un côté, et de l'autre côté faire inclure l'utilisation des nouveaux techniques à notre vie, en effet couramment n'importe qui peut utiliser les nouvelles sortes de technologie. Alors notre application doit reprendre aux besoins suivants :

- faciliter le travaille d'avocat.
- Faciliter la communication entre l'avocat et le client.
- réduction du temps de communication entre ces acteurs.
- L'organisation des données.
- Assurer une meilleure gestion et une cohérence de l'information.

3.3.4 Les grands choix techniques

Nous allons utiliser un certain nombre de techniques-clés qui sont principalement :

- Le langage JAVA pour la programmation de la logique applicative de ce projet dans une plateforme Android.

- Pour la modélisation on va utiliser la méthode de développement 2TUP à partir de langage standard UML.
- Système de gestion de bases de données relationnelles(MySQL).

3.3.5 Spécification des besoins fonctionnels

Après une étude du système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces besoins se regroupent dans les diagrammes des cas d'utilisation.

Ici on va décrire ces différents besoins, pour avoir ultérieurement présenté par le diagramme de cas d'utilisation :

- **Consulter Demande**

L'avocat consulte la liste des demandes des clients.

- **Confirmer Demande :**

L'avocat après la consultation des demandes des clients va envoyer un message de confirmation avec les papiers de dossier ou un message de refusé.

- **Consulter procuration**

L'avocat consulte les procurations des clients.

- **Gérer Client**

L'avocat doit gérer les clients (ajouter, modifier, supprimer) selon la liste des demandes.

- **Gérer séance tribunal**

L'avocat doit gérer les séances tribunal (ajouter, modifier, supprimer) selon la date des séances.

- **Gérer Affaire**

L'avocat doit gérer les affaires (ajouter, modifier, supprimer) selon la date des jugements.

- **Gérer Facture**

L'avocat doit gérer les factures (ajouter, modifier, supprimer) selon le type d'affaire.

- **Effectuer statistiques**

Les statistiques représentent un besoin essentiel qui permet d'avoir une vision globale.

Des différentes affaires, nous citons à titre d'exemples :

- ✓ Le nombre total des affaires.
- ✓ Le nombre total des affaires par type.
- ✓ Le nombre des affaires réalisées par mois.

- **Demander consultation**

Le client demande la consultation de la demande envoyer.

- **Etablir procuration**

Après chaque demande le client il faut éditer une procuration pour l'avocat.

- **Consulter Séance Tribunal**

Le client consulte la séance de son affaire.

- **Consulter Affaire**

Le client consulte la date de jugement.

- **Consulter Facture**

Le client consulte sa facture.

3.3.6 Spécification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé les besoins suivants :

- **L'authentification**

Chaque utilisateur de l'application doit s'authentifier par un nom d'utilisateur et un mot de passe, pour qu'il puisse utiliser le système.

- **Interfaces graphiques**

L'interface de cette application doit être simple et claire, il doit être bien organisé du point de vue graphique, le choix des couleurs, et des styles.

- **La rapidité d'accès**

Le système doit pouvoir répondre aux demandes des utilisateurs en temps réel.

- **La sécurité**

Les données des utilisateurs doit être protégé.

3.3.7 Identification des acteurs

Un acteur est un utilisateur type de système. Il représente une responsabilité par rapport au système ou un rôle plutôt qu'une personne physique. Il est donc une entité externe qui interagit directement avec le système en émettant et en recevant des messages. [8]

- **L'avocat**

- Superviseur et administrateur du système.
- L'avocat assure la consultation des demandes et confirme ces demande et consulte le dossier de chaque client.
- Créer des procédures pour chaque client, et toute procédure contient un ou plusieurs pièces jointes.
- Gérer séance tribunal, facture et Gérer historique d'affaire.
- Effectuer statistiques.

- **Le client**

Il a le droit de suivi l'avancement de son dossier. connaitre a tout moment sa situation financière, Ainsi, d'imprimer sa factur, Consulter la date de séance.

3.3.8 Identification des messages

Un message représente la spécification d'une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur. Un message est normalement associé à deux occurrences d'événements : un événement d'envoi et un événement de réception [9].

- **Messages émis par le système**

- L'Affichage de la liste des demandes.
- L'Affichage de la liste des clients.
- L'Affichage de la liste des procuration des client.
- Le formulaire (d'ajouter, modifier, supprimer) séance tribunal.

- Le formulaire (d'ajouter, modifier, supprimer) facture.
- Le formulaire (d'ajouter, modifier, supprimer) affaire.
- Résultat des statistiques.

• **Messages reçus par le système**

- La saisie des informations d'authentification.
- Demande la liste des demandes des clients.
- Demande la liste des clients..
- Demande d'établir procuration.
- Demande (d'ajouter, modifier, supprimer) séance tribunal.
- Demande (d'ajouter, modifier, supprimer) facture.
- Demande (d'ajouter, modifier, supprimer) Affaire.
- Demande de faire des statistiques.
- Demande de consulter facture.
- Demande de consulter affaire.
- Demande de consulter séance tribunal.

3.3.9 Le diagramme de contexte dynamique

Un diagramme de contexte dynamique c'est un diagramme de communication qui permet de positionner le système étudié dans son environnement. Ce diagramme précise les échanges d'informations qui sont réalisés entre notre système et les éléments extérieurs au système.

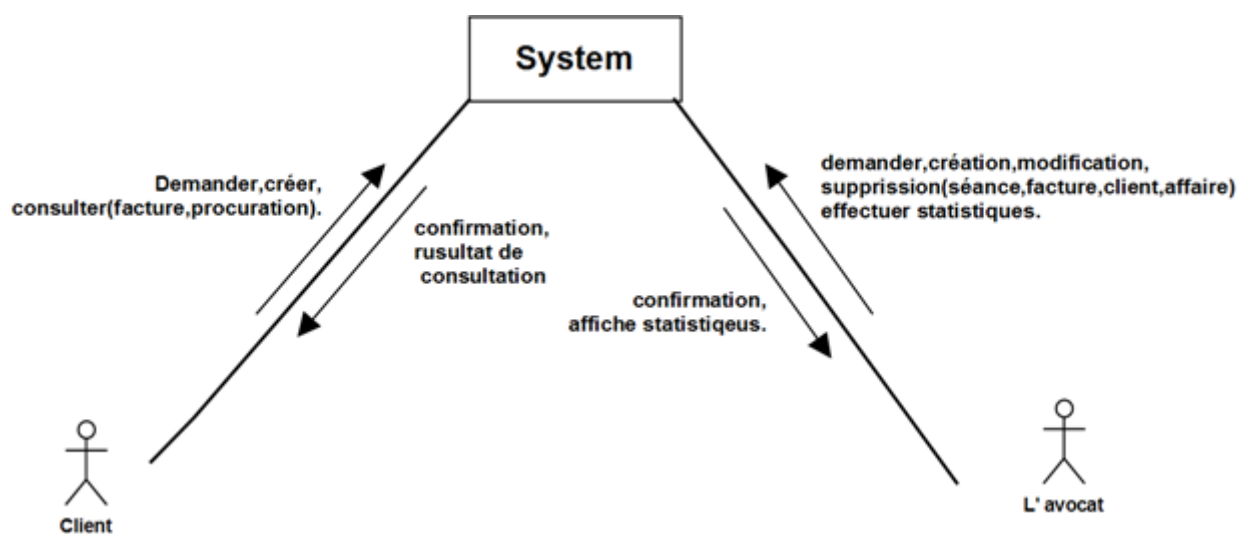


FIGURE 3.2 – Diagramme du contexte.

3.4 Conclusion

A la fin de ce chapitre, on a présenté les différents besoins nécessaires pour le fonctionnement de notre application que ce soit fonctionnel ou non fonctionnel, ainsi les divers acteurs interagissant avec le système. On peut dire que cet étape est vraiment nécessaire et basique pour avancé dans le développement de notre projet et pour la préparation de l'étape suivante.

Capture des besoins

4.1 Introduction

Après qu'on a défini les besoins fonctionnels nécessaires pour le fonctionnement de notre application, dans ce chapitre on présente ses besoins sous forme des cas d'utilisation par un diagramme de cas d'utilisation, ce qui nous donne une modélisation du système par cas d'utilisation. Ainsi, on va donner une fiche descriptive de chaque cas avec la description graphique correspondante.

4.2 Situation de la capture des besoins dans 2TUP

La deuxième étape du processus suivie c'est **Capture des besoins**, la figure suivante représente la situation de cette phase dans le processus 2TUP :

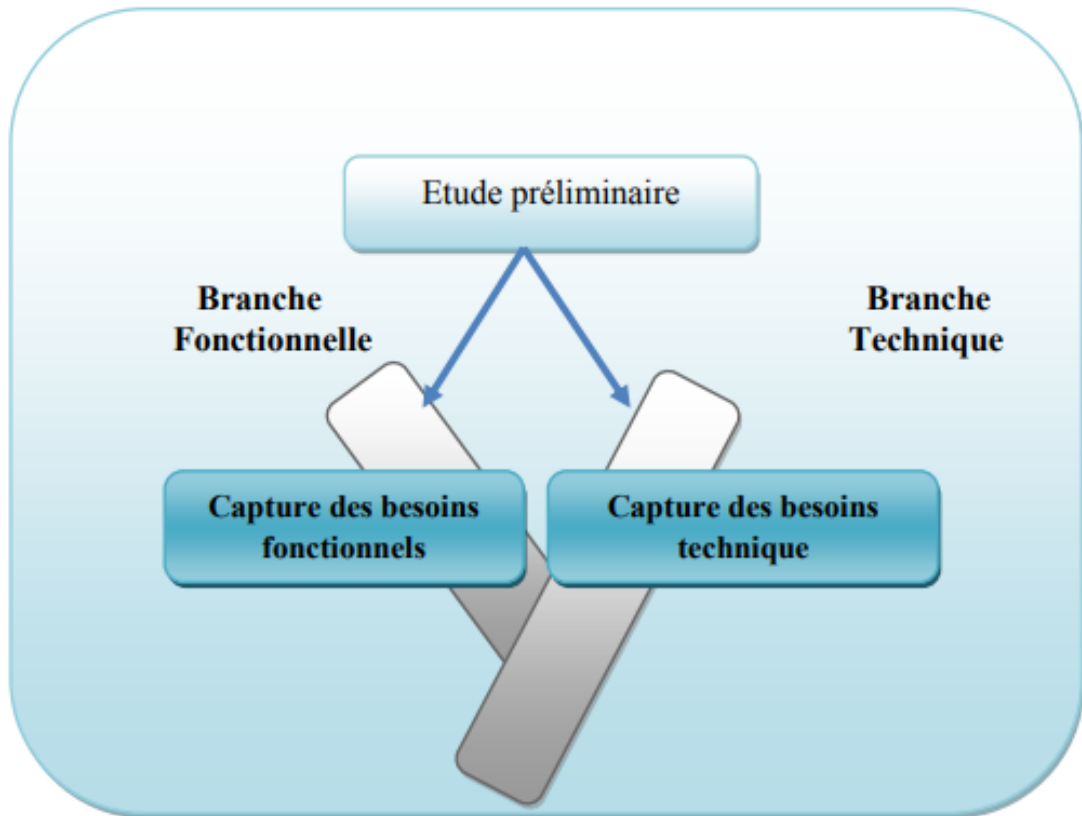


FIGURE 4.1 – la situation de la capture des besoins dans 2TUP.

4.3 Capture des besoins fonctionnels

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y de la méthode 2TUP. Elle formalise et détaille ce qui a été recueilli au cours de l'étude préliminaire. Cette phase représente un point de vue « fonctionnel » de l'architecture système. Elle est complétée au niveau de la branche droite du Y par la capture des besoins techniques et prépare l'étape suivante de la branche gauche : l'analyse.[8]

Dans cette phase on a permet de :

- Identifier les cas d'utilisations du système par ces acteurs.
- Décrire les cas d'utilisation.
- Identifier les classes candidates.
- Elaborer les diagrammes des classes participantes des cas d'utilisation.

4.3.1 Identification des cas d'utilisation

Nous avons identifié les cas d'utilisation et leurs acteurs lors de la première étape (étude préliminaire), nous allons directement les représenter graphiquement sur un diagramme de cas d'utilisation qui représente le fonctionnement du système vis-à-vis de l'utilisateur.

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. [8]

Diagramme de cas d'utilisation

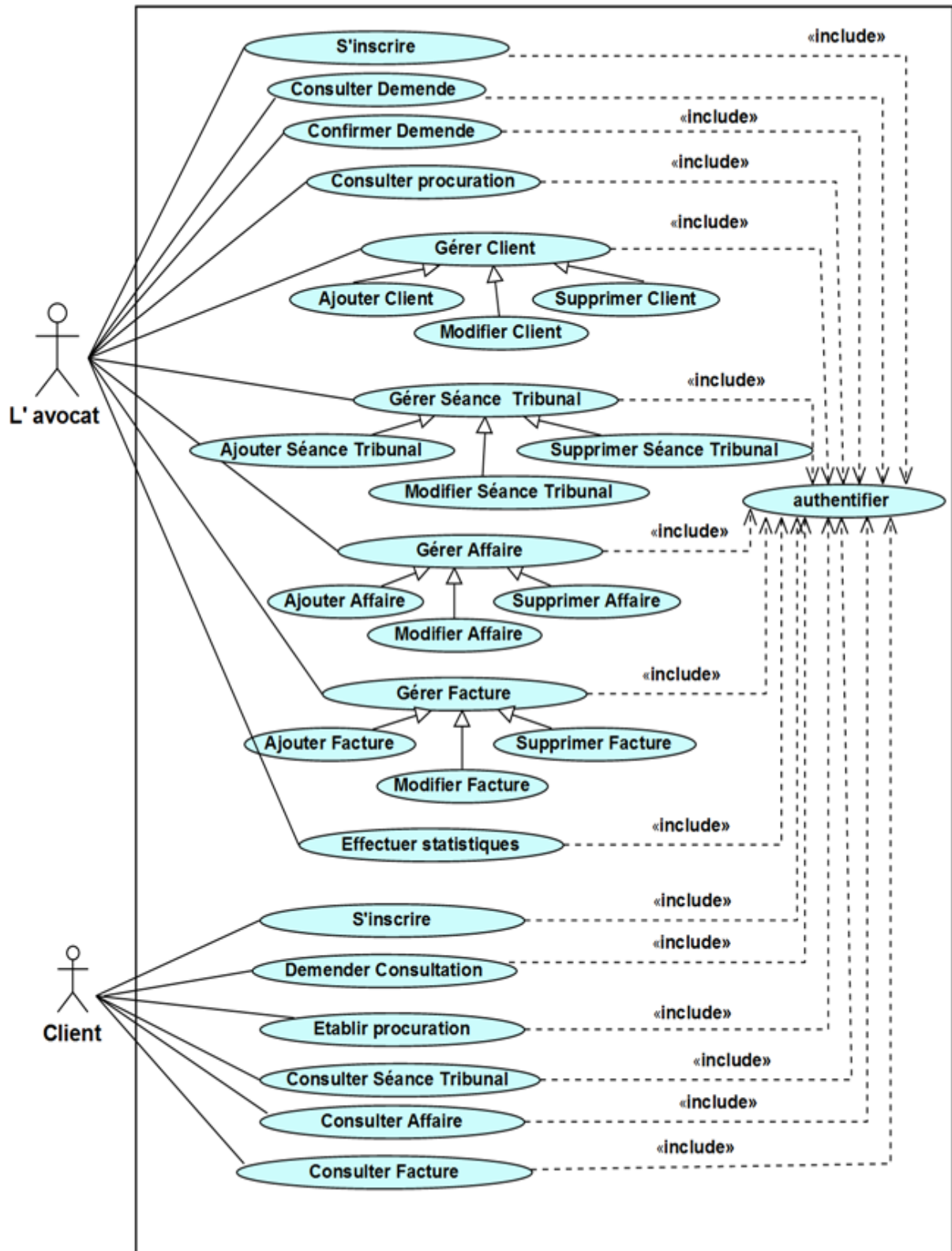


FIGURE 4.2 – Diagramme de cas d'utilisation générale

4.3.2 Description des cas d'utilisation

4.3.2.1 Description textuelle des cas d'utilisation :

Nous décrivons chaque cas de façon textuelle en lui associant un nom, un objectif, les acteurs qui y participent, les pré-conditions et des scénarios. Ensuite, les scénarios sont illustrés par les diagrammes d'activités et de séquences.

- **S'inscrire :**

Cas d'utilisation	S'inscrire.
Acteur	Utilisateur(client,avocat).
But	Ce cas permet à l'utilisateur de s'inscrire.
Poste condition	L'utilisateur est inscrit.
Scénario nominal	<p>1-L'utilisateur demande d'inscrire dans l'application.</p> <p>2-Le système affiche le formulaire d'inscription.</p> <p>3-L'utilisateur remplit les champs du formulaire et confirme l'inscription.</p> <p>4-Le système vérifie la validité des informations saisies et les enregistrées.</p> <p>5- Le système affiche un message de confirmation d'inscription «Inscription avec succès ».</p>
Scénario alternatif	<ul style="list-style-type: none"> • Les informations saisies sont invalide.le système affiche un message d'erreur. On reprend scénario nominal au point 2. • L'utilisateur existe déjà dans l'application.le système affiche un message d'erreur. On reprend scénario nominal au point 2. • L'utilisateur peut annuler l'opération en cours à n'importe quel moment pourvue qu'il ne l'a pas validé.

TABLE 4.1: Description textuelle du cas d'utilisation « **S'inscrire** »

- Consulter demande :

Cas d'utilisation	Consulter demande.
Acteurs	L'avocat
But	Permettre a l'avocat de consulte la liste des demandes.
Pré-condition	L'avocat est authentifié.
Poste condition	L'avocat consulte la liste des demandes.
Scénario-nominal	1-L'avocat consulte la liste des demandes. 2- Le système affiche la liste des demandes. 3- L' avocat sélectionne une demande. 4- Le system affiche la description d'affaire du client choisie .
Scénario alternatif	•Lorsque la Liste des demandes et vide le système affiche une notification «liste vide».

TABLE 4.2: Description textuelle du cas d'utilisation
« Consulter Demande » .

- Confirmer Demande :

Cas d'utilisation	Confirmer Demande.
Acteurs	L'avocat
But	Permettre a l'avocat de confirmer les demandes reçues.
Pré-condition	L'avocat est authentifié.
Poste condition	L'avocat consulte la liste des demandes.
Scénario-nominal	1- L' avocat demande au system la liste des demandes des clients. 2- Le système affiche la liste des demandes des clients. 3- L' avocat sélectionne une demande du client. 4- Le système affiche la demande du client. 5- L'avocat envoie un message de confirmation.
Scénario alternatif	•L' avocat envoie un message demande refusée. • Lorsque la liste des demandes et vide le système affiche une notification «liste vide».

TABLE 4.3: Description textuelle du cas d'utilisation
« Confirmer Demande »

- **Consulter procuration :**

Cas d'utilisation	Consulter procuration
Acteurs	L'avocat
But	Permettre a l'avocat de consulte les procurations.
Pré-condition	L'avocat est authentifié.
Poste condition	L'avocat consulte la liste des procurations.
Scénario-nominal	1- L' avocat demande la liste des procurations. 2- Le système affiche la liste des procurations. 3- L' avocat sélectionne une procuration. 4- Le system affiche la procuration du client.
Scénario alternatif	<ul style="list-style-type: none"> • Lorsque la Liste des procurations et vide le système affiche une notification «liste vide».

TABLE 4.4: Description textuelle du cas d'utilisation
« **Consulter procuration** »

- **Gérer Client :**

Cas d'utilisation	Gérer Client .
Acteurs	L'avocat
But	Permettre a l'avocat d' ajouter, modifier, supprimer un client.
Pré-condition	L'avocat est authentifié.
Poste condition	Les informations ont été enregistrées.
Scénario-nominal	<ul style="list-style-type: none"> • Commence lorsque l'avocat demande au système la gestion. des Client. 1- L'avocat accédé au système. 2- Le système présente l'état actuel. 3- L'avocat choisit une des actions suivantes : <ul style="list-style-type: none"> • Ajouter client. • Modifier client. • Supprimer client.

TABLE 4.5: Description textuelle du cas d'utilisation
« **Gérer Client** »

- **Ajouter Client :**

Cas utilisation	Ajouter Client .
Acteur	L'avocat.
But	Permettre a l'avocat d'ajouter un client.
Pré condition	L'avocat est authentifié.
Poste condition	Nouvelle client ajouté par le système.
Scénario nominal	1-L'avocat demande au système d'ajouter un client . 2-Le Système affiche un formulaire d' ajout. 3-L' avocat remplit le formulaire et enregistre les informations . 4-Le système vérifie les informations puis affiche une notification «Ajout avec succès »
Scénario alternatif	<ul style="list-style-type: none"> • Le cas ou le formulaire est mal rempli ou incomplet le system notifie l'avocat et lui propose de corriger les champs et retourne à l'étape 2. •Lorsque l'avocat choisit l'option annulé, le système affiche une notification« L'opération est annulée».

TABLE 4.6: Description textuelle du cas d'utilisation

« Ajouter Client ».

- **Modifier Client :**

Cas d'utilisation	Modifier Client .
Acteur	L'avocat.
But	Permettre a l'avocat de modifier les informations d'un client.
Pré condition	L'avocat est authentifié.
Poste condition	Les modifications sont enregistrées par le système.
Scénario nominal	<p>1- L'avocat demande au système une modification de client.</p> <p>2-Le système affiche la liste des clients.</p> <p>3- L'avocat sélectionné un client.</p> <p>4-Le système affiche un formulaire pour changer les informations de ce client.</p> <p>5-L'avocat saisit les nouvelles informations de client et demande la validation.</p> <p>6- Le système enregistré la modification puis affiche une notification « modification avec succès».</p>
Scénario alternatif	<ul style="list-style-type: none"> • Les informations sont incorrectes ou incomplètes,le système affiche un message d'erreur, reprise du scénario nominale au point 5. • lorsque la Liste des clients et vide le système affiche une notification «liste vide». • Lorsque l'avocat choisit l'option annulé, le système affiche une notification « L'opération a été annulée».

TABLE 4.7: Description textuelle du cas d'utilisation
« **Modifier Client** »

- **Supprimer Client :**

Cas utilisation	Supprimer Client.
Acteur	L'avocat.
But	Permettre a l'avocat de supprimer un client.
Pré condition	L'avocat est authentifié.
Poste condition	Le client est supprimé.
Scénario nominal	<ol style="list-style-type: none"> 1-L'avocat demande la suppression d'un client. 2-Le système lui affiche la liste des clients. 3-L'avocat sélectionne le client qu'il veut supprimer. 4-Le système demande la validation de la suppression. 5- L'avocat valide la suppression. 6-Le système supprime le client puis affiche une notification « suppression avec succès ».
Scénario alternatif	<ul style="list-style-type: none"> • lorsque la Liste des clients et vide le système affiche une notification «liste vide». •Lorsque l'avocat choisit l'option annulé, le système affiche une notification « L'opération a été annulée».

TABLE 4.8: Description textuelle du cas d'utilisation
« **supprimer Client** ».

- **Gérer séance tribunal :**

Cas d'utilisation	Gérer séance tribunal.
Acteurs	L'avocat
But	Permettre a l'avocat de ajouter, modifier, supprimer une séance tribunal.
Pré-condition	L'avocat est authentifié.
Poste condition	Les informations ont été enregistrées.
Scénario-nominal	<ul style="list-style-type: none"> • Commence lorsque l'avocat demande au système la gestion des séances tribunal. 1- L'avocat accédé au système. 2- Le système présente l'état actuel. 3- L'avocat choisit une des actions suivantes : <ul style="list-style-type: none"> • Ajouter séance tribunal. • Modifier séance tribunal. • Supprimer séance tribunal.

TABLE 4.9: Description textuelle du cas d'utilisation
« **Gérer séance tribunal** » .

- **Ajouter séance tribunal :**

Cas utilisation	Ajouter séance tribunal .
Acteur	L'avocat.
But	Permettre a l'avocat d'ajouter une séance du tribunal.
Pré condition	L'avocat est authentifié.
Poste condition	Nouvelle séance ajouté par le système.
Scénario nominal	<ol style="list-style-type: none"> 1- L'avocat demande au système d'ajouter une séance. 2. Le Système affiche un formulaire d' ajout. 3. L 'avocat remplit le formulaire et enregistre les informations 4-Le système vérifie les informations puis affiche une notification «Ajout avec succès »
Scénario alternatif	<ul style="list-style-type: none"> • Le cas ou le formulaire est mal rempli ou incomplet le system notifie l'avocat et lui propose de corriger les champs et retourne à l'étape 2. • Lorsque l'avocat choisit l'option annulé, le système affiche une notification« L'opération est annulée».

TABLE 4.10: Description textuelle du cas d'utilisation
« **Ajouter séance tribunal** ».

Modifier séance tribunal :

Cas d'utilisation	Modifier séance tribunal .
Acteur	L'avocat.
But	Permettre a l'avocat modifier les informations de la séance du tribunal.
Pré condition	L'avocat est authentifié.
Poste condition	Les modifications sont enregistrées par le système.
Scénario nominal	<ol style="list-style-type: none"> 1- L'avocat demande au système une modification de séance. 2-Le système affiche la liste des séances. 3- L'avocat sélectionné une séance. 4-Le système affiche un formulaire pour changer les informations de cette séance. 5-L'avocat saisit les nouvelles informations de séance et demande la validation. 6- Le système enregistré la modification puis affiche une notification « modification avec succès».
Scénario alternatif	<ul style="list-style-type: none"> • Les informations sont incorrectes ou incomplètes,Le système affiche un message d'erreur, Reprise du scénario nominale au point 5. • lorsque la Liste des séances et vide le système affiche une notification «liste vide». • Lorsque l'avocat choisit l'option annulé, le système affiche une notification « L'opération a été annulée».

TABLE 4.11: Description textuelle du cas d'utilisation
« **Modifier séance tribunal** ».

- **supprimer séance tribunal :**

Cas utilisation	supprimer séance tribunal.
Acteur	L'avocat.
But	Permettre a l'avocat de supprimer une séance tribunal.
Pré condition	L'avocat est authentifié.
Poste condition	La séance est supprimé.
Scénario nominal	<ol style="list-style-type: none"> 1-L'avocat demande la suppression d'une séance. 2-Le système lui affiche la liste des séances. 3-L'avocat sélectionne la séance qu'il veut supprimer. 4-Le système demande la validation de la suppression. 5- L'avocat valide la suppression 6-Le système supprime la séance puis affiche une notification « suppression avec succès ».
Scénario alternatif	<ul style="list-style-type: none"> •lorsque la Liste des clients et vide le système affiche une notification «liste vide». •Lorsque l'avocat choisit l'option annulé, le système affiche une notification « L'opération a été annulée».

TABLE 4.12: Description textuelle du cas d'utilisation
« **Supprimer séance tribunal** ».

- **Gérer Affaire :**

Cas d'utilisation	Gérer Affaire.
Acteurs	L'avocat
But	Permettre a l'avocat d'ajouter, modifier, supprimer une affaire.
Pré-condition	L'avocat est authentifié.
Poste condition	Les informations ont été enregistrées.
Scénario-nominal	<ul style="list-style-type: none"> • Commence lorsque l'avocat demande au système la gestion. d'affaire.. 1- L'avocat accédé au système. 2- Le système présente l'état actuel. 3- L'avocat choisit une des actions suivantes : <ul style="list-style-type: none"> • Ajouter Affaire. • Modifier Affaire. • Supprimer Affaire.

TABLE 4.13: Description textuelle du cas d'utilisation
« **Gérer Affaire** ».

- **Ajouter Affaire :**

Cas utilisation	Ajouter Affaire.
Acteur	L'avocat.
But	Permettre a l'avocat d'ajouter une affaire.
Pré condition	L'avocat est authentifié.
Poste condition	Nouvelle affaire ajouté par le système.
Scénario nominal	<ol style="list-style-type: none"> 1-L'avocat demande au système d'ajouter une affaire. 2.Le Système affiche un formulaire d' ajout. 3.L'avocat remplit le formulaire et enregistre les informations. 4-Le système vérifie les informations puis affiche une notification «Ajout avec succès »
Scénario alternatif	<ul style="list-style-type: none"> • Le cas ou le formulaire est mal rempli ou incomplet le system notifie l'avocat et lui propose de corriger les champs et retourne à l'étape 2. • Lorsque l'avocat choisit l'option annulé, le système affiche une notification« L'opération est annulée».

TABLE 4.14: Description textuelle du cas d'utilisation
« **Ajouter Affaire** ».

- **Modifier Affaire :**

Cas d'utilisation	Modifier Affaire .
Acteur	L'avocat.
But	Permettre a' l'avocat de modifier les informations d'une affaire.
Pré condition	L'avocat est authentifié.
Poste condition	Les modifications sont enregistrées par le système.
Scénario nominal	<p>1-L'avocat demande au système une modification d'affaire.</p> <p>2-Le système affiche la liste des affaires.</p> <p>3- L'avocat sélectionne une affaire.</p> <p>4-Le système affiche un formulaire pour changer les informations de cette affaire.</p> <p>5-L'avocat saisit les nouvelles information de l'affaire et demande la validation .</p> <p>6-Le système enregistré les modifications puis affiche une notification « modification avec succès».</p>
Scénario alternatif	<ul style="list-style-type: none"> • Les informations sont incorrectes ou incomplètes,Le système affiche un message d'erreur, reprise du scénario nominale au point 5. • lorsque la Liste des affaires et vide le système affiche une notification «liste vide». • Lorsque l'avocat choisit l'option annulé, le système affiche une notification « L'opération a été annulée».

TABLE 4.15: Description textuelle du cas d'utilisation
« **Modifier Affaire** ».

- **Supprimer Affaire :**

Cas utilisation	Supprimer Affaire .
Acteur	L'avocat.
But	Permettre a l'avocat de supprimer une affaire.
Pré condition	L'avocat est authentifié.
Poste condition	L'affaire est supprimée.
Scénario nominal	<p>1-L'avocat demande la suppression d une affaire.</p> <p>2-Le système lui affiche la liste des affaires.</p> <p>3-L'avocat sélectionne l affaire qu'il veut supprimer.</p> <p>4-Le système demande la validation de la suppression.</p> <p>5- L'avocat valide la suppression</p> <p>6-Le système supprime l'affaire puis affiche une notification « suppression avec succès ».</p>
Scénario alternatif	<ul style="list-style-type: none"> •lorsque la Liste des affaires et vide le système une notification «liste vide». •Lorsque l'avocat choisit l'option annulé, le système affiche une notification « L'opération a été annulée».

TABLE 4.16: Description textuelle du cas d'utilisation
« **Supprimer Affaire** ».

- **Gérer Facture :**

Cas d'utilisation	Gérer Facture.
Acteurs	L'avocat
But	Permettre a l'avocat d' ajouter, modifier, supprimer une facture.
Pré-condition	L'avocat est authentifié.
Poste condition	Les informations ont été enregistrées.
Scénario-nominal	<ul style="list-style-type: none"> • Commence lorsque l'avocat demande au système de la facture : <ol style="list-style-type: none"> 1- L'avocat accédé au système. 2- Le système présente l'état actuel. 3- L'avocat choisit une des actions suivantes : <ul style="list-style-type: none"> • Ajouter facture. • Modifier facture. • Supprimer facture.

TABLE 4.17: Description textuelle du cas d'utilisation
« Gérer facture ».

- **Ajouter Facture :**

Cas utilisation	Ajouter Facture .
Acteur	L'avocat.
But	Permettre a l'avocat d'ajouter une facture.
Pré condition	L'avocat est authentifié.
Poste condition	Nouvelle facture ajoutée par le système.
Scénario nominal	<ol style="list-style-type: none"> 1-L'avocat demande au système d'ajouter une facture. 2.Le Système affiche un formulaire d' ajout. 3.L'avocat remplit le formulaire et enregistre les informations. 4-Le système vérifie les informations puis affiche une notification «Ajout avec succès »
Scénario alternatif	<ul style="list-style-type: none"> • Le cas ou le formulaire est mal rempli ou incomplet le system notifie l'avocat et lui propose de corriger les champs et retourne à l'étape 2. • Lorsque l'avocat choisit l'option annulé, le système affiche une notification« L'opération est annulée».

TABLE 4.18: Description textuelle du cas d'utilisation
« Ajouter facture ».

- **Modifier Facture :**

Cas d'utilisation	Modifier Facture .
Acteur	L'avocat.
But	Permettre a l'avocat de modifier les informations de la facture.
Pré condition	L'avocat est authentifié.
Poste condition	Les modifications sont enregistrées par le système.
Scénario nominal	<p>1-L'avocat demande au système une modification de facture.</p> <p>2-Le système affiche la liste des factures.</p> <p>3- L'avocat sélectionne une facture.</p> <p>4-Le système affiche un formulaire pour changer les informations de cette facture.</p> <p>5-L'avocat saisit les nouvelles informations de la facture .</p> <p>6-Le système enregistré les modifications puis affiche une notification « modification avec succès».</p>
Scénario alternatif	<ul style="list-style-type: none"> • Les informations sont incorrectes ou incomplètes,Le système affiche un message d'erreur, reprise du scénario nominale au point 5. • lorsque Liste des factures et vide le système affiche une notification «liste vide». • Lorsque l'avocat choisit l'option annulé, le système affiche une notification « L'opération a été annulée».

TABLE 4.19: Description textuelle du cas d'utilisation
« **Modifier facture** ».

- **Supprimer Facture :**

Cas utilisation	Supprimer Facture .
Acteur	L'avocat.
But	Permettre a l'avocat de supprimer une facture.
Pré condition	L'avocat est authentifié.
Poste condition	La facture est supprimée.
Scénario nominal	<ol style="list-style-type: none"> 1-L'avocat demande la suppression d'une facture. 2-Le système lui affiche la liste des factures. 3-L'avocat sélectionne la facture qu'il veut supprimer. 4-Le système demande la validation de la suppression. 5- L'avocat valide la suppression. 6-Le système supprime la facture puis affiche une notification « suppression avec succès ».
Scénario alternatif	<ul style="list-style-type: none"> •lorsque la Liste des factures et vide le système affiche une notification «liste vide». •Lorsque l'avocat choisit l'option annulé, le système affiche une notification « L'opération a été annulée».

TABLE 4.20: Description textuelle du cas d'utilisation
« **Supprimer facture** ».

- **Effectuer Statistiques :**

Cas utilisation	Effectuer statistiques.
Acteur	L'avocat.
But	Permettre a l'avocat d'effectuer des statistique.
Pré condition	L'avocat est authentifié.
Poste condition	Les statistiques sont affichées.
Scénario nominal	1-L'avocat demande au système de faire des statistiques. 2-Le Système affiche le formulaire statistique. 3-L' avocat remplit le formulaire et enregistre les information. 4-Le système vérifie les informations puis affiche le résultat .
Scénario alternatif	<ul style="list-style-type: none"> • Le cas ou le formulaire est mal rempli ou incomplet le system notifie l'avocat et lui propose de corriger les champs et retourne à l'étape 2. • Lorsque l'avocat choisit l'option annulé, le système affiche une notification« L'opération est annulée».

TABLE 4.21: Description textuelle du cas d'utilisation
« **Effectuer statistiques** ».

- **Demander Consultation :**

Cas d'utilisation	Demander Consultation .
Acteurs	Client
But	Permettre au client de demander consultation.
Pré-condition	Le Client est authentifié.
Poste condition	La demande est envoyée.
Scénario-nominal	1-Le client demande consultation. 2- Le système affiche le formulaire permettant la demande de consultation. 3-Le client remplit le formulaire et la demande est envoyée. 4-Le système confirme l'opération.
Scénario alternatif	<ul style="list-style-type: none"> •Le client annule la demande créée. •Les informations sont incorrectes ou incomplètes,alors Le système affiche un message d'erreur.

TABLE 4.22: Description textuelle du cas d'utilisation
« **Demander Consultation** ».

- **Etablir Procuration :**

Cas d'utilisation	Etablir Procuration .
Acteurs	Client
But	Permettre au client d'établir procuration.
Pré-condition	Client est authentifié.
Poste condition	La procuration est prête pour l'impression.
Scénario-nominal	1-Le Client demande création d'une procuration.. 2- Le système affiche le formulaire qui contient l'information nécessaire pour la procuration. 3-Le client remplit le formulaire et l'envoie. 4-Le système confirmer l opération.
Scénario alternatif	<ul style="list-style-type: none"> •Lorsque le client choisit l'option annulé le système affiche une notification « L'opération est annulée». •Les informations sont incorrectes ou incomplètes Le système affiche un message d'erreur.

TABLE 4.23: Description textuelle du cas d'utilisation
« **Etablir procuration** ».

- **Consulter séance tribunal :**

Cas d'utilisation	Consulter séance tribunal .
Acteurs	Client
But	Permettre au client de consulter la séance tribunal.
Pré-condition	Le client est authentifié.
Poste condition	Client consulter séance tribunal.
Scénario-nominal	1- Le client demande de consulter séance tribunal. 2-Le système affiche les informations de la séance tribunal. 3- le client consulté la séance tribunal.
Scénario alternatif	<ul style="list-style-type: none"> • Annuler consultation.

TABLE 4.24: Description textuelle du cas d'utilisation
«**Consulter séance tribunal** » .

- Consulter affaire :

Cas d'utilisation	Consulter affaire .
Acteurs	Client
But	Permettre au client de consulter affaire.
Pré-condition	Le client est authentifié.
Poste condition	Client consulte l'affaire.
Scénario-nominal	1- Le client demande de consulte l'affaire. 2- Le système affiche les informations d'affaire. 3- Le client consulter l'affaire.
Scénario alternatif	• Annuler consultation.

TABLE 4.25: Description textuelle du cas d'utilisation
«Consulter affaire » .

- Consulter facture :

Cas d'utilisation	Consulter facture
Acteurs	Client
But	Permettre au client de consulter facture.
Pré-condition	L'avocat est authentifié.
Poste condition	Client consulte la facture.
Scénario-nominal	1- Le client demande de consulter facture. 2- Le système affiche les informations de la facture. 3- Le client consulter la facture.
Scénario alternatif	• Annuler consultation.

TABLE 4.26: Description textuelle du cas d'utilisation
«Consulter facture ».

4.3.2.2 Description graphique des cas d'utilisation :

S'inscrire :

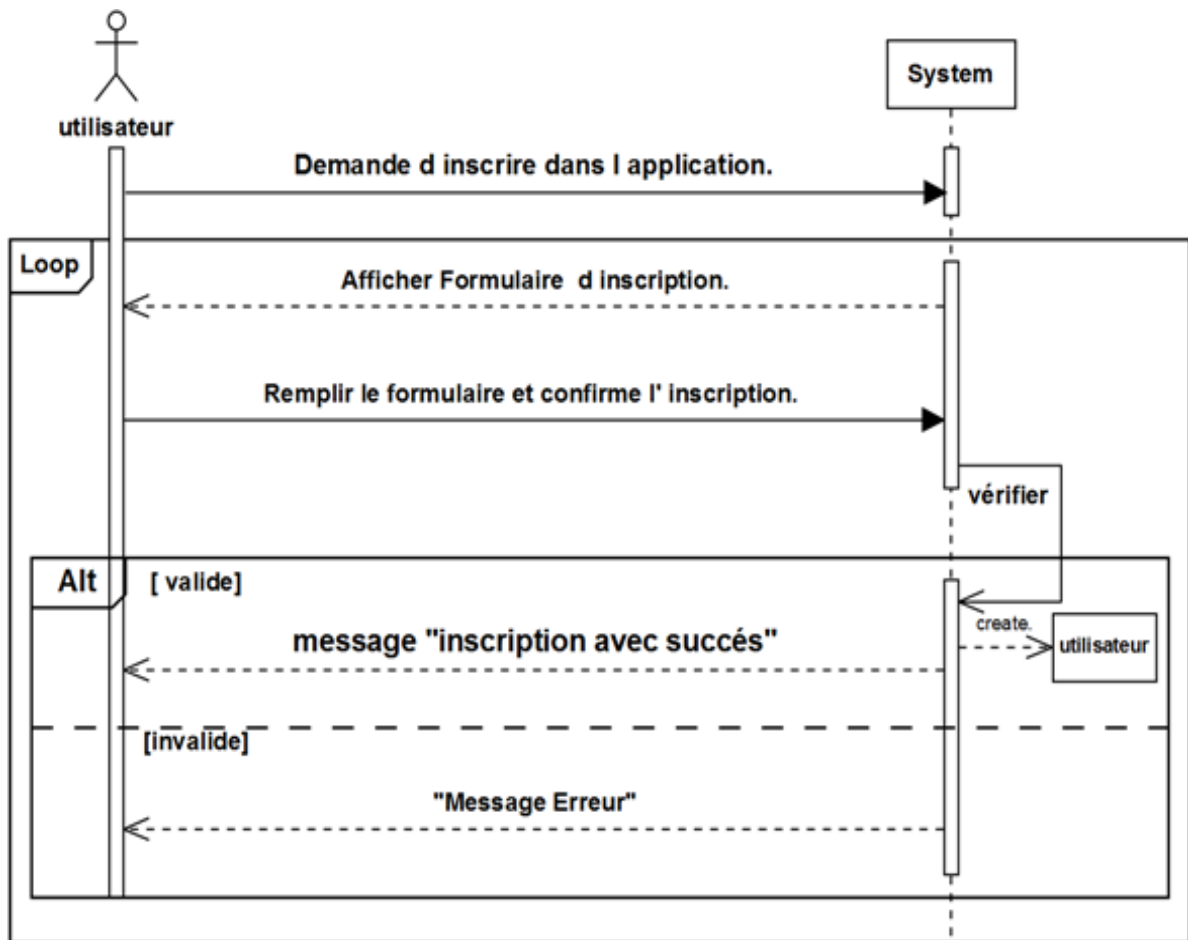


FIGURE 4.3 – Diagramme de séquence du cas «S'inscrire»

Consulter demande :

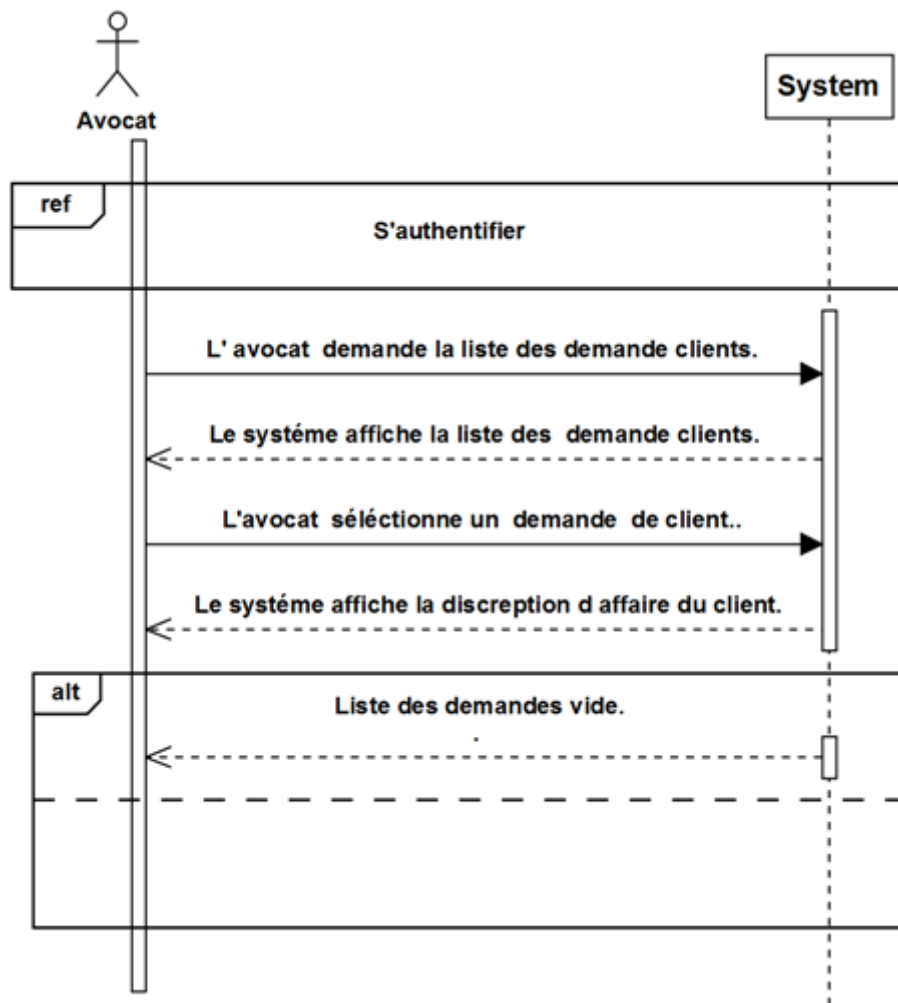


FIGURE 4.4 – Diagramme de séquence du cas consulter demande

Confirmer Demande :

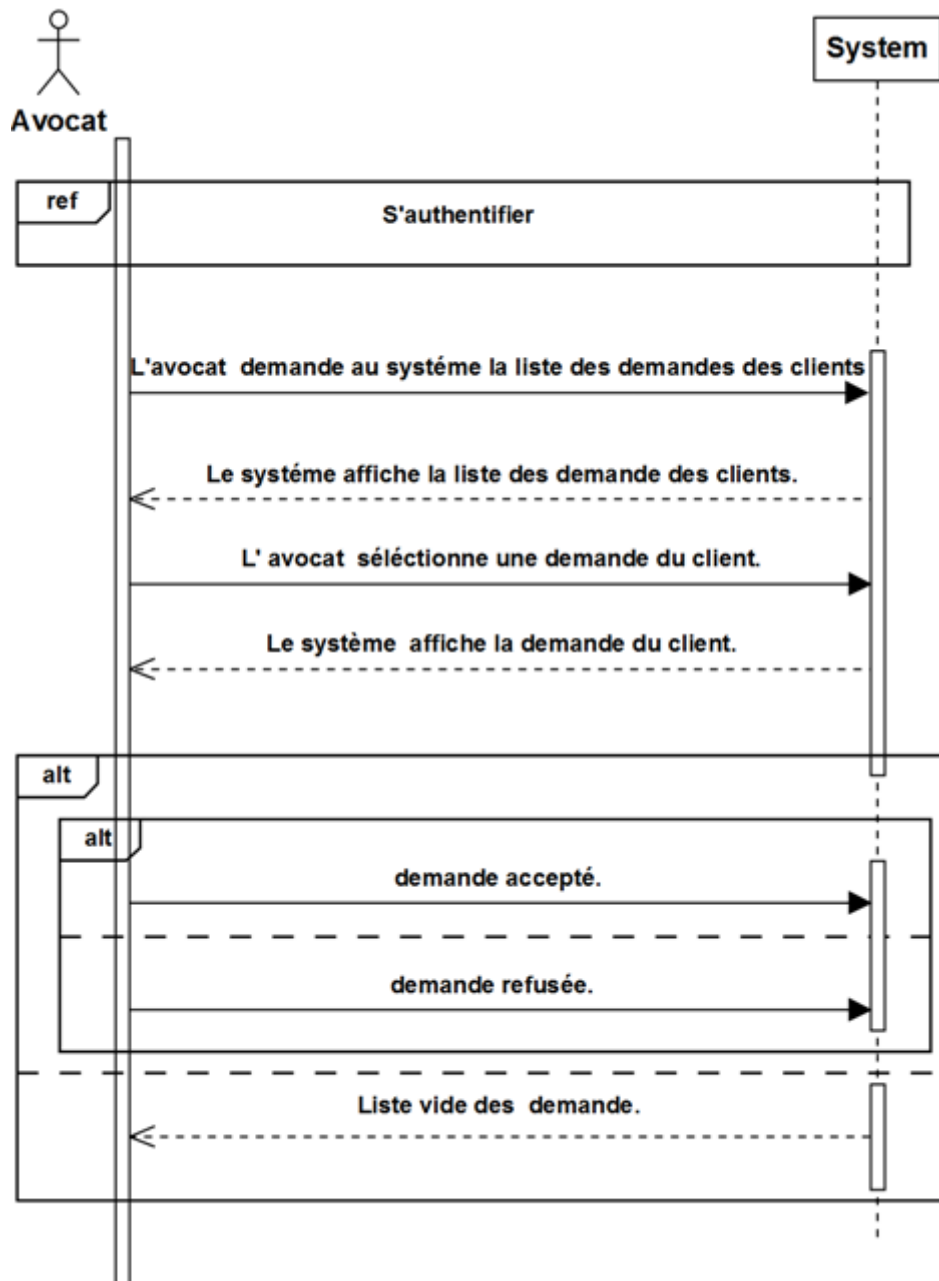


FIGURE 4.5 – Diagramme de séquence du cas confirmer demande

Consulter procuration :

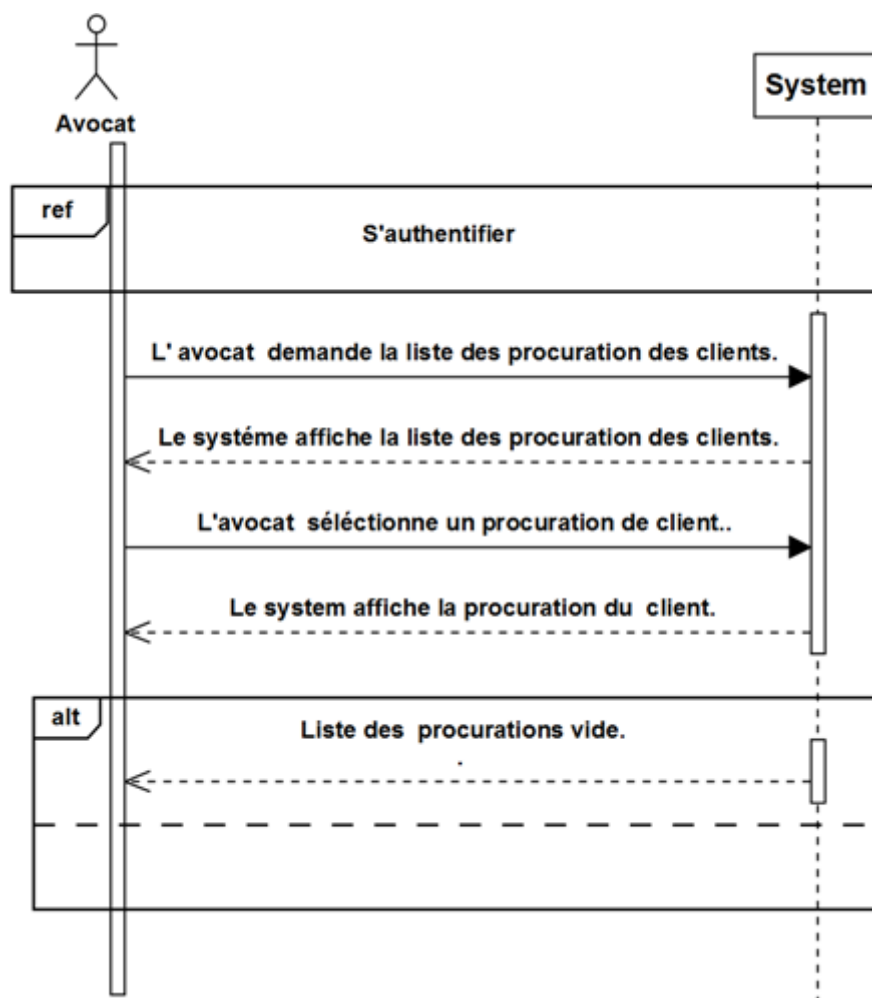


FIGURE 4.6 – Diagramme de séquence du cas consulter procuration

Gérer Client :

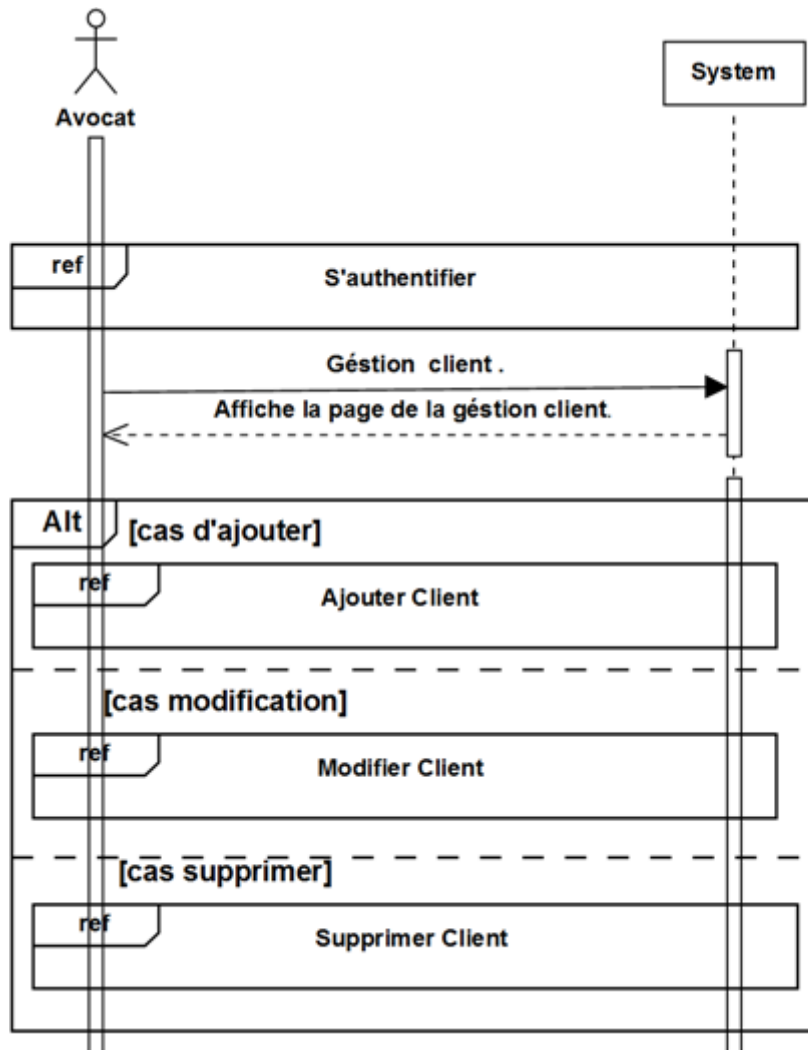


FIGURE 4.7 – Diagramme de séquence du cas gérer client

Ajouter Client :

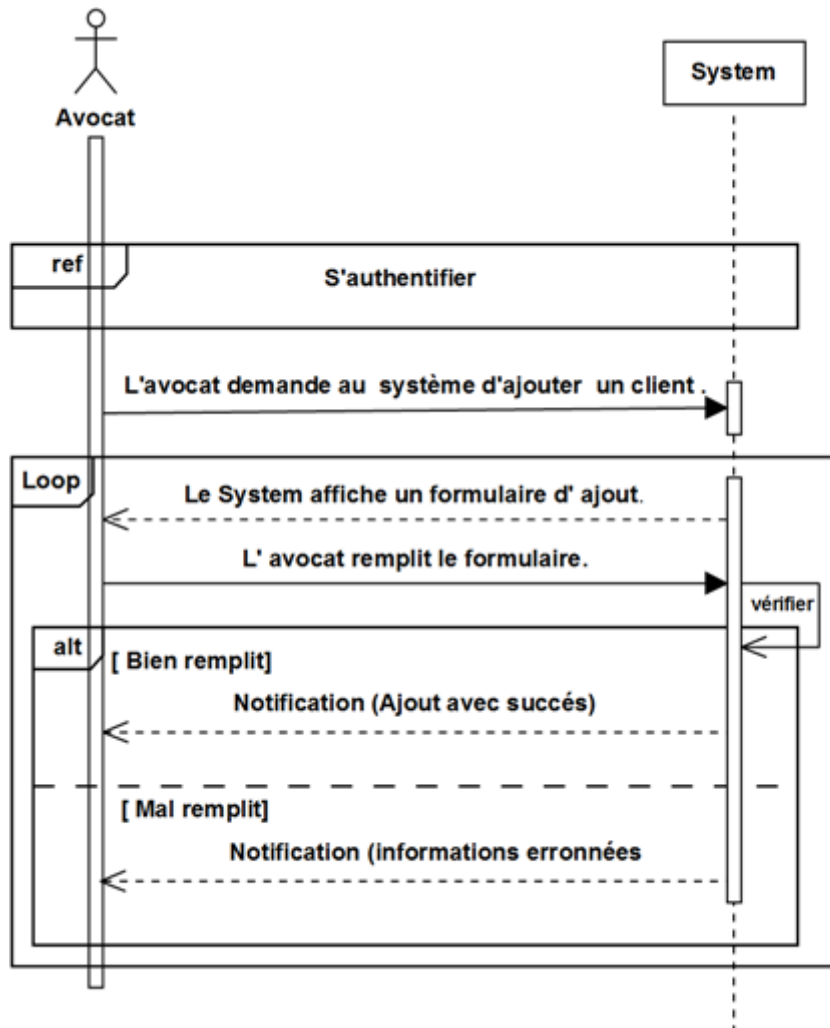


FIGURE 4.8 – Diagramme de séquence du cas ajouter client

Modifier Client :

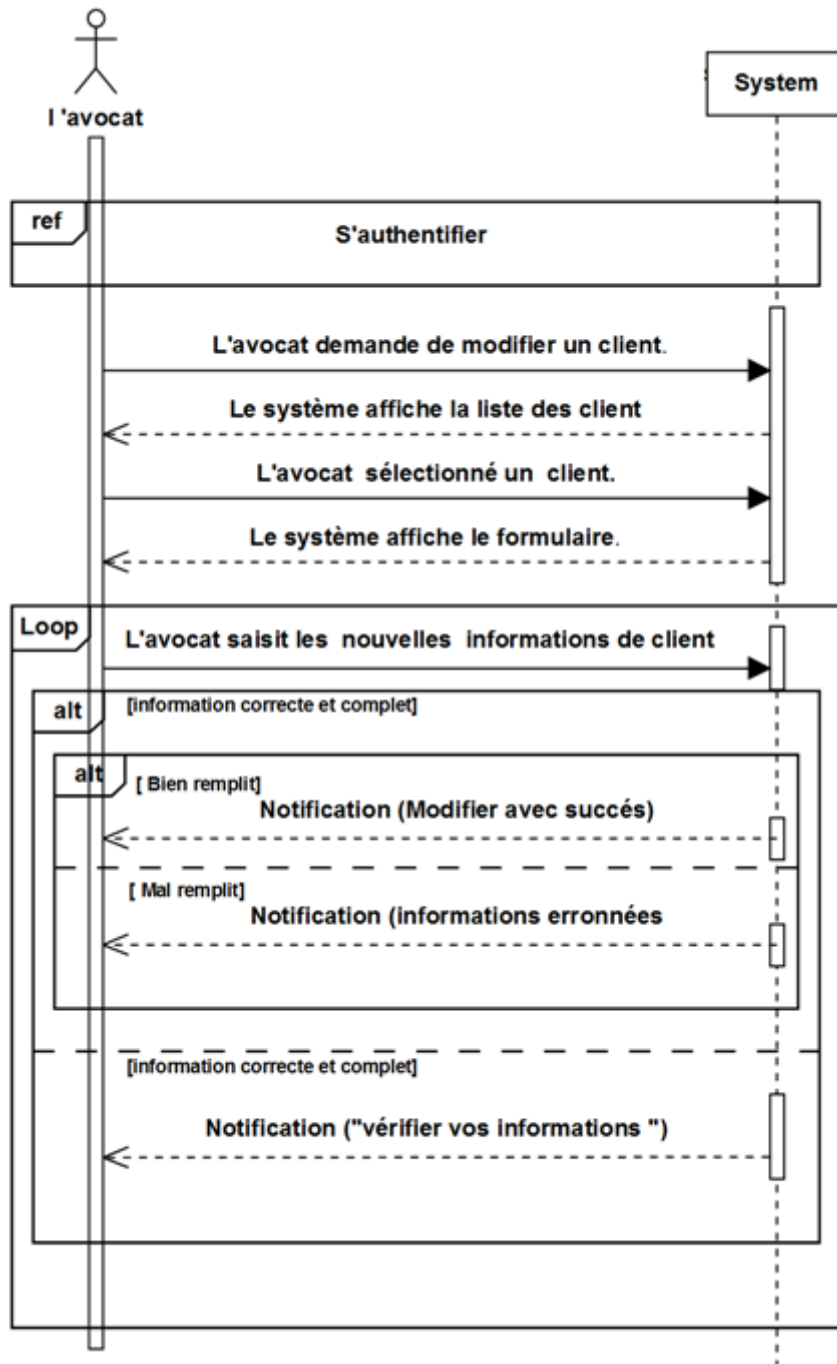


FIGURE 4.9 – Diagramme de séquence du cas Modifier Client

Supprimer Client :

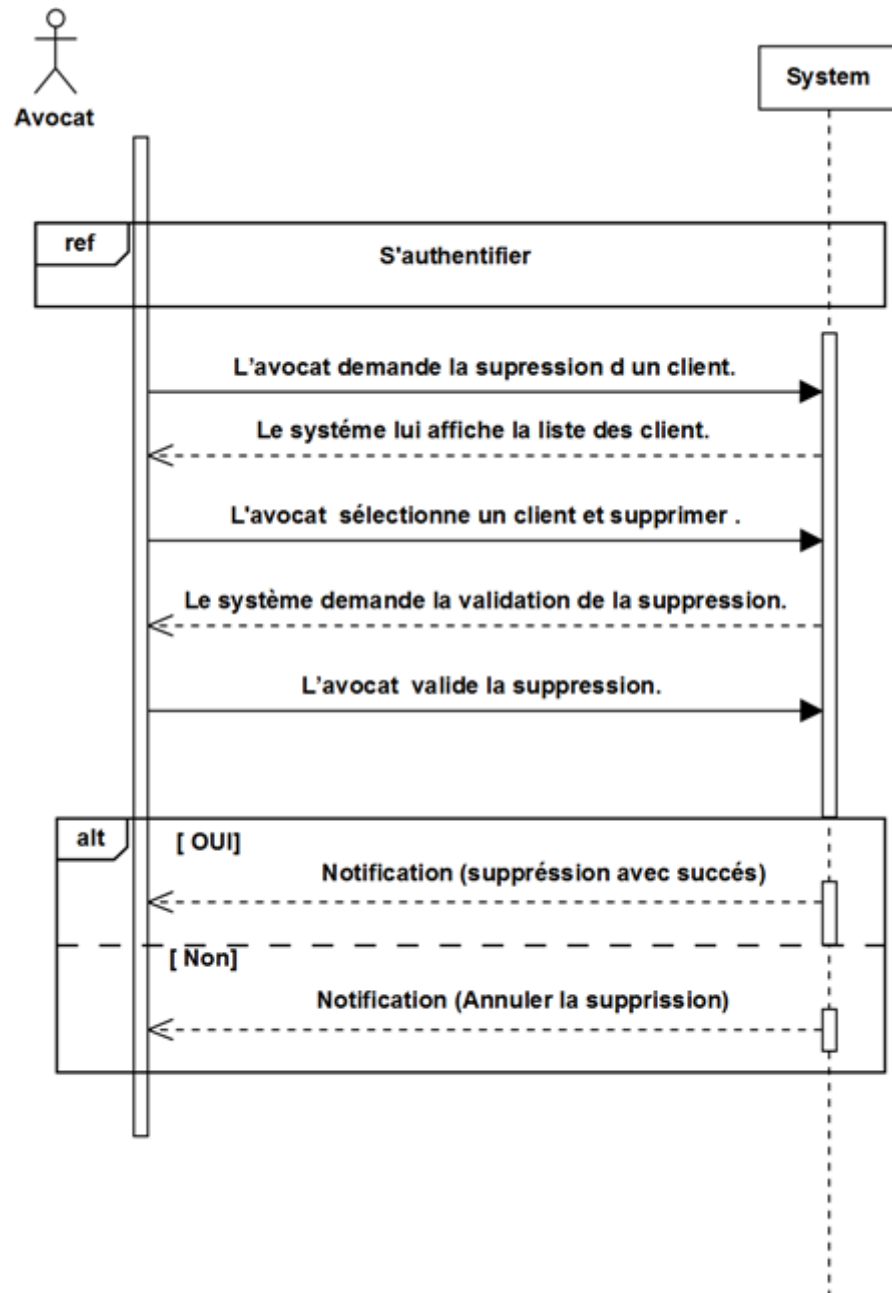


FIGURE 4.10 – Diagramme de séquence du cas supprimer client

Gérer Séance :

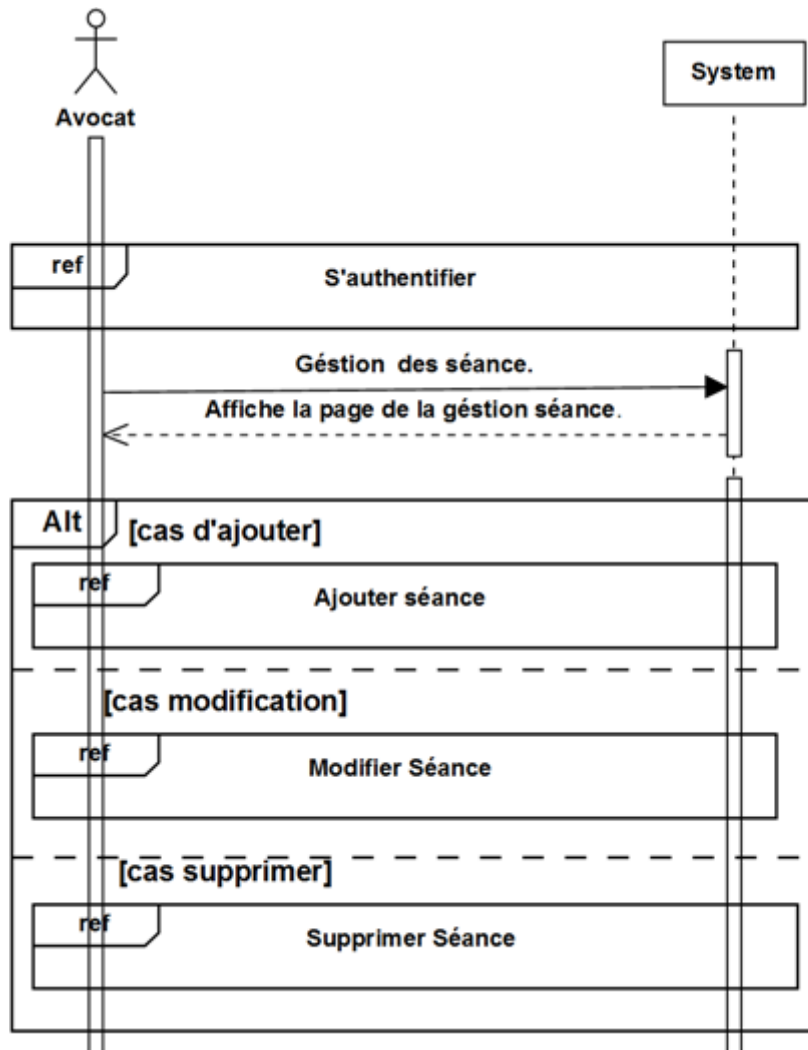


FIGURE 4.11 – Diagramme de séquence du cas gérer séance

Ajouter séance tribunal :

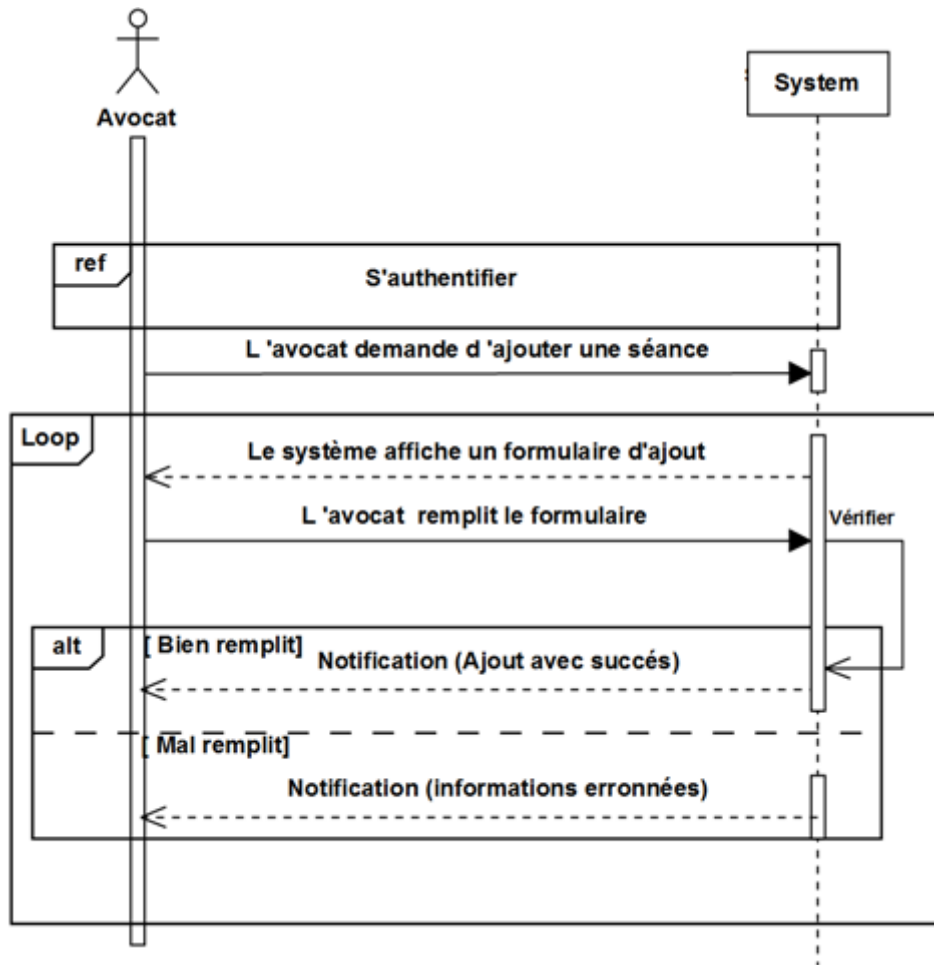


FIGURE 4.12 – Diagramme de séquence du cas ajouter séance tribunal

Modifier séance tribunal :

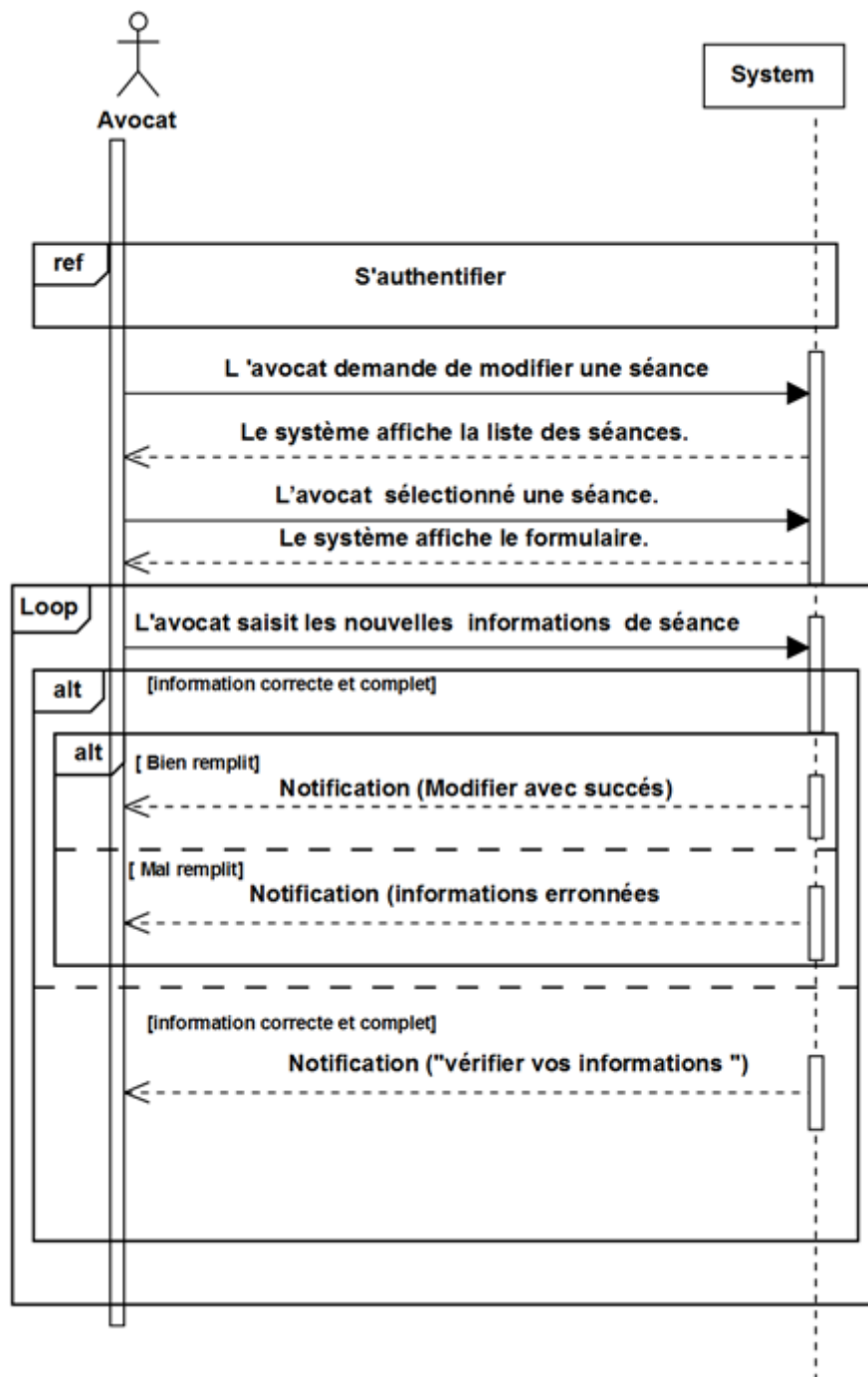


FIGURE 4.13 – Diagramme de séquence du cas modifier séance tribunal

Supprimer séance tribunal :

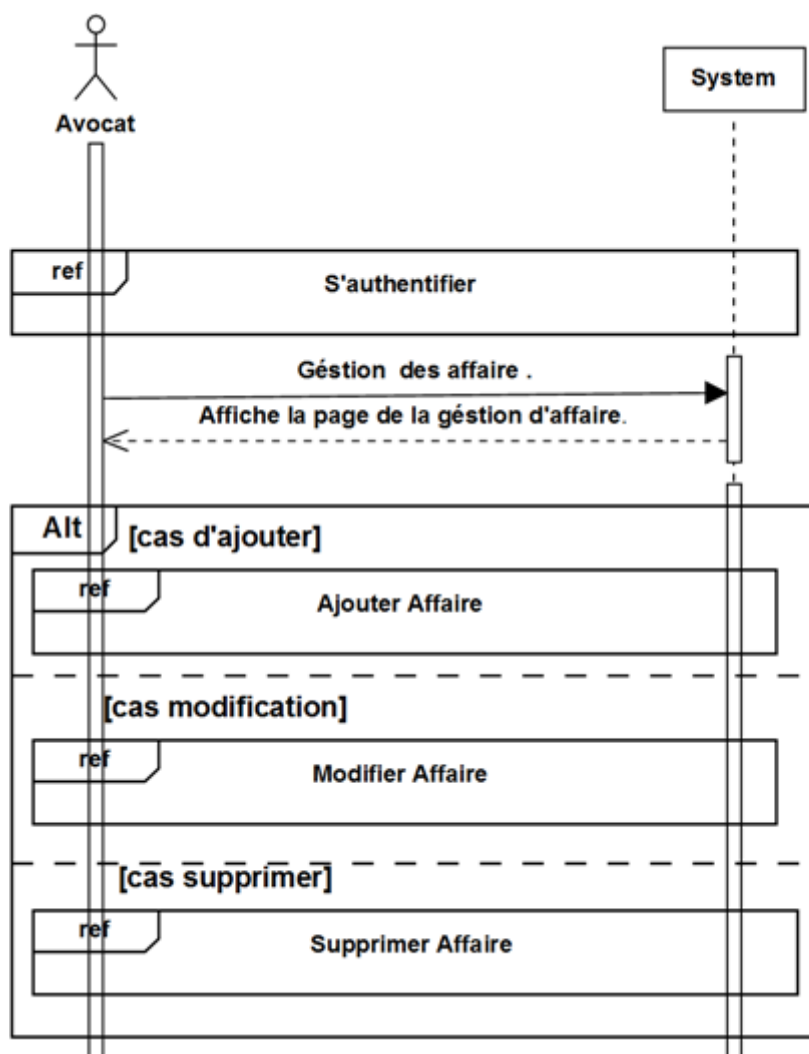


FIGURE 4.14 – Diagramme de séquence du cas supprimer séance tribunal

Gérer affaire :

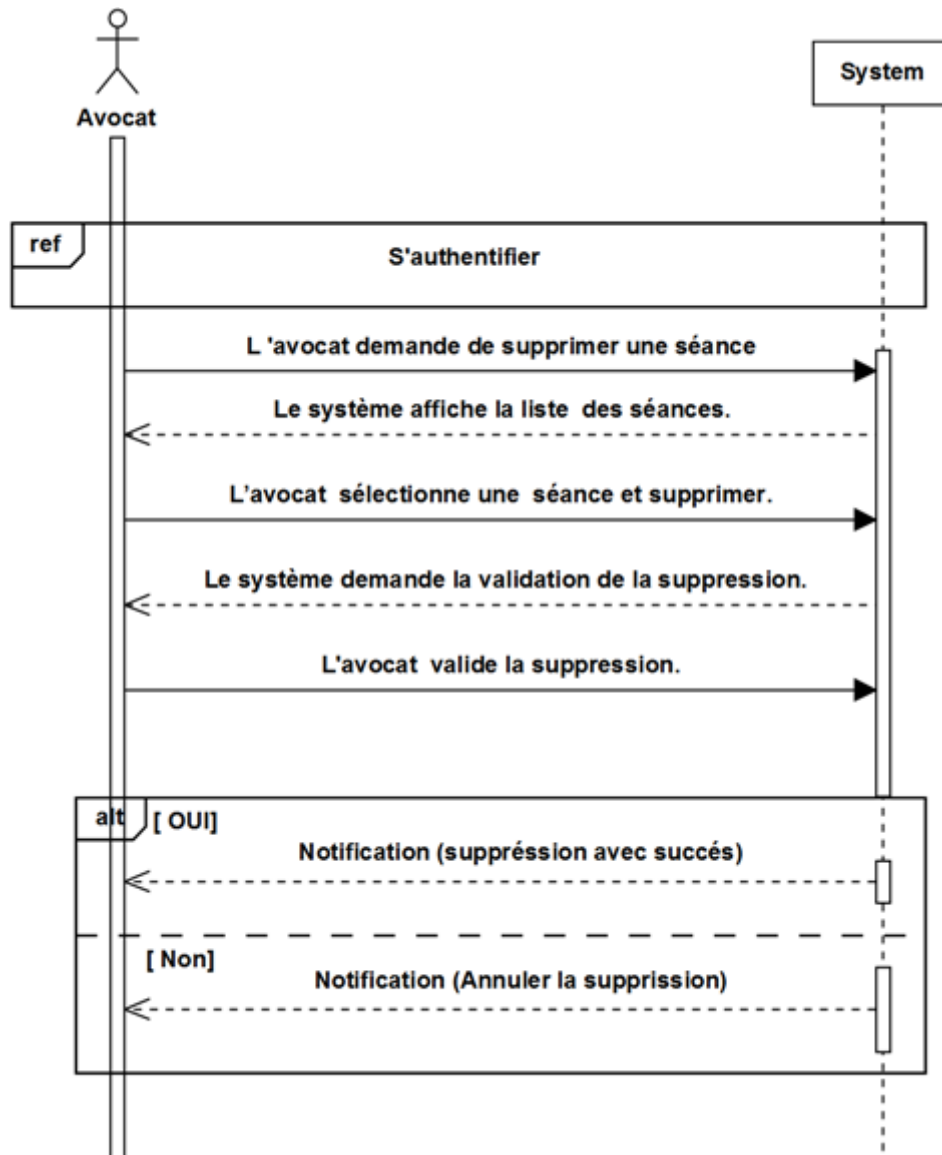


FIGURE 4.15 – Diagramme de séquence du cas gérer affaire

Ajouter Affaire :

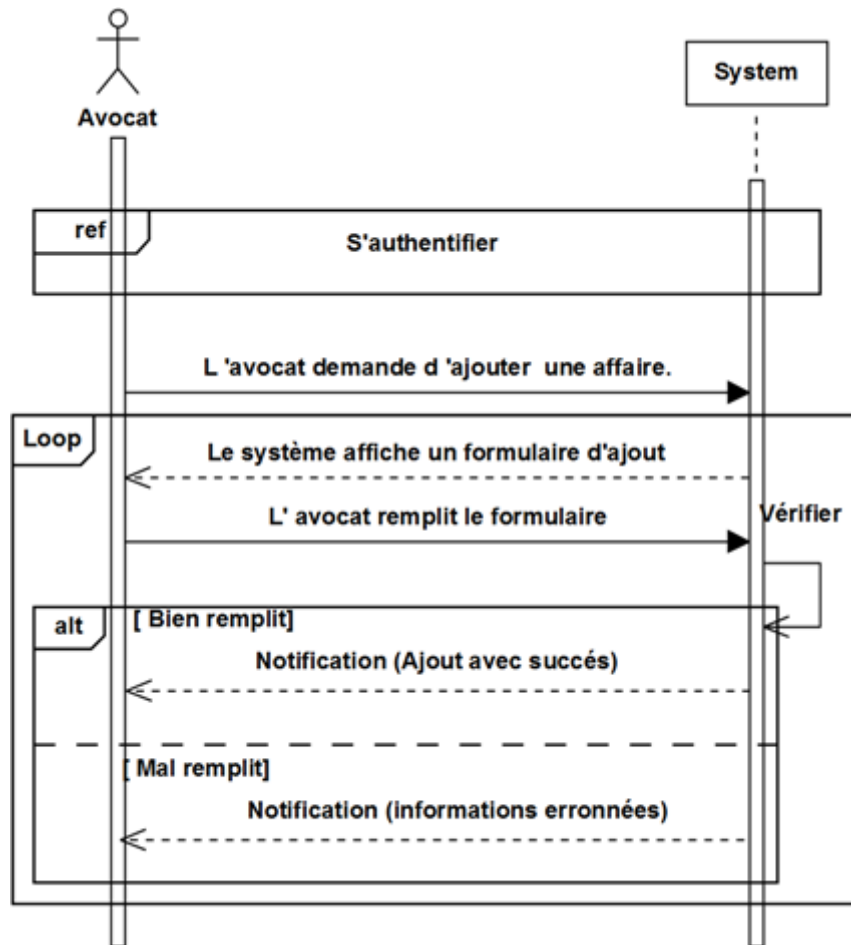


FIGURE 4.16 – Diagramme de séquence du cas ajouter affaire

Modifier Affaire :

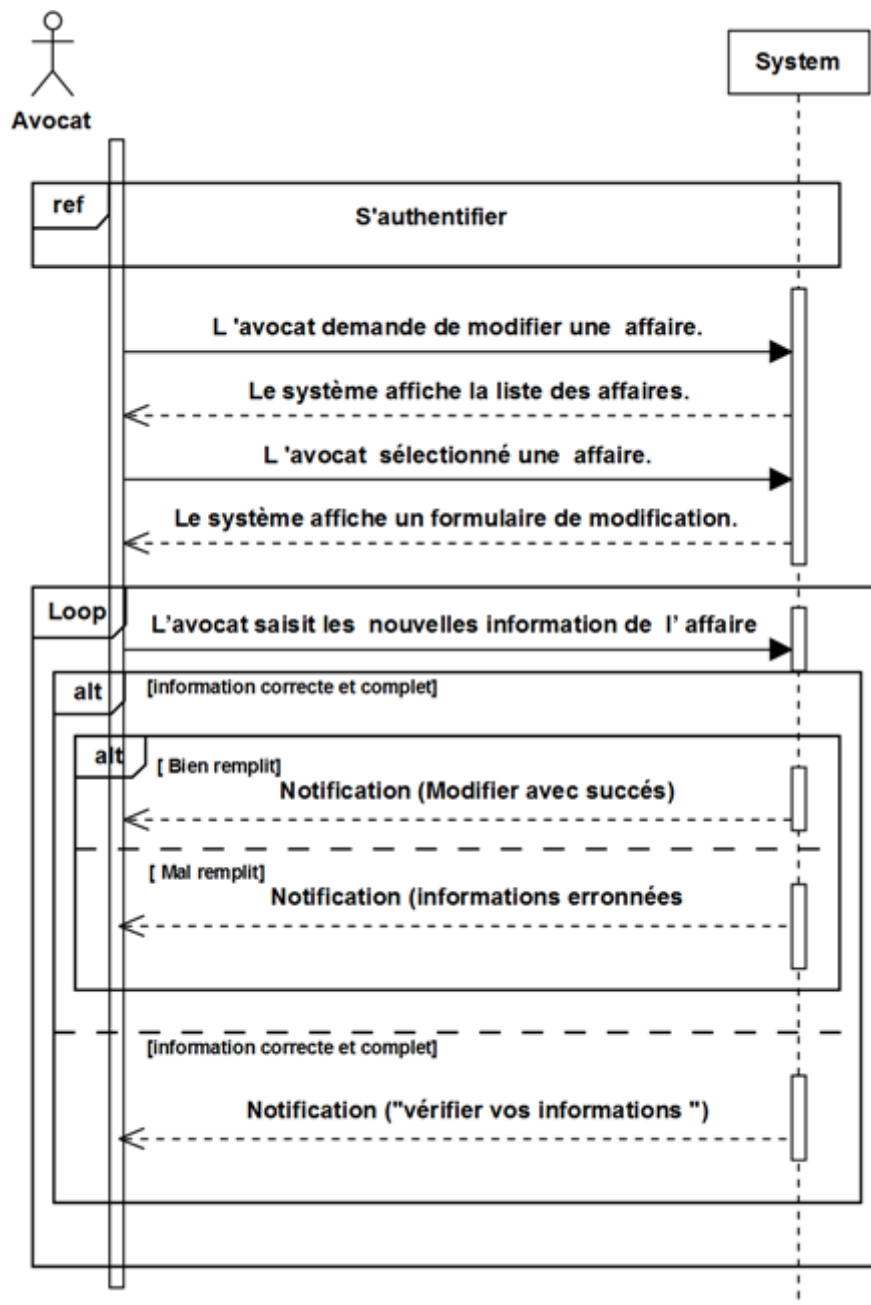


FIGURE 4.17 – Diagramme de séquence du cas modifier affaire

Supprimer Affaire :

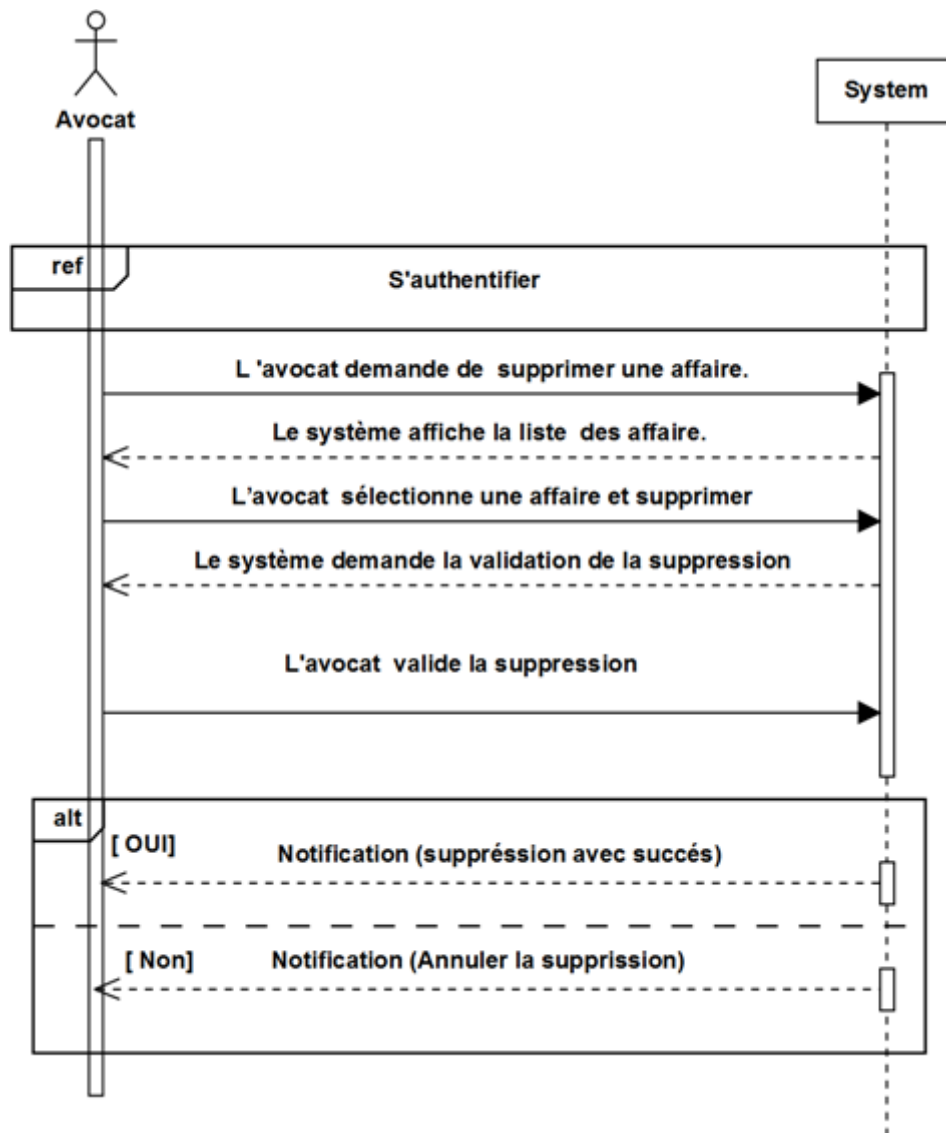


FIGURE 4.18 – Diagramme de séquence du cas supprimer affaire

Gérer Facture :

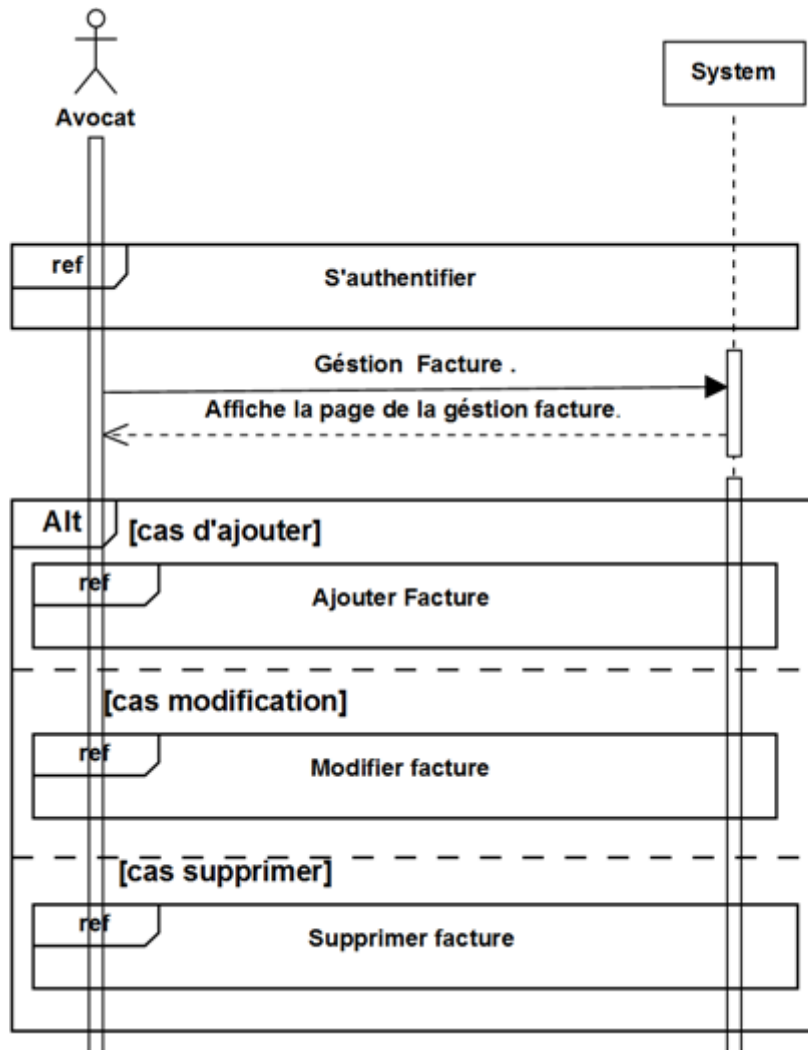


FIGURE 4.19 – Diagramme de séquence du cas gérer facture

Ajouter Facture :

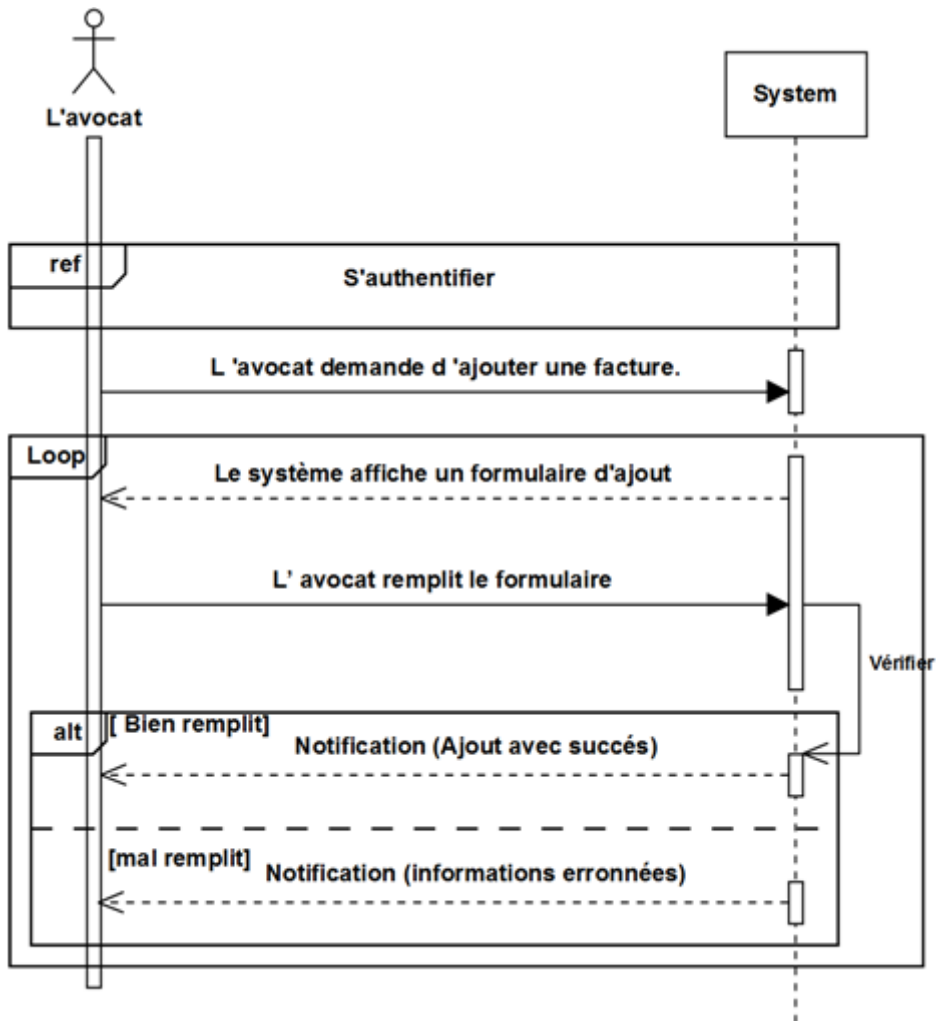


FIGURE 4.20 – Diagramme de séquence du cas ajouter facture

Modifier Facture :

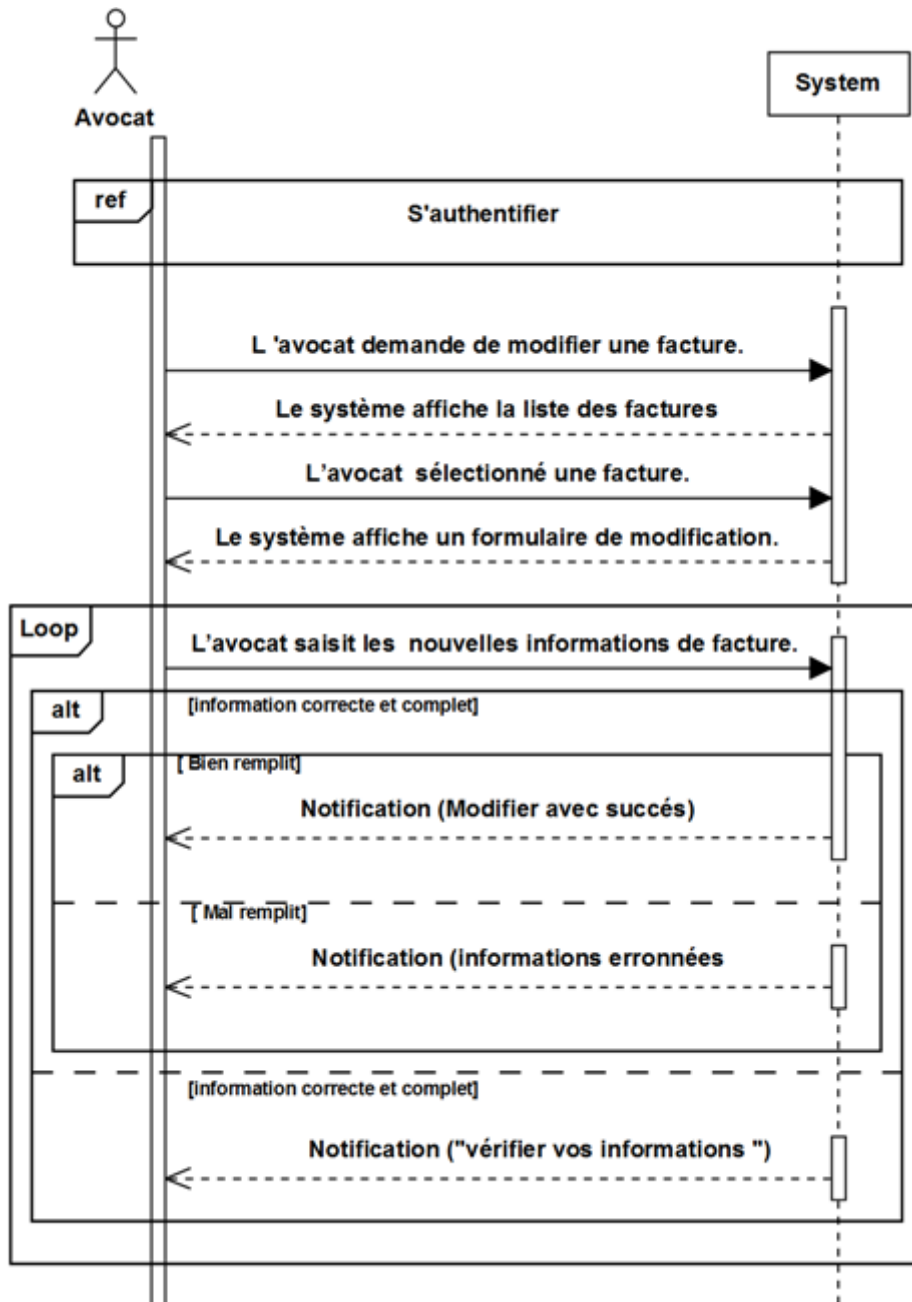


FIGURE 4.21 – Diagramme de séquence du cas modifier facture

Supprimer Facture :

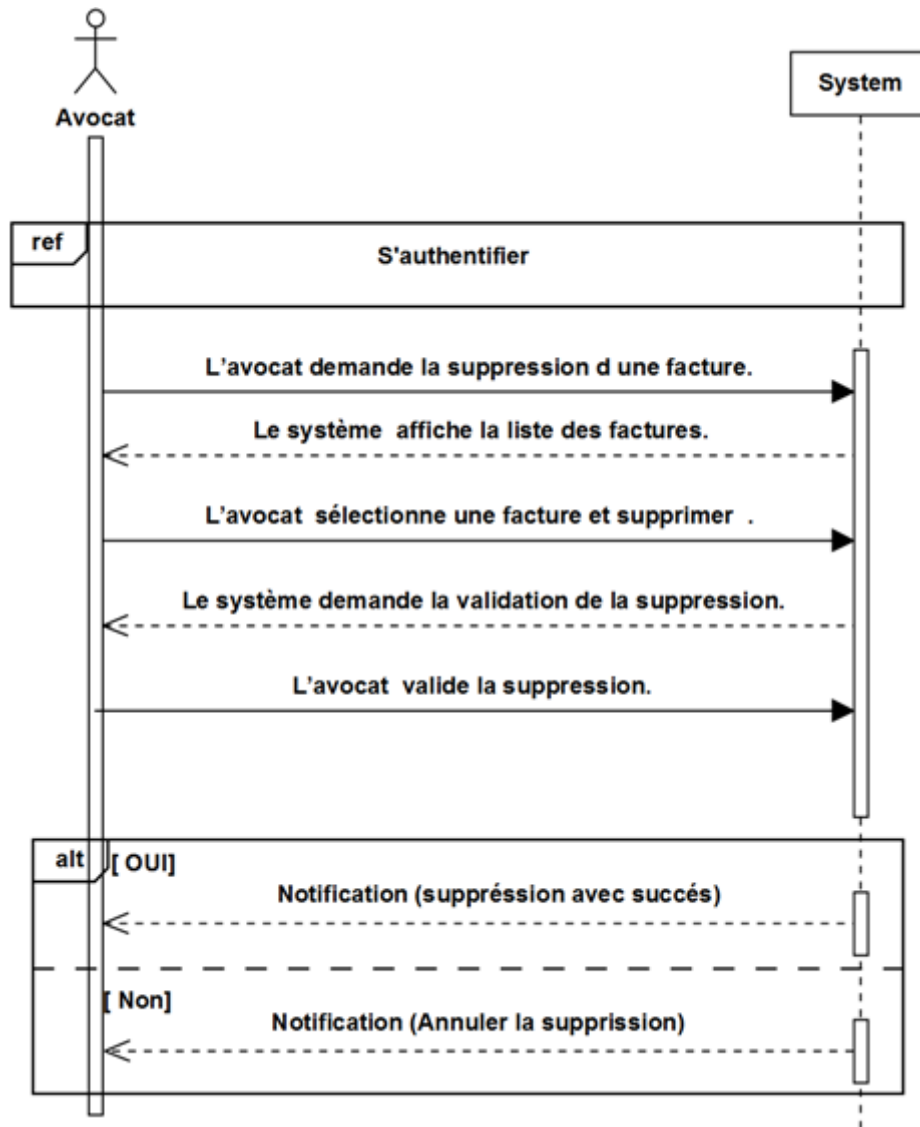


FIGURE 4.22 – Diagramme de séquence du cas supprimer facture

Effectuer Statistiques :

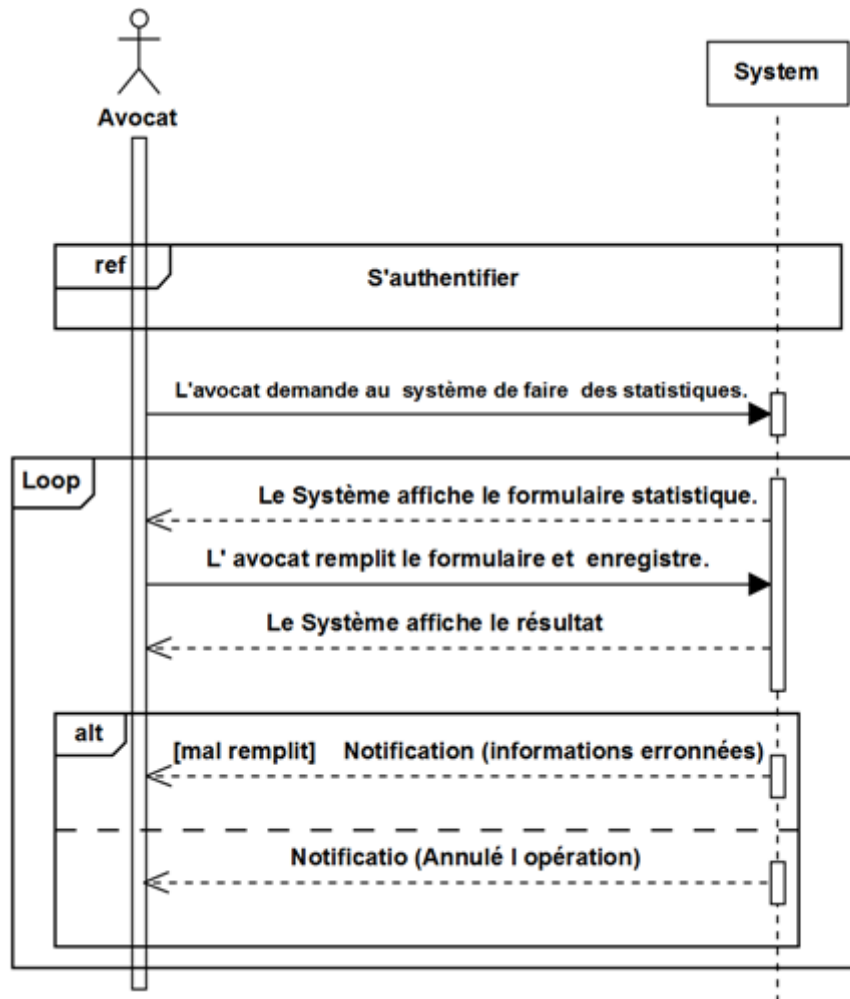


FIGURE 4.23 – Diagramme de séquence du cas Effectuer Statistiques

Demander Consultation :

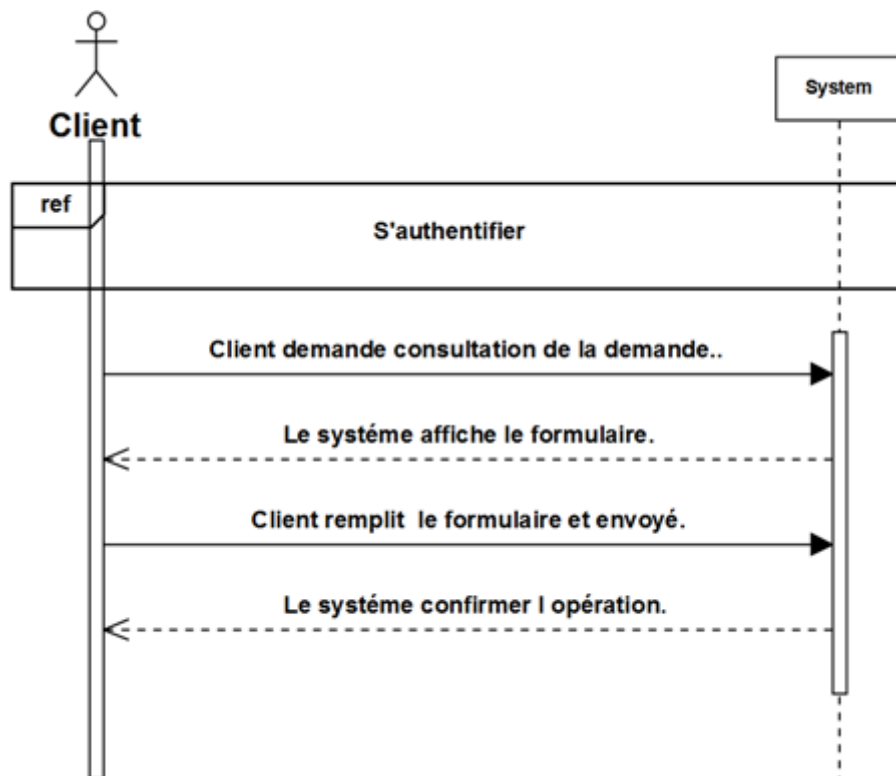


FIGURE 4.24 – Diagramme de séquence du cas demander consultation

Etablir Procuration :

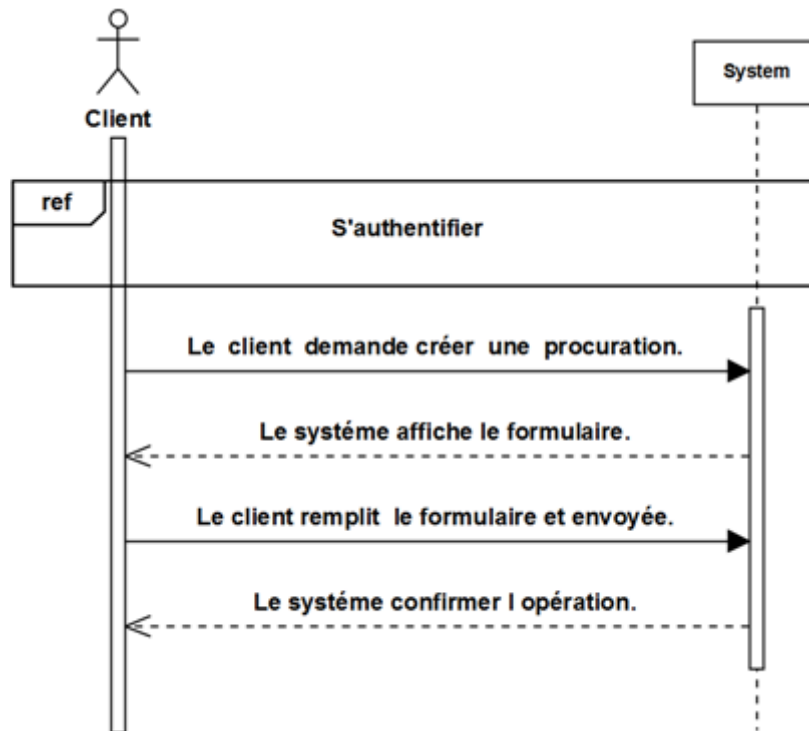


FIGURE 4.25 – Diagramme de séquence du cas Établir Procuration

Consulter séance tribunal :

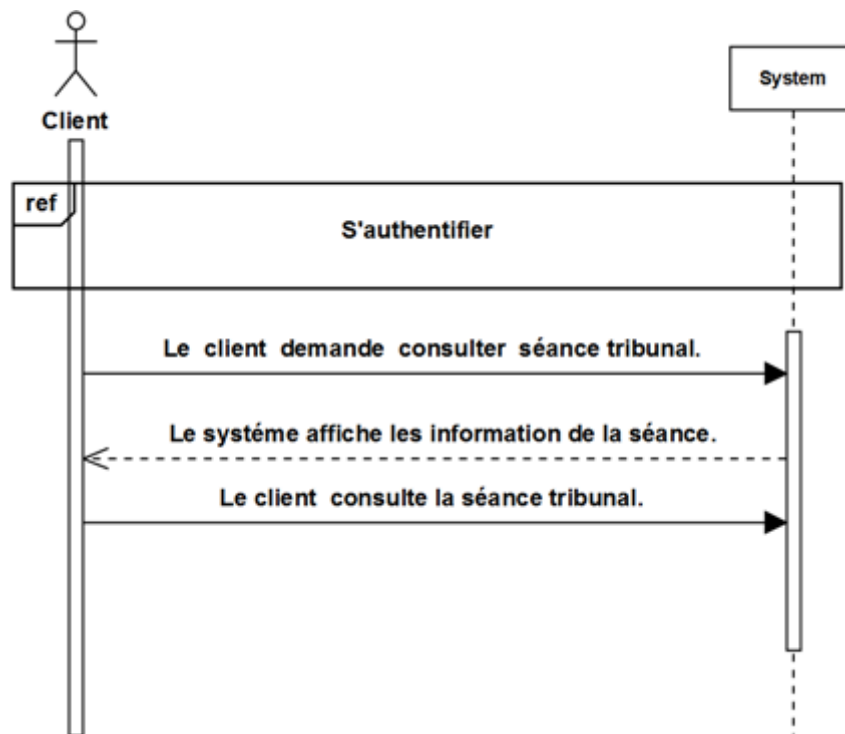


FIGURE 4.26 – Diagramme de séquence du cas consulter séance tribunal

Consulter Affaire :

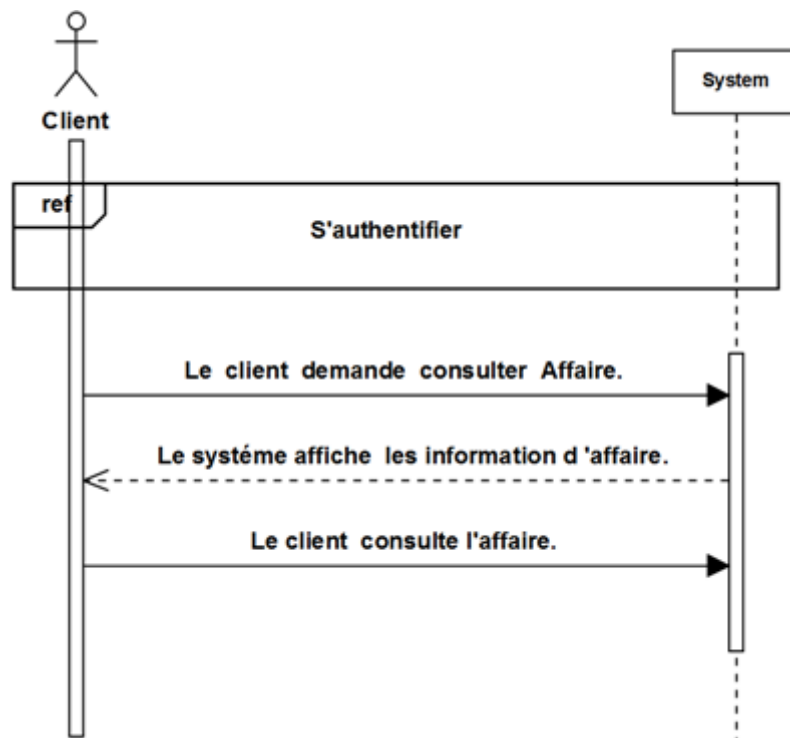


FIGURE 4.27 – Diagramme de séquence du cas consulter affaire

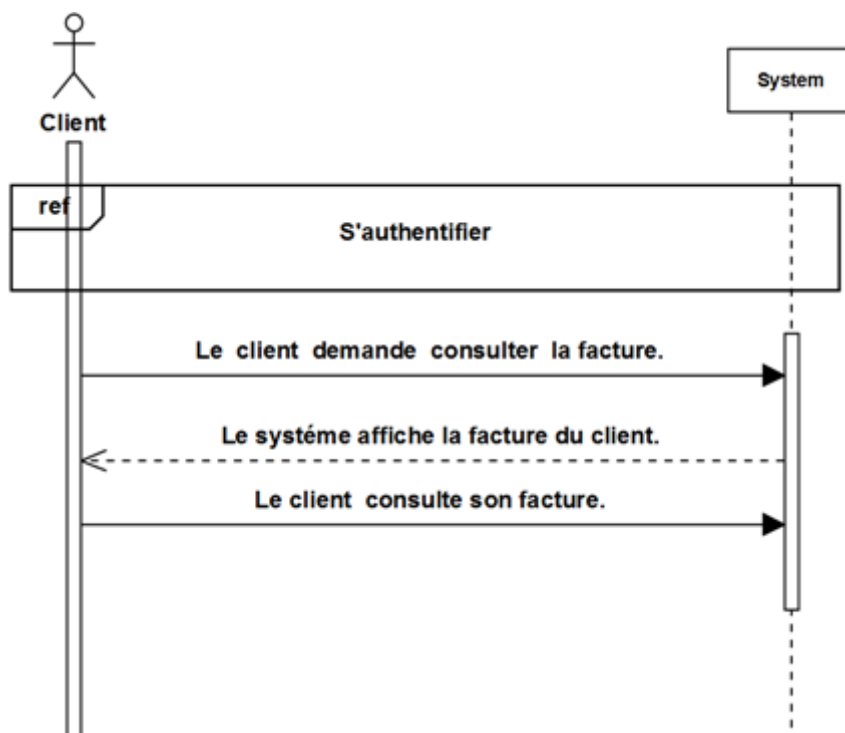
Consulter Facture :

FIGURE 4.28 – Diagramme de séquence du cas consulter facture

4.3.3 Identification des classes candidates :

Cette phase va préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d'analyse. La technique utilisée pour identifier les classes candidates est la suivante :

- Chercher les noms communs importants dans les descriptions textuelles des cas d'utilisation.
- Vérifier les propriétés « objet » de chaque concept (identité, propriétés, comportement), puis définir ses responsabilités. [9]

•Liste des classes candidates :

le tableau ci-dessous résume la liste des classes candidates de notre système :

La liste des classes candidates
- Demande.
- Procuration .
- Client .
- Séance.
-Affaire .
-Facture .

TABLE 4.27: liste des classes candidates.

4.3.4 Diagramme de classe participantes :

Dans cette section tous les diagrammes de classe participante de notre système sont représentés par les figures qui suit :

• Cas d'utilisation « s'inscrire » :

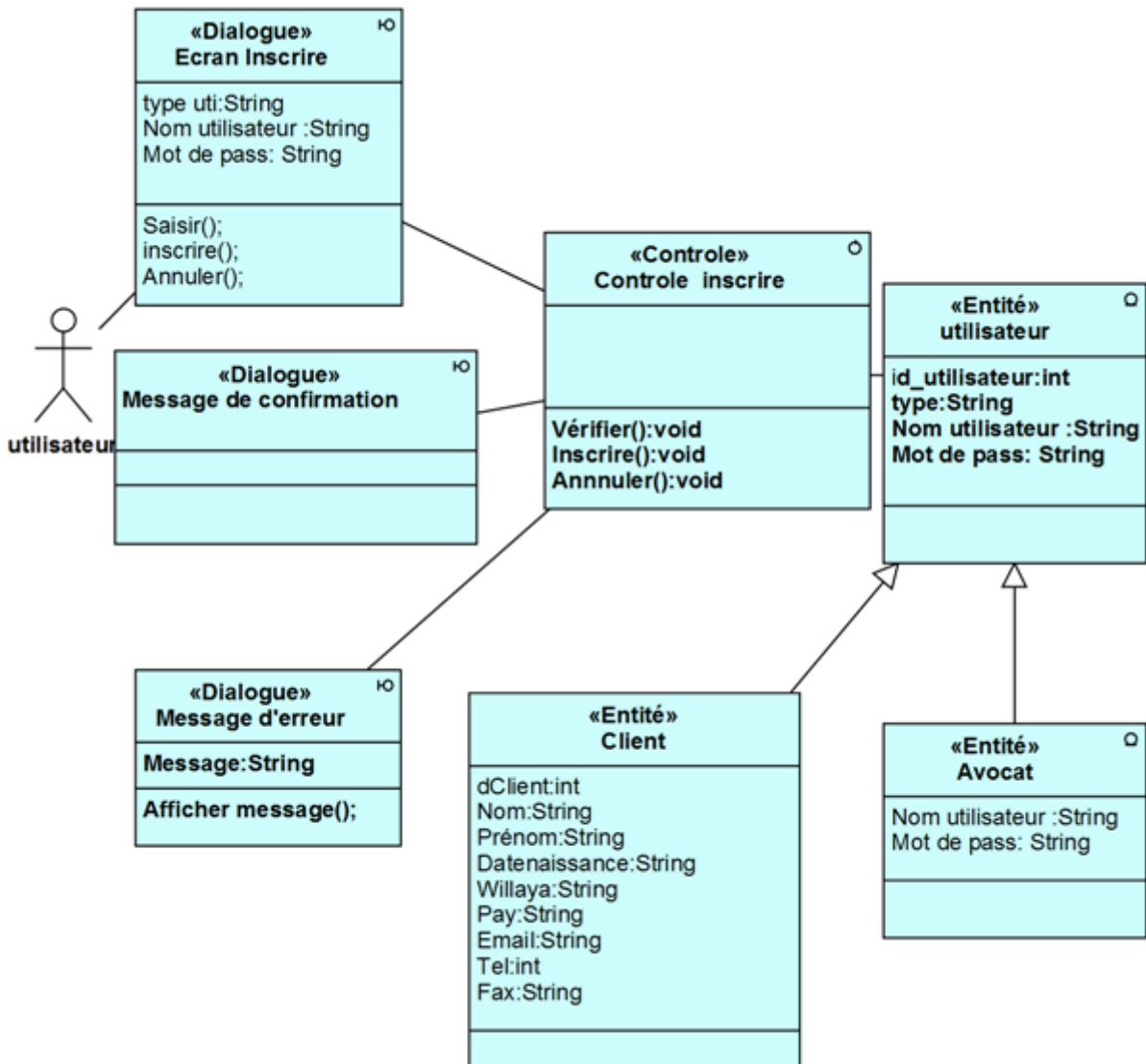


FIGURE 4.29 – Diagramme de classe participantes pour le cas d'utilisation «s'inscrire»

• Cas d'utilisation « s'authentifier » :

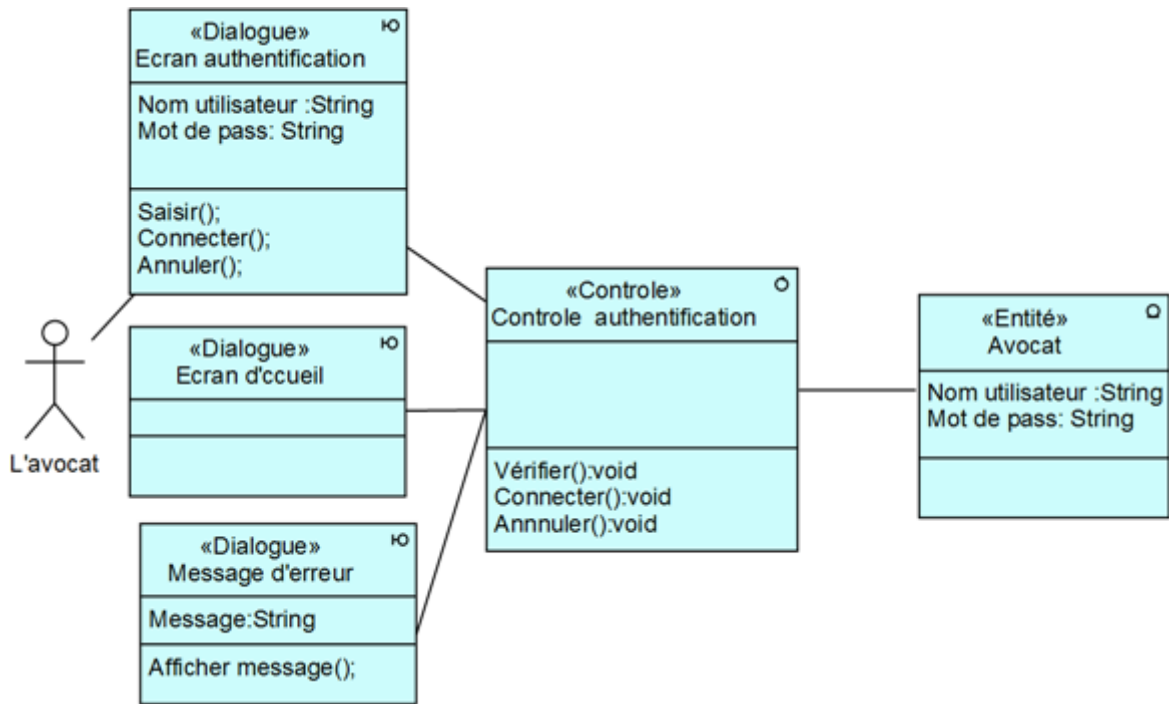


FIGURE 4.30 – Diagramme de classe participantes pour le cas d'utilisation «s'authentifier»

• Cas d'utilisation « Consulter demande » :

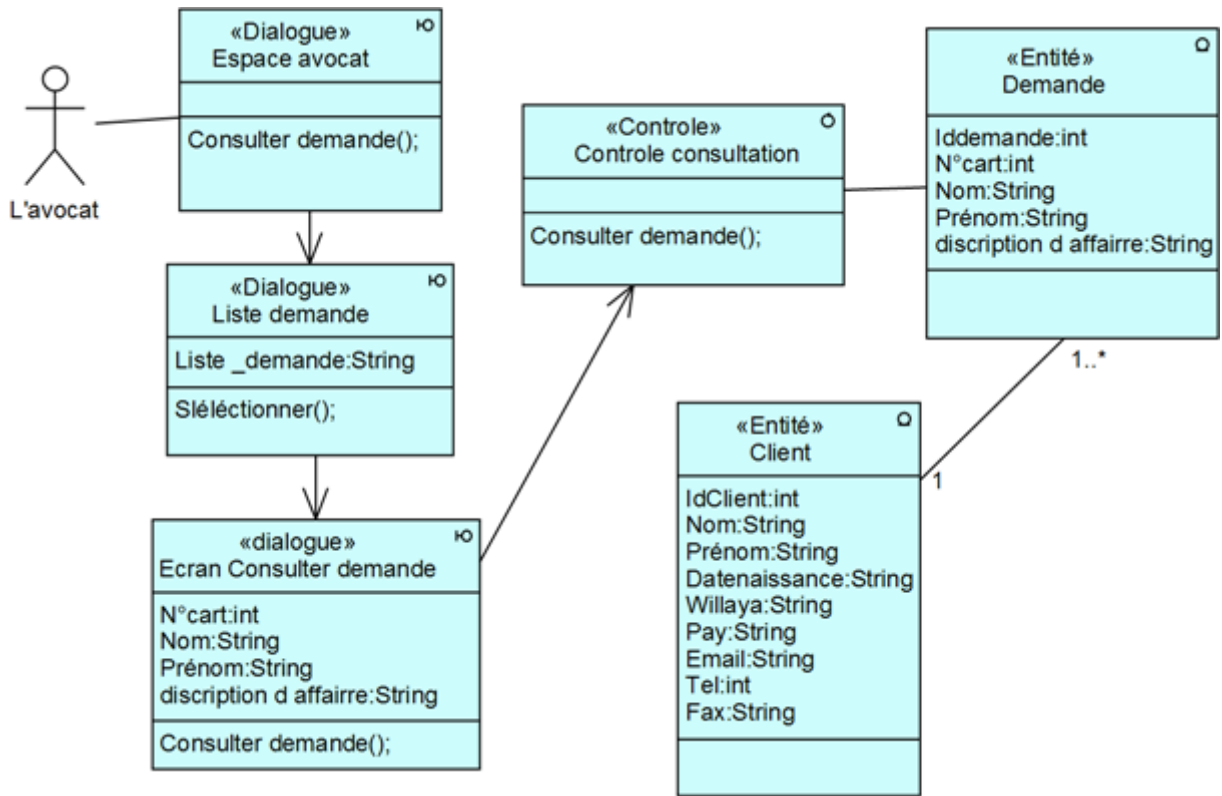


FIGURE 4.31 – Diagramme de classe participantes pour le cas d'utilisation « Consulter demande »

• Cas d'utilisation « Confirmer Demande » :

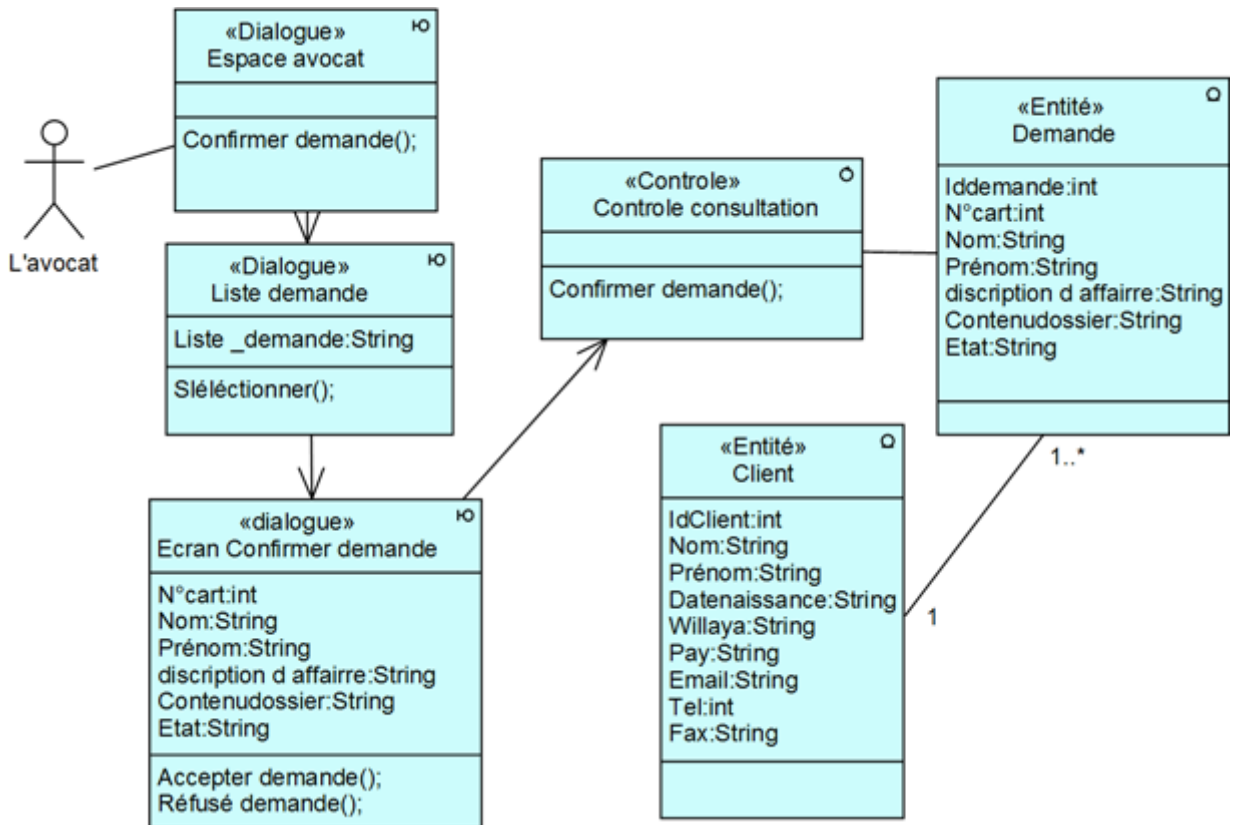


FIGURE 4.32 – Diagramme de classe participantes pour le cas d'utilisation «Confirmer Demande »

• Cas d'utilisation « Ajouter client » :

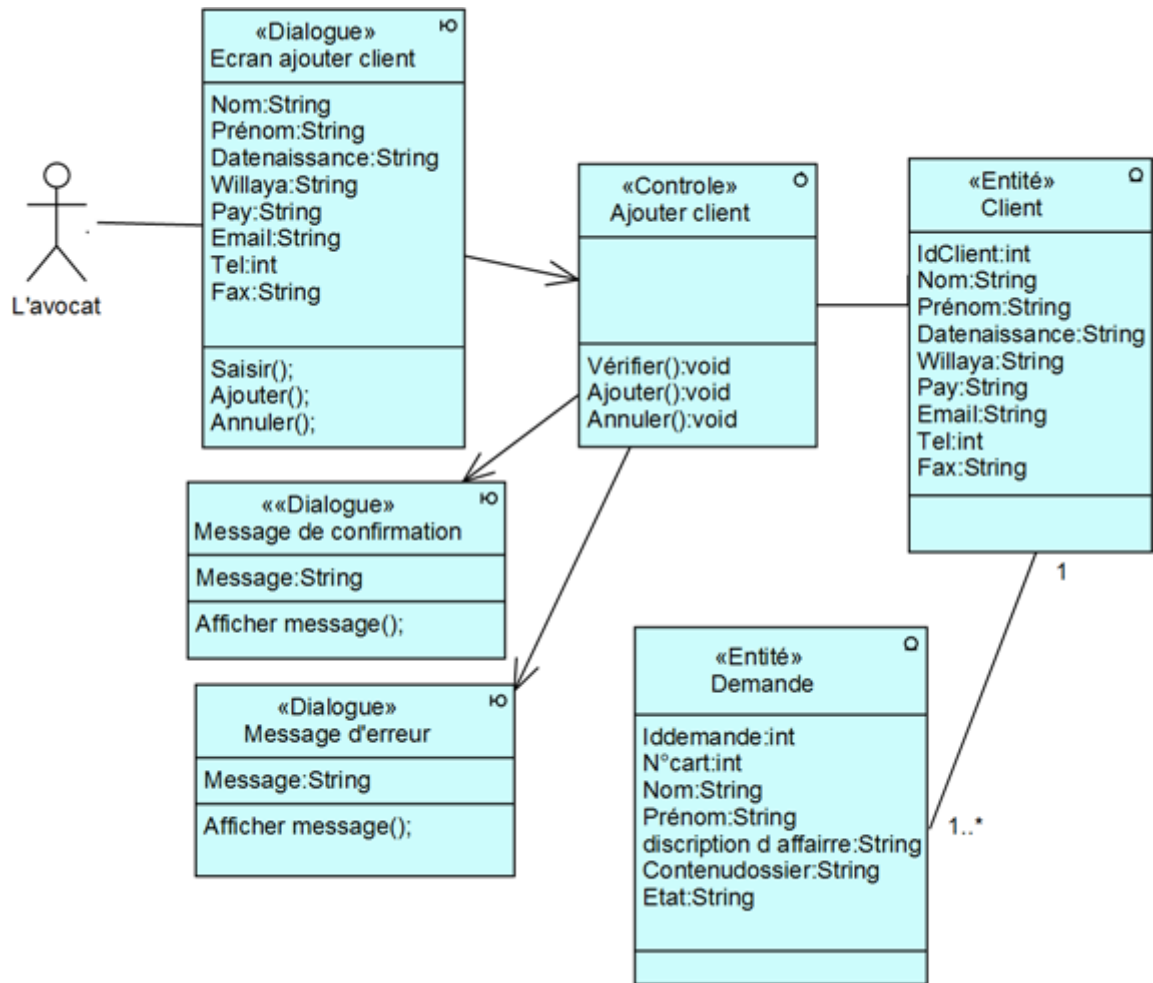


FIGURE 4.33 – Diagramme de classe participantes pour le cas d'utilisation «Ajouter client »

• Cas d'utilisation « Modifier Client » :

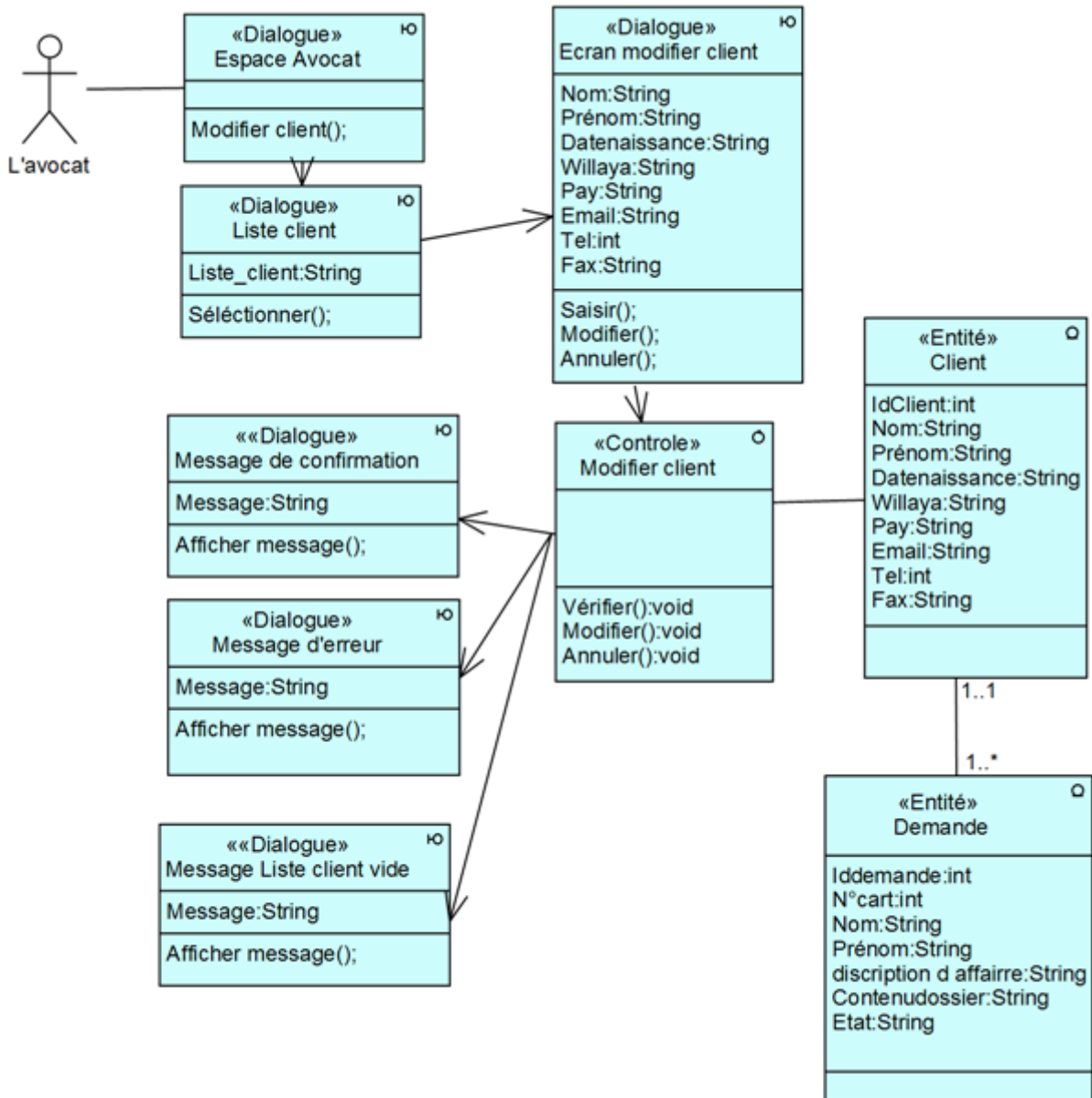


FIGURE 4.34 – Diagramme de classe participantes pour le cas d'utilisation «Modifier Client »

• Cas d'utilisation « Supprimer Client » :

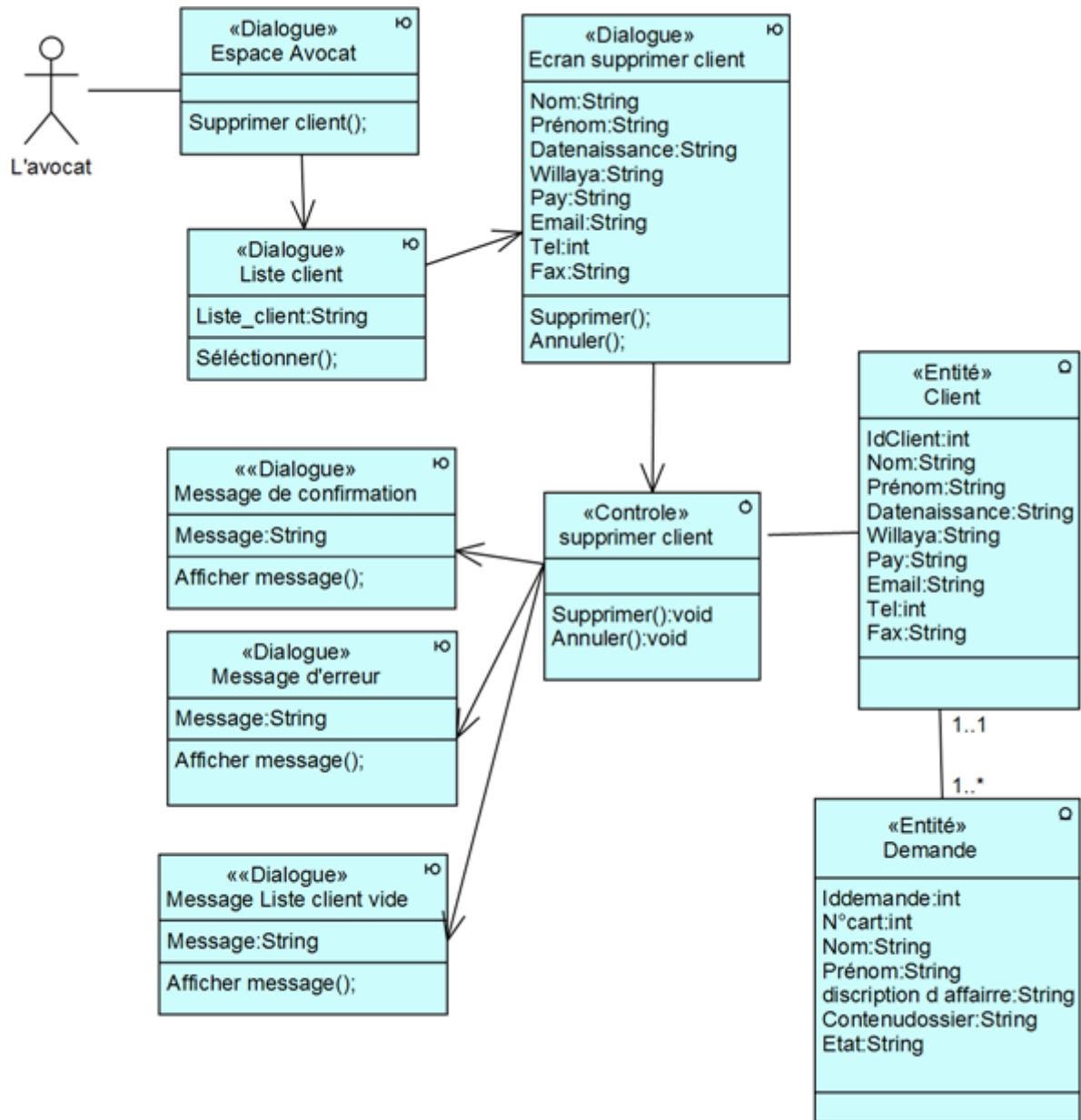


FIGURE 4.35 – Diagramme de classe participantes pour le cas d'utilisation «Supprimer Client»

• Cas d'utilisation « Ajouter Séance Tribunal » :

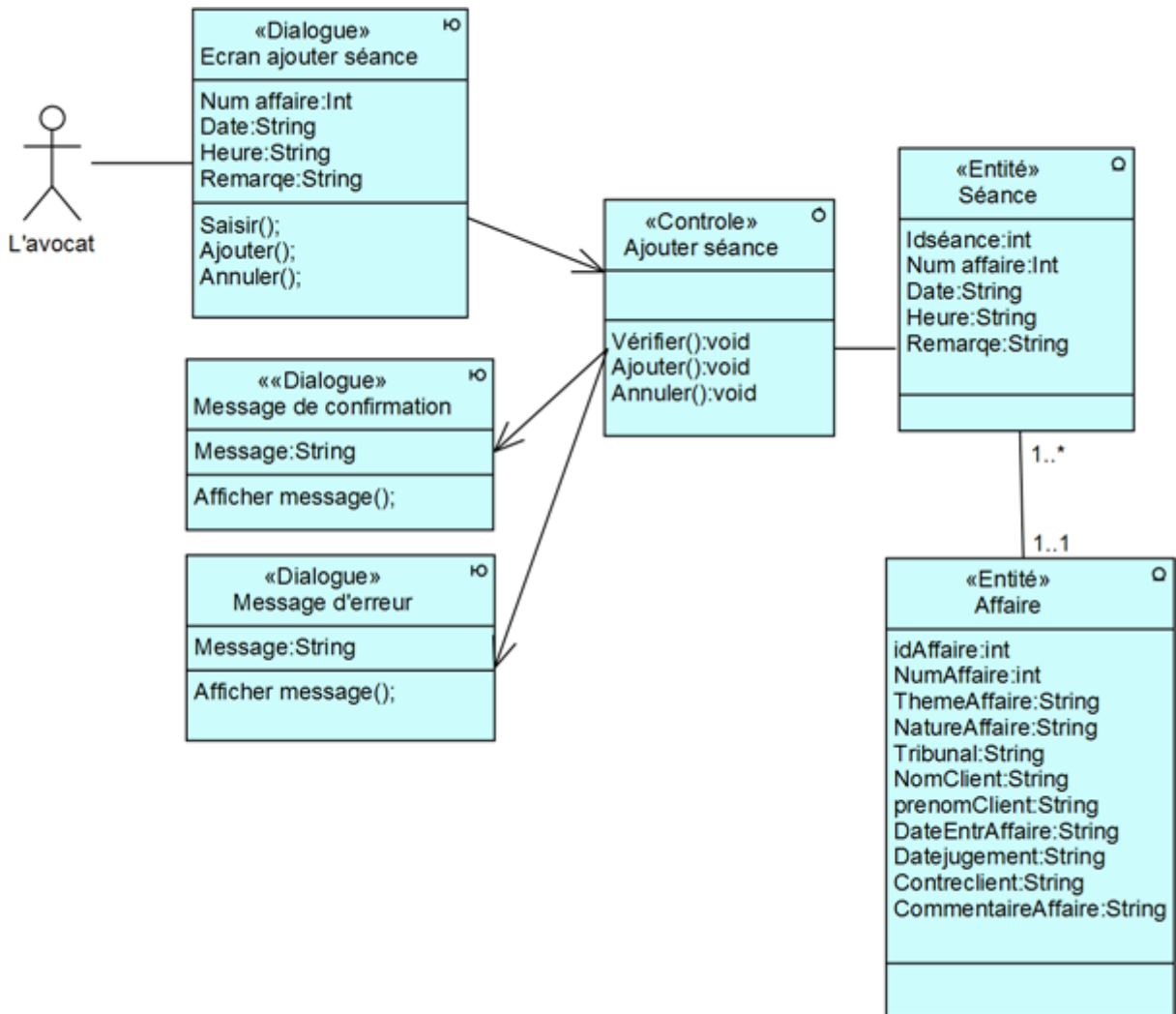


FIGURE 4.36 – Diagramme de classe participantes pour le cas d'utilisation «Ajouter Séance Tribunal»

• Cas d'utilisation « Modifier Séance Tribunal » :

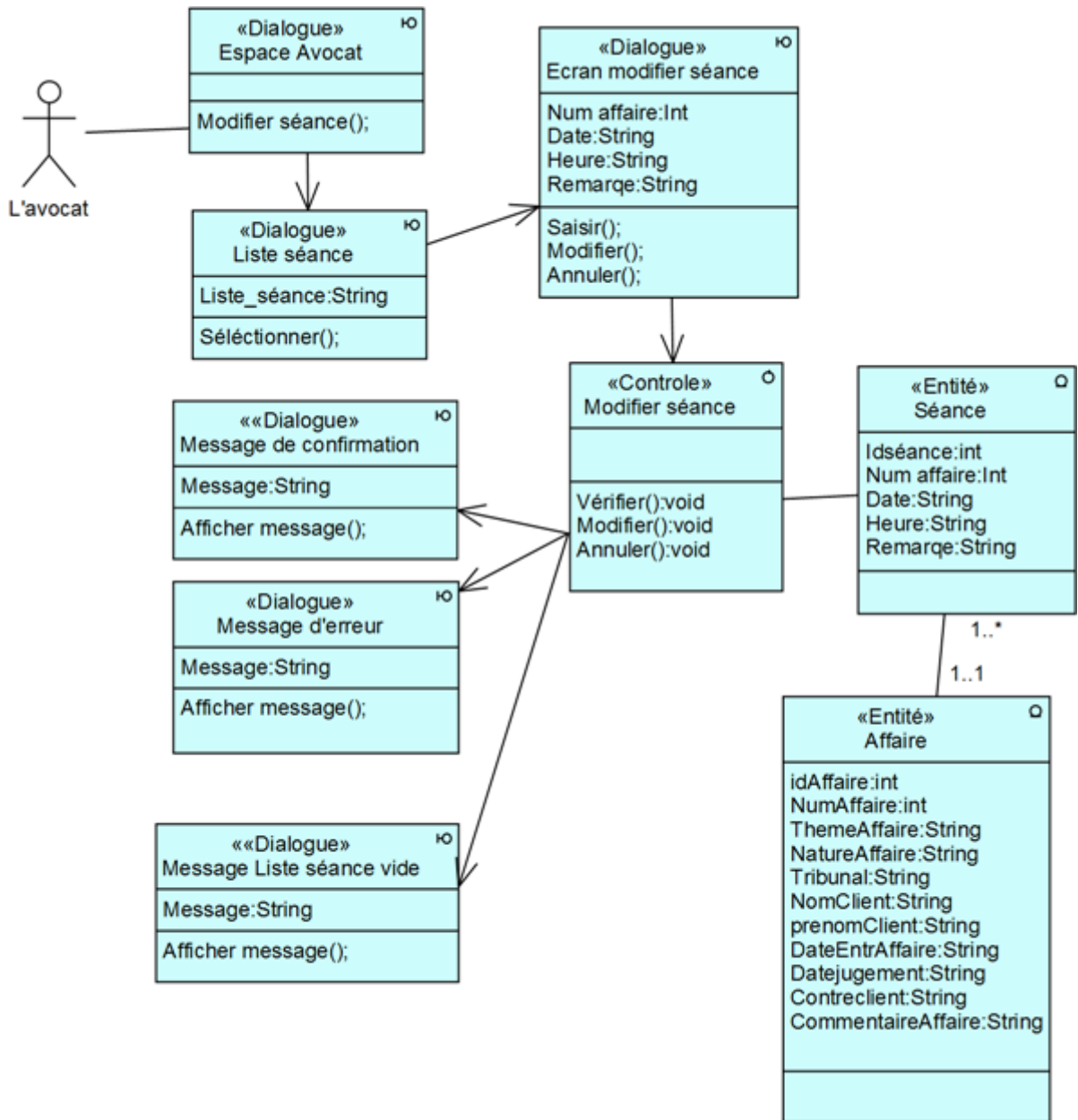


FIGURE 4.37 – Diagramme de classe participantes pour le cas d'utilisation «Modifier Séance Tribunal»

• Cas d'utilisation « Supprimer Séance Tribunal » :

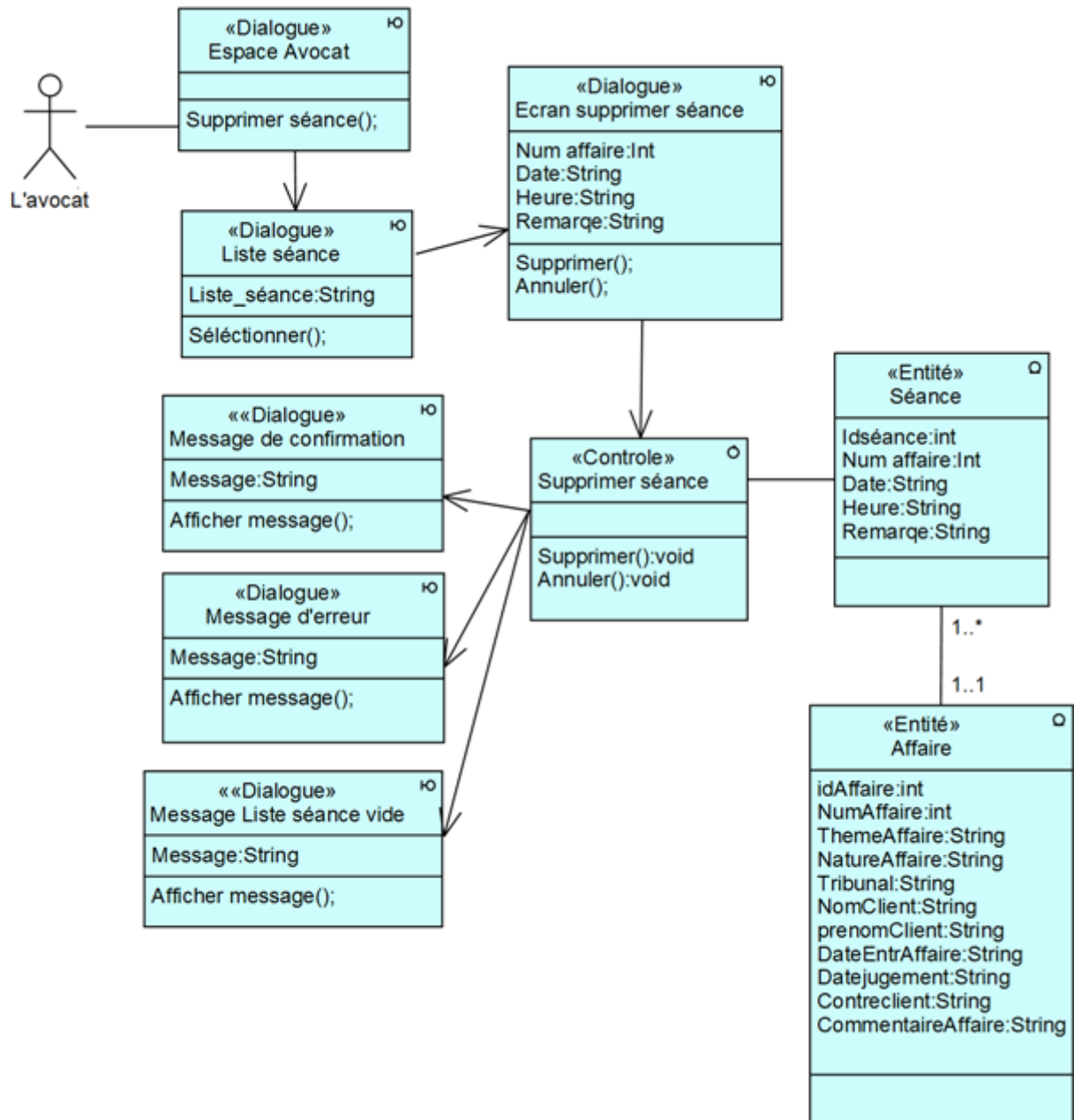


FIGURE 4.38 – Diagramme de classe participantes pour le cas d'utilisation «Supprimer Séance Tribunal»

• Cas d'utilisation « Ajouter Affaire » :

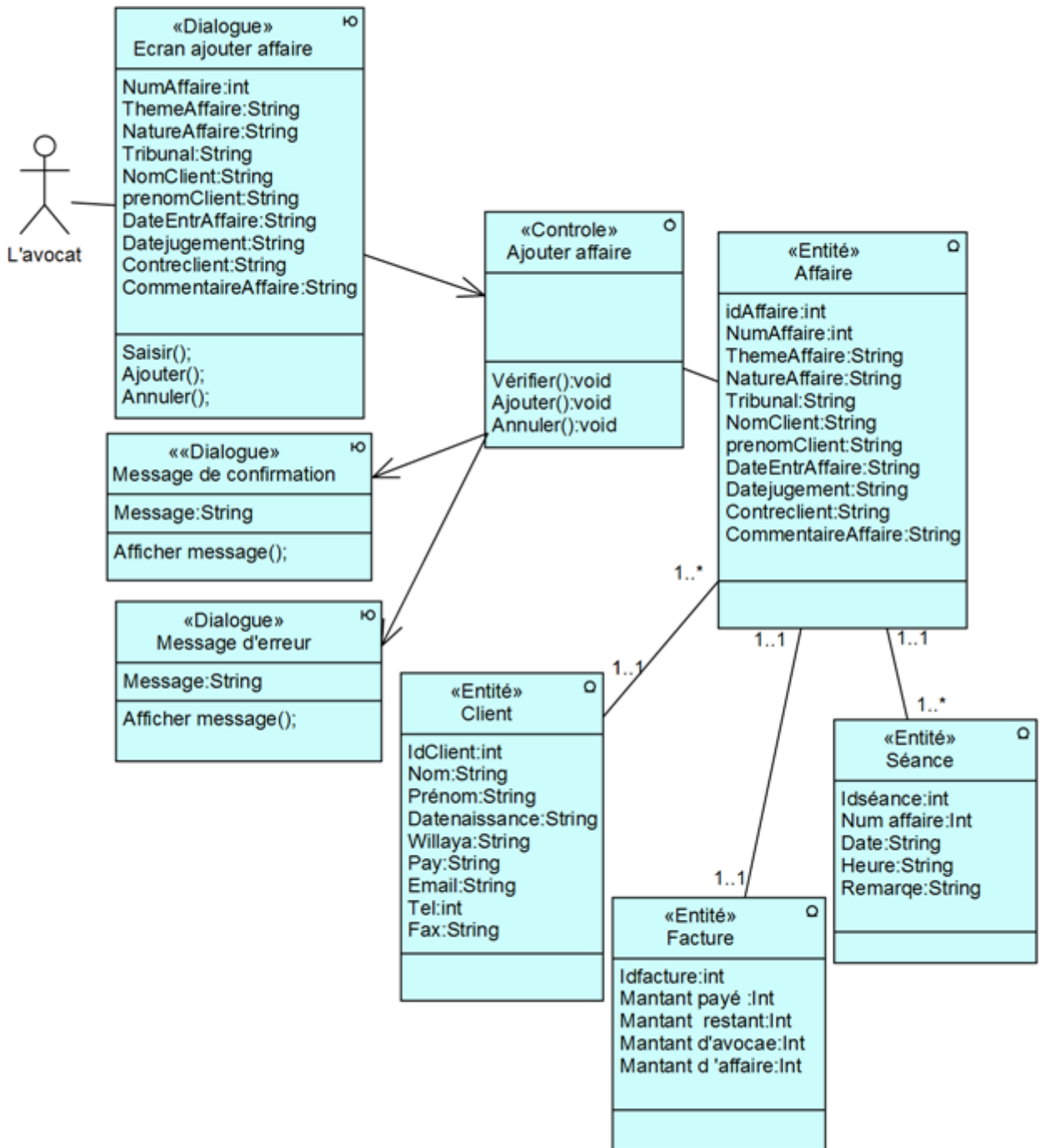


FIGURE 4.39 – Diagramme de classe participantes pour le cas d'utilisation «Ajouter Affaire»

• Cas d'utilisation « Modifier Affaire » :

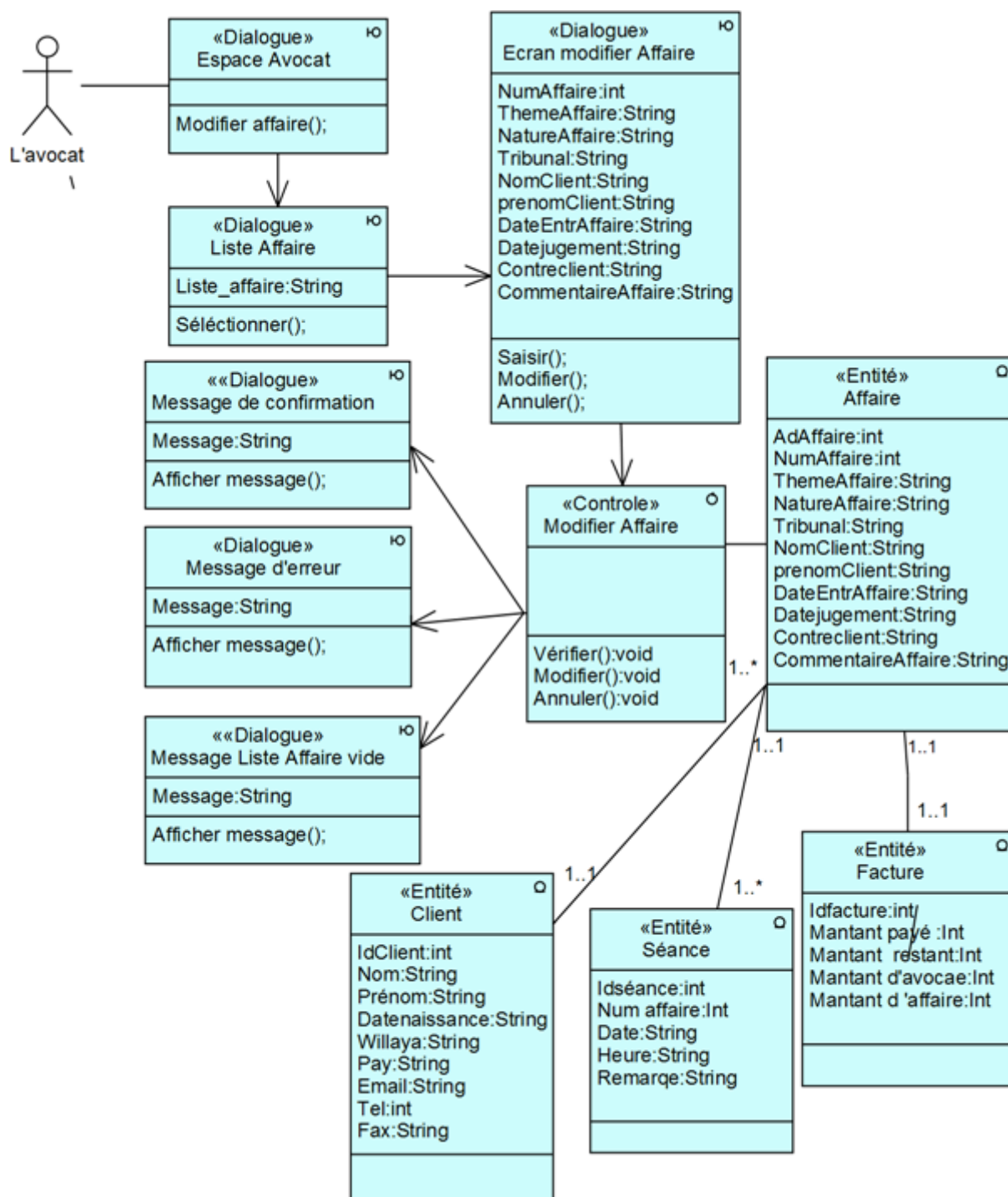


FIGURE 4.40 – Diagramme de classe participantes pour le cas d'utilisation «Modifier Affaire»

• Cas d'utilisation « Supprimer Affaire » :

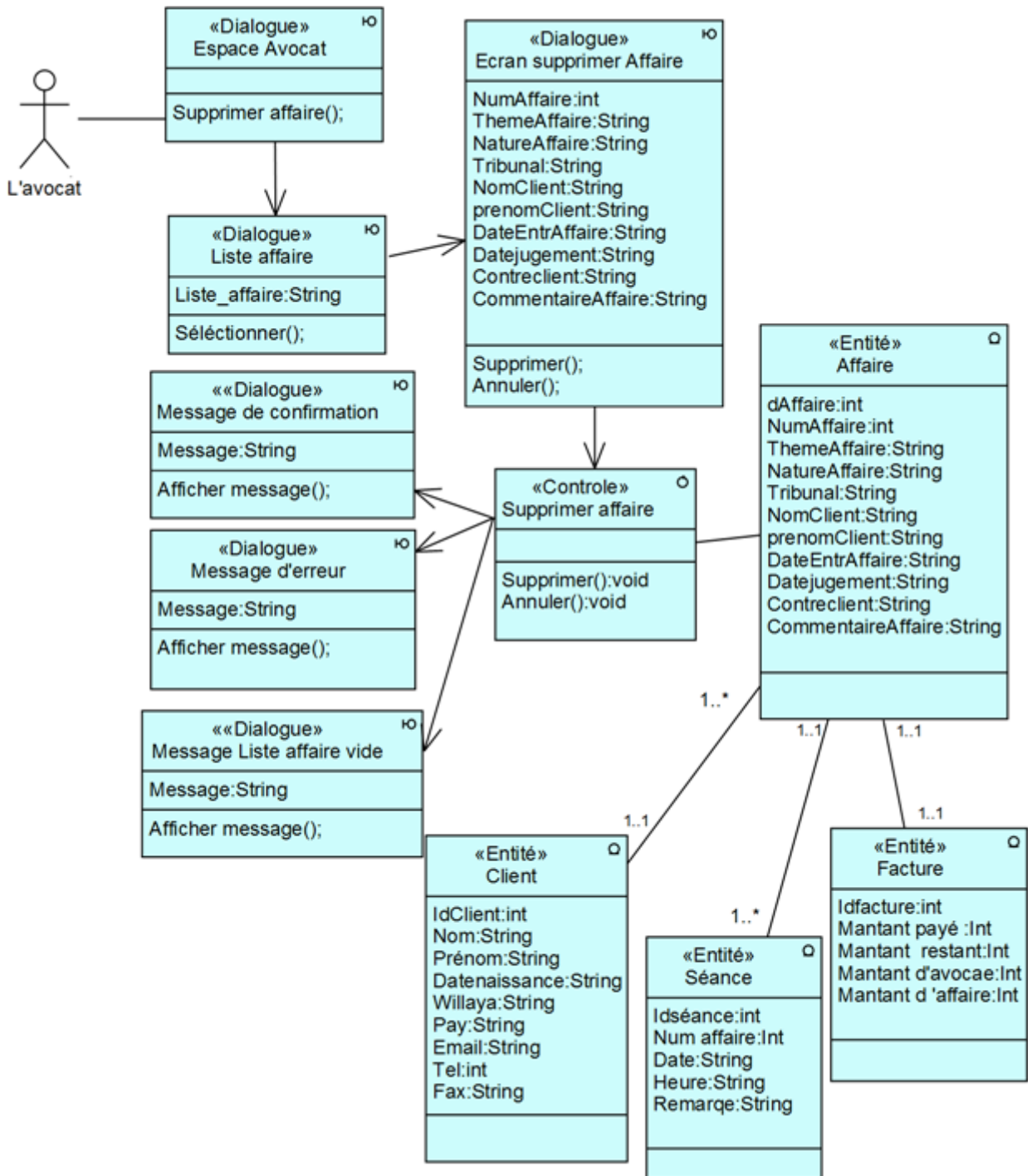


FIGURE 4.41 – Diagramme de classe participantes pour le cas d'utilisation «Supprimer Affaire»

• Cas d'utilisation « Ajouter facture » :

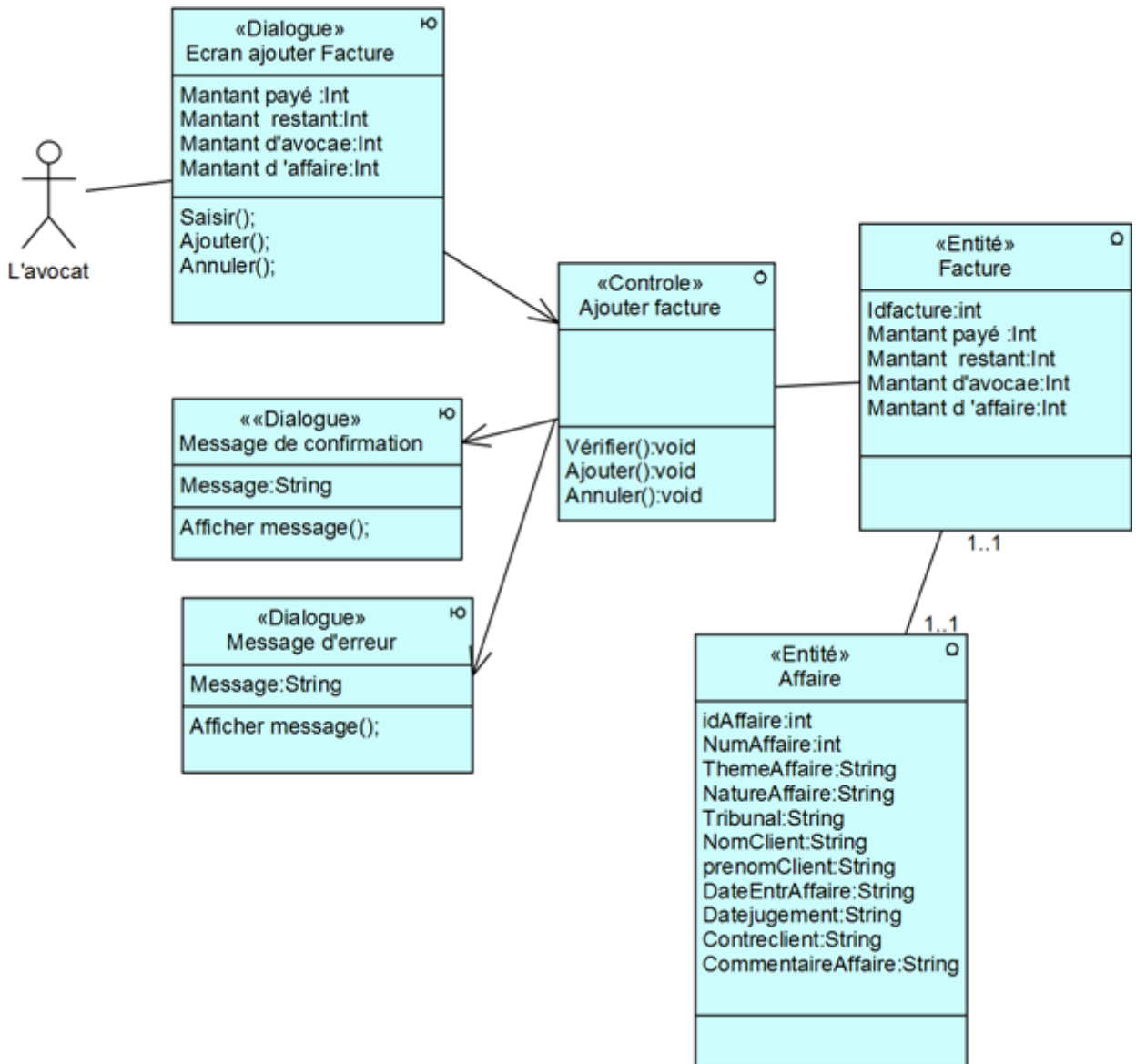


FIGURE 4.42 – Diagramme de classe participantes pour le cas d'utilisation «Ajouter facture»

• Cas d'utilisation « Modifier facture » :

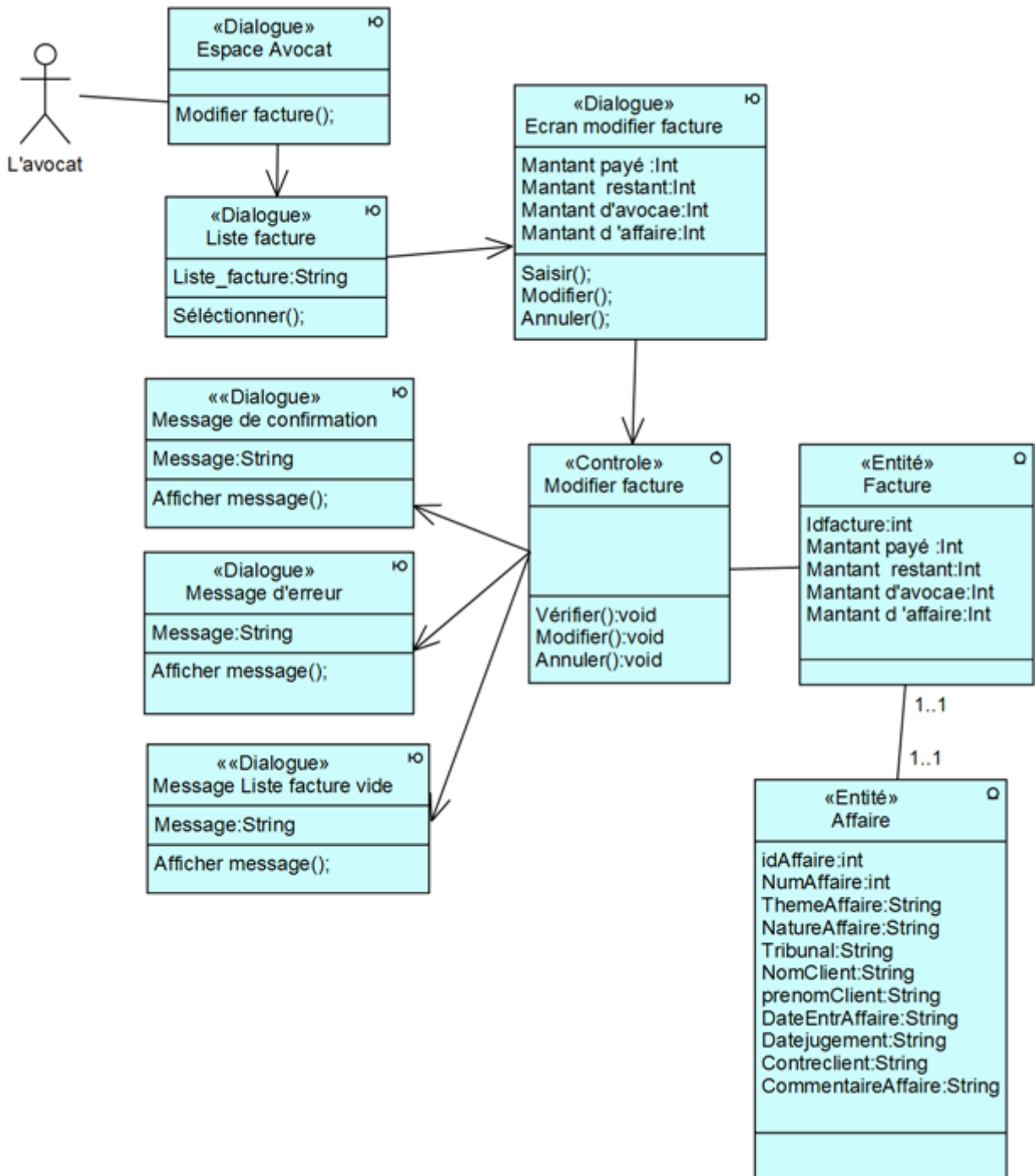


FIGURE 4.43 – Diagramme de classe participantes pour le cas d'utilisation «Modifier facture»

• Cas d'utilisation « Supprimer facture » :

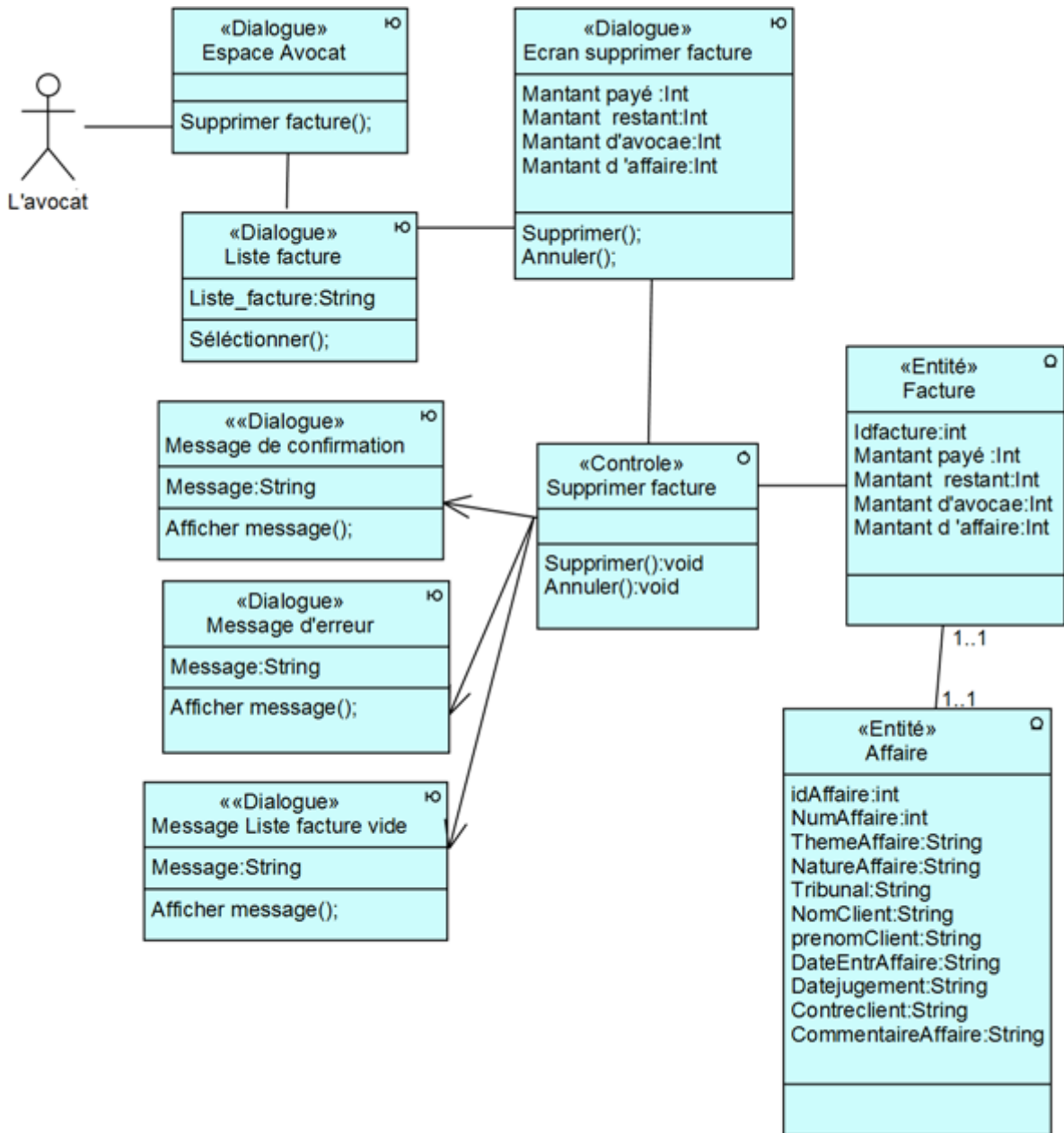


FIGURE 4.44 – Diagramme de classe participantes pour le cas d'utilisation «Supprimer facture»

• Cas d'utilisation « Demander consultation » :

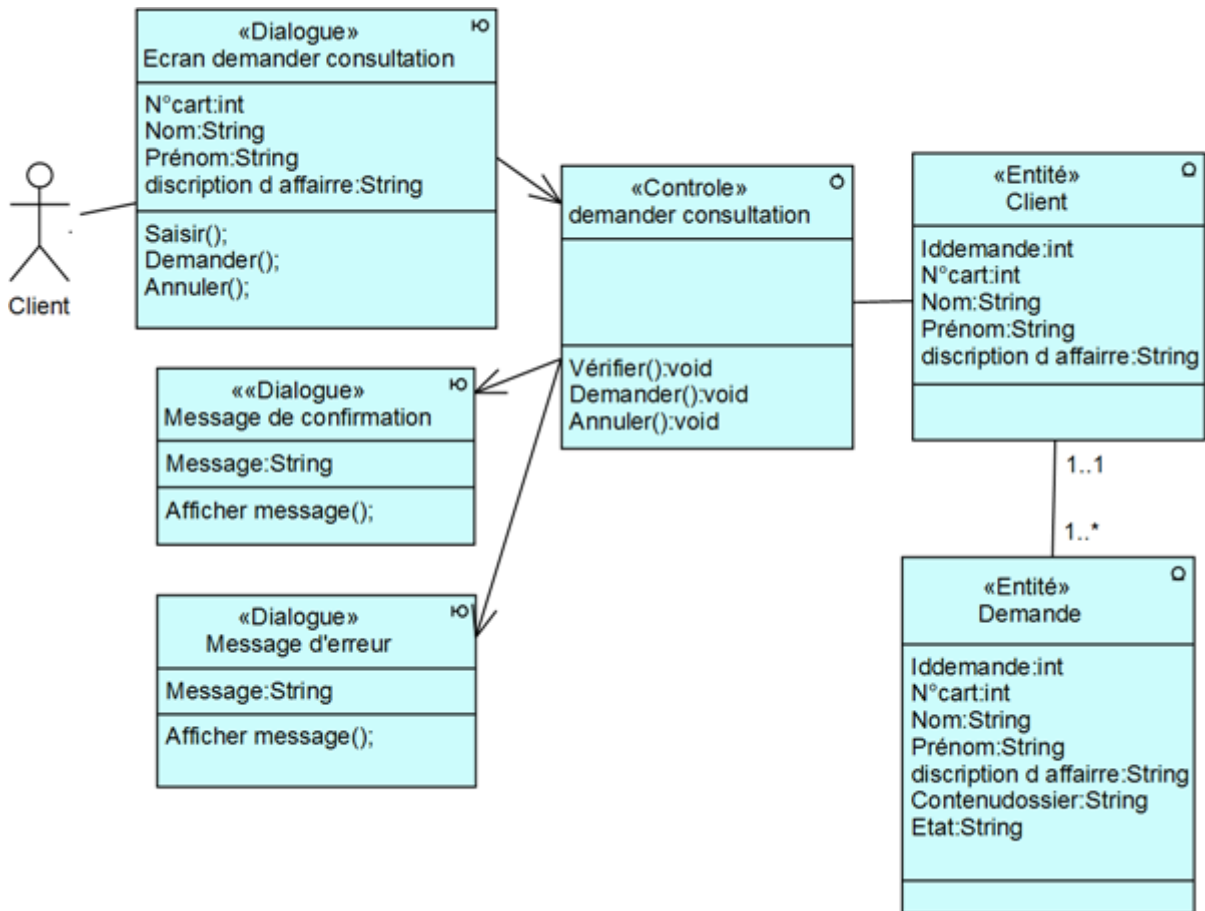


FIGURE 4.45 – Diagramme de classe participantes pour le cas d'utilisation «Demander consultation»

• Cas d'utilisation «Etablir procuration » :

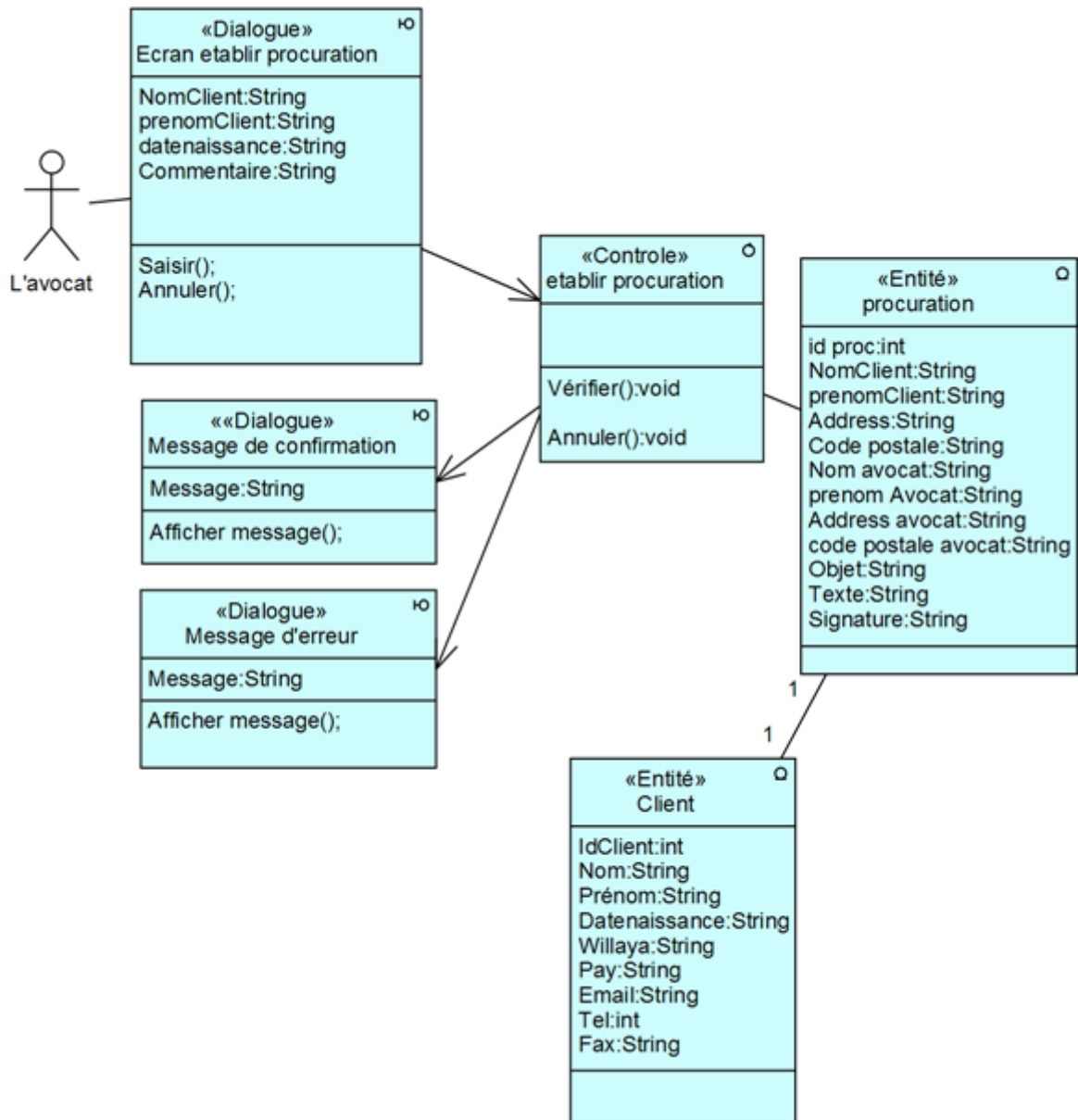


FIGURE 4.46 – Diagramme de classe participantes pour le cas d'utilisation « Etablir procuration»

• Cas d'utilisation « Consulter séance tribunal » :

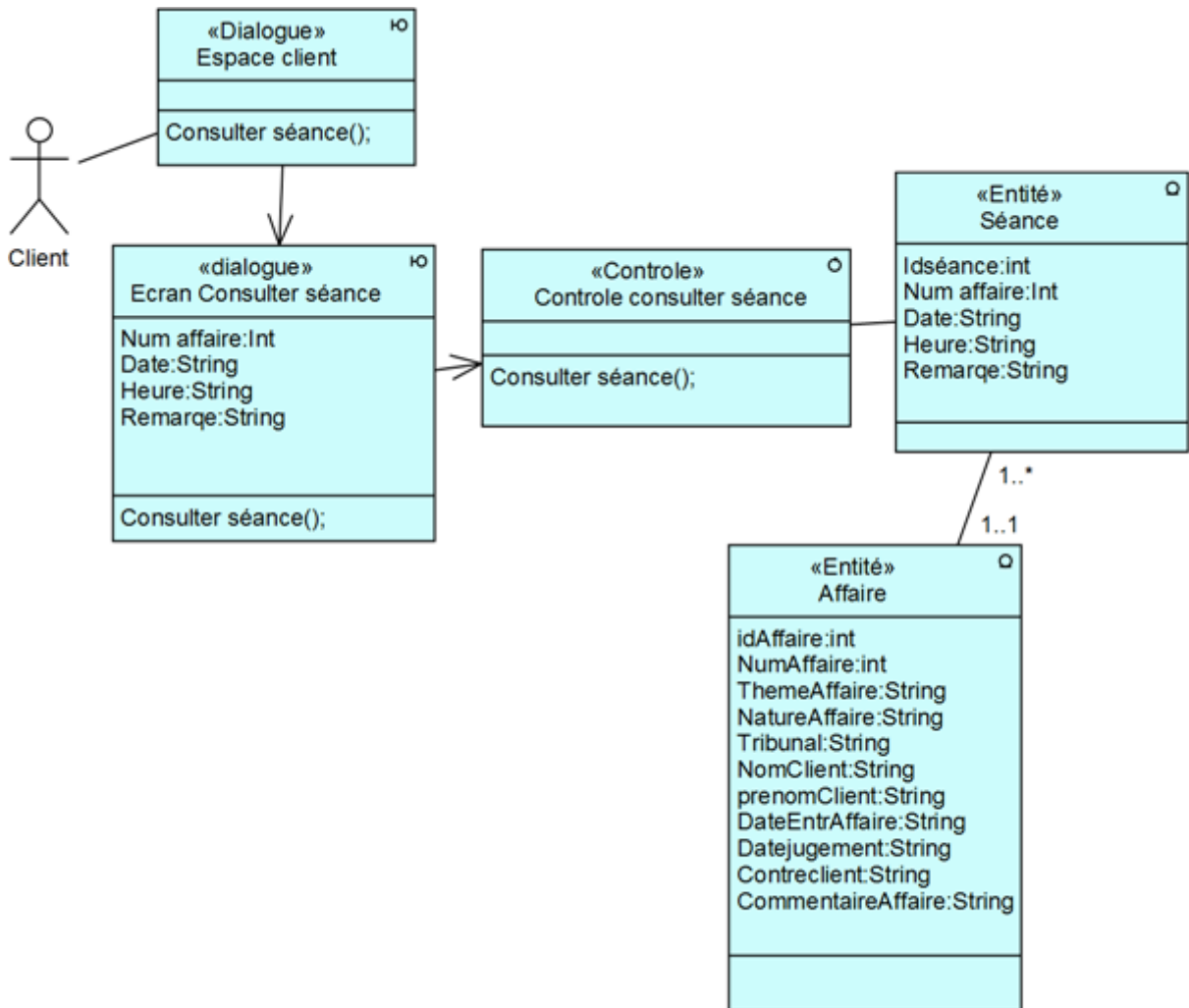


FIGURE 4.47 – Diagramme de classe participantes pour le cas d'utilisation « Consulter séance tribunal »

• Cas d'utilisation «Consulter Affaire » :

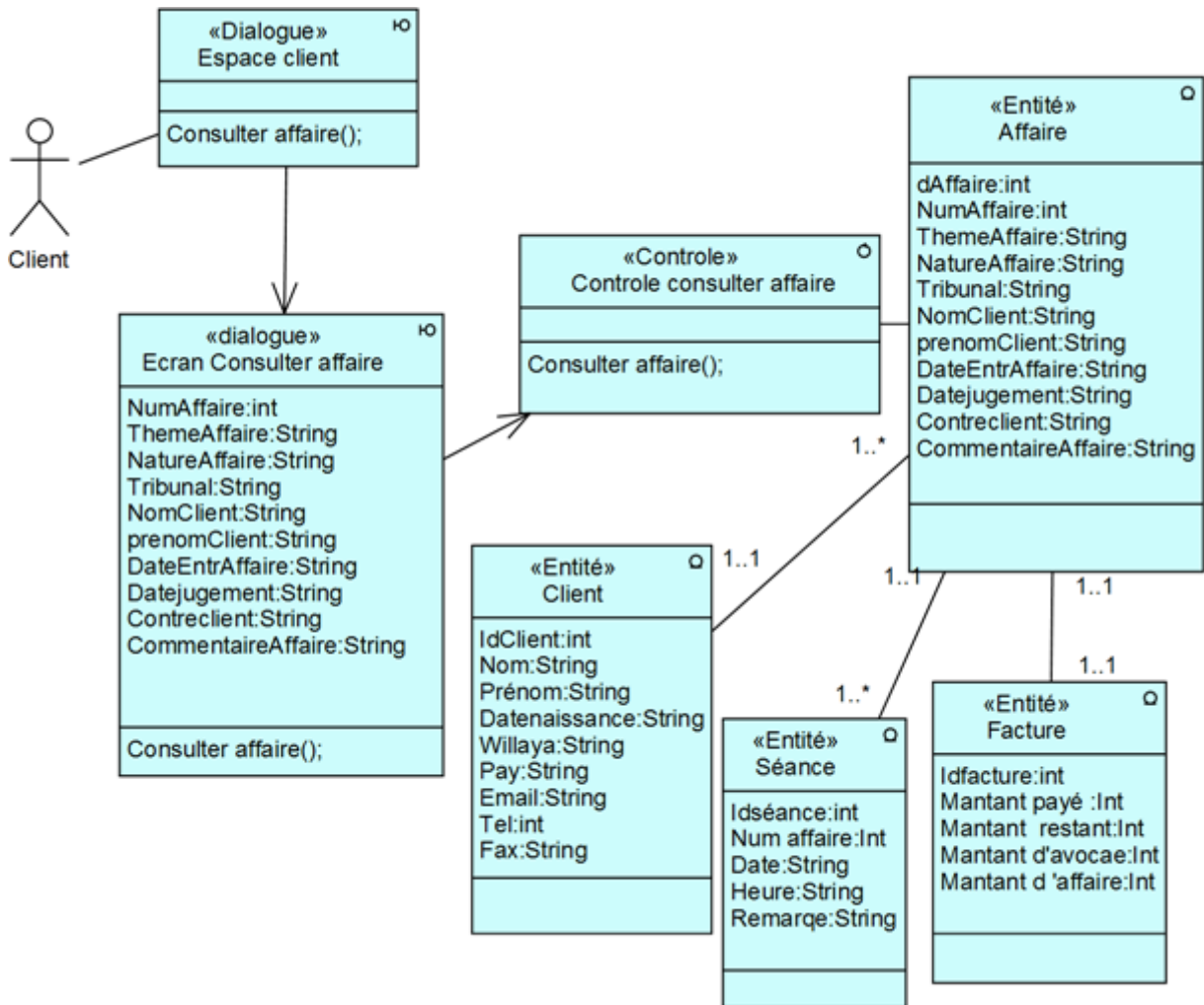


FIGURE 4.48 – Diagramme de classe participantes pour le cas d'utilisation «Consulter Affaire»

• Cas d'utilisation «Consulter Facture » :

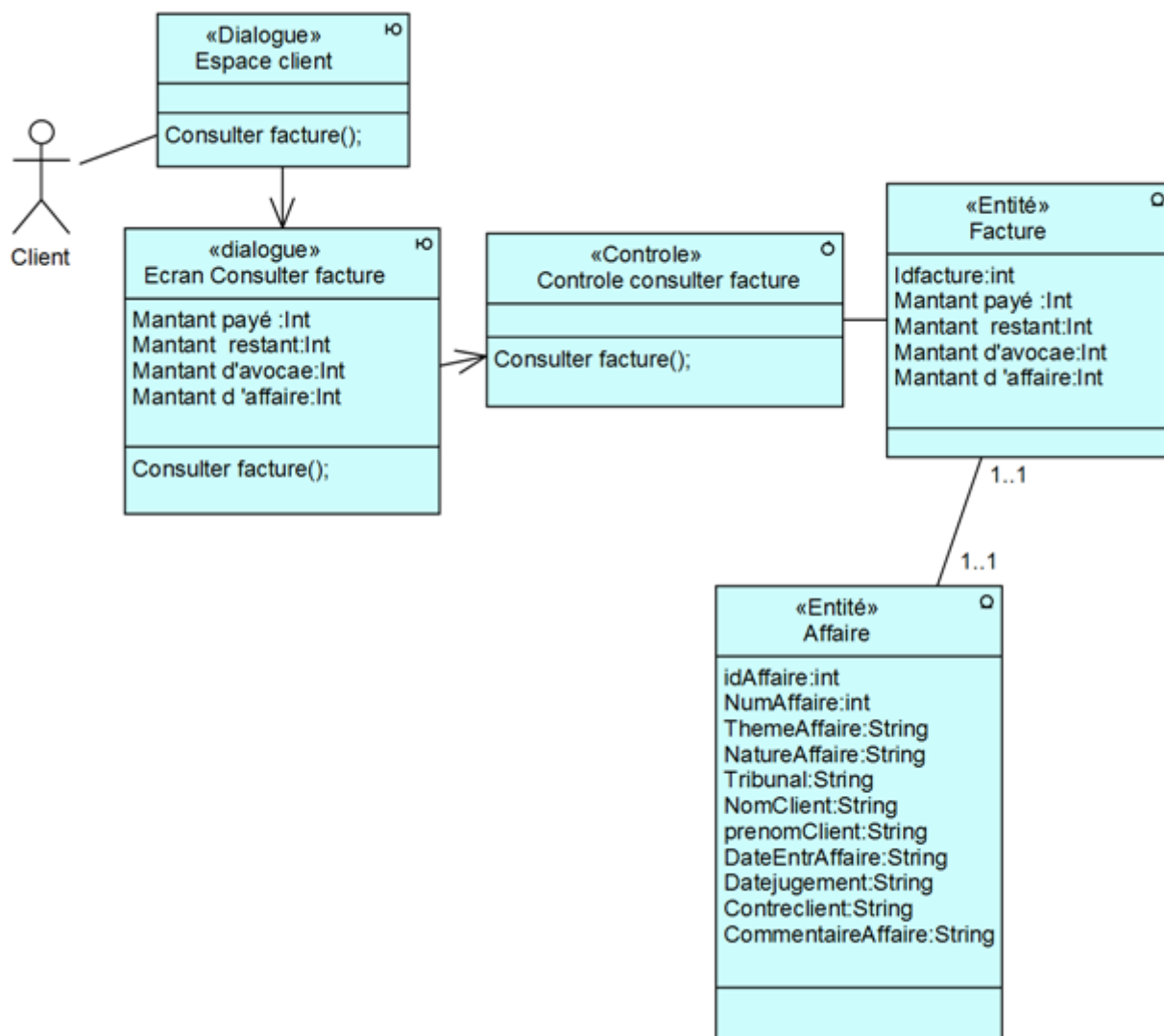


FIGURE 4.49 – Diagramme de classe participantes pour le cas d'utilisation «Consulter Facture»

4.4 Capture des besoins technique :

Après avoir développé précédemment les besoins fonctionnels qui constituent la branche gauche du cycle Y, nous allons présenter la seconde branche du cycle en occurrence des besoins techniques du système.

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. Le modèle de spécification s'exprime suivant deux points de vue qui sont :

- La spécification technique du point de vue matériel.
- La capture des spécifications logicielles. [8]

4.4.1 La spécification technique du point de vue matériel :

Les prérequis techniques ont été exprimés dans l'étude préliminaire, lors de l'expression des besoins opérationnels et de celle des choix stratégiques de développement. Ces choix impliquent des contraintes relatives à la configuration du réseau matériel. Elles sont de nature géographique, organisationnelle, et technique. Elles concernent les performances d'accès aux données, la sécurité du système, la volumétrie. [8]

- **Style d'architecture en niveaux**

Le style d'architecture en niveaux spécifie le nombre de niveaux géographiques et organisationnels où vont se situer les environnements d'exécution du système. [2]

Dans notre système La configuration géographique impose le développement d'une solution client/serveur à deux niveaux.

- **Spécification d'architecture 2 niveaux :**

Une architecture à deux niveaux aussi appelée client-serveur, est une architecture où un Poste client exécute une application se trouvant sur un serveur. cela effectué comme un dialogue entre le client et le serveur dans lequel :

- **Client** : demande une ressource.
- **Serveur** : la lui fournit directement.

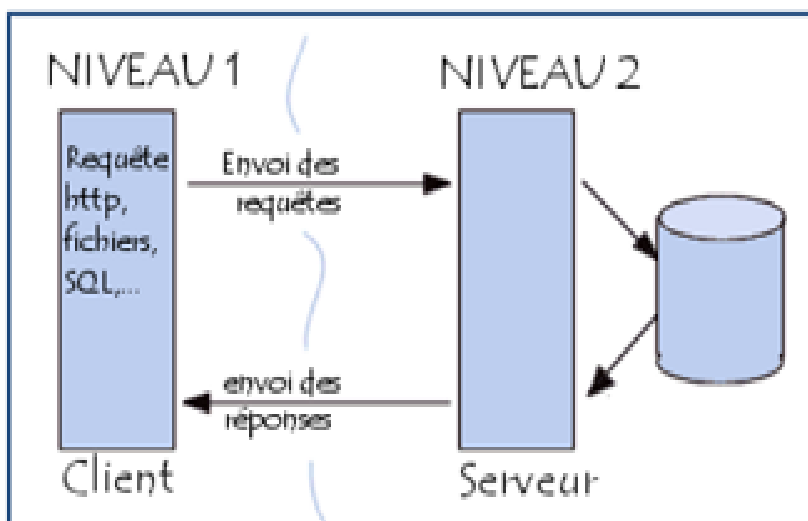


FIGURE 4.50 – Architecture à 2 niveaux

4.4.2 La capture des spécifications logicielles :

Une fois que les spécifications techniques et d'architecture sont exprimées, on peut s'intéresser aux fonctionnalités propres du système technique en procédant à une spécification logicielle. Dans ce cas, on utilise les cas d'utilisation de manière différente que pour la spécification fonctionnelle. C'est pourquoi nous avons introduit le concept d'exploitant et de cas d'utilisation technique.

- **Exploitant :**

L'exploitant est un acteur au sens d'UML, si ce n'est qu'il ne bénéficie que des fonctionnalités techniques du système.

- **Cas d'utilisation technique :**

un cas d'utilisation technique est destiné à l'exploitant, c'est une séquence d'action produisant une valeur ajoutée opérationnelle ou purement techniques. [8]

- ✓ **Identification des exploitants du système**

Les exploitants de notre système sont la majorité des acteurs de la branche fonctionnelle : l'avocat et le client.

- ✓ **Identification des cas d'utilisation technique**

Les principaux cas d'utilisation techniques de notre système sont illustrés par la figure suivante :

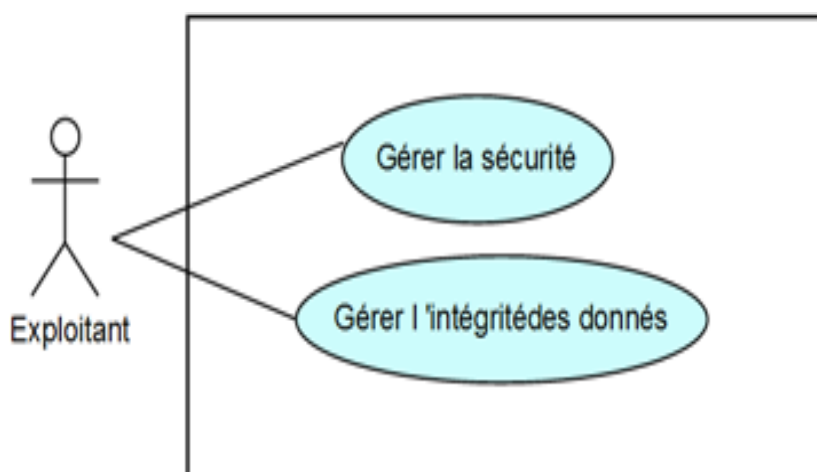


FIGURE 4.51 – Modèle de spécification logicielle du système.

Description des cas d'utilisation techniques :

- 1- « Gérer la sécurité »

- **S'authentifier :**

Cas d'utilisation	S'authentifier.
Acteur	Utilisateur.
But	Ce cas permet de vérifier l'autorisation d'accès à des fonctionnalités de l'application.
Pré condition	L'utilisateur possède un compte.
Poste condition	L'utilisateur est authentifié.
Scénario nominal	<p>1-L'utilisateur demande l'accès à l'application.</p> <p>2-Le système affiche le formulaire d'authentification.</p> <p>3-L'utilisateur introduit son identifiant et son mot de passe et confirme l'authentification.</p> <p>4-Le système vérifie les informations d'authentification.</p> <p>5-Le système renvoie l'interface correspondante.</p>
Scénario alternatif	<ul style="list-style-type: none"> • L'identifiant et/ou mot de passe incorrect. Le système affiche un message d'erreur. Reprise de l'enchaînement du scénario nominal au point 2. • L'utilisateur peut annuler l'opération en cours à n'importe quel moment pourvu qu'il ne l'a pas validé.

TABLE 4.28: Description textuelle du cas d'utilisation « **S'authentifier** »

• Diagramme de séquence :

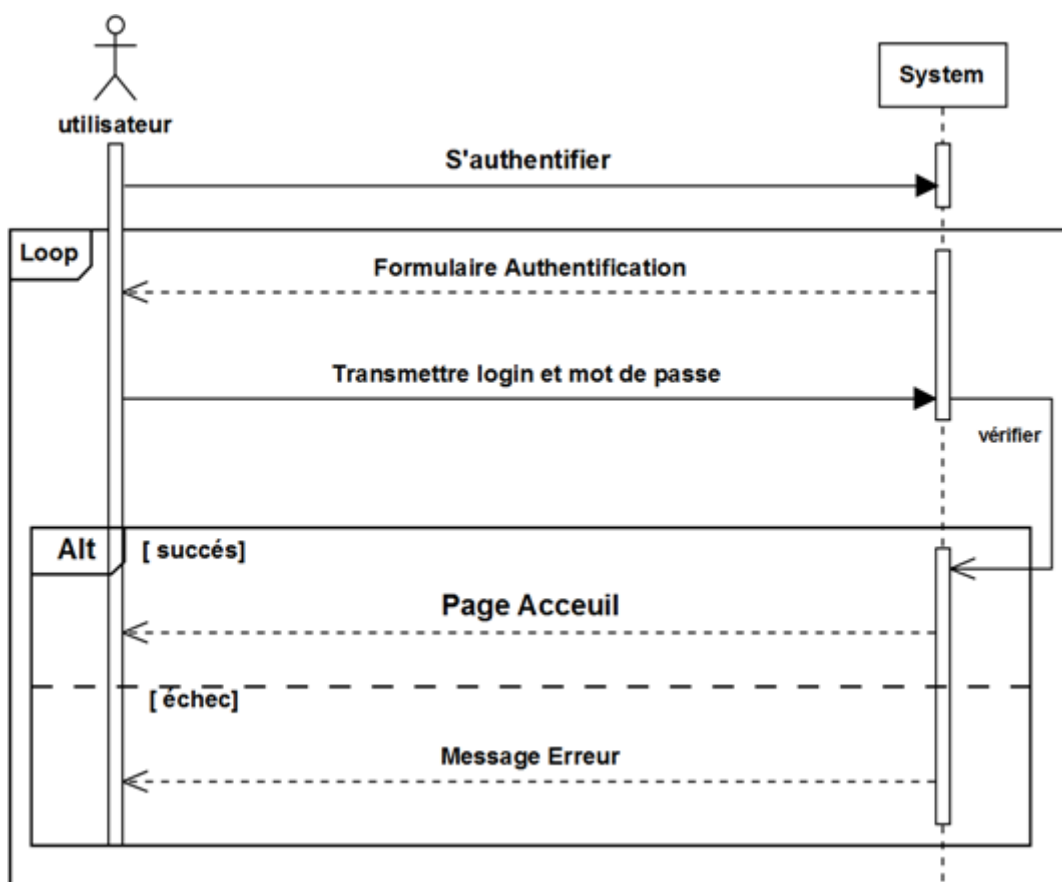


FIGURE 4.52 – Diagramme de séquence du cas s'authentifier

• 2- « Gérer l'intégrité des données »

Cas d'utilisation	Gérer l'intégrité des données.
Acteur	L'exploitant.
But	Vérifier l'intégrité des données.
Pré condition	L'exploitant s'authentifie.
Scénario nominal	1-L'exploitant saisit les données. 2-Le système contrôle la validité de chaque champ.
Scénario alternatif	• Le système affiche une notification « champs non valide ». Le scénario reprend au point 1.

TABLE 4.29: Description textuelle du cas d'utilisation « Gérer l'intégrité des données.»

- Diagramme séquence :

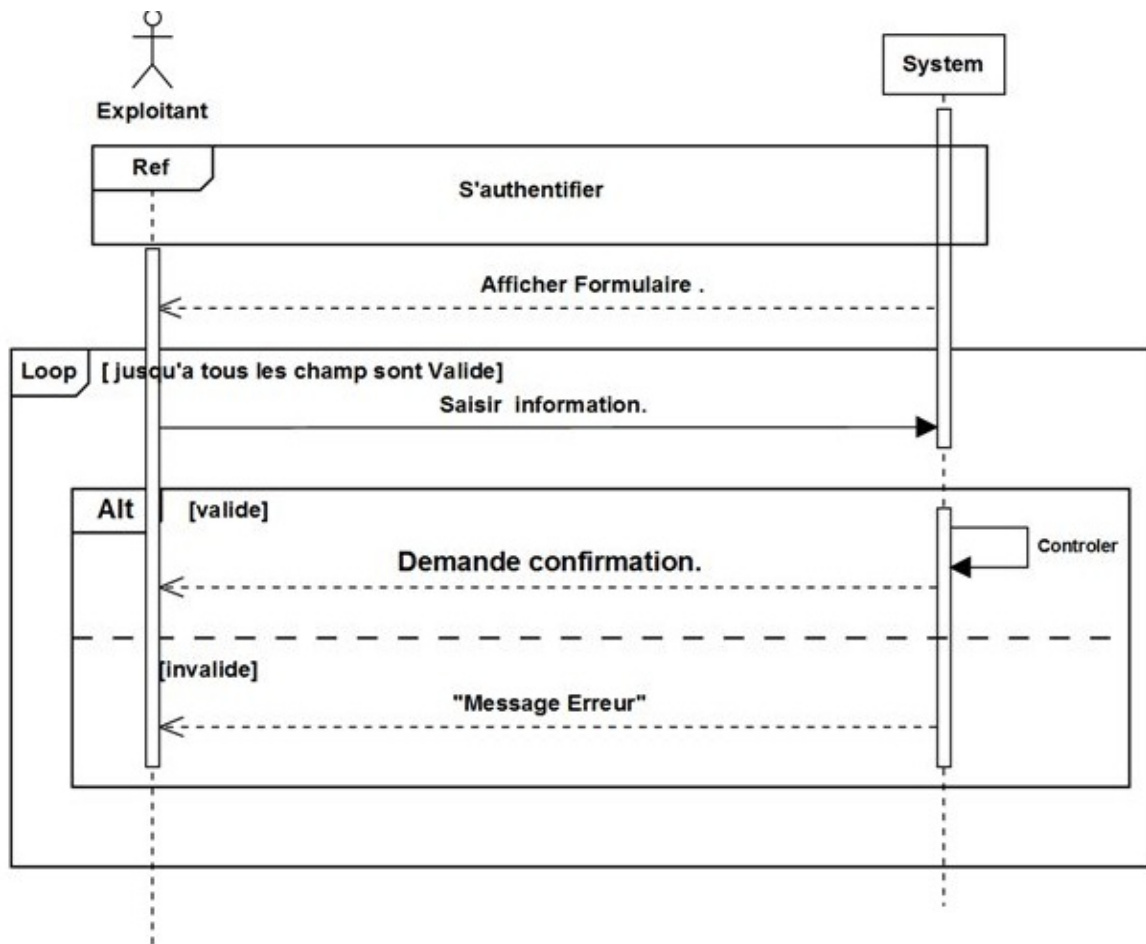


FIGURE 4.53 – Diagramme de séquence de cas d'utilisation « Gérer l'intégrité des données ».

4.5 Conclusion

La phase de capture des besoins est une étape qui œuvre d'un côté pour compléter les recueils des besoins opérés pendant la phase d'étude préliminaire, et d'un autre côté, elle donne une première vue pour la prochaine étape concernant l'analyse, afin d'identifier les classes candidates du modèle statique qui présentent une des approches orienté objet. La phase de capture des besoins techniques couvre par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description des métiers des utilisateurs ni de la description applicative.

Analyse et Conception

5.1 Introduction

Après la phase de l'étude préliminaire et capture des besoins étudiés dans le précédent chapitre, dans le présent chapitre nous présentons les principales activités pour les deux phases d'analyse et de conception.

5.2 Analyse :

La phase d'analyse représente la deuxième étape de la branche gauche du cycle en Y. Elle est consacrée à l'analyse d'objet de notre système juste après l'étude préliminaire et la capture des besoins fonctionnels et techniques. Elle consiste à étudier précisément les spécifications fonctionnelles afin d'obtenir une idée de ce que va réaliser le système en termes de métiers. Elle est constituée de 3 activités :

- Le découpage en catégories ;
- Le développement du modèle statique ;
- Le développement du modèle dynamique ; [8]

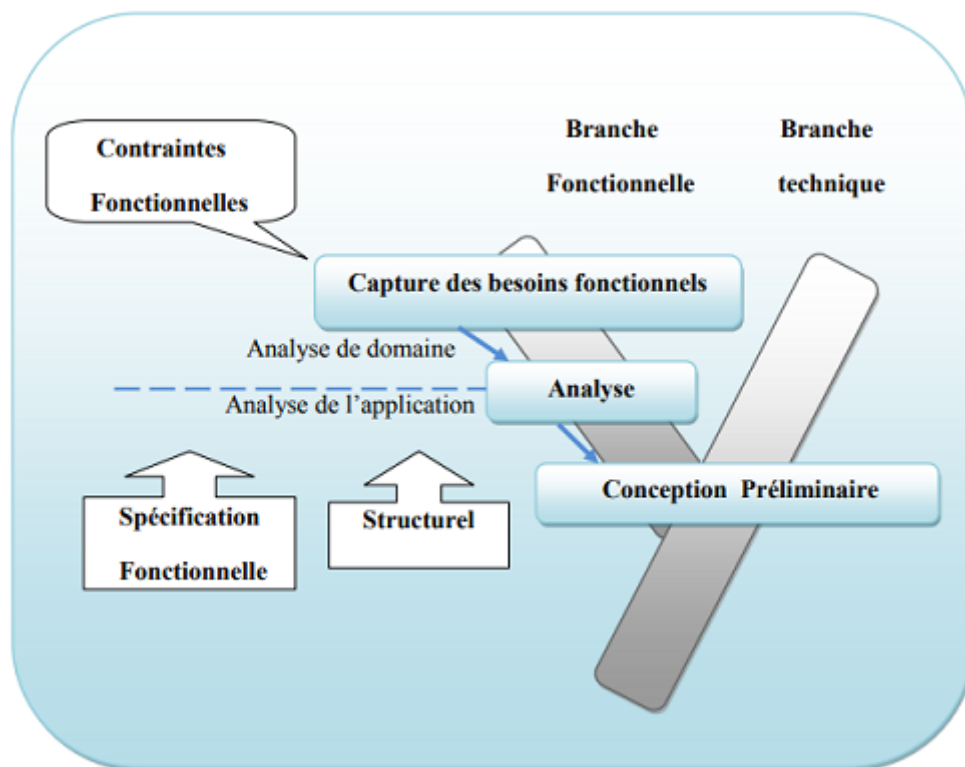


FIGURE 5.1 – Situation de la phase d’analyse dans 2TUP.

5.2.1 Découpage en catégories :

Le découpage en catégories constitue la première activité de l’étape d’analyse, permet de déterminer les classes fondamentales du projet en utilisant les diagrammes de classes participantes dégagés dans l’étape de captures des besoins fonctionnels, Il utilise la notion de package pour définir des catégories de classes d’analyse et découper le modèle UML en blocs logiques les plus indépendants possibles. [8]

Le découpage en catégories se fait en 3 étapes :

5.2.1.1 La répartition des classes candidates en catégories :

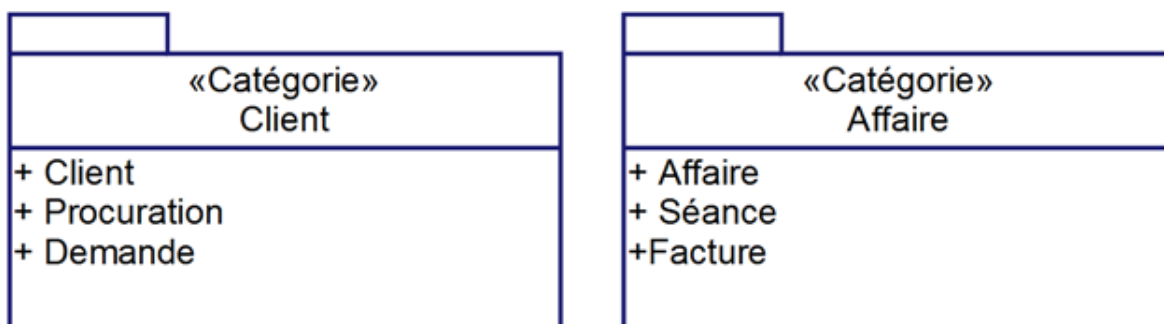


FIGURE 5.2 – Découpage en catégories

5.2.1.2 Élaboration des diagrammes de classes préliminaires par catégorie :

- Catégorie « Client »

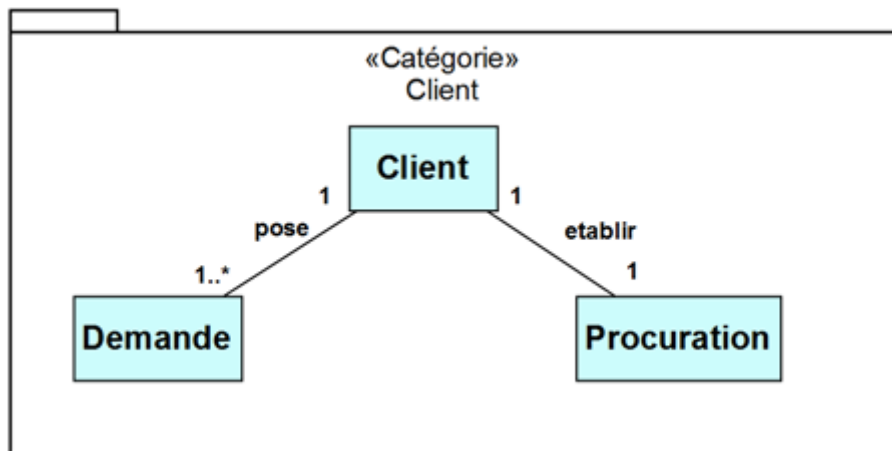


FIGURE 5.3 – Diagramme des classes préliminaire de la Catégorie « Client »

- Catégorie « Affaire »

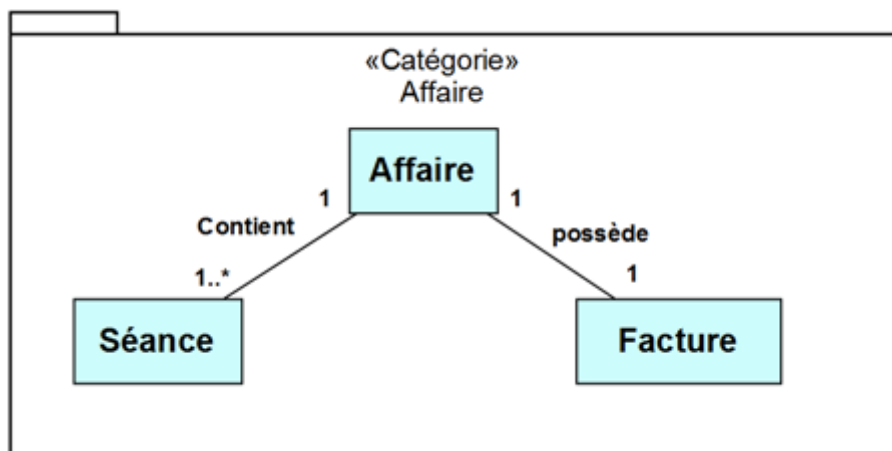


FIGURE 5.4 – Diagramme des classes préliminaire de la Catégorie « Affaire »

5.2.1.3 Dépendance entre catégories :

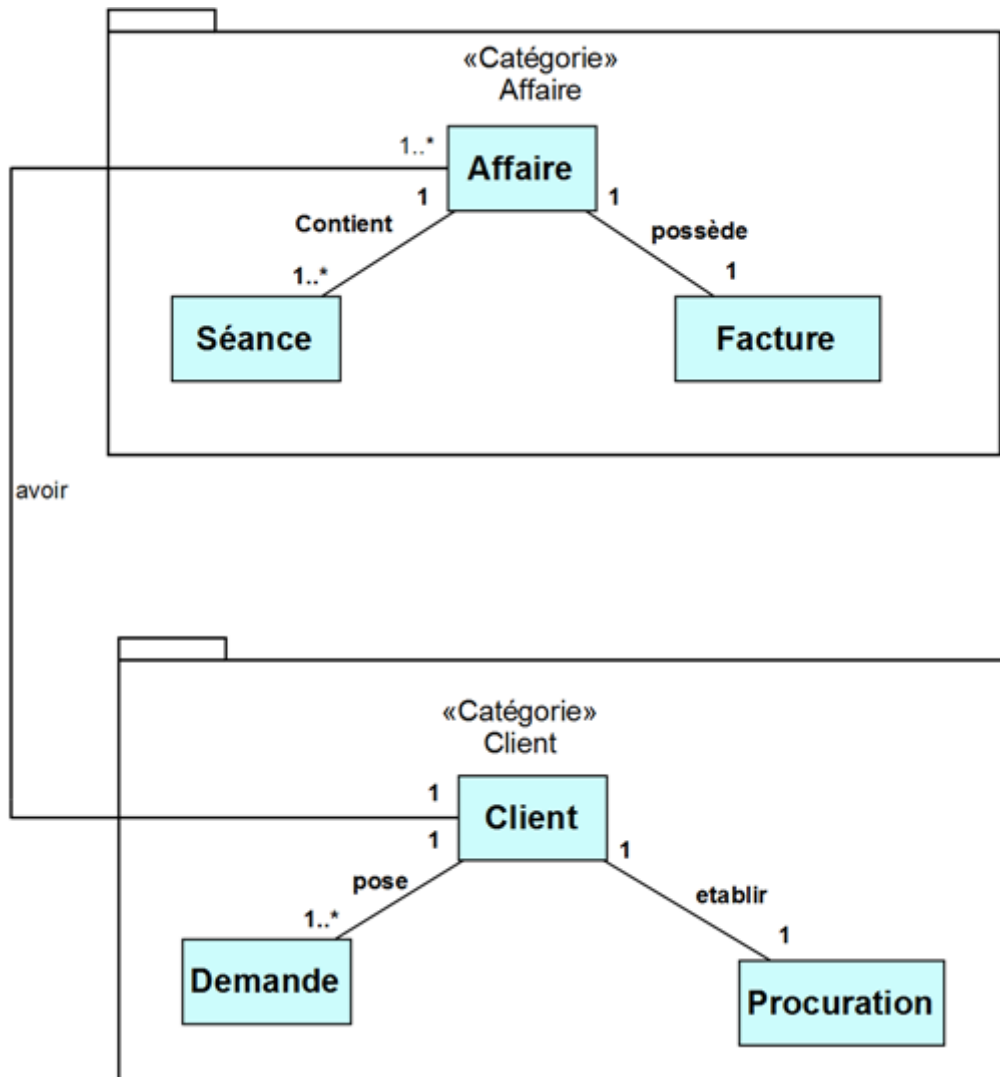


FIGURE 5.5 – Dépendance entre catégories

5.2.2 Développement du modèle statique :

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y et succède au découpage en catégories. Il consiste à détailler, compléter et optimiser les diagrammes de classes participantes obtenus et réorganisés lors du découpage en catégories. [8]

5.2.2.1 Diagramme de classe de la catégorie « Client » :

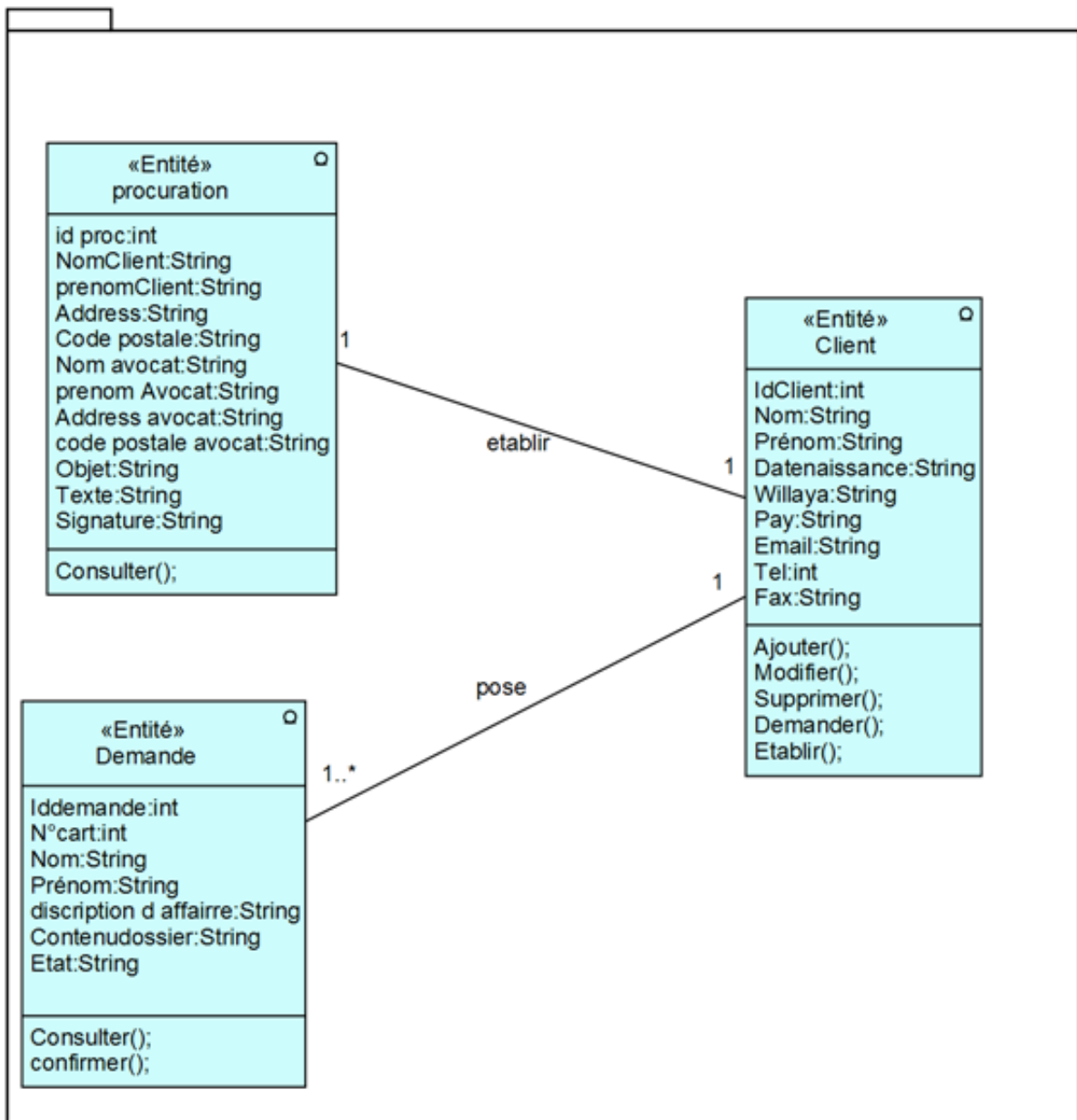


FIGURE 5.6 – Diagramme de classe de la catégorie « Client »

5.2.2.2 Diagramme de classe de la catégorie « Affaire » :

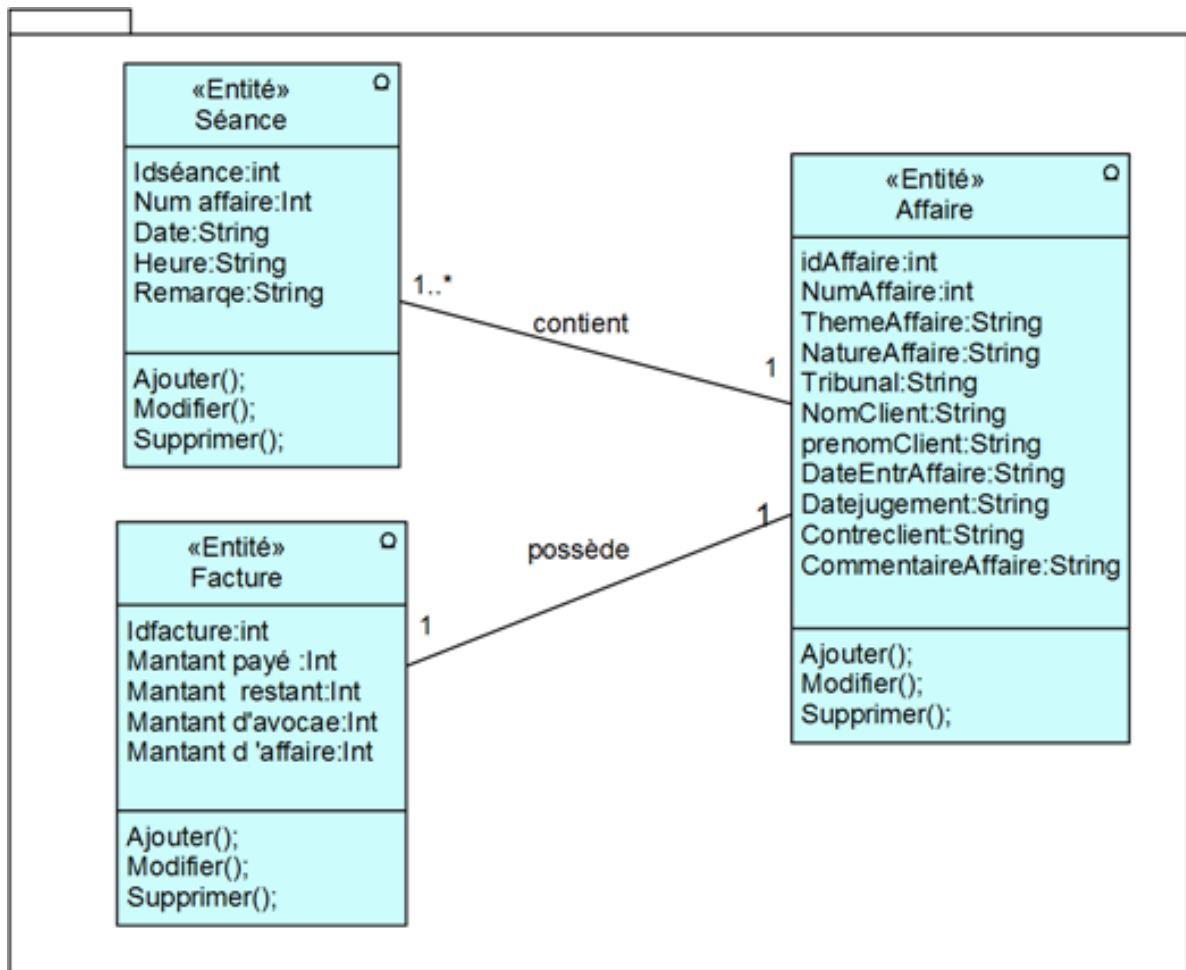


FIGURE 5.7 – Diagramme de classe de la catégorie « Affaire »

5.2.3 Développement du modèle dynamique :

Le développement du modèle dynamique constitue la troisième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y. Il s'agit d'une activité itérative, fortement couplée avec l'activité de modélisation statique. [8]

5.2.3.1 Diagrammes de séquence détaillés :

Nous avons vu à la partie de capture des besoins fonctionnels qu'un cas d'utilisation décrit un ensemble de scénarios. Lors de l'étape de détermination des besoins fonctionnels, un scénario représente une séquence d'interactions entre le système et ses acteurs. Le système est alors considéré comme une boîte noire. Maintenant que nous avons développé le modèle statique d'analyse, nous allons remplacer le système par une collaboration d'objets dans chaque scénario.

• Diagrammes de séquence détaillés « S'authentifier » :

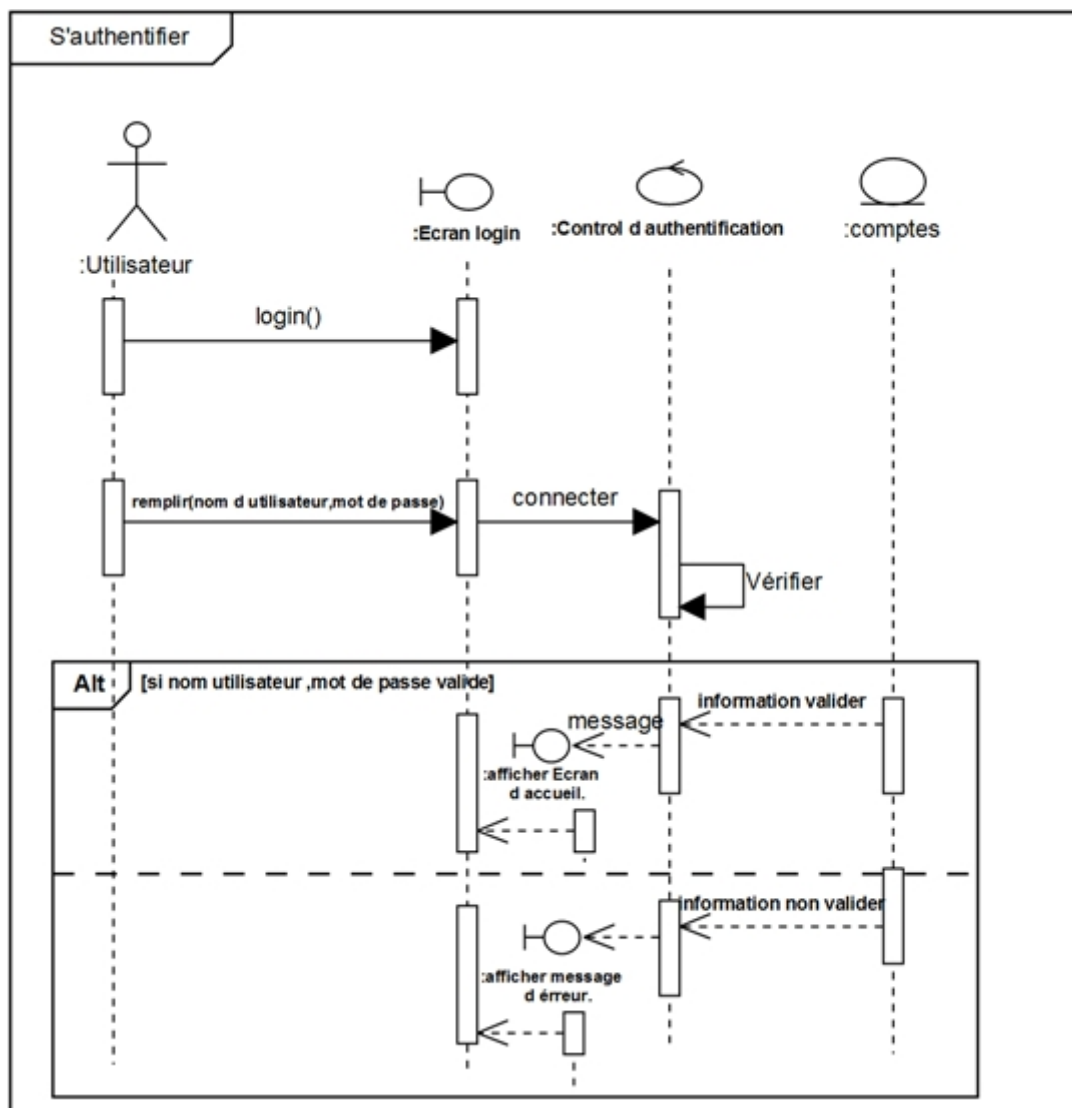


FIGURE 5.8 – Diagrammes de séquence détaillés « S'authentifier »

- Diagrammes de séquence détaillés « Consulter Demande » :

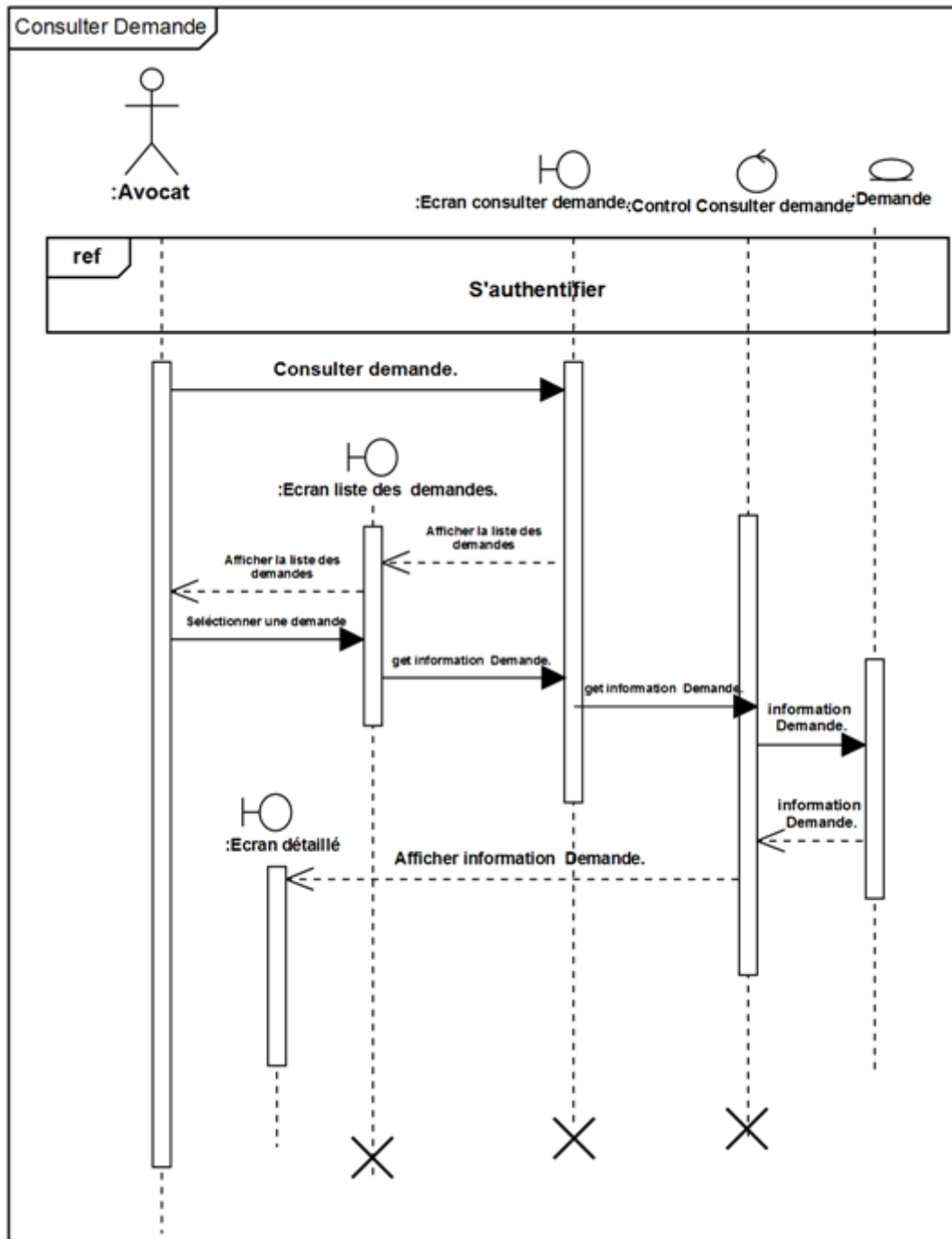


FIGURE 5.9 – Diagrammes de séquence détaillés « Consulter Demande »

• Diagrammes de séquence détaillés « Confirmer Demande » :

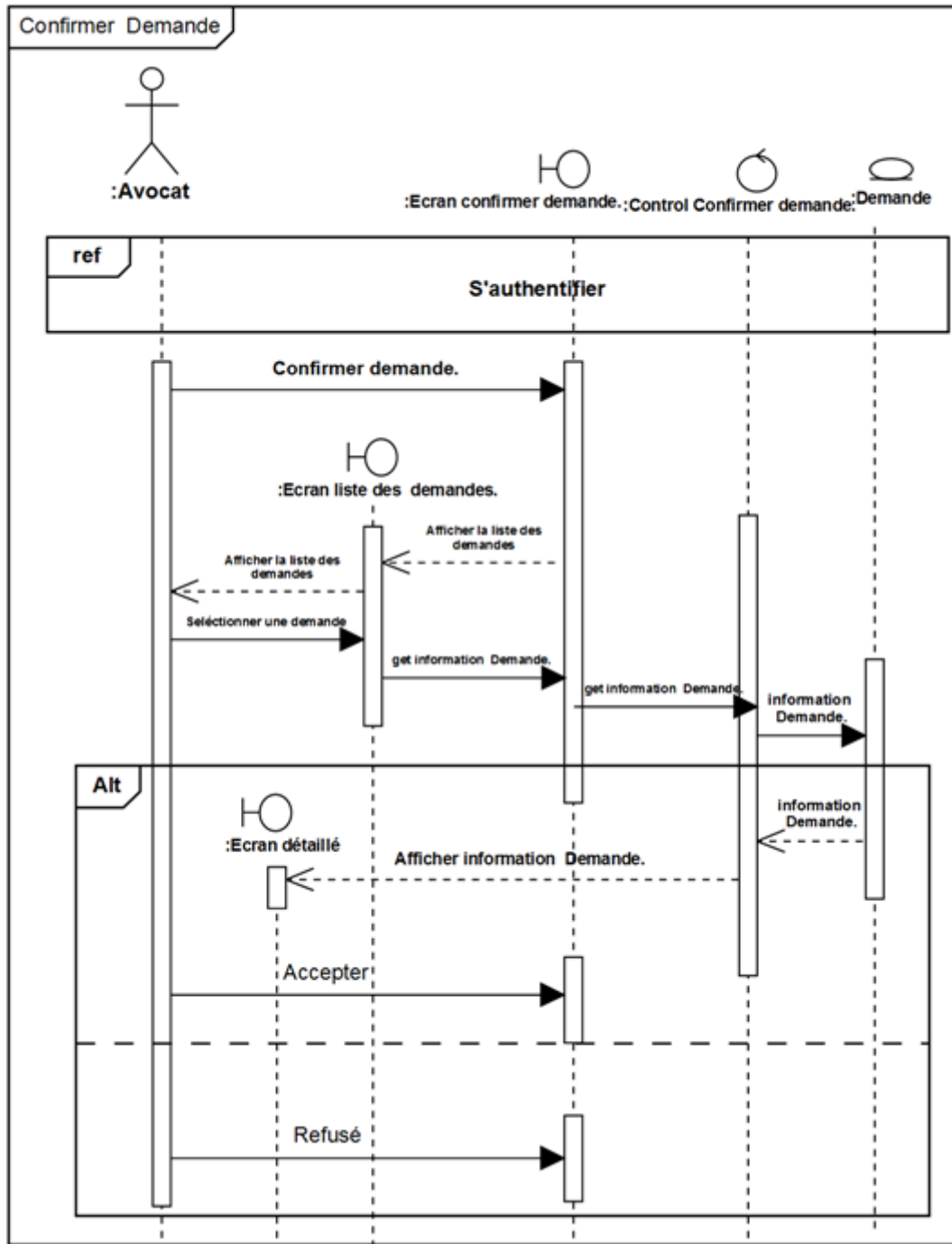


FIGURE 5.10 – Diagrammes de séquence détaillés « Confirmer Demande »

- Diagrammes de séquence détaillés « Ajouter Client » :

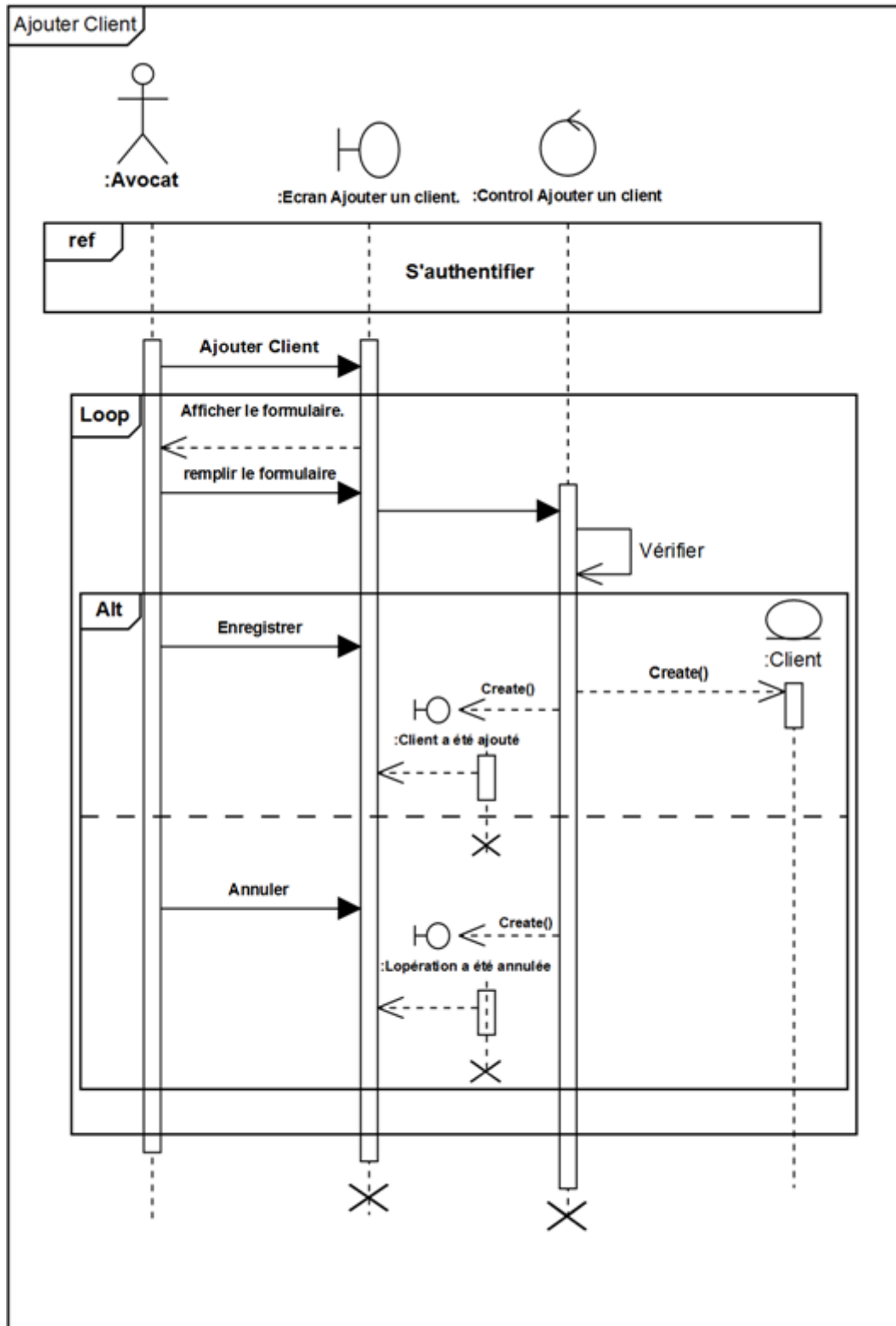


FIGURE 5.11 – Diagrammes de séquence détaillés « Ajouter Client »

- Diagrammes de séquence détaillés « Modifier Client » :

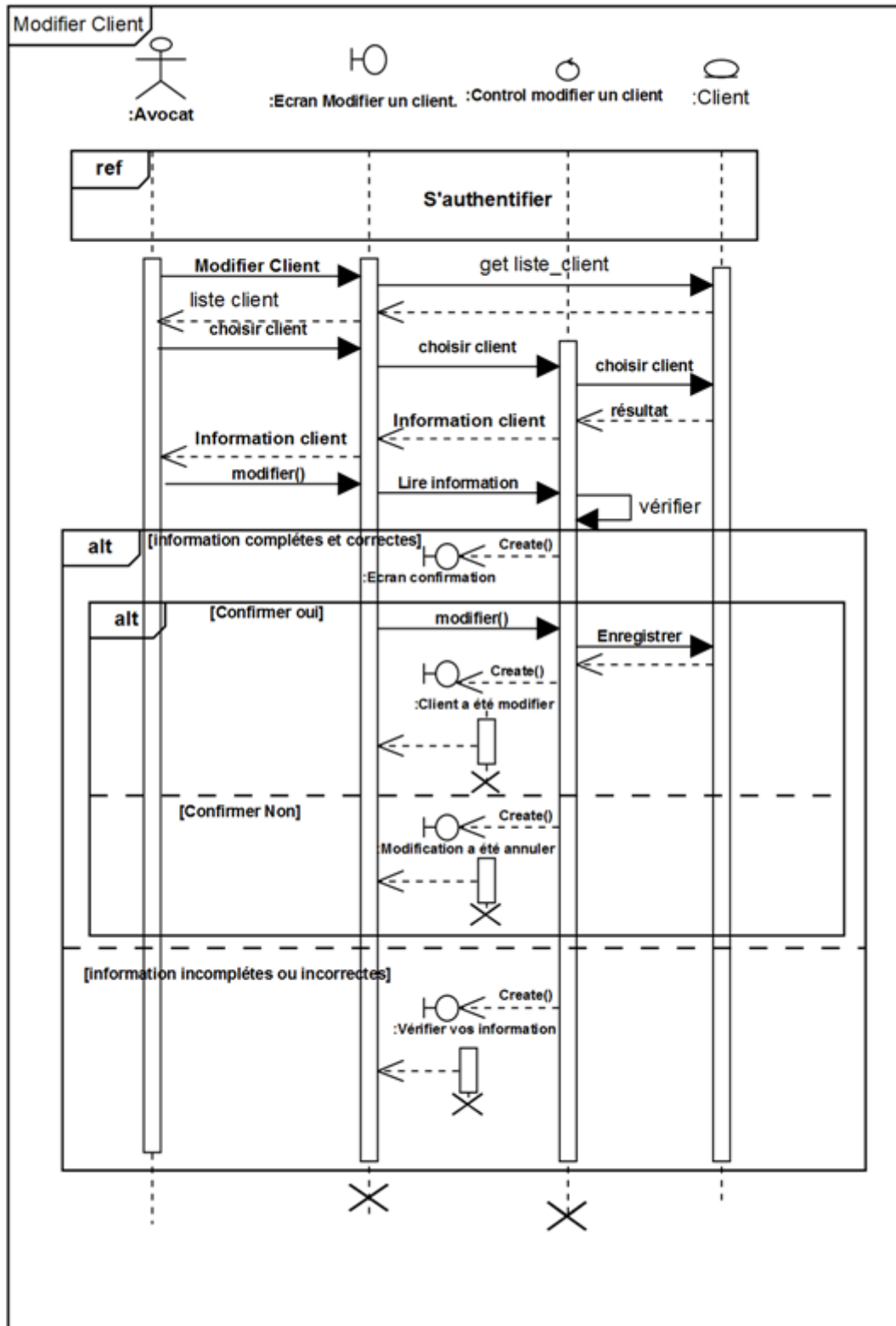


FIGURE 5.12 – Diagrammes de séquence détaillés « Modifier Client »

• Diagrammes de séquence détaillés « Supprimer Client » :

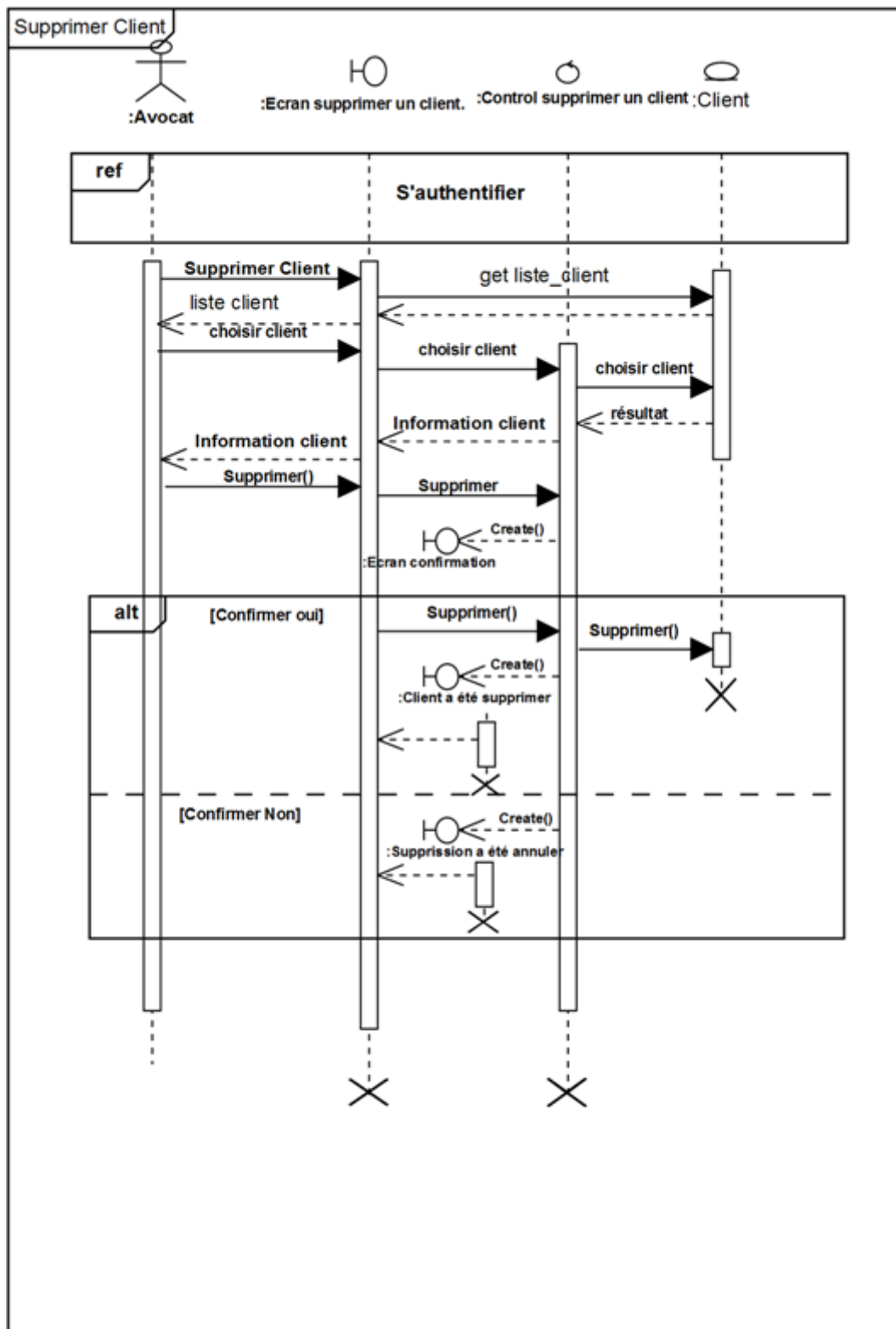


FIGURE 5.13 – Diagrammes de séquence détaillés « Supprimer Client »

- Diagrammes de séquence détaillés « Ajouter affaire » :

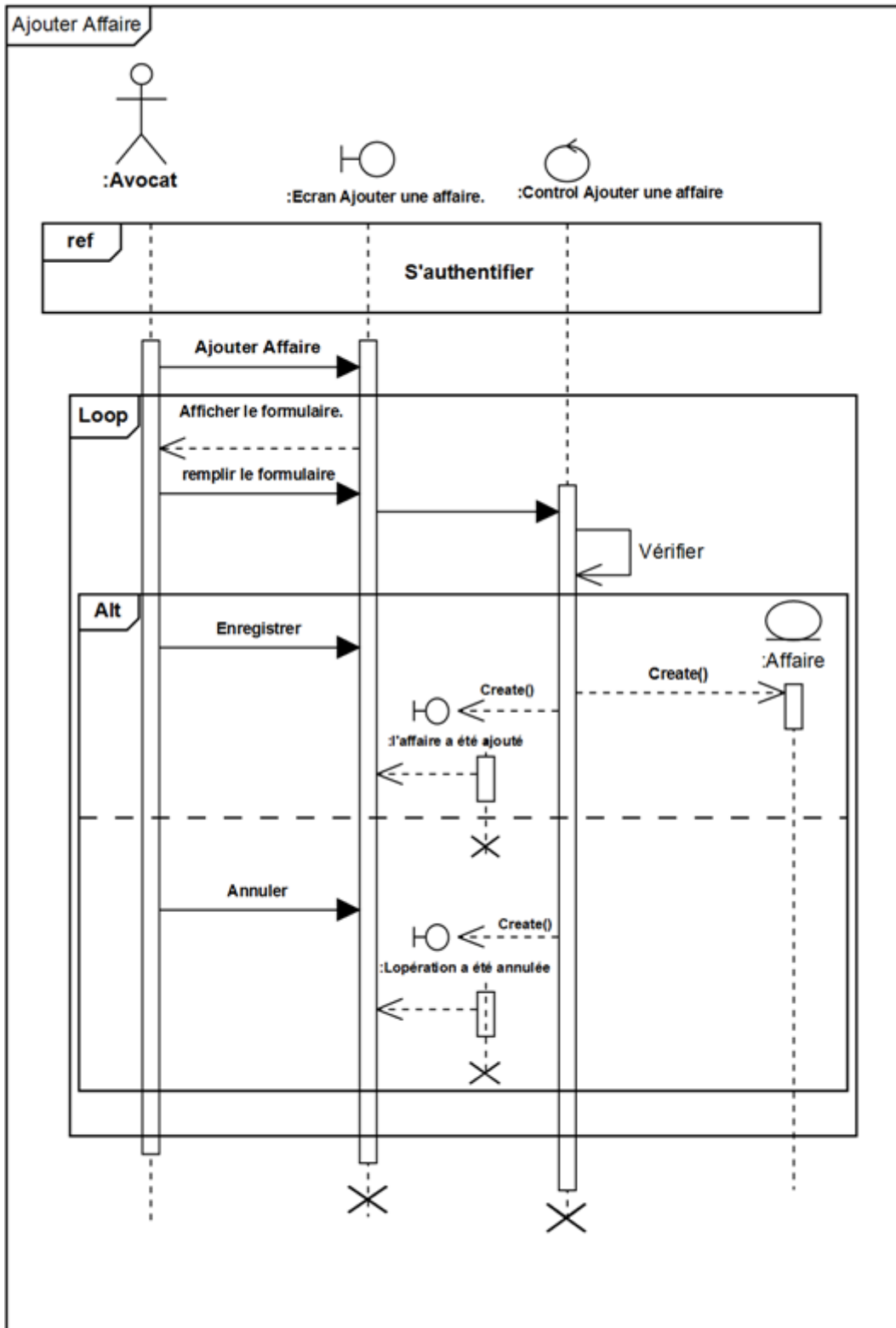


FIGURE 5.14 – Diagrammes de séquence détaillés « Ajouter affaire »

• Diagrammes de séquence détaillés « Modifier Affaire » :

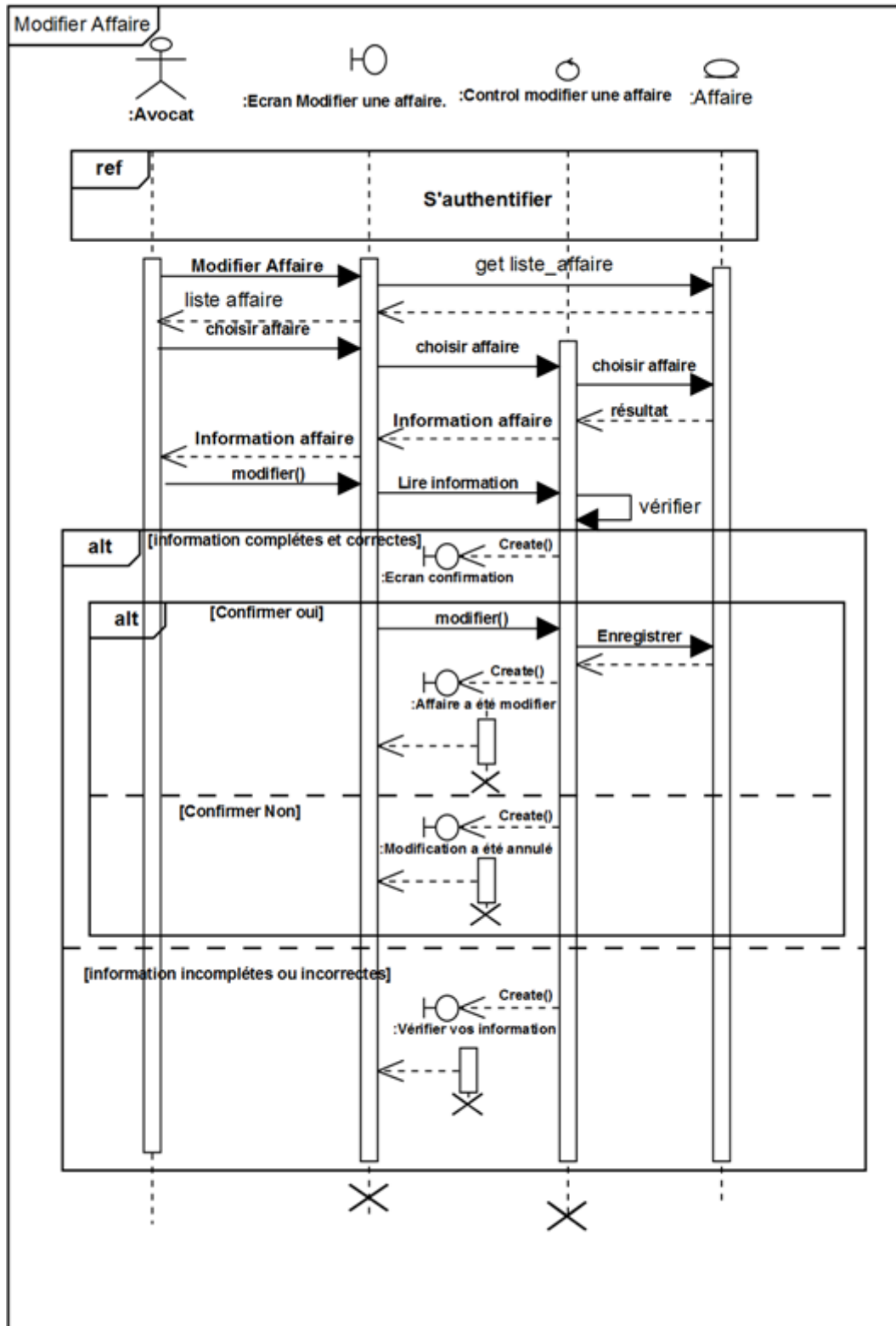


FIGURE 5.15 – Diagrammes de séquence détaillés « Modifier affaire »

• Diagrammes de séquence détaillés « Supprimer Affaire » :

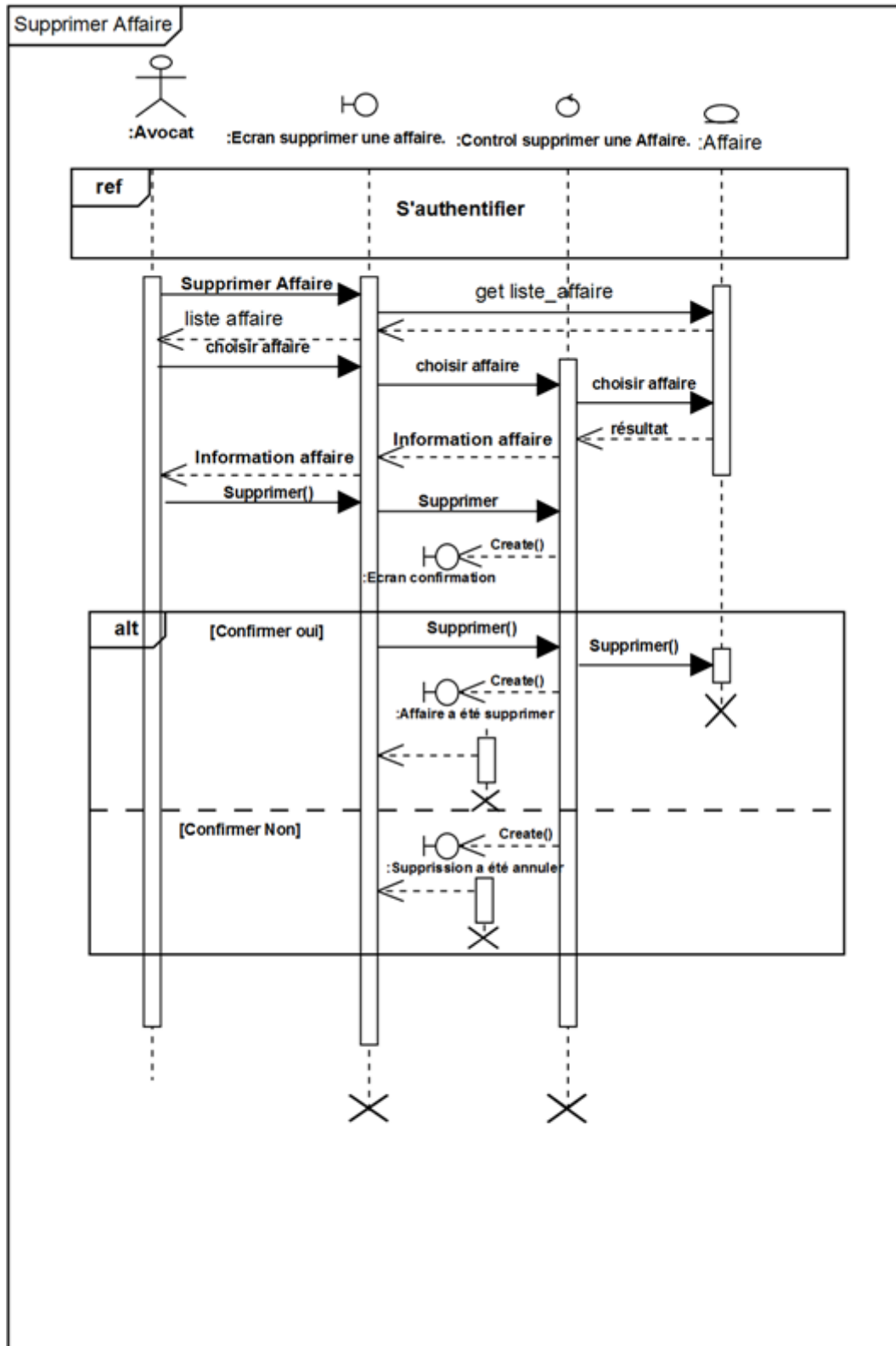


FIGURE 5.16 – Diagrammes de séquence détaillés « Supprimer affaire »

• Diagrammes de séquence détaillés « Demander Consultation » :

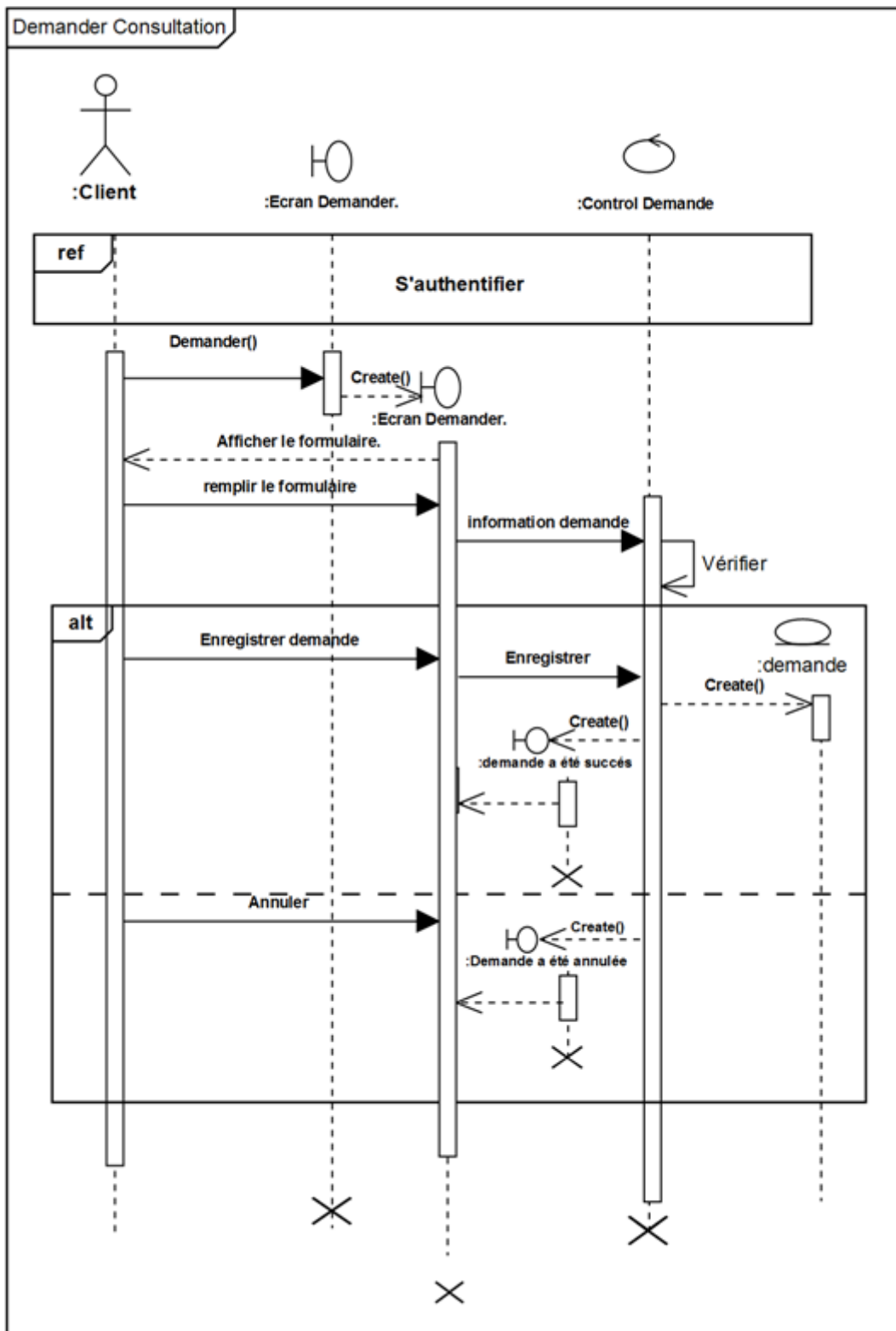


FIGURE 5.17 – Diagrammes de séquence détaillés « Demander Consultation »

- Diagrammes de séquence détaillés « Consulter séance tribunal » :

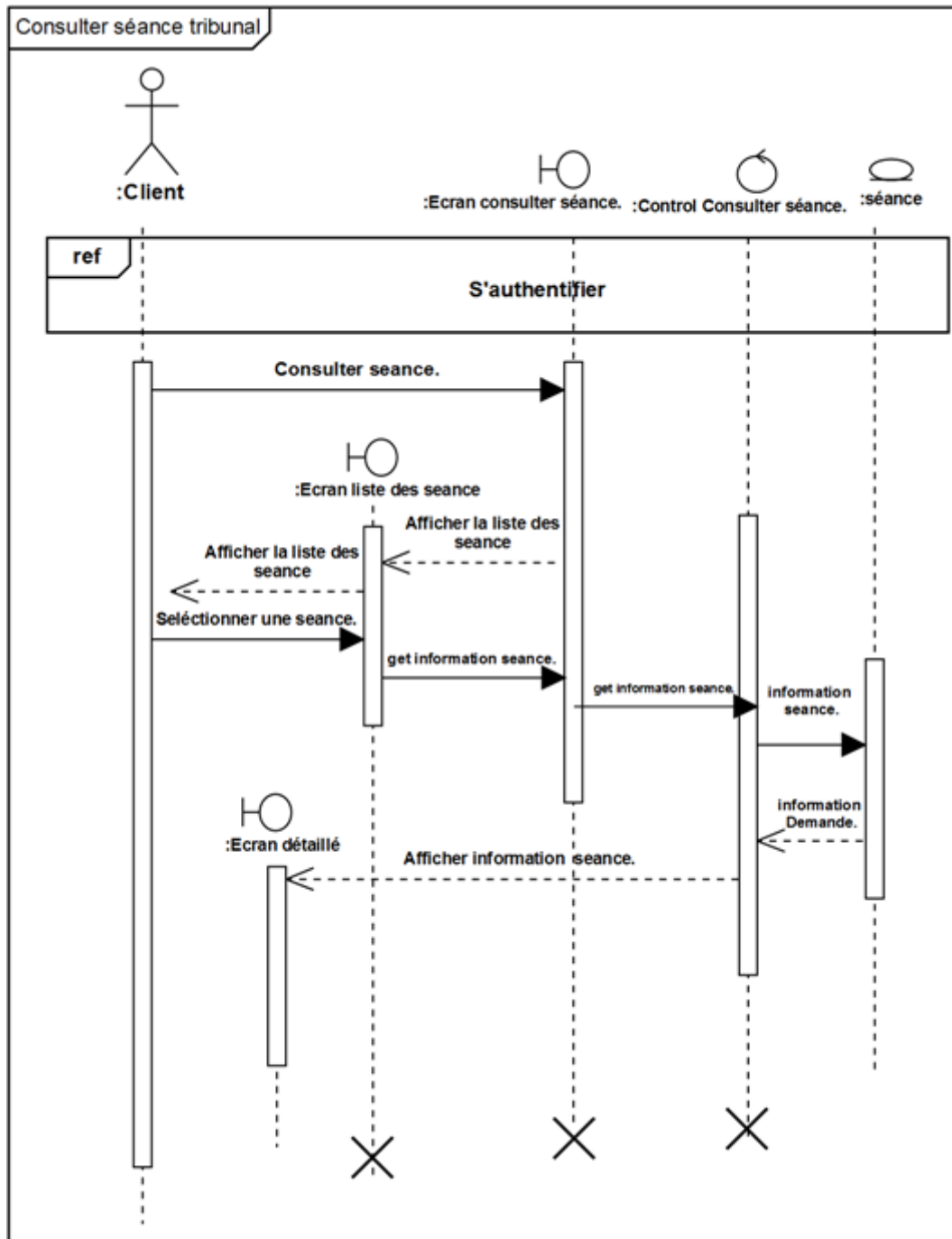


FIGURE 5.18 – Diagrammes de séquence détaillés « Consulter séance tribunal »

5.3 Conception :

La conception est la phase ultime de la modélisation avec UML. Après la modélisation des besoins puis l'organisation de la structure de la solution, la conception vient construire et documenter précisément les classes, les interfaces, et les méthodes qui constituent le codage de la solution. Le noyau de cette étape est de savoir transformer le modèle objet en modèle relationnel. [8]

5.3.1 Dictionnaire de données :

- les classes et les attributs :

Classes	Descriptions	Attributs	Types
Client	Identifiant client Nom client Prénom client Date de naissance client Wilaya de client Email client Téléphone client Fax client	idClient Nom Prénom Datenaissance Wilaya Email Tel Fax	int varchar(50) varchar(50) varchar(50) varchar(50) varchar(50) int varchar(50)
Demande	Identifiant demande Numéro de carte Nom de client Prénom client Description d'affaire Contenu du dossier Etat de demande	iddemande Ncart NomCl prénomCl descript_affaire Contenudossier Etat	int int int int varchar(50) varchar(50) varchar(50)
Procuration	Identifiant procuration Nom du client Prénom du client L'adress du client Le code postale Nom d'avocat Prénom d'avocat L'address de l'avocat Le code postale de l'avocat L'objet Le texte La signature	id Proc NomClient prenomClient Address Code postale Nom avocat prenom avocat Address avocat Code postale avocat Objet Texte Signature	int varchar(50) varchar(50) varchar(50) varchar(50) varchar(50) varchar(50) varchar(50) varchar(50) varchar(50) varchar(50) varchar(50) varchar(50)
Séance	Identifiant séance Num d'affaire Date de séance Heure de séance	id_séance Num_affaire Date Heure	varchar(50) int varchar(50) varchar(50)
Affaire	Identifiant affaire Numéro de l'affaire Thème d'affaire Nature d'affaire	id_affaire Num_affaire Thème_affaire Nature_affaire	int int varchar(50) varchar(50)

	Le tribunaie	tribunal	varchar(50)
	Nom Client	nomCient	varchar(50)
	Prénom Client	prénomClient	varchar(50)
	Date entrer de l'affaire	DateEnteAffaire	varchar(50)
	Date de jugement	Datejuement	varchar(50)
	Contre client	Contreclient	varchar(50)
	Commentaire d'affaire	CommentaireAffaire	varchar(50)
Facture	Identifiant facture	id_facture	int
	Mantant payé	Mantant_payé	int
	Mantant restant	Mantant_restant	int
	Mantant d'avocat	Mantant_d'avocat	int
	Mantant d'affaire	Mantant_d'affaire	int

TABLE 5.1: Dictionnaire de données

5.3.2 Diagramme de classe détaillé :

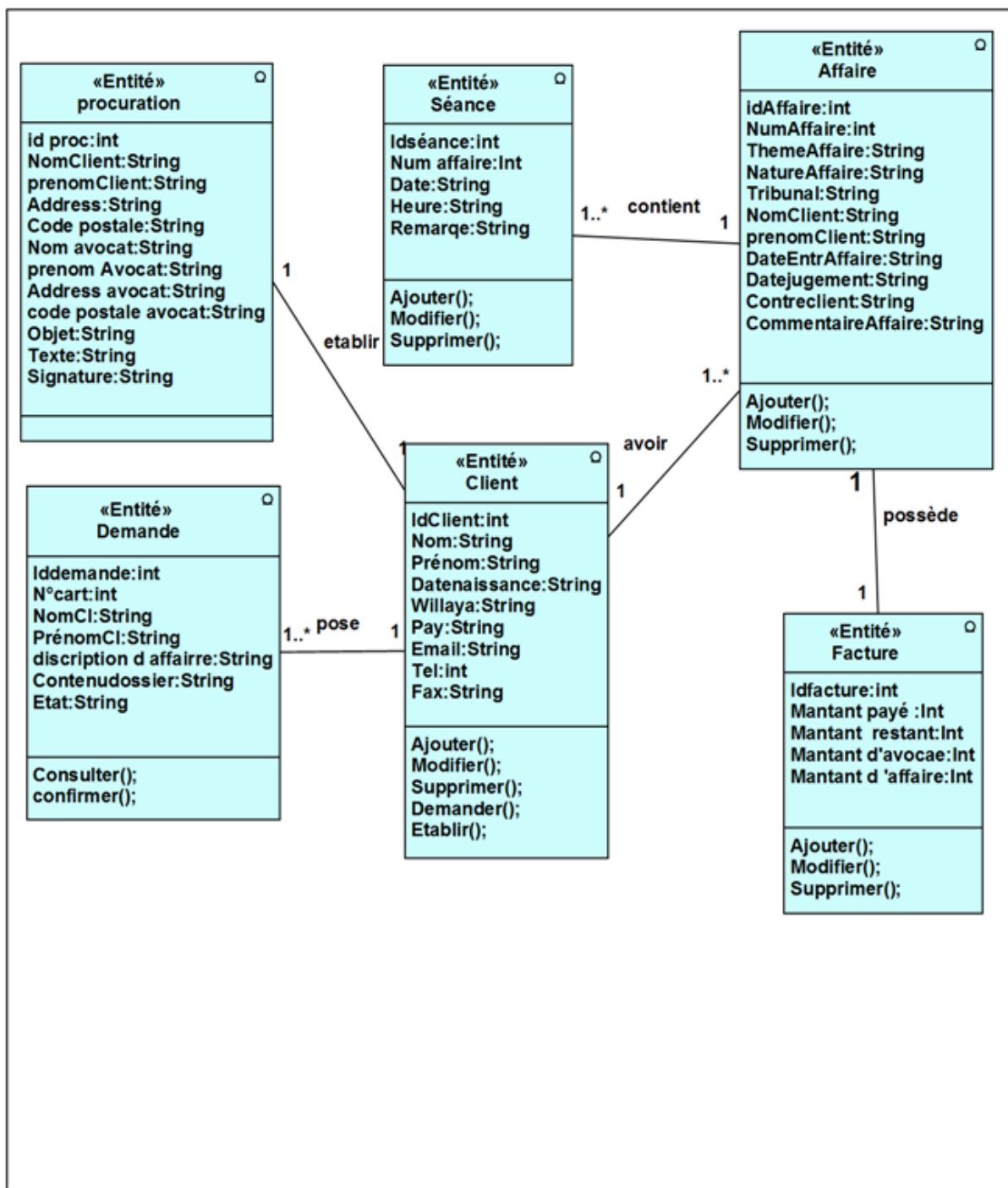


FIGURE 5.19 – Diagramme de classe détaillé

5.3.3 Le passage de modèle objet au modèle relationnel :

L'utilisation d'un SGBDR impose un enchaînement de représentation entre la structure des classes et la structure des données relationnelles, les deux structures ayant des analogies, les équivalences exprimées au tableau sont utilisées pour en réaliser le rapprochement. Une classe définie une structure de données à laquelle souscrivent des instances ; elle correspond donc à une table du modèle relationnel :

Chaque attribut donne lieu à une colonne, chaque instance stocke les données dans une ligne et son ID sert de clé primaire.[11]

Modèle objet	Modèle relationnel
Classe	Table
Attribut de type simple	Colonne
Attribut de type composé	Colonne ou clé étrangère
Instance	T_uplet
ID	Clé primaire
Association	Clé étrangère
héritage	Clé primaire identique sur plusieurs tables

TABLE 5.2: « Équivalence entre les concepts objets et relationnels.

5.3.3.1 Les règles de passage :

La réalisation du modèle relationnel peut être faite à partir de la description conceptuelle. Ce passage de modèle doit respecter un certain nombre de règle.

- **Transformation des classes :**

chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.

- **Transformation des associations :**

Les règles de transformation des associations dépendent des multiplicités maximales des associations nous distinguons trois familles d'associations :

- **Association 1..* :**

il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association l'attribut porte le nom de la clé primaire de la relation père de l'association.

– **Association *.* et classes-associations :**

la classe-association devient une relation la clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association.

– **Association 1-1**

il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un l'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule.[12]

5.3.3.2 Les relations de la base de données :

En se basant sur les règles ci-dessus, nous avons converti les classes entités et leurs associations, à des tables dans la base données.

NB : pour la notation, nous avons choisi de mettre en gras les clés primaires et de mettre # au début de chaque clé étrangère.

Les tables générées sont :

-**Client**(**idClient** , Nom, Prénom,DateNaissance,Wilaya,Pay, Email,Tel,Fax) ;
 -**Demande** (**iddemande**,#idClient,Ncart , NomCl, prénomCl,descriptiond'affaire , contenu dossier , Etat) ;
 -**Procuration** (**id proc**,NomClient,prenomClient ,Address, Code postale ,Nom Avocat,prenom Avocat,Address Avocat,Code postale Avocat, Objet ,Texte,Signature) ;
 -**Séance** (**idséance**,#Num affaire ,Date, Heure, Remarque) ;
 -**Facture** (**idfacture**,Montant payé,Montant restant ,Montant d'avocat,Montant d'affaire) ;
 -**Affaire** (**idAffaire**,NumAffaire,ThemeAffaire ,NatureAffaire,Tribunale NomClient,prenomClient,DateEntrAffaire,Datejujement,Contreclient,commentaire_ Affaire) ;

TABLE 5.3: Les relations de la base de données de système

5.4 Conclusion :

Au cours de ce chapitre, nous avons présenté d'une part l'étape d'analyse qui nous a permis de passer d'une structuration fonctionnelle via les cas d'utilisations et les packages à une structuration objet via les classes et les catégories .

D'autre part nous avons détaillé les différentes classes du système pour passer à la conception de la base de donnée, cette dernière étape est primordiale pour l'élaboration du modèle relationnelle qui définit l'ensemble des tables de la base de données de notre système.

Le chapitre suivant, quant à lui, sera consacré à la phase de développement de notre application.

Réalisation

6.1 Introduction

Après avoir réalisé la conception appropriée à notre application, nous allons dans ce chapitre décrire le processus de réalisation de notre application. Ceci en spécifiant l'environnement de développement, l'implémentation de la base de données et un aperçu sur les interfaces de notre application.

6.2 Environnement de développement

Afin de développer des applications sous Android un ensemble d'outils est nécessaire :

6.2.1 Le langage java

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprises et mobiles. [17]



FIGURE 6.1 – Logo du langage Java

6.2.2 Le langage XML

Le langage XML (eXtended Markup Language) est un format général de documents orienté texte. Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de documents que pour la transmission de données entre applications. La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer et poser le vocabulaire et la syntaxe des données qu'il va contenir. [13]

6.2.3 Le SDK (Software Development Kit)

C'est un kit de développement basé sur le langage Java. Il s'agit des outils que Google a fournis pour interagir avec Android pour la réalisation des applications. Le SDK propose, en plus, un émulateur Android. Ce dernier permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée à une version de l'émulateur, permettant au développeur de tester et voir exactement à quoi ressemblera son application sur un matériel réel. [14]

6.2.4 L'IDE Android studio

Android Studio est un environnement de développement pour des applications android. Il permet principalement d'éditer des fichiers JAVA et les fichiers de configuration d'une application android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolution simultanément. [14]



FIGURE 6.2 – Logo de l'IDE Android Studio

6.3 Les bases de données

Pour implémenter notre base des données, nous avons utilisé l'environnement de création de base de données PHPMyAdmin et le système de gestion de base de données MySQL.

6.3.1 Wamp server

WampServer est une plateforme de développement Web sous Windows. Il permet de développer des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données.



FIGURE 6.3 – Logo du WampServer

6.3.2 Langage de Manipulation de Bases de Données MySQL

MySQL est un Système de Gestion de Bases de Données (SGBD) client/serveur open source qui gère, stocke et manipule une grande quantité de données en les organisant sous forme de tables, il prend aussi en charge les fonctionnalités de protection et de sécurité et fournit un ensemble d'interfaces de programmation (dont une avec PHP) facilitant l'accès aux données. ce système repose sur Le langage SQL qui permet de décrire des tables et d'interagir avec elle (insertion de donnée et demande de donnée). [15]



FIGURE 6.4 – Logo de Langage de Manipulation de Bases de Données MySQL

6.3.3 PHP

PHP (Hypertext Preprocessor) est une plate-forme composée d'un langage de programmation très complet et de nombreux outils pour le développement. Elle s'adapte très rapidement aux technologies émergentes et se voit de plus en plus utilisée dans des développements web dynamiques professionnels et Open Source.

Voici quelques-unes de ses caractéristiques principales :

- Très bon compromis entre fiabilité et rapidité d'exécution ;
- Langage procédural et un langage orienté objet ;
- Outil très complet, doté de nombreuses fonctionnalités, extensions et bibliothèques. [16]

6.3.4 Notepad++

Notepad++ est un éditeur de texte très léger, très puissant et libre (licence GPL). Il est parfait pour programmer avec des langages ne nécessitant pas d'environnement de développement (html, css, javascript, php) ou en ayant un peu pratique (python, processing), ou pour du traitement de données. [16]



FIGURE 6.5 – Logo de Notepad++

6.4 Présentation de quelques interfaces de notre application :

Dans cette partie du mémoire, nous allons présenter les principales fonctionnalités de notre application qui fera par la description de quelques interfaces.

- **La fenêtre « Authentification »**

Au premier lancement de l'application, la première fenêtre qui s'affiche est la fenêtre « Authentification » suivante :

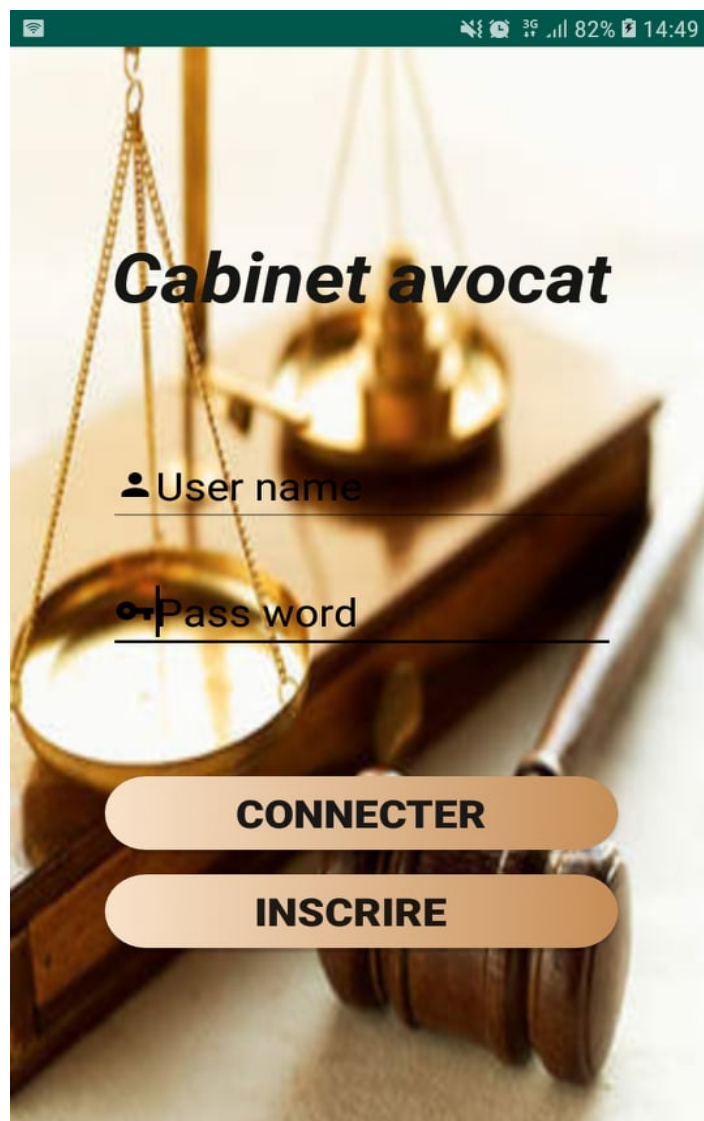
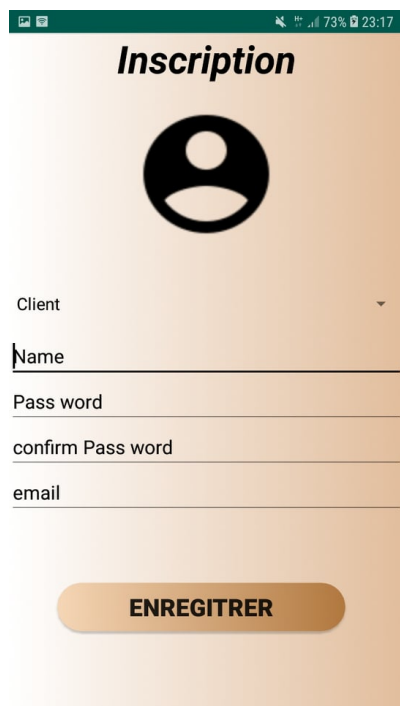


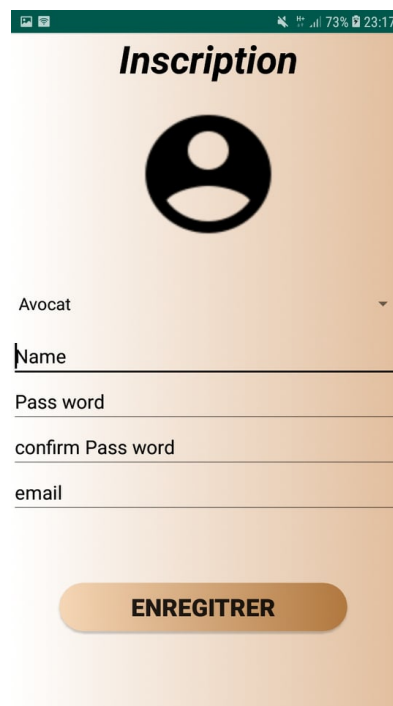
FIGURE 6.6 – Fenêtre d'authentification

Dans le cas de la première utilisation de l'application, l'utilisateur doit inscrire « créer un nouveau compte ». Après avoir taper sur ce dernier, l'application demande à l'utilisateur de choisir le type d'utilisateur (Avocat/Client).dans l interface suivant



The screenshot shows a mobile application interface for registration. At the top, the title "Inscription" is displayed in bold black text. Below the title is a large black circular icon representing a person. Underneath the icon is a dropdown menu currently set to "Client". Below the dropdown are five input fields: "Name", "Pass word", "confirm Pass word", and "email". At the bottom of the form is a rounded rectangular button with the text "ENREGISTRER" in bold black capital letters. The background is a light beige color.

FIGURE 6.7 – Fenêtre d’inscription



The screenshot shows a mobile application interface for registration, similar to Figure 6.7. The title "Inscription" is at the top. Below it is a large black circular icon representing a person. Underneath the icon is a dropdown menu currently set to "Avocat". Below the dropdown are five input fields: "Name", "Pass word", "confirm Pass word", and "email". At the bottom of the form is a rounded rectangular button with the text "ENREGISTRER" in bold black capital letters. The background is a light beige color.

FIGURE 6.8 – Fenêtre d’inscription

En suite, il valide et remplit les champs demandés, puis cliquer sur enregistrer. Si l'utilisateur possède déjà un compte, il doit saisir correctement son identifiant et son mot de passe et cliquer sur « se connecter » pour accéder à son compte (Fenêtre Accueil Avocat/Client).

- **Accueil « Avocat »**

Après avoir saisis le login et le mot de passe correctement et cliquer sur « se connecter », l'application vérifie dans la base de données quel type d'utilisateur (Avocat/Client) est connecté, puis une autre fenêtre apparaît (fenêtre Accueil). Cette dernière donne la possibilité au avocat de faire consulter demande, gere client..

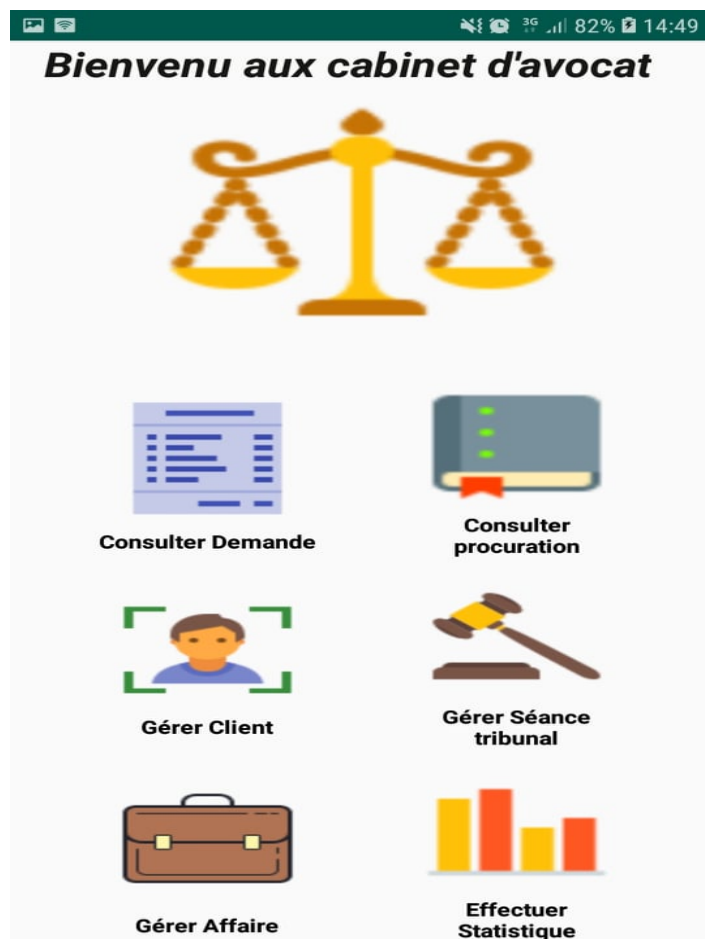


FIGURE 6.9 – Accueil Avocat

- Accueil « Client »

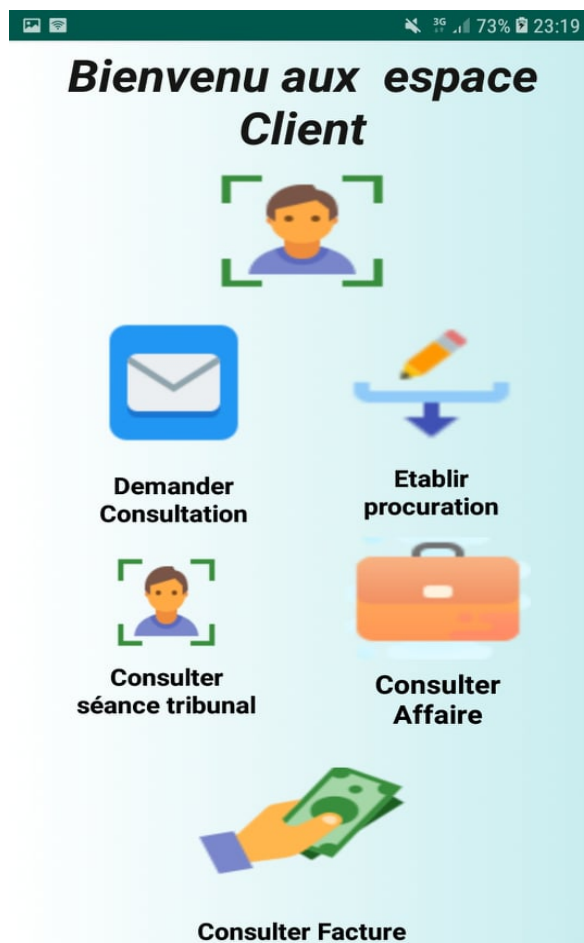


FIGURE 6.10 – Accueil Client

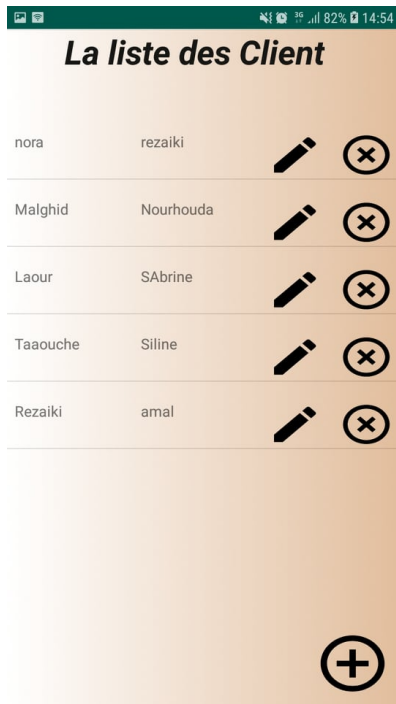


FIGURE 6.11 – Fenêtre liste des client

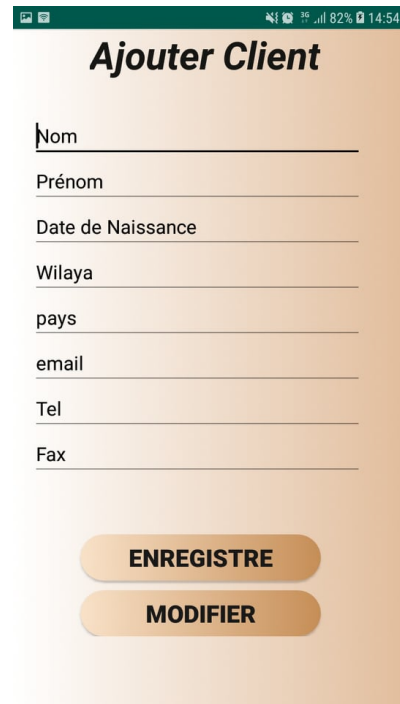


FIGURE 6.12 – Fenêtre Ajouter client

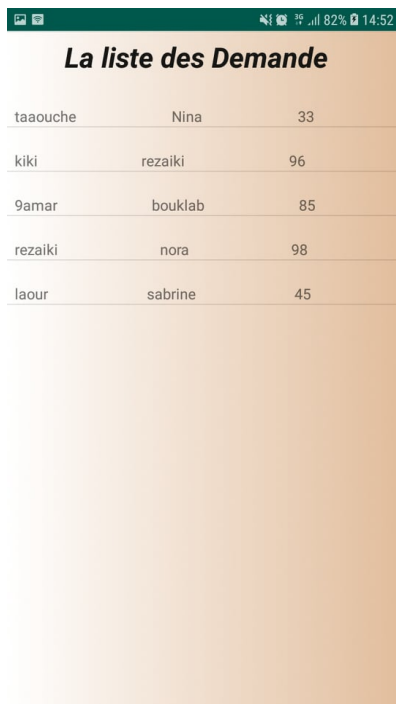


FIGURE 6.13 – Fenêtre liste des demande client



FIGURE 6.14 – Fenêtre discription d'affaire



FIGURE 6.15 – Fenêtre Ajouter séance tribunaie



FIGURE 6.16 – Fenêtre Modifier séance tribunaie

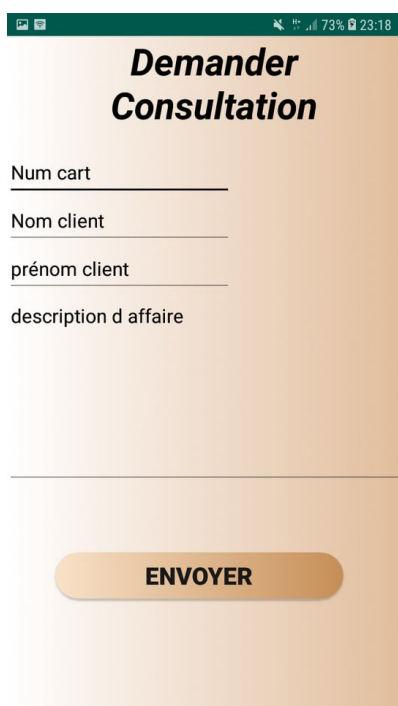


FIGURE 6.17 – Fenêtre demander consultation

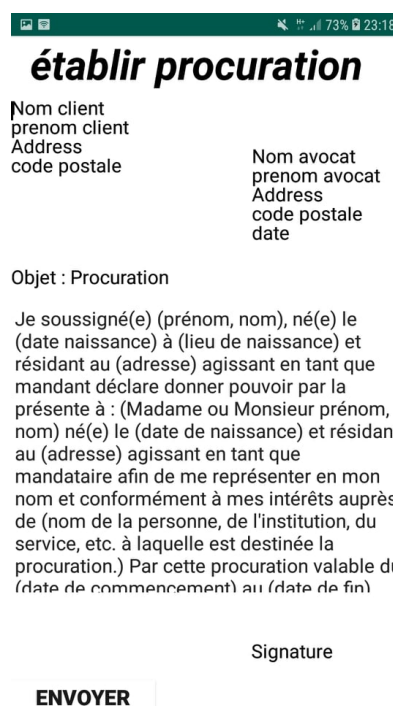


FIGURE 6.18 – Fenêtre établir procuration

6.5 Conclusion

La phase de réalisation est l'étape la plus délicate dans le développement de notre application. Dans ce chapitre, nous avons présenté les aspects pratiques liés à la réalisation de notre projet, à savoir les outils de développement nécessaire. En dernier, nous avons illustré quelques interfaces que comprend notre application.

Conclusion Générale

Conclusion générale

Dans le tout de cette étude nous avons abordé le développement d'une application mobile pour la gestion d'un cabinet d'avocat ; de deux point de vue modélisation et réalisation, avec le processus 2TUP.

Sur le plan modélisation, nous sommes basés sur le processus 2TUP « 2Track Unified Process ». La conception a été illustrée par les différents diagrammes UML2.0, qui nous ont permis de spécifier et documenter le système, notamment le diagramme des cas d'utilisation permet de capturer et représenter les besoins, le diagramme de classe a permis de représenter l'aspect statique du système. Et enfin, les diagrammes de séquence sont utilisés dans les phases de capture de besoins fonctionnels et d'analyse.

Sur le plan réalisation, nous avons choisi le langage Java sous la plateforme Android avec MYSQL pour la gestion de la base de données. En outre, ce projet nous a permis d'enrichir nos connaissances dans des domaines variés comme : L'orienté objet, UML, 2TUP, ...

A travers cette étude nous avons pu mettre en pratique et développer les connaissances que nous avons acquises durant notre cursus universitaire, et en particulier dans les systèmes d'information, les bases de données et le génie logiciel.

Cependant, nous avons senti une grosse pression vue le temps qui est relativement court, donc nous n'avons pas pu réaliser tout ce qu'on a espérer faire. Cependant des perspectives d'amélioration de notre application restent envisageables telles que :

- Enrichir notre application par l'envoi automatique des dossier.
- Migrer notre base de donné d'une base de donné centralisé à une base de donné Mobile.

Bibliographie

- [1] **Samuel, Pierre**. Réseaux et systèmes informatiques mobiles : Fondements, architectures et applications[en ligne].Montréal :Edition revue et augmentée, 2003 , 601 Format.
- [2] -tech- smartphone[en ligne]. disponible sur <<https://www.futura-sciences.com/tech/definitions/smartphone-smartphone-1954/>>. (Consulté le 15/02/2019).
- [3] **B.Bathelot**.Définition : Application mobile [en ligne].(MAJ le 8 Mar 2017)disponible sur :< <https://www.definitions-marketing.com/definition/application-mobile/>>.
- [4] **Ferhat Yasser, Laouar Djaafar**.Réalisation d'une application mobile de gestion de la vie estudiantine pour les Smartphones Android. STIC.Mila :Centre Universitaire 'Abedlhafid boudouf 'de Mila,2013,158p.
- [5] **Damien Guignard, Julien Chable, Emmanuel Robles** :Programmation Android : De la conception au déploiement, avec le SDK Google Android 2, Eyrolles 2011.
- [6] **Dewas,Bilal**.Développement d'application mobile :Institu des sciences et technologie –licence d'informatique,2017,70p.
- [7] **Joseph Gabant,Dunod** .Merise et uml pour la modélisation,mars 2004.
- [8] **pascal Roques et Franck Vallée** .UML en action de l'analyse des besoins a la conception en java,2000.
- [9] **pascal Roques et Franck Vallée** .UML en action De l'analyse des besoins à la conception 4 éme édition.

- [10] **pascal Roques et Franck Vallée** . UML en action de l'analyse des besoins à la conception J2EE
- [11] **jaques guyot** . de UML à SQL, Edition système et information.
- [12] **pascal Roques** . UML en action, Eyrolles 2004
- [13] **CARTON O** .L'essentiel de XML, cours de M2 Pro à l'Université Paris Diderot.
- [14] [http ://www.memoireonline.com/03/12/5548/Rapport-de-stage-sur-le-projet-Locate-mycar-google-map-android.html](http://www.memoireonline.com/03/12/5548/Rapport-de-stage-sur-le-projet-Locate-mycar-google-map-android.html)
- [15] **Philippe Rigaux** .Pratique de MySQL et PHP Conception et réalisation de sites web dynamiques", DUNOD, 4ème édition, 2009.
- [16] **Guillaume Ponçon** .Best practices PHP5", EYROLLES, 2005.
- [17] **J. M. Doudoux**. Developper en java avec Eclipse version 0.80.007. 2011.
- [18] [https ://www.renson-avocats.be/role-et-les-missions-avocat.html](https://www.renson-avocats.be/role-et-les-missions-avocat.html).
- [19] [https ://www.dictionnaire-juridique.com](https://www.dictionnaire-juridique.com).
- [20] xtrait du Journal Officiel de la République Algérienne / N 67 du 24 octobre 2004).