

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane MIRA de Bejaia

Faculté des Sciences Exactes



Mémoire de fin de cycle
En vue d'obtenir le diplôme de master professionnel en Génie
Logiciel

Conception et réalisation d'une application web pour la gestion des Notifications et Ordonnancement

Cas d'étude : Impôts de Bejaia

Réaliser Par

BOUANANI Yacine
FORTAS El Aziz

Devant le jury composé de :

Président : ATMANI Mouloud
Examinatrice : ZIDANI Ferroudja
Promotrice : BACHIRI Lina

Promotion : 2019/2020

Remercîment

En préambule à ce mémoire, nous remercions Allah qui nous a doté d'une grande volonté et qui nous a aidé et nous a donné le courage, la force et la patience d'accomplir ce travail.

Ce mémoire n'aurait jamais pu voir le jour sans le soutien actif d'un certain nombre de personnes que nous tenons à remercier, toutes celles et ceux qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent ici l'expression de nos remerciements les plus chaleureux :

À nos chers parents pour tous les sacrifices consentis à notre égard et leur énorme soutien.

À toutes nos familles et nos proches amis (es) qui par leurs prières et leurs encouragements ont pu surmonter tous les obstacles.

À notre Encadrante, en l'occurrence Mme BACHIRI Lina pour l'attention qu'elle nous a accordé et pour nous avoir orientés dans le bon sens quant à l'élaboration de ce projet.

Aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.



Dédicace

On a le plaisir de dédier ce modeste travail :

A nos chers parents qui nous ont soutenus durant la réalisation de ce projet et pour tout le mal qu'ils se sont donnés afin de nous faciliter la tâche, en témoignage de la profonde affection qu'on leurs porte.

A nos frères et sœurs à qui nous souhaitons un avenir prospère.

A toutes nos familles.

A nos chers amis(es) pour leurs présences et encouragements, qu'ils trouvent là toute notre reconnaissance.

Table des matières

Table des matières	I
Liste des figures	IV
Liste des tableaux	VI
Liste d'abréviations	VII
Introduction générale.....	1

Chapitre I: Technologies web

I.1 Introduction.....	4
I.2 Technologies du web.....	4
I.2.1 Site web.....	4
I.2.2 Architecture des applications web.....	4
I.2.3 Application Client/serveur.....	6
I.2.4 Serveur web	6
I.2.5 Fonctionnement d'une application web.....	6
I.2.6 Serveur d'application et la base de données	7
I.2.7 Hébergement d'une application web	9
I.2.8 Méthodes d'analyse et de conception	9
I.2.9 Le choix de la méthode	11
I.2.10 Sécurité des applications Web	11
I.3 Conclusion	12

Chapitre II : Étude de l'existant et spécification des besoins

II.1 Introduction	14
II.2 Etude de l'existant	14
II.2.1 Direction des impôts.....	14
II.2.2 Activités	16
II.2.3 Etude des fiches	16
II.2.4 Etude des postes de travail.....	18
II.3 Critique de l'existant	18
II.3.1 Problématique	18
II.3.2 Solution proposer	19

<i>II.4 Expression des besoins</i>	19
<i>II.4.1 Besoins fonctionnels</i>	19
<i>II.4.2 Besoins non fonctionnels</i>	20
<i>II.5 Modélisation des besoins du système</i>	20
<i>II.5.1 Description du contexte de l'application</i>	20
<i>II.5.2 Identification des cas d'utilisation</i>	22
<i>II.5.3 Description textuelle des cas d'utilisation</i>	23
<i>II.6 Conclusion :</i>	32

Chapitre III: Conception

<i>III.1 Introduction</i>	34
<i>III.2 Réalisation des diagrammes de séquence</i>	34
<i>III.2.1 Définition</i>	34
<i>III.2.2 Diagramme de séquence de cas « authentification »</i>	34
<i>III.2.3 Diagramme de séquence de cas « gérer les message »</i>	35
<i>III.2.4 Diagramme de séquence de cas « gérer les fiches »</i>	37
<i>III.2.5 Diagramme de séquence de cas « gérer les utilisateurs »</i>	38
<i>III.2.6 Diagramme de séquence de cas « gérer réclamation préalable »</i>	42
<i>III.2.7 Diagramme de séquence de cas « gérer recours »</i>	43
<i>III.3 Rédaction du dictionnaire des données</i>	44
<i>III.4 Réalisation des diagrammes de classes</i>	48
<i>III.4.1 Règles de gestion entre les classes</i>	48
<i>III.4.2 Principes et définitions de base</i>	48
<i>III.4.3 Diagramme de classe</i>	49
<i>III.5 Modèle relationnel</i>	50
<i>III.5.1 Règles de passage au modèle relationnel</i>	50
<i>III.5.2 Modèle relationnel pour notre cas</i>	51
<i>III.6 Conclusion</i>	51

Chapitre IV: Réalisation

<i>IV.1 Introduction</i>	53
<i>IV.2 Outils et environnement de développement</i>	53
<i>IV.2.1 Code Visual Studio</i>	53
<i>IV. 2.2 WampServer</i>	53

<i>IV. 2.3 Système d'exploitation Windows</i>	53
<i>IV.2.4 Serveur apache</i>	54
<i>IV.2.5 MySQL</i>	54
<i>IV.2.6 JavaScript</i>	54
<i>IV.2.7 PHP</i>	55
<i>IV.2.8 CSS3</i>	55
<i>IV.2.9 HTML</i>	56
<i>IV.2.10 Visual Paradigm pour uml</i>	56
<i>IV.2.11 Bootstrap3</i>	56
<i>IV.3 Organigramme de l'application</i>	57
<i>IV.3 .1 Coté administrateur</i>	57
<i>IV.3.2 Coté Simple Employé</i>	57
<i>IV.3.3 Coté contribuable</i>	58
<i>IV.4 Description des interfaces de l'application</i>	59
<i>IV.4.1 Espace Contribuable</i>	60
<i>IV.4.2 Espace Admin, utilisateur (simple employé)</i>	64
<i>IV.5 Conclusion</i>	69
<i>Conclusion générale</i>	71
<i>Webographie</i>	72
<i>Bibliographie</i>	74

Liste des figures

Figure I.1:Architecture 3-tiers d'une application web	5
Figure I.2:Processus de traitement des pages web dynamique	7
Figure I.3:Processus d'interrogation de la base de données	8
Figure I.4: Cycle de vie du processus unifié	10
Figure II.1: Organigramme de la direction de la wilaya de Bejaia	15
Figure II.2: Organigramme de la sous-direction du contentieux.	16
Figure II.3: Diagramme de contexte.	22
Figure II.4: Diagramme de cas d'utilisation « Authentification ».	24
Figure II.5: Diagramme de cas d'utilisation « Gérer les utilisateurs ».	25
Figure II.6: Diagramme de cas d'utilisation « afficher les statistiques ».	26
Figure II.7: Diagramme de cas d'utilisation « gérer les fiches des contribuables ».	27
Figure II.8: Diagramme de cas d'utilisation « gérer les messages ».	28
Figure II.9: Diagramme de cas d'utilisation « gérer réclamation préalable ».	29
Figure II.10: Diagramme de cas d'utilisation « gérer recours ».	30
Figure II.11: Diagramme global des cas d'utilisations.	31
Figure III.1: Diagramme de séquence de cas d'utilisation « Authentification ».	35
Figure III.2: Diagramme de séquence de cas d'utilisation « ajouter message ».	36
Figure III.3: Diagramme de séquence de cas d'utilisation « supprimer message ».	37
Figure III.4: Diagramme de séquence de cas d'utilisation « ajouter une feuille d'instruction».	38
Figure III.5: Diagramme de séquence de cas d'utilisation « ajouter utilisateur ».	39
Figure III.6: Diagramme de séquence de cas d'utilisation « rechercher utilisateur ».	40
Figure III.7: Diagramme de séquence de cas d'utilisation « supprimer utilisateur ».	41
Figure III.8: Diagramme de séquence de cas d'utilisation « modifier utilisateur ».	42
Figure III.9: Diagramme de séquence de cas d'utilisation « ajouter une réclamation ».	43
Figure III.10: Diagramme de séquence de cas d'utilisation « ajouter recours ».	44
Figure III.11: Diagramme de classe.	50
Figure IV.1: Logo du Code Visual Studio.	53
Figure IV.2 : Logo du wampServer.	53
Figure IV.3: Logo du système exploitation Windows.	54
Figure IV.4: Logo du Serveur apache.	54
Figure IV.5: Logo du MySQL.....	54
Figure IV.6: Logo du langage JavaScript.....	55
Figure IV.7: Logo du langage PHP.	55
Figure IV.8: Logo du langage CSS3.	55
Figure IV.9: Logo du langage Html.	56
Figure IV.10: Logo du logiciel Visual Paradigm.	56
Figure IV.11: Logo du langage Bootstrap.....	57
Figure IV.12: Organigramme de l'application coté administrateur	57

Figure IV.13 : Organigramme de l'application coté simple employé.....	58
Figure IV.14: Organigramme de l'application coté contribuable	59
Figure IV.15: Page authentification.	60
Figure IV.16: Ajouter contribuable.....	61
Figure IV.17: Page d'accueil coté contribuable.	61
Figure IV.18: Interface réclamation.	62
Figure IV.19: Interface messagerie.	63
Figure IV.20: Interface ajouter message.	63
Figure IV.21: Interface authentification Employés.....	64
Figure IV.22: Interface accueil « coté administrateur ».....	65
Figure IV.23: Page d'accueil simple employé.	65
Figure IV.24: Interface statistiques.	66
Figure IV.25: Interface nombre de contribuables.	66
Figure IV.26: Interface feuille d'instruction.	67
Figure IV.27: Interface de dégrèvement (admission totale).....	68
Figure IV.28: Interface messagerie.....	69

Liste des tableaux

Tableau II.1: Légende des messages des diagrammes de contexte dynamique.	21
Tableau III.2: Dictionnaire des données.	47

Liste d'abréviations

BDD : Base De Données.

CSS : Cascading Style Sheet.

HTTP : HyperText Transfer Protocol.

HTML : HyperText Markup Language.

MySql: My Structured Query Language.

PHP : Personal Home Page.

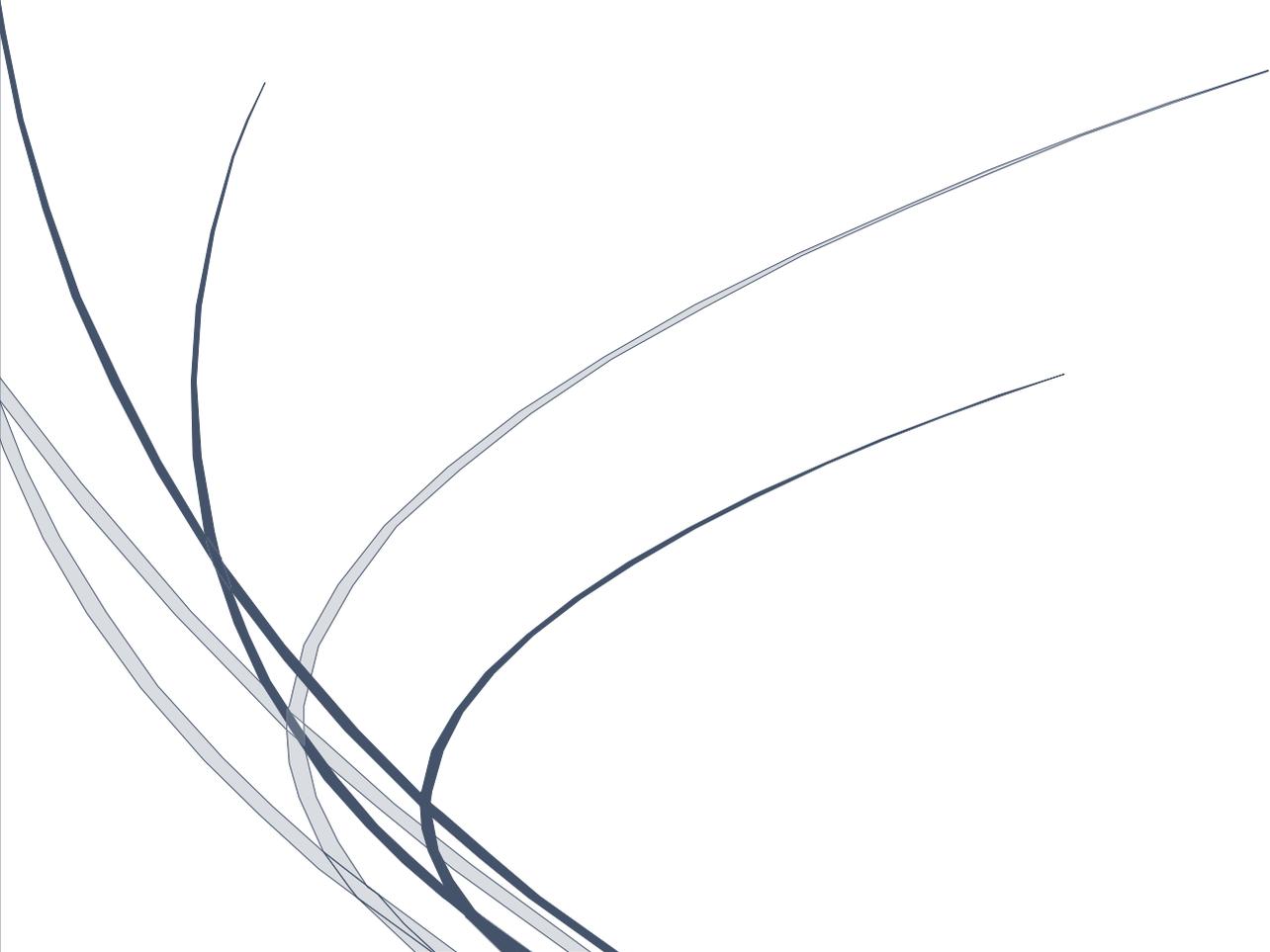
SGBD : System de Gestion de Base de Données.

SQL : Structured Query Language.

UML: Unified Modeling language.



Introduction Générale



Introduction générale

Dans un monde de plus en plus moderne, l'informatique devient presque indispensable dans tous les domaines de vie à savoir, la science, l'industrie, l'administration, le commerce...

Aujourd'hui, plusieurs sociétés essaient de profiter au maximum possible de l'informatique afin d'améliorer leurs productivités et de faire face à quelques problèmes pénibles qui peuvent constituer un obstacle de progression. En effet, la présence des applications Web est devenue indispensable en société, car c'est une bonne façon d'innover et de gérer ses activités.

Notre principal objectif dans ce projet de fin de cycle est la conception et la réalisation d'une application web pour la direction des impôts de Bejaia (plus précisément pour la sous-direction du contentieux) permettant de gérer ce service du contentieux de la manière très facile. Ceci pour répondre à un ensemble de besoins notamment : la communication en ligne entre les impôts et les contribuables (réclamation, recours), gestion des utilisateurs et les statistiques.

Ce travail est divisé en quatre chapitres :

Le premier chapitre, intitulé « Technologies web », sera consacré à la présentation des technologies web où seront décrits le fonctionnement, l'architecture et la sécurité des applications web.

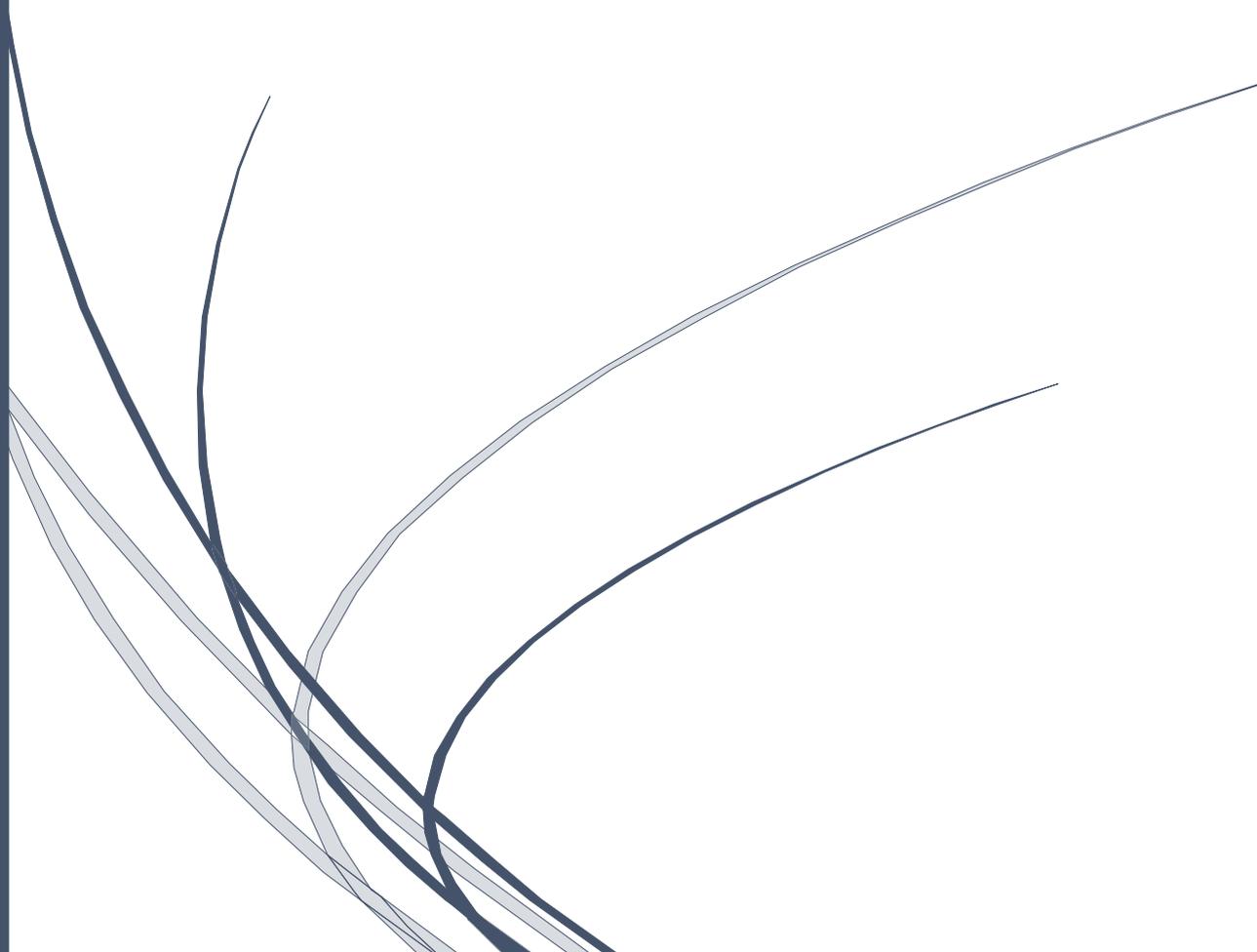
Le deuxième chapitre, intitulé « Étude de l'existant et spécification des besoins », sera consacré à l'étude de l'existant, L'identification des besoins fonctionnels et non fonctionnels auxquels doit répondre notre application, ainsi que l'identification des acteurs et leur cas d'utilisation. Dans ce chapitre, les messages échangés entre les acteurs et le système seront modélisés par un diagramme de contexte, quant aux cas d'utilisations seront décrit textuellement avant d'être modélisés avec un diagramme de cas d'utilisations.

Le troisième chapitre, intitulé « Conception », portera sur le fonctionnement du futur système en répondant aux besoins identifiés dans le chapitre précédent. Il décrira les cas d'utilisation sous forme de diagramme de séquence détaillé. Un diagramme de classes sera aussi présenté pour modéliser notre application. Ce diagramme sera transformé par la suite en modèle relationnel de données.

Dans le dernier chapitre, intitulé « réalisation », nous présenterons les outils et l'environnement de développement et les langages utilisés, nous passerons par la suite à la présentation de l'organigramme d'interaction et les différentes interfaces de notre application.

Nous terminerons ce mémoire par une conclusion générale, pour résumer l'essentiel du travail accompli.

Chapitre I : Technologies web



I.1 Introduction

Le développement des applications web présentent certaines particularités, au niveau technique et ergonomique. Cette spécificité nous oblige, au moment de la conception, à préconiser des méthodes de conception et des méthodes de travail dédiées à ce genre d'applications.

Ce chapitre est consacré pour définir quelques généralités sur les technologies web.

I.2 Technologies du web

Une application web est un ensemble de pages qui interagissent avec les utilisateurs, les unes avec les autres, ainsi qu'avec les différentes ressources d'un serveur web, notamment les bases de données [B1].

I.2.1 Site web

Un site web est composé d'un ensemble de documents structurés, nommés pages web, stockés (hébergés) sur un ordinateur (serveur) connecté au réseau mondial [B1].

Il existe deux types de sites web :

Un site statique : constitué des pages web avec du contenu fixe. Chaque page est codée en HTML (HyperText Markup Language) et affiche les mêmes informations à chaque visiteur. La modification de ces pages se fait par une intervention de l'administrateur du site web [B1].

Un site dynamique : les pages du site qui le compose peuvent être amodiables par les visiteurs.

De plus, entre deux visites sur le même site, le contenu de la page peut être différent sans action de l'administrateur du site Internet. Le serveur qui fait fonctionner le site utilise une technologie de Scripting (comme PHP : Hypertext Preprocessor) ainsi qu'une base de données comme MySQL (My Structured Query Language) [B1].

I.2.2 Architecture des applications web

Une architecture est un modèle générique et conceptuel qui se rapporte à un sujet et qui représente la fonctionnalité, la structure, le positionnement, l'interrelation des différents types d'éléments (hardware, logiciels, infrastructure) qui la composent [W6].

En règle générale, une application est découpée en 3 niveaux (couches) d'abstraction [B2] :

La couche présentation (Vue): La couche présentation est chargée du traitement de l'interaction avec l'utilisateur. C'est un rôle d'affichage et d'interaction.

La couche métier (Contrôleur) : La couche métier effectue les traitements applicatifs. Elle effectue de plus le tampon entre la présentation et les données. Elle effectue aussi les règles de gestion de l'application.

La couche accès aux données (Model) : La partie donnée stocke les données pérennes de l'entreprise ou de l'application.

Différentes architectures pour une application web :

Architecture 1- tiers : La plus simple de l'architecture de base de données est un niveau où le client, le serveur et la base de données résident tous sur la même machine. Chaque fois que vous installez une base de données dans votre système et y accédez pour vous exercer aux requêtes SQL, il s'agit d'une architecture à 1 niveau. Mais une telle architecture est rarement utilisée en production

Architecture 2- tiers : Une architecture à deux niveaux est une architecture de base de données où la couche de présentation s'exécute sur un client (PC, mobile, tablette, etc.).

Les données sont stockées sur un serveur.

Architecture 3- tiers : Est une extension de l'architecture à 2 niveaux. L'architecture à 3 niveaux comprend les couches suivantes :

- Couche de présentation (votre PC, tablette, mobile, etc.).
- Couche d'application (serveur).
- Serveur de base de données.

Dans notre application nous avons utilisé une architecture à 3-tiers qui est illustrée dans la figure (I.1) ci-dessous :

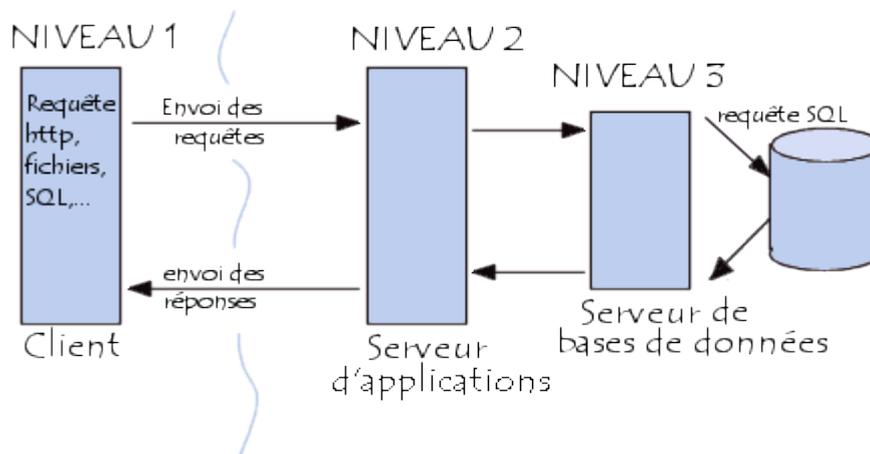


Figure I.1: Architecture 3-tiers d'une application web [W4].

I.2.3 Application Client/serveur

Une application web est dite basée sur un modèle Client - Serveur si la base de données se trouve sur un serveur qui ne sert qu'à ça. Pour interagir avec cette base de données, il faut

utiliser un logiciel "client" qui va interroger le serveur et transmettre la réponse que le serveur lui aura donnée [B2].

Le serveur peut être installé sur une machine différente du client.

Par conséquent, lorsque nous installons un SGBD (système de gestion de base des données), nous installons en réalité deux choses :

- Le serveur
- Le client

Chaque requête (insertion/modification/lecture de données) est faite par l'intermédiaire du client. Nous pouvons donc avoir besoin d'un langage pour discuter avec le client, pour lui donner les requêtes que nous souhaitons effectuer.

Dans le cas de notre application, nous avons MySQL comme système de gestion des bases de données, et le SQL (Structured Query Language) comme langage de système de gestion des bases de données.

I.2.4 Serveur web

Un serveur Web est un programme qui utilise le protocole HTTP pour fournir les fichiers qui constituent les pages Web que les utilisateurs ont demandés via des requêtes transmises par les clients HTTP de leurs ordinateurs. Des ordinateurs et des appliances dédiés peuvent également jouer le rôle des serveurs Web [W5].

I.2.5 Fonctionnement d'une application web

Lorsqu'un serveur web reçoit une requête de page web statique, il transmet simplement cette page au navigateur requérant. En revanche, lorsque le serveur web reçoit une requête de page dynamique, il transmet cette page à une extension logicielle spéciale chargée d'achever la page. Ce logiciel spécial est appelé serveur d'application.

Le serveur d'application lit le code de la page, termine cette page en fonction des instructions figurantes dans le code, puis en retire le code.

Il en résulte une page statique que le serveur d'application renvoie au serveur web, lequel transmet alors cette page au navigateur requérant.

Le navigateur reçoit uniquement du code HTML pur lorsque la page lui est transmise.

Un serveur d'application nous permet de travailler avec des ressources côté serveur telles que les bases de données. Une page dynamique peut, par exemple, ordonner au serveur d'application d'extraire des données de la base de données et de les insérer dans le code HTML de la page.

L'instruction d'extraction des données de la base est nommée requête de base de données. Une requête est composée de critères de recherches rédigés dans un langage de base de données appelé SQL. La requête SQL est rédigée dans les scripts ou les balises côté serveur de la page [B2].

Toutes ces actions sont résumées dans le schéma présenté dans la figure (I.2) sise ci-dessous :

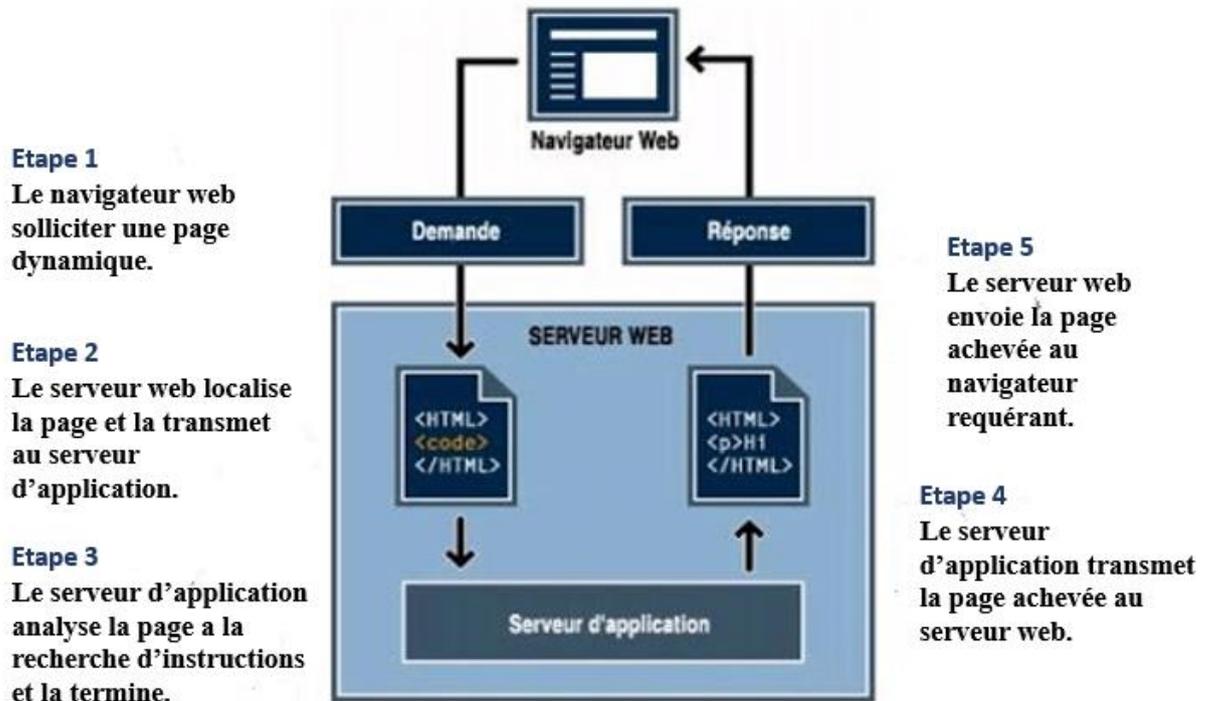


Figure I.2 : processus de traitement des pages web dynamique [B2].

I.2.6 Serveur d'application et la base de données

Un serveur d'application nous permet de travailler avec des ressources côté serveur telles que les bases de données. Une page dynamique peut, par exemple, ordonner au serveur d'application d'extraire des données de la base de données et de les insérer dans le code HTML de la page.

L'instruction d'extraction des données de la base est nommée requête de base de données. Une requête est composée de critères de recherches rédigés dans un langage de base de données appelé SQL. La requête SQL est rédigée dans les scripts ou les balises côté serveur de la page.

Un serveur d'application ne peut pas communiquer directement avec une base de données car le format propriétaire de cette dernière rend les données indéchiffrables, de la même manière qu'un document Microsoft Word ouvert dans Bloc-notes est indéchiffrable. Le pilote de base de données constitue l'unique moyen de communication entre le serveur d'application et la base de données, ce logiciel agit comme un interprète entre le serveur d'application et la base de données.

Une fois que le pilote a établi la communication, la requête est exécutée par rapport à la base de données et un jeu d'enregistrements est créé.

Un jeu d'enregistrements est un ensemble de données extraites d'une ou de plusieurs tables de base de données. Le jeu d'enregistrements est renvoyé au serveur d'application et les données sont utilisées pour compléter la page dynamique. Voici un exemple simple de requête de base de données rédigée en SQL :

```
SELECT last name, first name, fit points from employees.
```

Cette instruction crée un jeu d'enregistrements à trois colonnes et le remplit de lignes comportant le nom, le prénom et les points de mise en forme de tous les employés de la base de données. Voici une illustration du processus d'interrogation de la base de données via des requêtes et de la transmission des données au navigateur [B2].

Toutes ces actions sont résumées dans le schéma présenté dans la figure (I.3) sise ci-dessous :

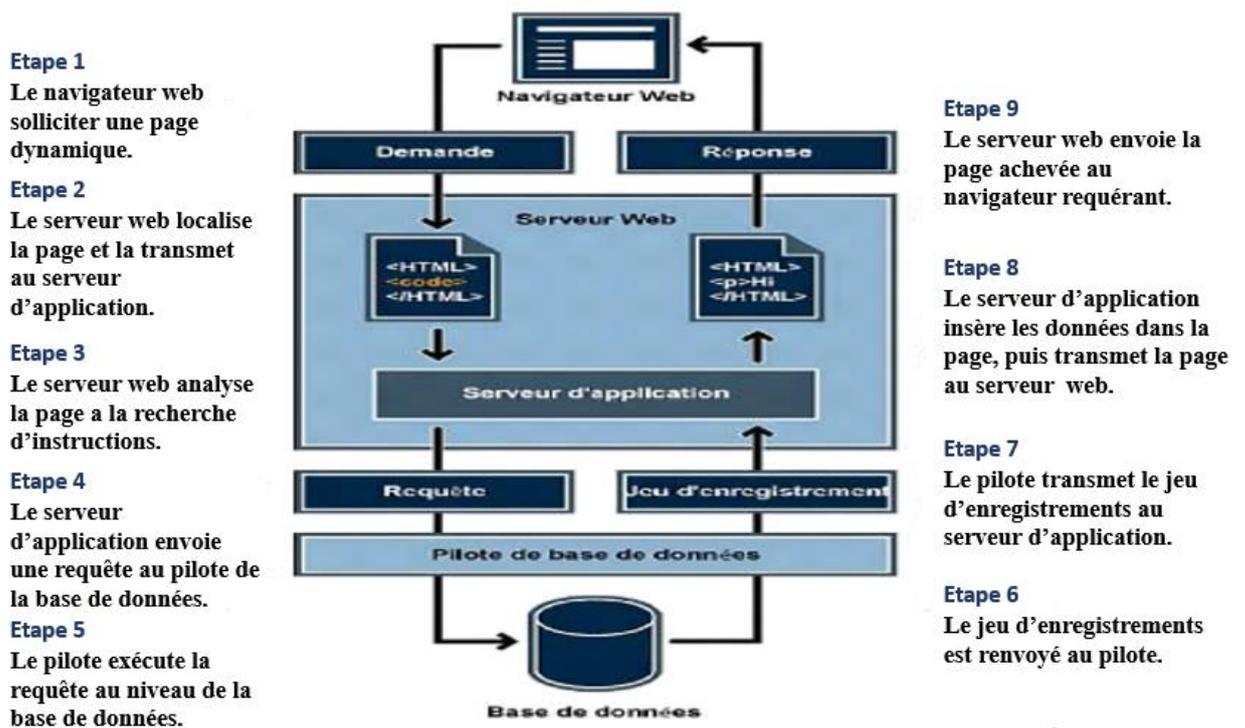


Figure I.3 : processus d'interrogation de la base de données [B2].

Généralement et après la réalisation et l'implémentation d'une application, une dernière étape reste à accomplir c'est l'hébergement de l'application afin qu'elle devienne accessible et manipulable par les utilisateurs.

I.2.7 Hébergement d'une application web

Définition :

L'hébergement est un service visant à rendre un site ou une application web accessible sur internet. Afin que les pages soient visibles par tout le monde, il faut qu'elles soient stockées sur un serveur [W7].

Un hébergeur web est un prestataire de service qui met à votre disposition un espace de stockage sur un serveur connecté en permanence à Internet. Un hébergeur web permet de mettre à votre disposition des sites web conçus et de rendre votre nom de domaine accessible [W7].

Il existe plusieurs types d'hébergements dont :

1. **Hébergement gratuit** : Ce type d'hébergement offre un service basique, avec un espace de stockage assez limité, avec une fiabilité qui n'est pas toujours optimale.
2. **Hébergement payant** : Ce type d'hébergement propose un service de qualité, avec de bonnes capacités de stockage ainsi qu'une grande fiabilité. Le support sera également plus disponible et plus réactif en cas de problème et aidera avec des réponses personnalisées.

I.2.8 Méthodes d'analyse et de conception

Une méthode d'analyse et de conception a pour objectif de permettre la formalisation des étapes préliminaires du développement d'un système, afin de rendre ce développement plus fidèle aux besoins du client [W8].

- La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité, d'extensibilité, etc.
- La phase de conception permet de décrire de manière non ambiguë, en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation [W8].

Il existe plusieurs méthodes d'analyses, tel que UP (Unified Process).

I.2.8.1 Définition du processus unifié

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise [W11].

I.2.8.2 Caractéristiques du processus unifié [B3]

- 1) **Centre sur l'architecture** : L'architecture peut être considérée comme l'ensemble de vues du système qui vont provenir des besoins de l'entreprise et des différents intervenants.
- 2) **Pilote par les cas d'utilisation** : Le modèle des cas d'utilisations guide le processus unifié et décrit les fonctionnalités du système.
- 3) **Itératif et incrémental** : Les itérations se succèdent dans un ordre logique permettant de donner lieu à un incrément et donc d'établir un développement plus optimisé et efficace.

I.2.8.3 Cycle de vie du processus unifié [W1]

Le processus unifié se déroule en quatre phases :

- 1) Incubation/ création/début : Décrire la vision du produit final.
- 2) Elaboration : Détailler les fonctionnalités.
- 3) Construction : Concevoir et implanter le produit.
- 4) Transition : Transmettre une version du produit aux utilisateurs.

Chaque phase répète un nombre de fois une série d'itérations. Et chaque itération est composée de cinq activités :

- 1) Capture des besoins.
- 2) Analyse.
- 3) Conception.
- 4) Implémentation.
- 5) Test.

Toutes ces actions sont résumées dans le schéma présenté dans la figure (I.4) sise ci-dessous :

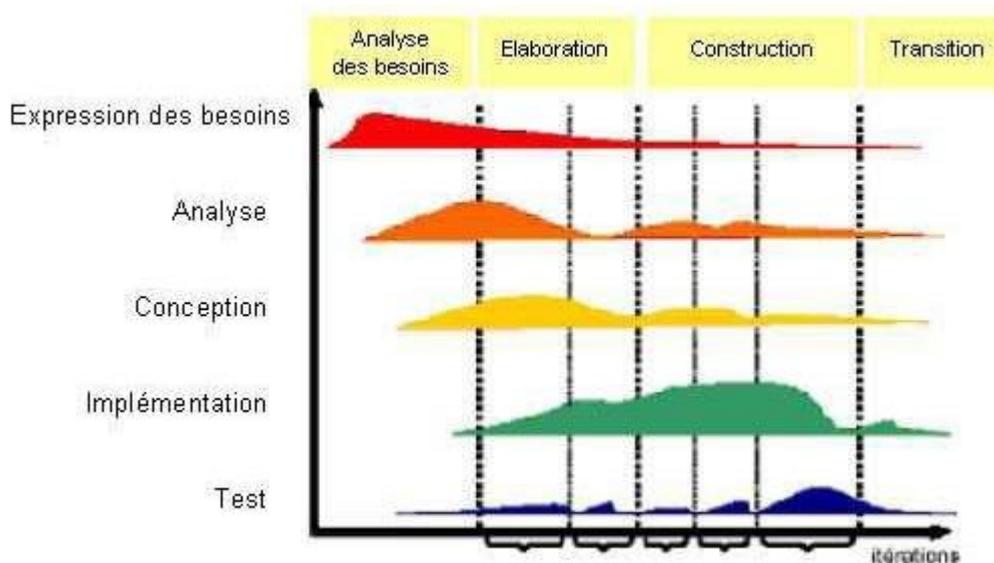


Figure I.4 : Cycle de vie du processus unifié [W1].

I.2.8.4 Langage de modélisation du processus unifié

L'UP s'appuie sur UML qui est un langage de modélisation graphique et textuel destiné à comprendre et à décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vues [B3].

UML définit 13 types de diagrammes, divisés en deux catégories [W1] :

Les diagrammes structurels ou statiques rassemblent :

1. Diagramme de classes (Class diagram).
2. Diagramme d'objets (Object diagram).

3. Diagramme de composants (Component diagram).
4. Diagramme de déploiement (Deployment diagram).
5. Diagramme des paquetages (Package diagram).
6. Diagramme de structure composite (Composite Structure Diagram).

Les diagrammes comportementaux rassemblent :

1. Diagramme des cas d'utilisation
2. Diagramme états-transitions (State Machine Diagram).
3. Diagramme d'activité (Activity Diagram).
4. Diagramme de séquence (Sequence Diagram).
5. Diagramme de communication (Communication Diagram).
6. Diagramme global d'interaction (Interaction Overview Diagram).
7. Diagramme de temps (Timing Diagram).

I.2.9 Le choix de la méthode

Pour la réalisation de notre application, notre choix a été porté sur le Processus unifié. En effet, le processus unifié est une solution de développement logiciel adaptée à tout type de projet. Ces traits distinctifs tiennent compte de trois notions :

- Piloté par les cas d'utilisation.
- Centré sur l'architecture.
- Itératif et incrémental.

Le langage de modélisation que nous avons utilisé est UML, qui est une partie intégrante de la démarche UP. Ces diagrammes sont largement utilisés dans chaque étape et phase de ce processus de développement.

I.2.10 Sécurité des applications Web

Session : Une session correspond à un groupe d'interactions effectuées par les utilisateurs sur votre site Web au cours d'une période donnée. Par exemple, une session peut inclure plusieurs pages vues, événements, interactions sur les réseaux sociaux [W9].

Principe d'une session PHP

- L'utilisation d'une session revient à pouvoir garder des variables d'une page à l'autre, de manière transparente.
- Lors de l'initialisation de la session, génération d'un identifiant unique par le serveur, transmis au client par un cookie.
- À chaque nouvelle requête du client, l'identifiant est transmis, et le serveur utilise les données liées à ce client.
- Les sessions ont une durée de vie limitée (configurable), en cas d'inactivité prolongée, les données sont effacées du serveur [W10].

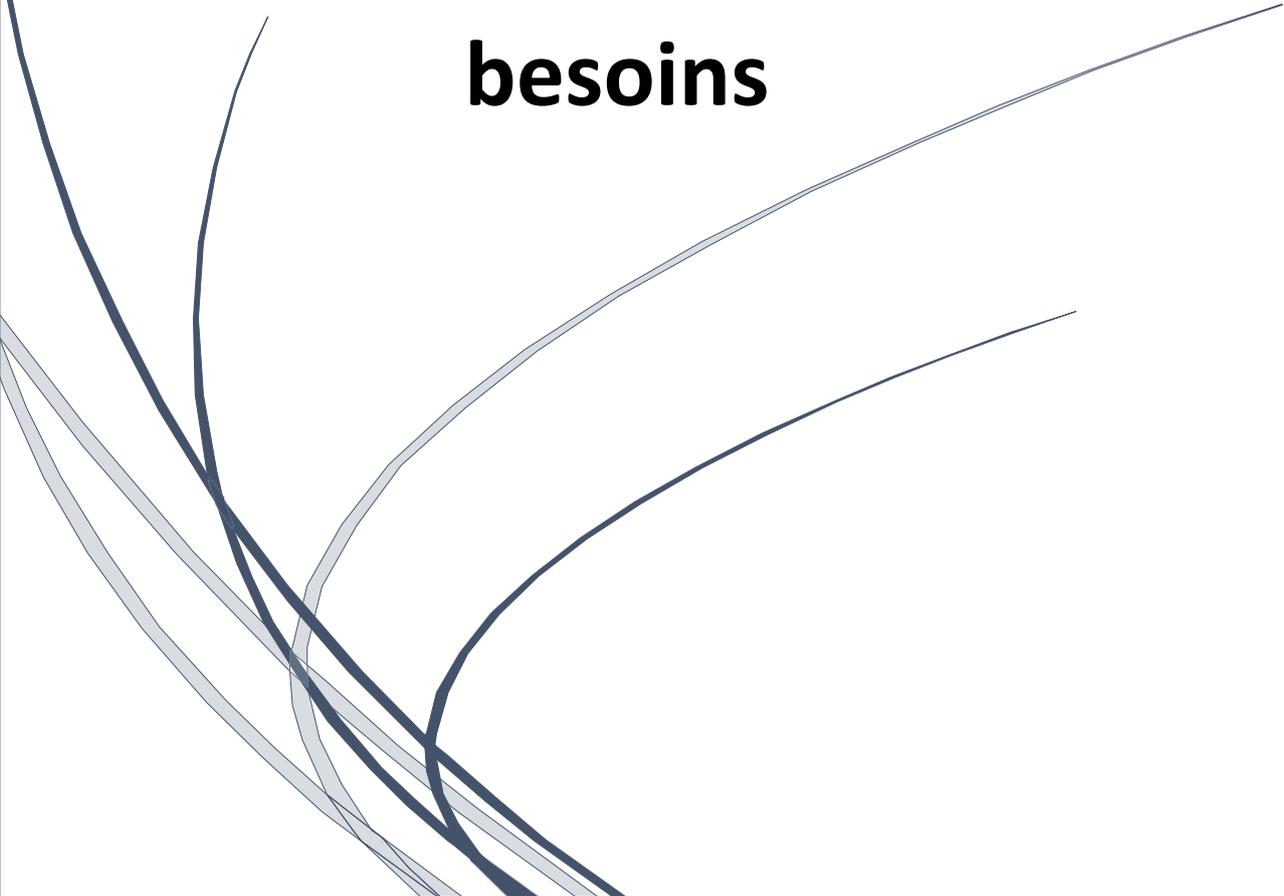
I.3 Conclusion

Dans ce chapitre, nous avons vu quelques notions sur les technologies web tel que les applications web, et nous avons présenté notre méthode de conception.

Dans le chapitre suivant, nous allons entamer l'étude de l'existant et spécification des besoins afin de définir les acteurs qui interagissent avec notre système, et réaliser quelques diagrammes UML à savoir le diagramme de cas d'utilisation.



Chapitre II : Étude de l'existant et spécification des besoins



II.1 Introduction

Dans ce chapitre, nous avons démarré par l'étude de l'existant ensuite critique de l'existant, ce qui nous amène à identifier les possibilités du système et les besoins fonctionnel et non fonctionnel de notre application web que nous projetons dans des diagrammes des cas d'utilisations détaillés et globales.

II.2 Etude de l'existant

L'étude de l'existant, est une phase importante pour bien comprendre le système actuel et définir ses objectifs.

II.2.1 Direction des impôts

L'impôt, constitue un des prélèvements obligatoires effectué par voie d'autorité et par la puissance publique (l'État et les collectivités territoriales) sur les ressources des personnes vivant sur son territoire ou y possédant des intérêts [W2].

Elle est chargée, de [W3]:

- Assurer l'autorité hiérarchique des centres des impôts et des centres de proximité des impôts.
- Organiser la collecte des éléments nécessaires à l'élaboration des prévisions fiscales.
- Analyser et d'évaluer périodiquement, l'activité des services relevant de sa compétence, d'en dresser synthèse et de proposer toute mesure de nature à améliorer leur action.
- La prise en charge des rôles et des titres de recettes et de recouvrement des impôts, taxes et redevances.
- Constituer et de gérer le fonds documentaire de la direction de wilaya et d'en assurer la diffusion et la vulgarisation.
- Organisation de l'accueil et de l'information des contribuables.

II.2.1.1 Organigramme de la direction de la wilaya de Bejaia

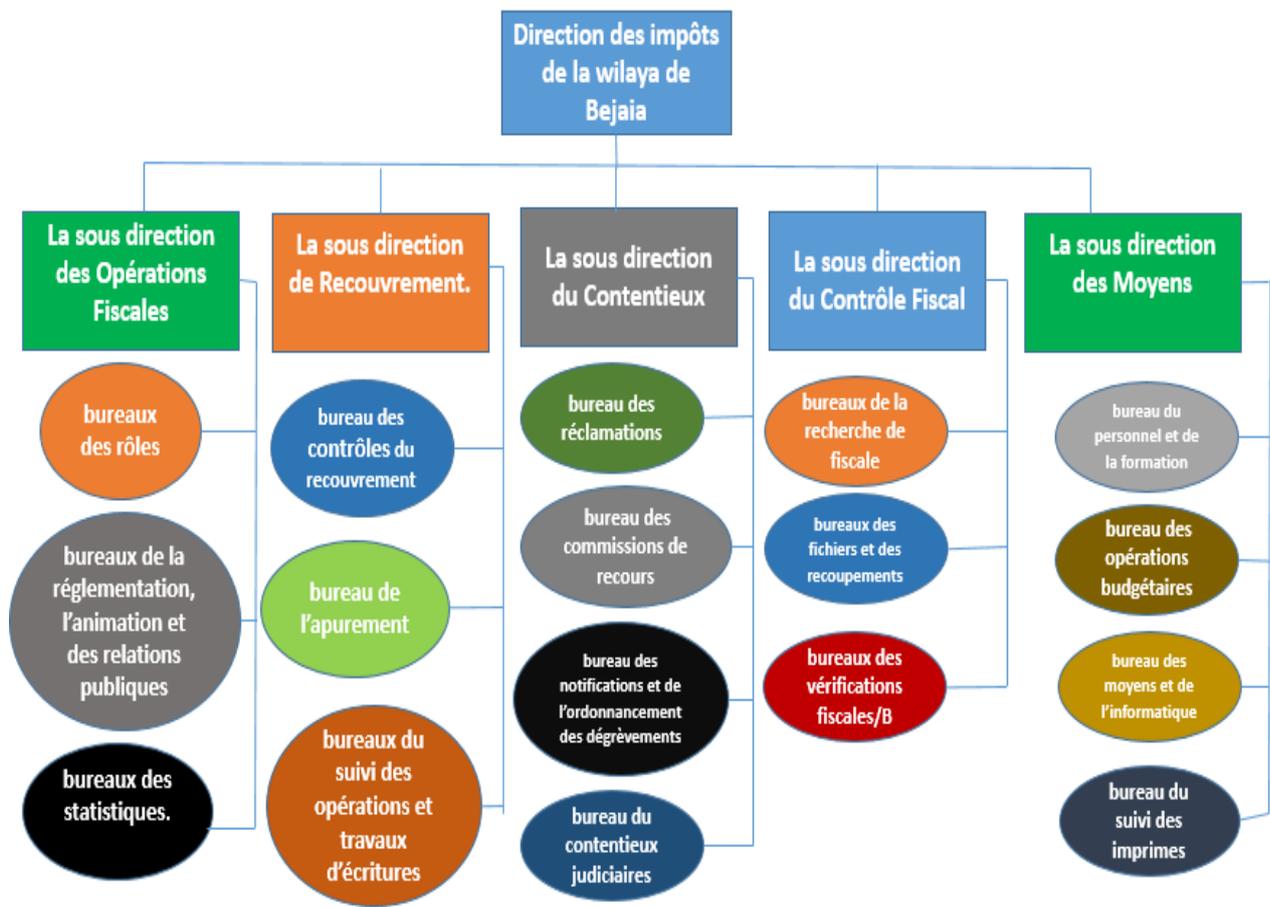


Figure II.1 : Organigramme de la direction de la wilaya de Bejaia.

II.2.1.2 Sous-direction du contentieux

Elle est chargée d'assurer les tâches suivantes:

- Le traitement des réclamations introduites, au titre des deux phases administratives sont :
 - ✓ La phase des recours contentieux.
 - ✓ La phase Gracieuse et la notification des décisions.
- Le traitement des demandes de remboursement des précomptes – TVA.
- La formalisation des dossiers de dépôt de plainte ou de recours en appel et la défense, devant les juridictions compétentes, des intérêts de l'administration fiscale.

II.2.1.3 Organigramme de la sous-direction du contentieux

La sous-direction du contentieux est composée de :

- Bureau des réclamations.
- Bureau du contentieux judiciaire et des commissions de recours.
- Bureau des notifications et de l'ordonnancement.

- Bureau de la recherche de l'information fiscale, des vérifications fiscales et du contrôle des évaluations.

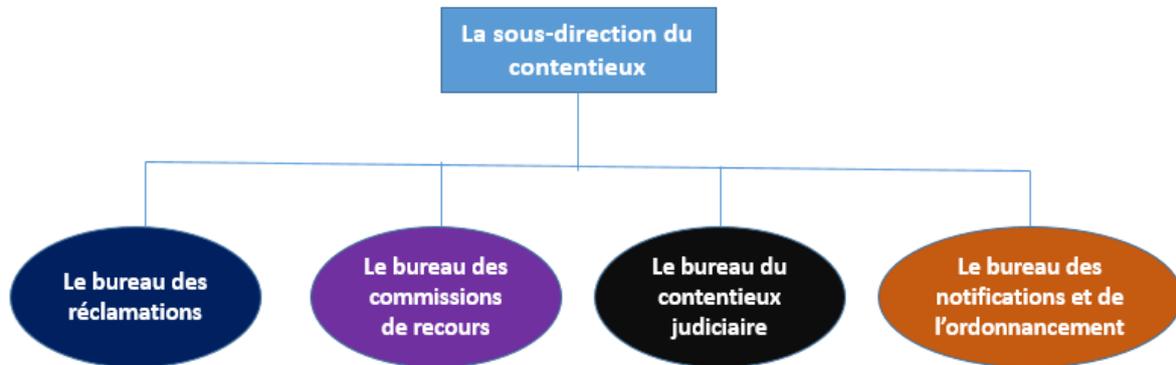


Figure II.2 : Organigramme de la sous-direction du contentieux.

II.2.2 Activités

Avant d'aborder cette étape importante, il s'avère nécessaire de définir le périmètre de notre étude.

Dans le cadre de ce projet, les principales activités étudiées sont :

- ❖ Gérer les utilisateurs de l'application.
- ❖ Gérer les contribuables.
- ❖ Gérer les statistiques.
- ❖ Gérer les réclamations.
- ❖ Gérer les recours.
- ❖ Gérer les messages.

II.2.3 Etude des fiches

Les fiches les plus utilisées dans le bureau des notifications et ordonnancement sont : feuille d'instruction, avis d'admission partielle, avis d'admission totale, avis de rejet, demande de réclamation, demande de recours.

Dans chaque fiche on distingue les informations spécifiées suivantes :

Feuille d'instruction :

- ❖ Nom du contribuable.
- ❖ Prénom du contribuable.
- ❖ Profession du contribuable.
- ❖ Adresse du contribuable.
- ❖ Type de l'impôt.

- ❖ Wilaya du contribuable.
- ❖ Montant de l'impôt.
- ❖ Montant du dégrèvement.
- ❖ Analyse de la demande.
- ❖ La date.
- ❖ Le motif de la décision.

Avis d'admission totale

- ❖ Nom du contribuable.
- ❖ Prénom du contribuable.
- ❖ Numéro d'identification fiscale.
- ❖ Date de la décision.
- ❖ Type de l'impôt.
- ❖ Montant de dégrèvement ou restitution.

Avis d'admission partielle

- ❖ Nom du contribuable.
- ❖ Prénom du contribuable.
- ❖ Numéro d'identification fiscale.
- ❖ Date de décision.
- ❖ Type de l'impôt.
- ❖ Montant de dégrèvement ou restitution.
- ❖ Le motif de la décision.

Avis de rejet

- ❖ Nom du contribuable.
- ❖ Prénom du contribuable.
- ❖ Numéro d'identification fiscale.
- ❖ Date de la décision.
- ❖ Type de l'impôt.
- ❖ Le motif de la décision.

Demande de recours

- ❖ Nom du contribuable.
- ❖ Prénom du contribuable.
- ❖ Adresse du contribuable.
- ❖ Numéro de téléphone.
- ❖ La date.
- ❖ Nom de la société.
- ❖ Objet de la demande.
- ❖ Le contenu de la demande.

Demande de la réclamation

- ❖ Nom du contribuable.
- ❖ Prénom du contribuable.
- ❖ Adresse du contribuable.

- ❖ Numéro de téléphone.
- ❖ La date.
- ❖ Nom de la société.
- ❖ Objet de la demande.
- ❖ Le contenu de la demande.

II.2.4 Etude des postes de travail

Chef de service : Le chef de service peut ajouter, modifier, et supprimer un utilisateur (simple employé) de l'application web et peut accéder à toutes les gestions de l'application.

Simple employé : Lorsque l'administrateur (le Chef de service) ajoute un simple employé, il lui accorde un login et mot de passe, comme le simple employé considéré maintenant comme un utilisateur alors il va assurer certaines tâches suivantes :

- ❖ Réception des contribuables.
- ❖ La rédaction des feuilles d'instruction des contribuables.
- ❖ Repense pour les réclamations et les recours

II.3 Critique de l'existant

Les logiciels EXCEL et WORD de Microsoft Corporation, bien qu'ils soient puissants ne satisfont pas à tous les besoins des utilisateurs de cette application. En effet, certaines fonctionnalités ne se font pas de manière automatique et nécessite une intervention manuelle qui est la plupart du temps fastidieuse.

II.3.1 Problématique

L'administration de la sous-direction du contentieux désire synchroniser ses services d'une manière performante, facile et efficace afin de satisfaire l'employeur et garantir le maintien de l'activité pour détecter les problèmes existants.

Nous avons interrogé le chef de service du contentieux qui nous a cité quelques anomalies et pour localiser leur source, nous nous sommes mis en pratique avec lui. Et après une observation continue, nous avons pu observer les insuffisances suivantes:

- Volume important des informations traitées manuellement, ce qui provoque parfois des erreurs dans le bureau des documents.
- Recherche difficile sur les registres qui engendre une perte de temps.
- Insécurité des informations.
- Possibilité d'erreur dans le remplissage des différents documents et registres.
- Possibilité d'erreur dans le calcul des statistiques.
- Nombre important des archives qui engendre une difficulté de stockage.
- Détérioration des archives à force de leur utilisation trop fréquente.
- Prenant en considération l'absence d'avoir les moyens d'échange des informations entre l'administration et les contribuables.
- Comment les contribuables prennent des réclamations en ligne ?
- Comment les contribuables prennent des recours en ligne ?

II.3.2 Solution proposer

Nous constatons que la solution informatique est la plus adéquate puisqu'elle répond mieux aux anomalies souvent fréquentées dans la gestion manuelle. Ainsi, nous avons décidé de concevoir une application web qui va gérer les activités de ce service et qui va permettre par la suite de minimiser le support papier et d'améliorer la rapidité de l'accès à l'information.

Pour cela, nous avons assigné à notre étude les objectifs suivants :

- Réduire le temps de traitement des données.
- Faciliter la recherche et l'accès aux informations.
- Stockage des informations sur des supports informatiques ce qui assurera leur sécurité.
- Gain de temps dans le calcul des statistiques.
- Automatiser les tâches qui se traitent manuellement.
- Proposer une bonne codification.
- La prise des réclamations des contribuables en ligne.
- La prise des recours des contribuables en ligne.
- Etablir une communication par messagerie entre la sous-direction du contentieux et les contribuables.

II.4 Expression des besoins

On distingue deux types de besoins :

- Les besoins fonctionnels qui conduisent à l'élaboration des cas d'utilisation.
- Les besoins techniques (non-fonctionnels) qui aboutissent à la rédaction d'une des exigences de système pour sa réalisation et son bon fonctionnement.

II.4.1 Besoins fonctionnels

Administrateur : Est un acteur qui peut faire toutes les tâches du chef de service, mais durant tout le travail nous utiliserons que l'acteur 'Administrateur'. Il doit pouvoir :

- S'authentifier.
- Gérer les utilisateurs.
- Gérer les statistiques.
- Gérer les messages.
- Gérer les fiches.

a. **Utilisateur** : Est un acteur qui peut faire toutes les tâches du simple employé, mais durant tout le travail nous utiliserons que l'acteur 'Utilisateur'. Il doit pouvoir :

- L'authentification.
- Gérer les statistiques.
- Gérer les fiches.

- Gérer les messages.

b. **Contribuable** : Personne qui contribue aux charges publiques en payant des impôts.

- L'authentification.
- Rédiger une réclamation.
- Rédiger un recours.

II.4.2 Besoins non fonctionnels

- L'application doit permettre de gérer les accès des utilisateurs selon un privilège et un état d'activation de chaque compte.
- Prévenir contre l'accès direct avec les liens URL et définir un délai de temps pour la fermeture de session non active.
- Le logiciel doit fonctionner sur les différents systèmes d'exploitation.
- Le code doit être clair pour permettre des futures évolutions ou améliorations.
- L'ergonomie : l'application offre une interface conviviale et facile à utiliser.
- La sécurité : l'application doit respecter la confidentialité des données.

II.5 Modélisation des besoins du système

Pour la modélisation des besoins, nous avons opté pour les diagrammes suivants :

- Le diagramme de contexte qui va nous permettre d'avoir une vue générale de notre système.
- Les diagrammes de cas d'utilisation qui vont aider à identifier les acteurs à recenser les cas d'utilisation.

II.5.1 Description du contexte de l'application

La description du contexte de l'application se fait à travers les deux étapes suivantes :

- L'identification des messages qui transiteront entre les acteurs et le système.
- Le diagramme de contexte.

II.5.1.1 Identification des messages échangés entre les acteurs vers le système

Un message représente la spécification d'une communication unidirectionnelle entre objets et qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur.

Un message est normalement associé à deux occurrences d'évènement : un évènement d'envoi et un évènement de réception [B4].

Les messages entre les acteurs et le système sont donnés dans le tableau (II.1) :

Administrateur		Application
<ul style="list-style-type: none"> • Demande d'authentification. • Demande d'ajout, modifier, rechercher, supprimer d'un utilisateur. • Demande d'afficher des statistiques. • Demande d'ajoute une feuille d'instruction. • Demande d'ajoute, afficher, supprimer un message. 		
Utilisateur		Application
<ul style="list-style-type: none"> • Demande d'authentification. • Demande d'afficher des statistiques. • Demande d'ajoute une feuille d'instruction. • Demande d'ajoute, afficher, supprimer un message. 		
Contribuable		Application
<ul style="list-style-type: none"> • Demande d'authentification. • Demande d'ajoute une réclamation. • Demande d'ajoute un recours. • Demande d'ajoute, afficher, supprimer un message. 		
Administrateur		Contribuable
<ul style="list-style-type: none"> • Ajouter des messages. 		
Utilisateur		Contribuable
<ul style="list-style-type: none"> • Ajouter des messages. 		

Tableau II.1 : Légende des messages des diagrammes de contexte dynamique.

II.5.1.2 Modélisation du contexte

Tous les messages identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme voir la figure (II.3).

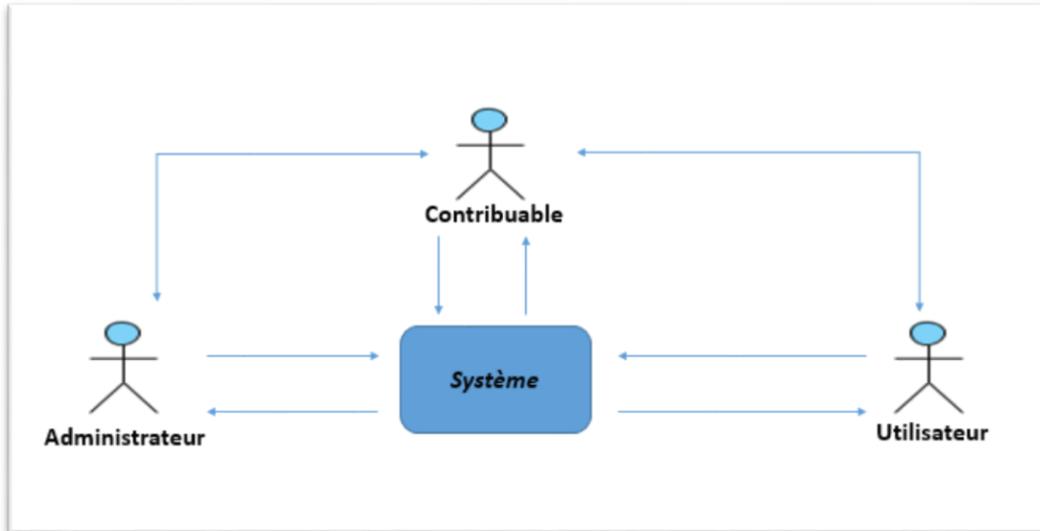


Figure II.3 : Diagramme de contexte.

II.5.2 Identification des cas d'utilisations

Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie [W12].

Cas d'utilisation 1 : « Authentification ».

Cas d'utilisation 2 : « Gérer les utilisateur » ce cas est scindé en quatre scenarios.

- **Scenario 1 :** ajouter un utilisateur.
- **Scenario 2 :** rechercher un utilisateur.
- **Scenario 3 :** modifier un utilisateur.
- **Scenario 4 :** supprimer un utilisateur.

Cas d'utilisation 3 : « afficher les statistique » ce cas est scindé en six scenarios.

- **Scenario 1 :** afficher nombre de réclamations.
- **Scenario 2 :** afficher nombre de recours.
- **Scenario 3:** afficher membre de contribuables.
- **Scenario 4:** afficher nombre de rejets.
- **Scenario 5:** afficher nombre d'admissions.
- **Scenario 6:** afficher nombre d'admissions partielles.

Cas d'utilisation 4 : « gérer les fiches » ce cas est scindé un seul scenario.

- **Scenario 1** : ajouter une feuille d'instruction.

Cas d'utilisation 5 : « gérer les messages » ce cas est scindé en deux scenarios.

- **Scenario 1** : Ajouter un message.
- **Scenario 2** : supprimer un message.

Cas d'utilisation 6 : « réclamation préalable » ce cas est scindé un seul scenario.

- **Scenario 1** : rédiger une réclamation.

Cas d'utilisation 7 : « recours » ce cas est scindé un seul scenario.

- **Scenario 1** : rédiger un recours.

II.5.3 Description textuelle des cas d'utilisations

Elle est organisée de la façon suivante :

- **Un sommaire d'identification** : Va résumer les propriétés du cas d'utilisation.
 - **Une description des enchaînements** : Préconditions au déclenchement du cas d'utilisation doivent être spécifiées, un scénario nominal décrivant celui-ci additionné à des scénarios alternatifs et d'exceptions [W13].
- ✓ **Cas d'utilisation 1** : « Authentification »
- **Acteur principale** : Administrateur, utilisateur (employé simple), contribuable.
 - **Objectif** : vérifier l'identité et lui attribuer les privilèges.
 - **Précondition** : néant.
 - **Post-condition** : Application accessible.
 - **Scenario nominal** : L'utilisateur saisit l'adresse de l'application dans le navigateur. La page d'authentification s'affiche en lui demandant d'introduire login et le mot de passe.
L'utilisateur introduit login et le mot de passe.
Le système vérifie login et mot de passe, ensuite oriente l'utilisateur vers la page d'accueil.
 - **Scenario alternatif** :

Le système détecte une erreur dans les données saisies.

Le système recharge la page d'authentification avec un message d'erreur en lui indiquant que le mot de passe ou le login est incorrecte.

Les détails de diagramme de Cas d'utilisation « Authentification » sont donnés par la figure ci-dessus (Figure II.4) :

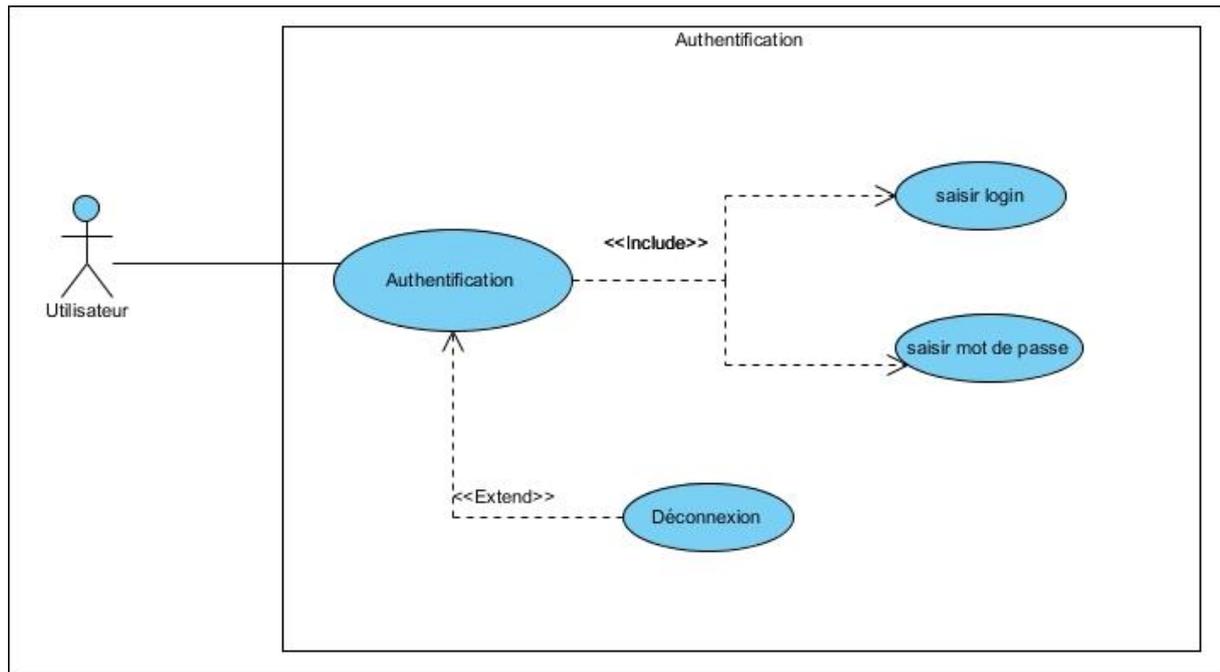


Figure II.4 : Diagramme de cas d'utilisation « Authentification ».

✓ **Cas utilisation 2 : « Gérer les utilisateurs »**

- **Acteur principal :** administrateur.
- **Objectif :** Ce cas d'utilisation permet d'ajouter un nouvel utilisateur de le rechercher, modifier ou de le supprimer.
- **Précondition :** S'authentifier.

L'administrateur accède à son espace.

- **Post-condition :** Mis à jour de la base de données.
- **Scénario nominal :**

L'administrateur demande d'accéder à l'interface gestion des utilisateurs.

Le système affiche l'interface gestion des utilisateurs.

L'utilisateur remplir les champs de formulaire puis valide l'opération.

Le système sauvegarde la mise à jour de l'utilisateur dans la base de donnée puis, affiche un message de confirmation.

- **Scénario alternatif :**

Le formulaire n'est pas valide : le système affiche le message d'erreur en lui indiquant les champs erronés.

L'utilisateur existe déjà : le système affiche un message « l'utilisateur existe ».

Les détails de diagramme de cas d'utilisation « Gérer les utilisateurs » sont donnés par la figure ci-dessus (Figure II.5) :

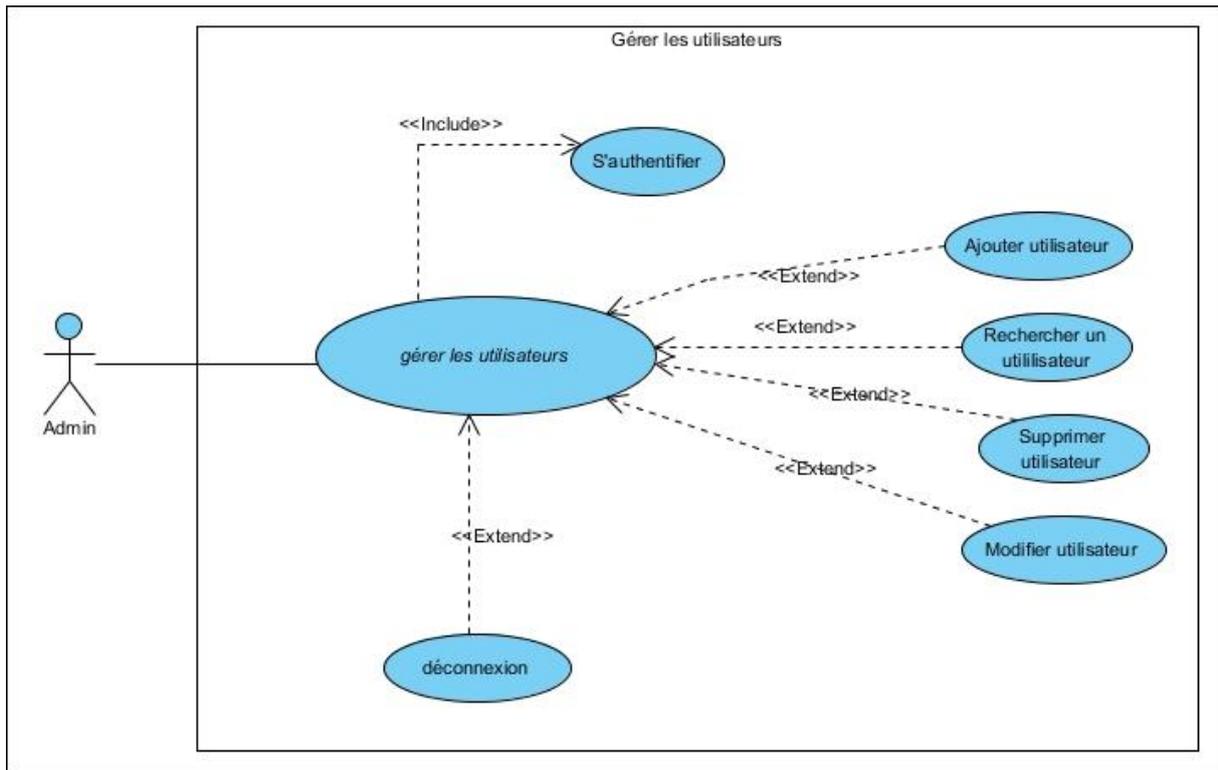


Figure II.5 : Diagramme de cas d'utilisation « Gérer les utilisateurs ».

✓ **Cas utilisation 3 : « Afficher les statistiques »**

- **Acteur principal :** Administrateur, utilisateur (simple employé).
- **Objectif :** Ce cas d'utilisation permet d'afficher les statistiques.
- **Précondition :** S'authentifier.

L'administrateur ou l'utilisateur (simple employé) accède à son espace.

- **Post-condition :** Mis à jour de la base de données.
- **Scénario nominal :**

L'administrateur ou l'utilisateur (simple employé) demande d'accéder à l'interface statistique.

Le système affiche l'interface statistique.

L'utilisateur remplir les deux champs (date de début et date de fin) puis valide l'opération.

Le système affiche le résultat.

- **Scénario alternatif :**

Les deux champs ne sont pas valides : le système affiche le message d'erreur en lui indiquant les champs erronés.

Les détails de diagramme de Cas d'utilisation « afficher les statistiques » sont donnés par la figure ci-dessus (Figure II.6) :

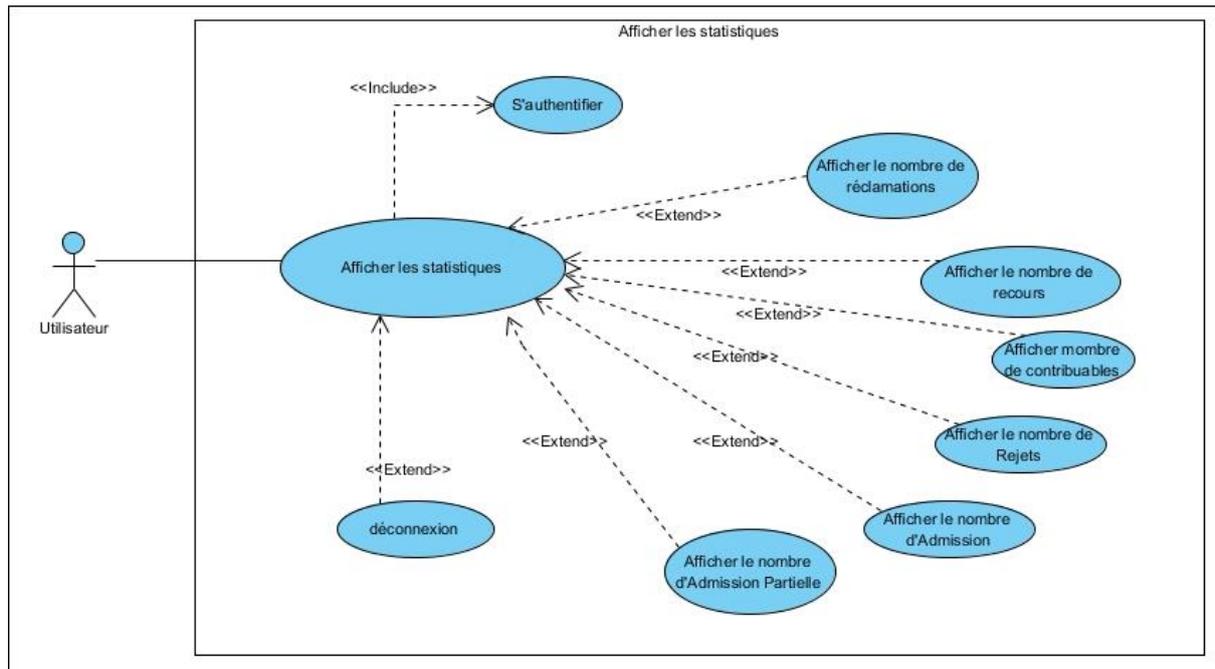


Figure II.6 : Diagramme de cas d'utilisation « afficher les statistiques ».

✓ **Cas d'utilisation 4 :** « gérer les fiches »

- **Acteur principal :** Administrateur, utilisateur (simple employé).
- **Objectif :** Ce cas d'utilisation permet de créer un nouveau formulaire (réponse pour le contribuable qui a envoyé une réclamation).
- **Précondition :** S'authentifier.

L'administrateur ou l'utilisateur (simple employé) accède à son espace.

- **Post-condition :** Mis à jour de la base de données.
- **Scenario nominal :**

L'administrateur demande d'accéder à l'interface gérer les fiches.
 Le système affiche l'interface gérer les fiches.
 L'utilisateur remplir les champs de formulaire (feuille d'instruction) puis valide l'opération.
 Le système sauvegarde la mise à jour des fiches dans la base de données puis affiche un message de confirmation.

Scenario alternatif :

Formulaire n'est pas valide : le système affiche un message d'erreur en lui indiquant les champs erronés.

Les détails de diagramme de cas d'utilisation « gérer les fiches des contribuable » sont donnés par la figure ci-dessus (Figure II.7) :

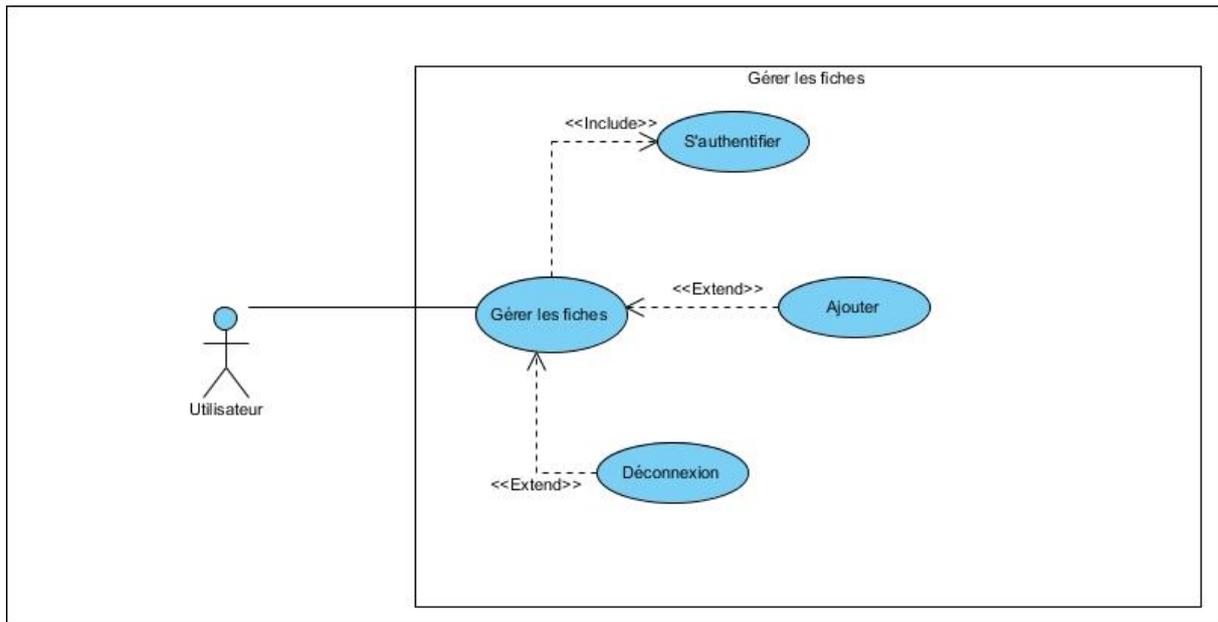


Figure II.7 : Diagramme de cas d'utilisation « gérer les fiches des contribuables ».

✓ **Cas d'utilisation 5 :** « gérer les messages »

- **Acteur principal :** Administrateur, contribuable et utilisateur (simple employé).
- **Objectif :** Ce cas d'utilisation permet d'échanger les messages entre le contribuable et l'utilisateur (simple employé).
- **Précondition :** S'authentifier.

Le contribuable ou bien l'administrateur accède à son espace.

- **Post-condition :** Message envoyé.
- **Scénario nominal :**

L'administrateur ou le contribuable choisit d'envoyer un message.
Le système affiche le formulaire.
L'administrateur ou le contribuable remplit et valide le formulaire.
Le système affiche le message de succès.

Les détails de diagramme de cas d'utilisation « gérer les messages » sont donnés par la figure ci-dessus (Figure II.8) :

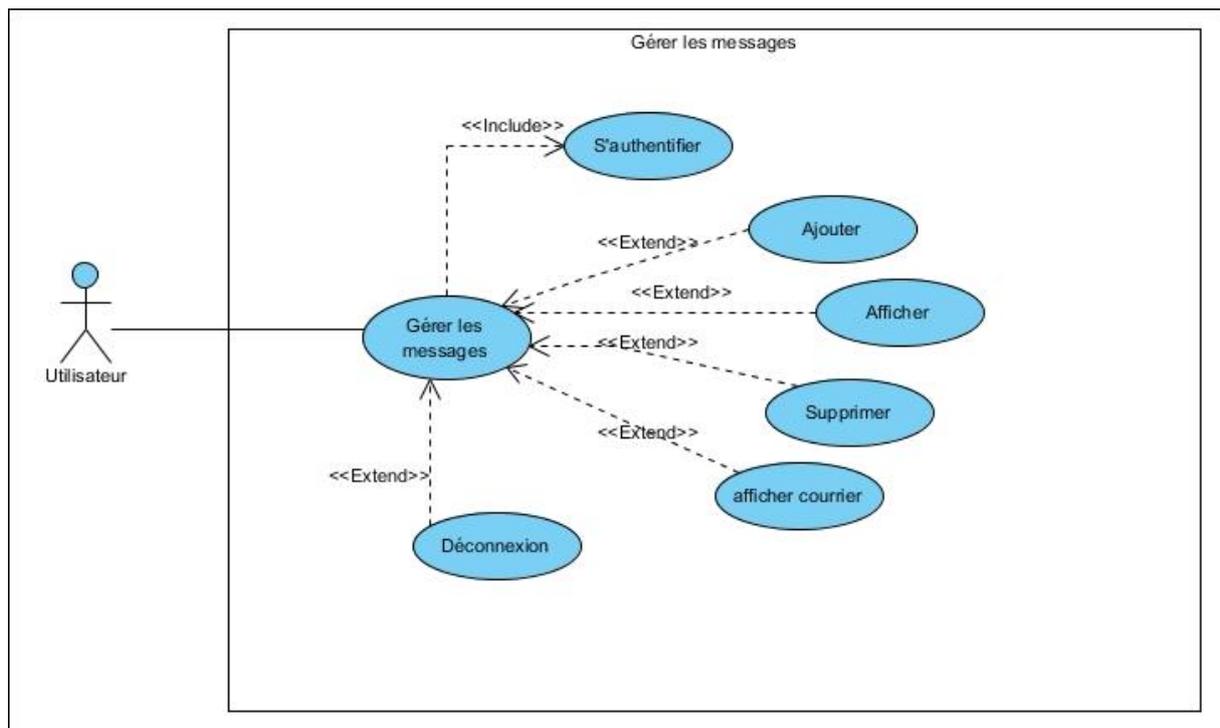


Figure II.8: Diagramme de cas d'utilisation « gérer les messages ».

✓ **Cas d'utilisation 6 :** « rédiger réclamation préalable »

- **Acteur principal :** Contribuable.
- **Objectif :** Ce cas d'utilisation permet de créer un nouveau formulaire (une demande).
- **Précondition :** S'authentifier.

Le contribuable accède à son espace.

- **Post-condition :** Mis à jour de la base de données.
- **Scenario nominal :**

Le contribuable demande d'accéder à l'interface « rédiger réclamation préalable ».

Le système affiche l'interface « gérer réclamation préalable ».

Le contribuable remplir les champs de formulaire puis valide l'opération.

Le système sauvegarde la mise à jour des fiches dans la base de données puis affiche un message de confirmation.

- **Scenario alternatif :**

Formulaire n'est pas valide : le système affiche un message d'erreur en lui indiquant les champs erronés.

Les détails de diagramme de cas d'utilisation « gérer réclamation préalable » sont donnés par la figure ci-dessus (Figure II.9) :

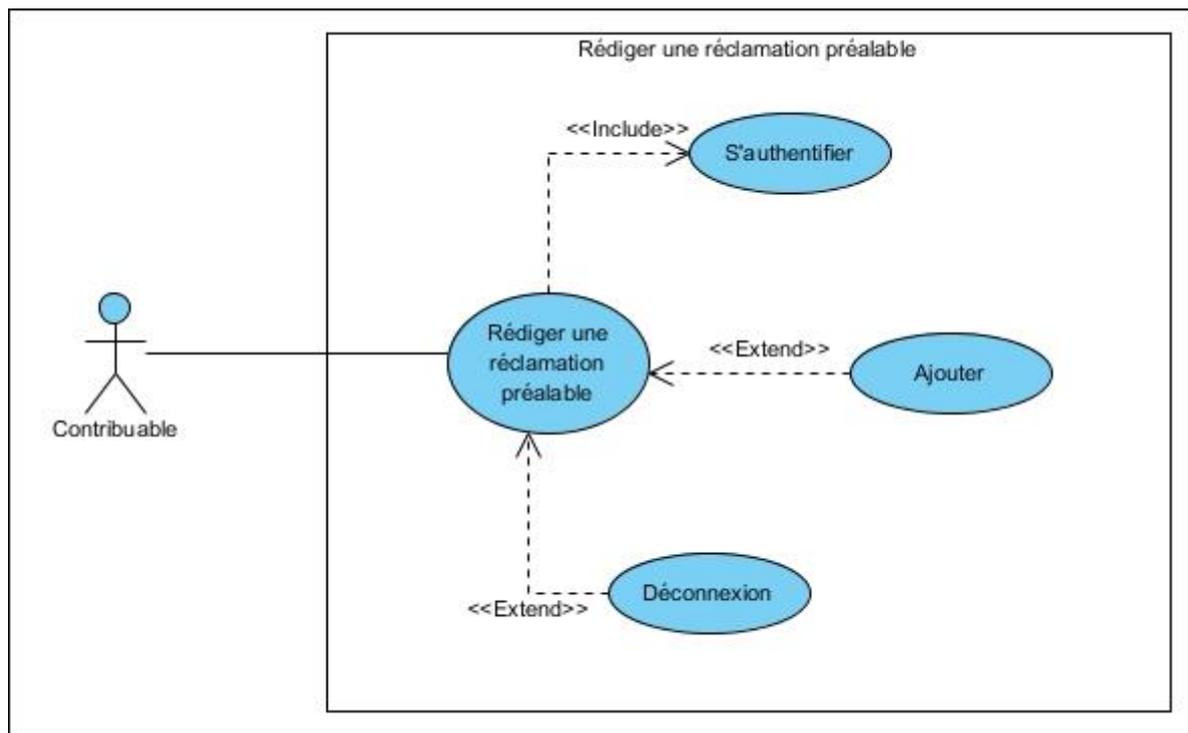


Figure II.9 : Diagramme de cas d'utilisation « gérer réclamation préalable ».

- ✓ **Cas d'utilisation 7 :** « rédiger recours »
- **Acteur principal :** Contribuable.
- **Objectif :** Ce cas d'utilisation permet de créer un nouveau formulaire (demande).
- **Précondition :** S'authentifier.

Le contribuable accède à son espace.

- **Post-condition :** Mis à jour de la base de données.
- **Scenario nominal :**

Le contribuable demande d'accéder à l'interface rédigé recours.
 Le système affiche l'interface rédigé recours.
 L'utilisateur remplir les champs de formulaire puis valide l'opération.
 Le système sauvegarde la mise à jour des fiches dans la base de données puis affiche un message de confirmation.

- **Scenario alternatif :**

Formulaire n'est pas valide : le système affiche un message d'erreur en lui indiquant les champs erronés.

Les détails de diagramme de cas d'utilisation « gérer recours » sont donnés par la figure ci-dessus (Figure II.10) :

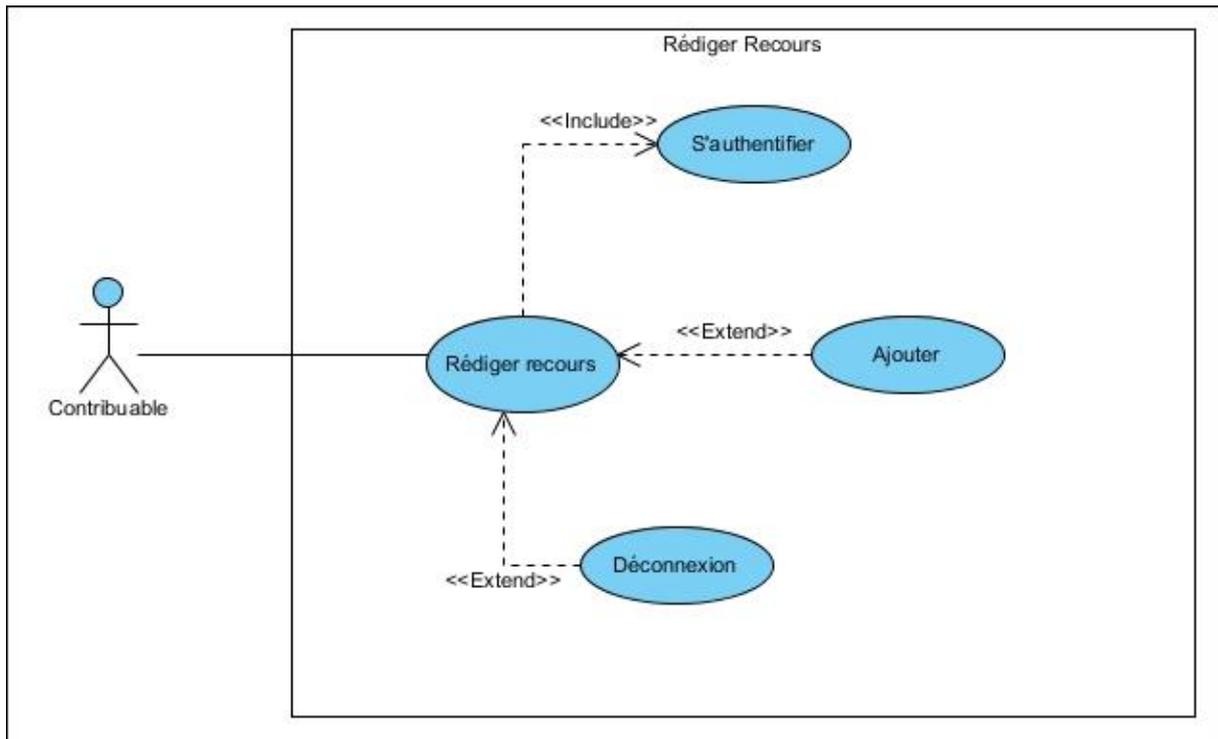


Figure II.10 : Diagramme de cas d'utilisation « gérer recours ».

Ainsi le diagramme global des cas d'utilisation peut se traduire comme suit voir la figure (II.11) :

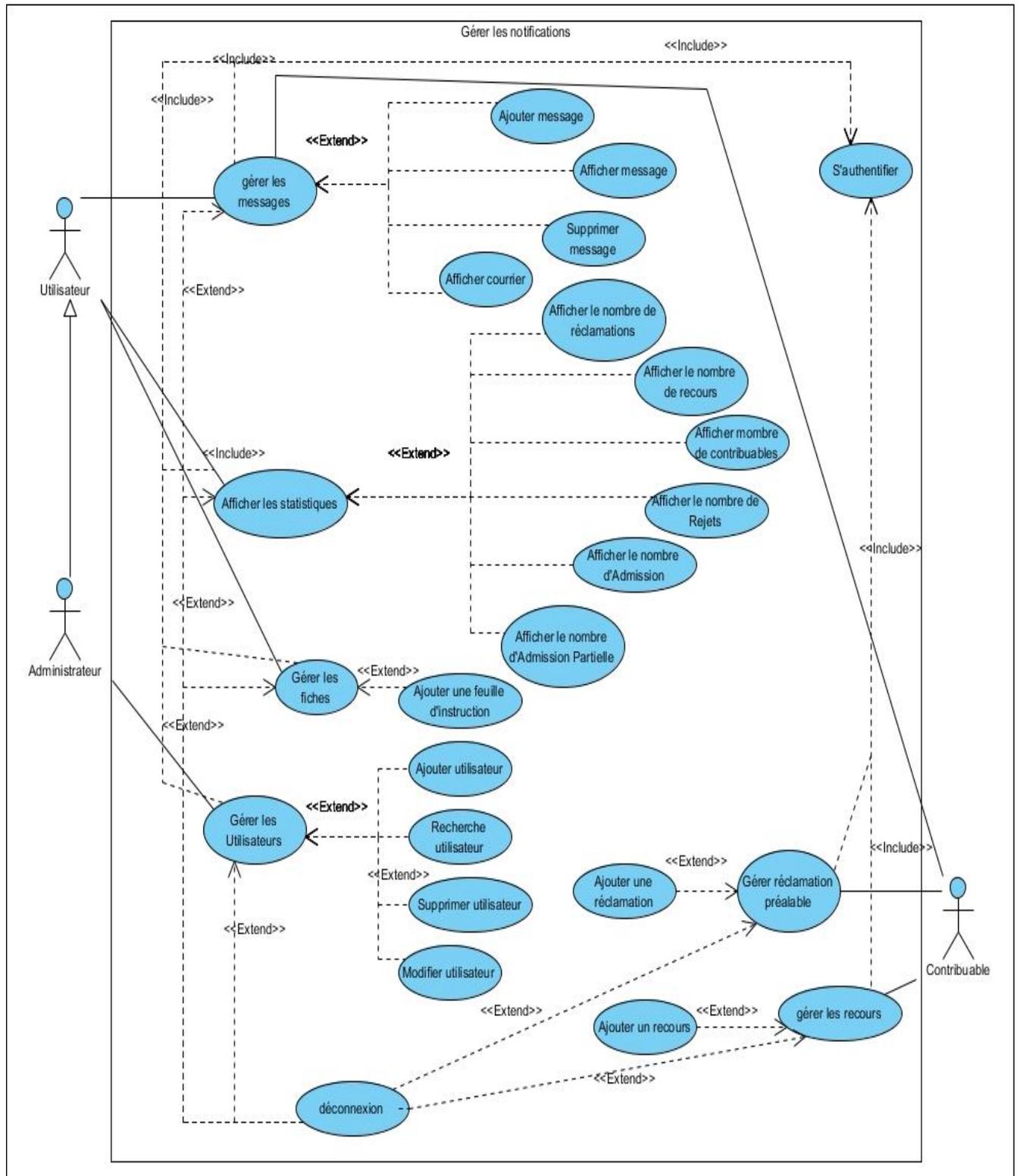


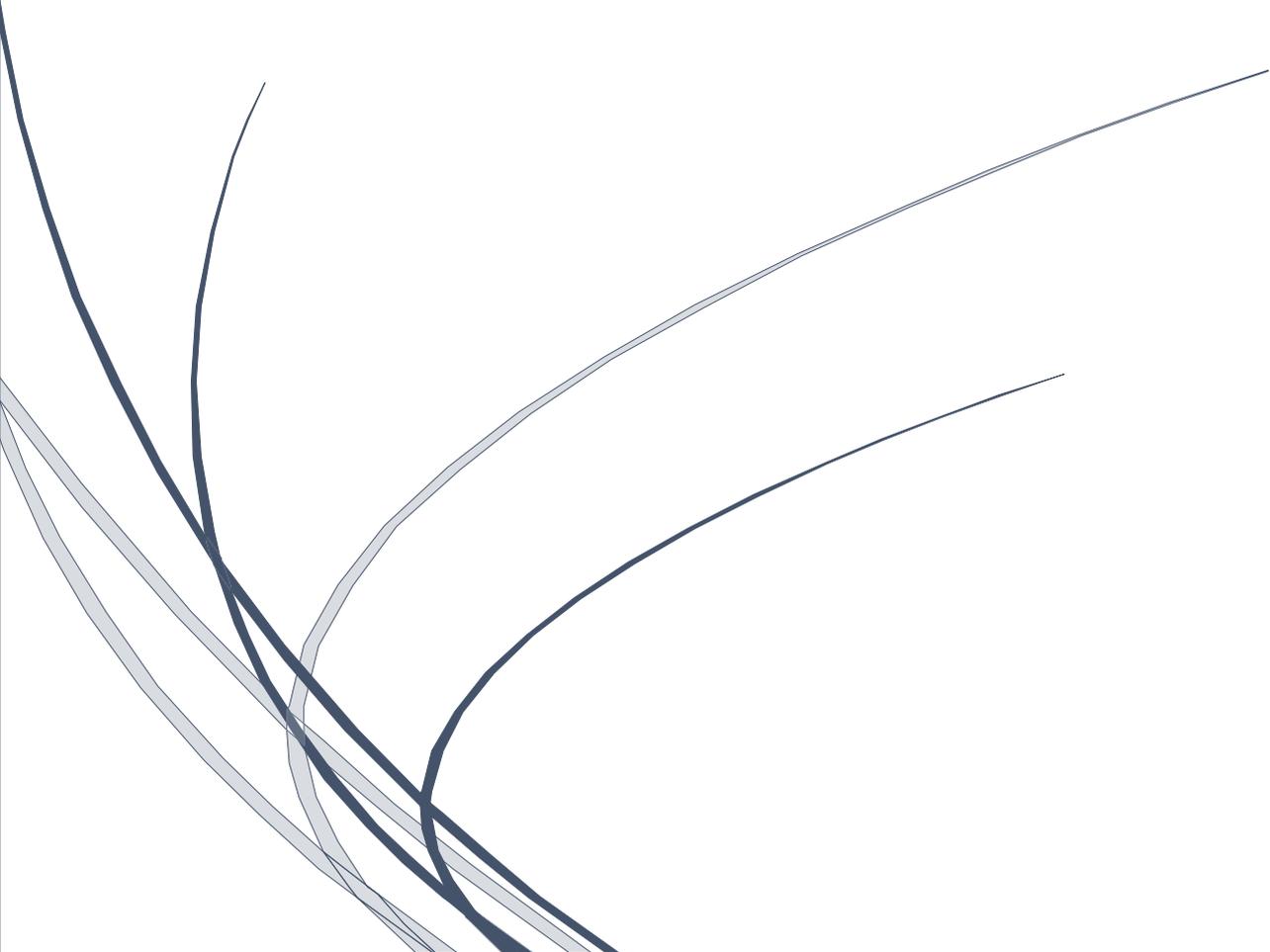
Figure II.11 : diagramme global des cas d'utilisations.

II.6 Conclusion :

Dans ce chapitre, nous avons présenté d'une façon globale, une étape essentielle du système élaboré pour l'analyse de notre application en suivant le processus de normalisation UML. Nous avons montré le principe de fonctionnement. Nous avons aussi présenté le diagramme de contexte et expliquer les diagrammes de cas d'utilisation, retenu pour notre système.



Chapitre III : Conception



III.1 Introduction

Dans le cycle de vie de notre projet, la conception représente une phase primordiale et déterminante pour produire une application de haute qualité.

Dans ce qui, suit nous allons présenter les diagrammes de séquences et les diagrammes des classes, et nous allons construire le modèle relationnel à partir des classes de l'application, et cela en suivant les règles du passage au model relationnel.

III.2 Réalisation des diagrammes de séquence

III.2.1 Définition

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs [W14]:

- Les objets au cœur d'un système interagissent en s'échangent des messages.
- Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

III.2.2 Diagramme de séquence de cas d'utilisation « authentification ».

➤ Description textuelle

A chaque authentification, l'utilisateur doit saisir son login et son mot de passe, d'où l'usage de l'opérateur "loop". L'utilisateur valide sa demande de connexion, deux cas peuvent se présenter : si les informations saisis sont correctes, le système affiche l'espace correspondant de l'utilisateur, s'ils sont incorrects, le système réaffiche le formulaire d'authentification avec un message d'erreur, ce qui justifie l'utilisation de l'opérateur "alt".

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.1) :

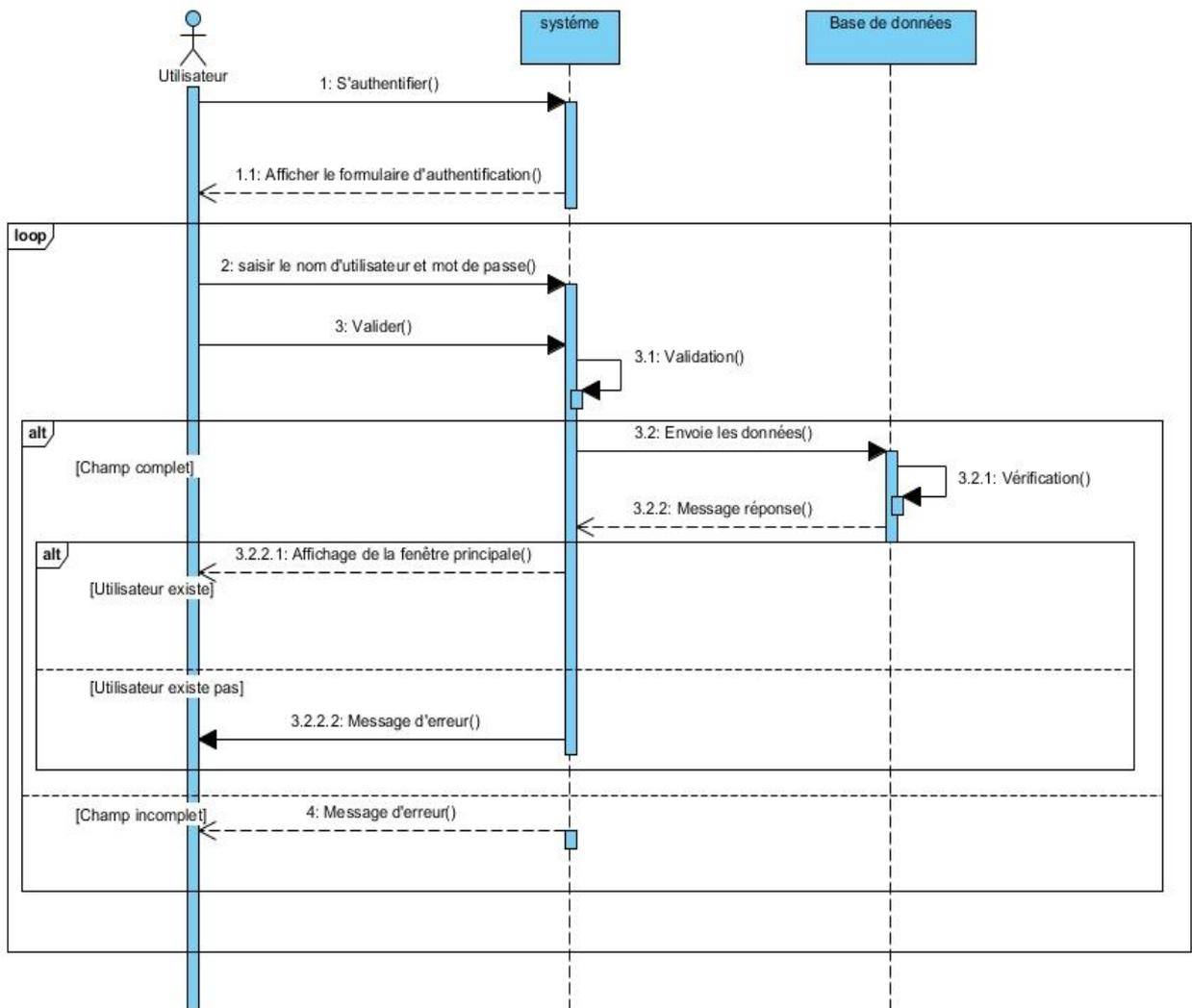


Figure III.1 : Diagramme de séquence de cas d'utilisation « Authentification ».

III.2.3 Diagramme de séquence de cas d'utilisation « gérer les messages »

➤ Description textuelle du scénario « ajouter un message » :

Une fois que l'utilisateur (l'administrateur/le contribuable /un simple employé) est dans son espace, il choisit d'ajouter un message. Le système lui affiche le formulaire d'ajout, par la suite l'utilisateur le remplit et le valide. A la fin, le système vérifie la validité des informations saisies, autrement l'utilisateur sera averti de l'échoue de l'envoi de message.

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.2) :

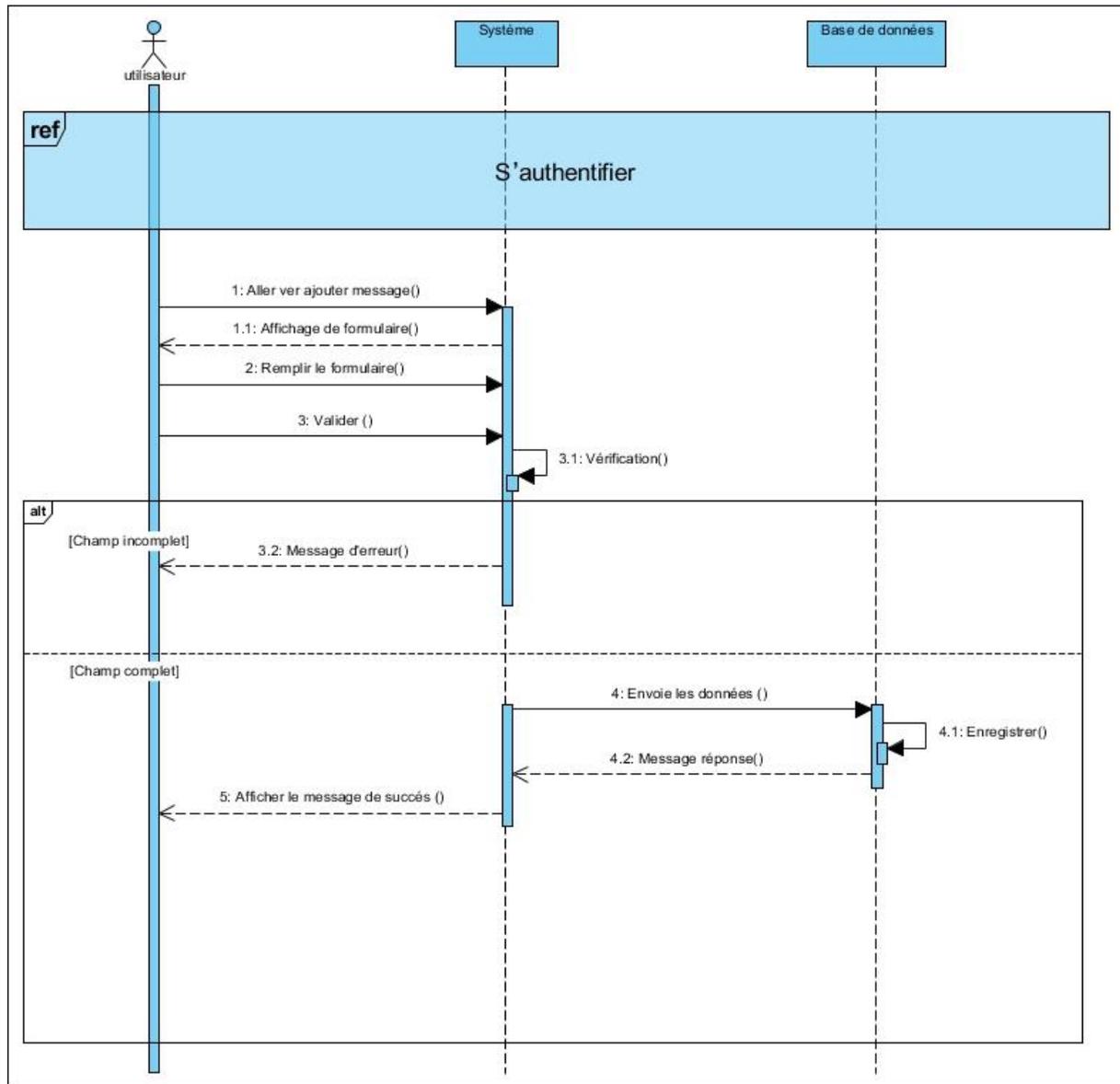


Figure III.2 : Diagramme de séquence de cas d'utilisation « ajouter message ».

➤ Description textuelle du scénario « supprimer un message »:

L'utilisateur (l'administrateur/le contribuable/un simple employé) a l'opportunité de supprimer un message dans la base de données, il doit tout d'abord accéder à la gestion des messages ensuite supprimer un message. Par la suite l'utilisateur saisit l'identifiant de message à supprimer puis il valide.

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.3) :

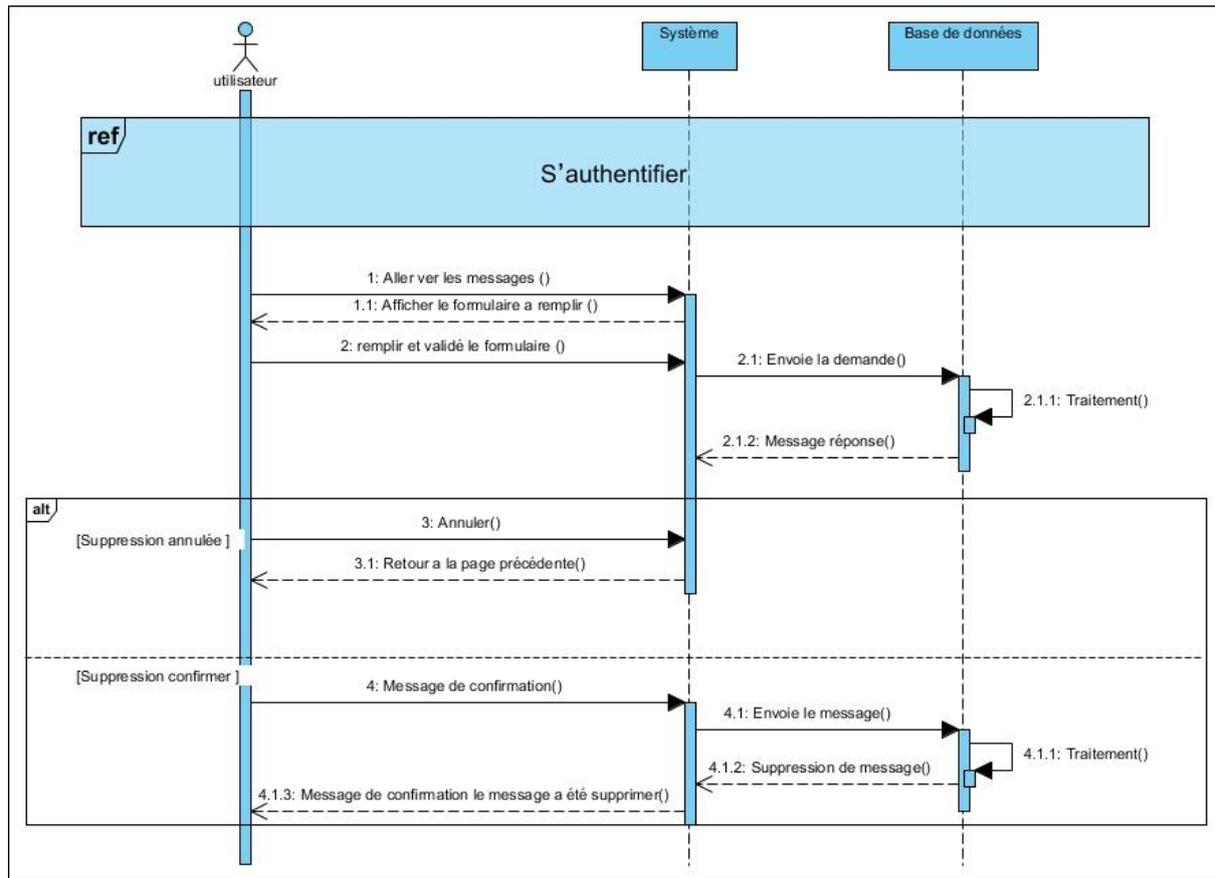


Figure III.3 : Diagramme de séquence de cas d'utilisation « supprimer message ».

III.2.4 Diagramme de séquence de cas d'utilisation « gérer les fiches »

➤ Description textuelle du scénario « ajouter une feuille d’instruction » :

Si l’administrateur ou un simple employé veut saisir une feuille d’instruction, il saisit les informations du contribuable et la réponse pour sa réclamation dans le formulaire, le système vérifie les informations et répond soit par un message d’erreur ou bien il génère automatiquement une admission totale ou une admission partielle ou bien un rejet.

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.4) :

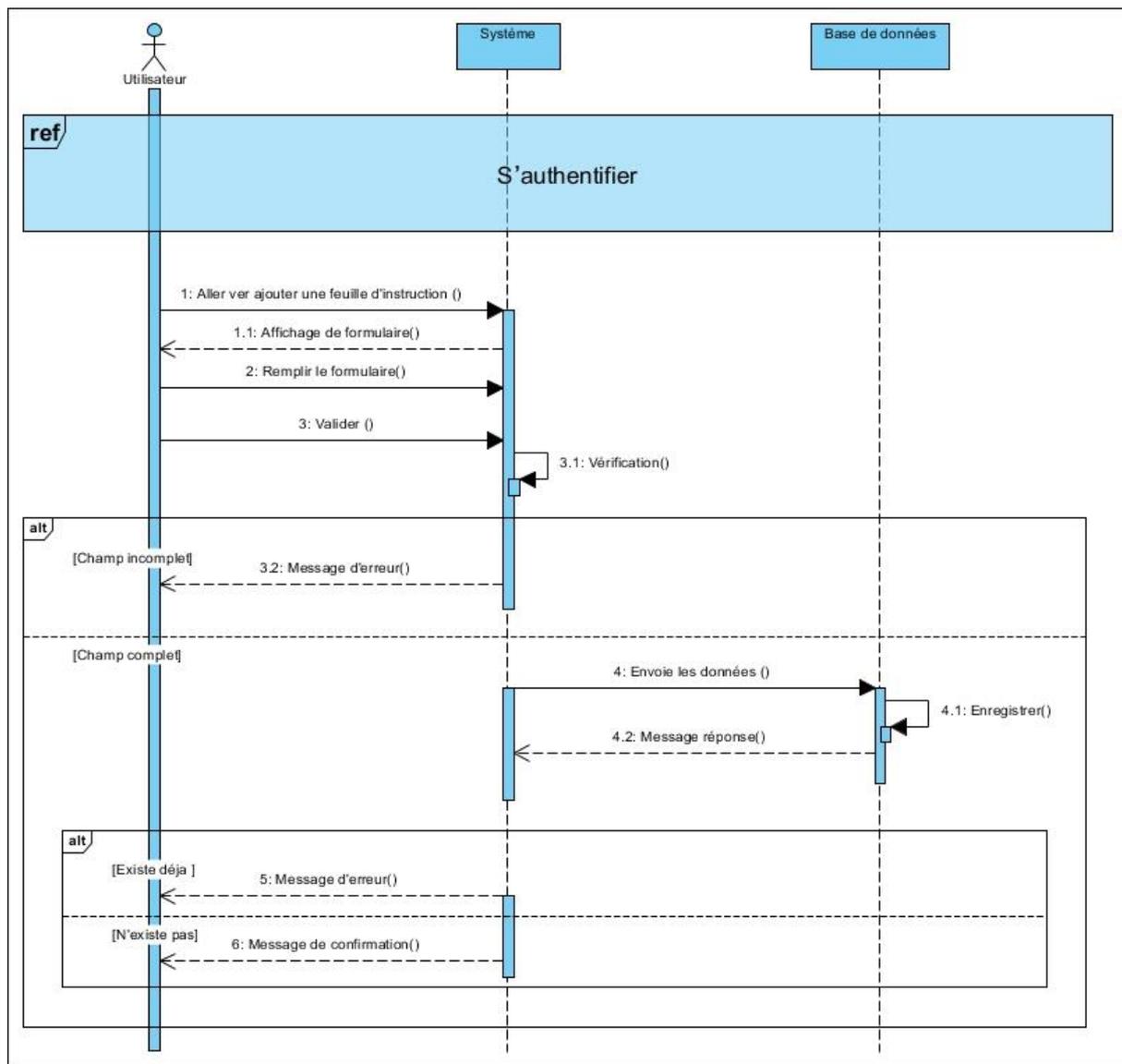


Figure III.4 : Diagramme de séquence de cas d'utilisation « ajouter une feuille d'instruction ».

III.2.5 Diagramme de séquence de cas d'utilisation « gérer les utilisateurs »

➤ Description textuelle du scénario « ajouter un utilisateur » :

Si l'administrateur veut saisir un nouvel utilisateur, il saisit les informations de l'utilisateur dans le formulaire, le système vérifie les informations et répond soit par un message d'erreur ou bien par un message de confirmation d'ajout.

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.5) :

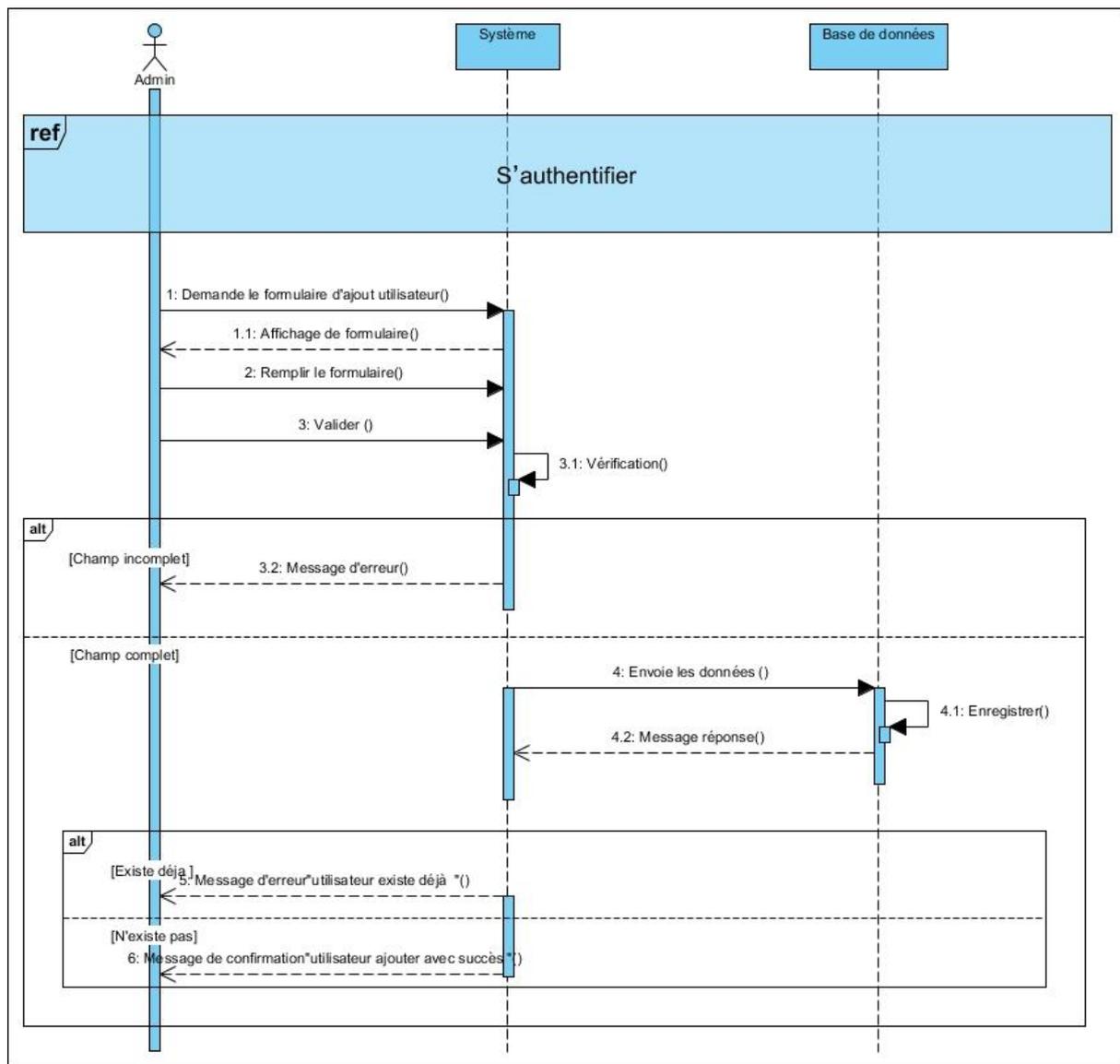


Figure III.5 : Diagramme de séquence de cas d'utilisation « ajouter utilisateur ».

➤ Description textuelle du scénario « rechercher un utilisateur »:

Si l'administrateur désire de chercher un utilisateur dans la base de données, il suffit juste de saisir le login correspondant à cet utilisateur. Si l'utilisateur existe dans la base de données, le système affiche les informations correspondantes à l'utilisateur, Sinon il affiche « l'utilisateur introuvable ».

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.6) :

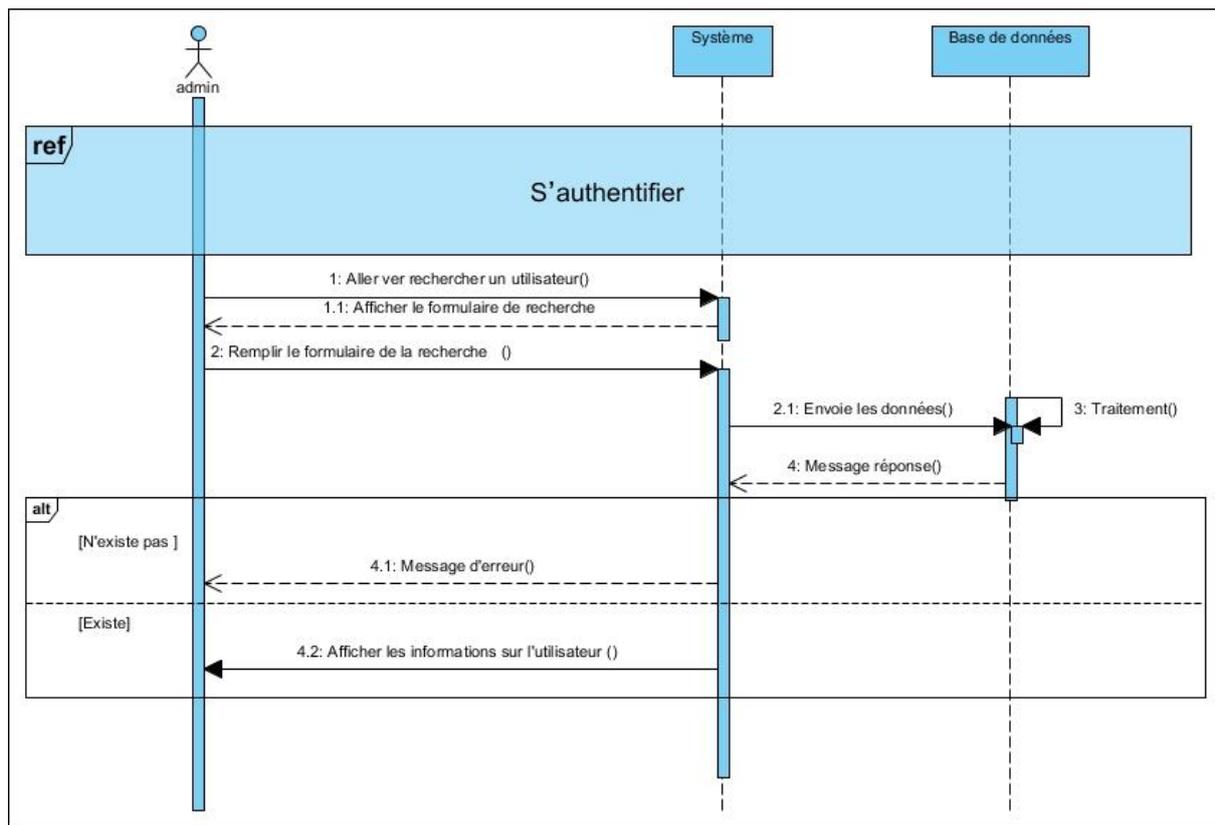


Figure III.6 : Diagramme de séquence de cas d'utilisation « rechercher utilisateur ».

➤ **Description textuelle du scénario « supprimer un utilisateur »:**

L'administrateur a l'opportunité de supprimer un utilisateur dans la base de données. Pour cela, il doit tout d'abord accéder à l'interface « gestion des utilisateurs », puis il saisit le login de l'utilisateur à supprimer ensuite il valide.

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.7) :

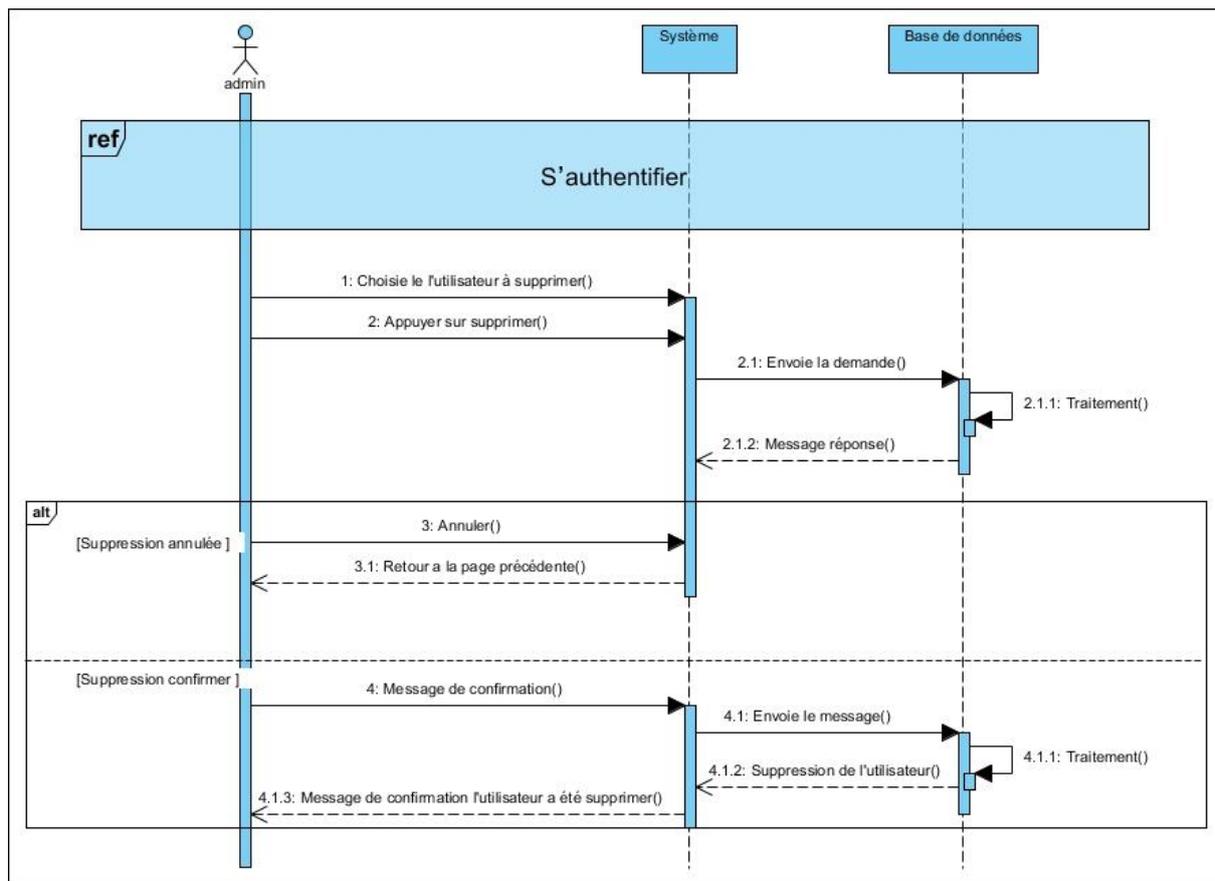


Figure III.7 : Diagramme de séquence de cas d’utilisation « supprimer utilisateur ».

➤ Description textuelle du scénario « modifier un utilisateur » :

Lorsque l’administrateur veut modifier un utilisateur, il doit tout d’abord cliquer sur le bouton « gestion des utilisateur », puis modifie un utilisateur, en remplissant les champs requis et enfin envoyer les données au serveur pour les enregistrer. Après traitement du serveur un message de succès ou d’erreur sera envoyé.

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.8) :

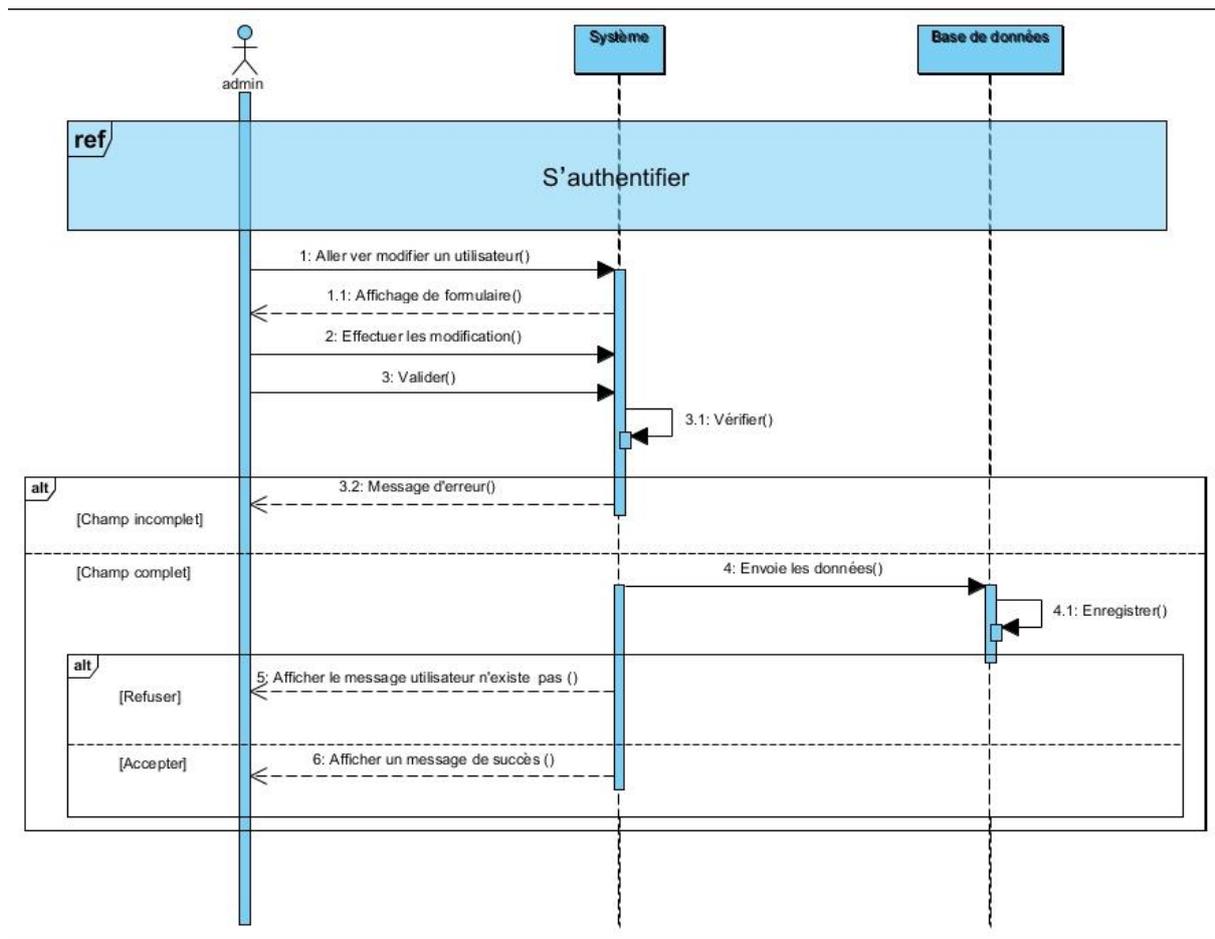


Figure III.8 : Diagramme de séquence de cas d'utilisation « modifier utilisateur ».

III.2.6 Diagramme de séquence de cas d'utilisation « gérer réclamation préalable »

➤ Description textuelle du scénario « ajouter une réclamation » :

Si le contribuable veut contester (faire une réclamation), il doit accéder à l’ajout d’une réclamation. Tout d’abord, il saisit les informations dans le formulaire, le système vérifie les informations puis répond soit par un message d’erreur, soit par message de succès.

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.9) :

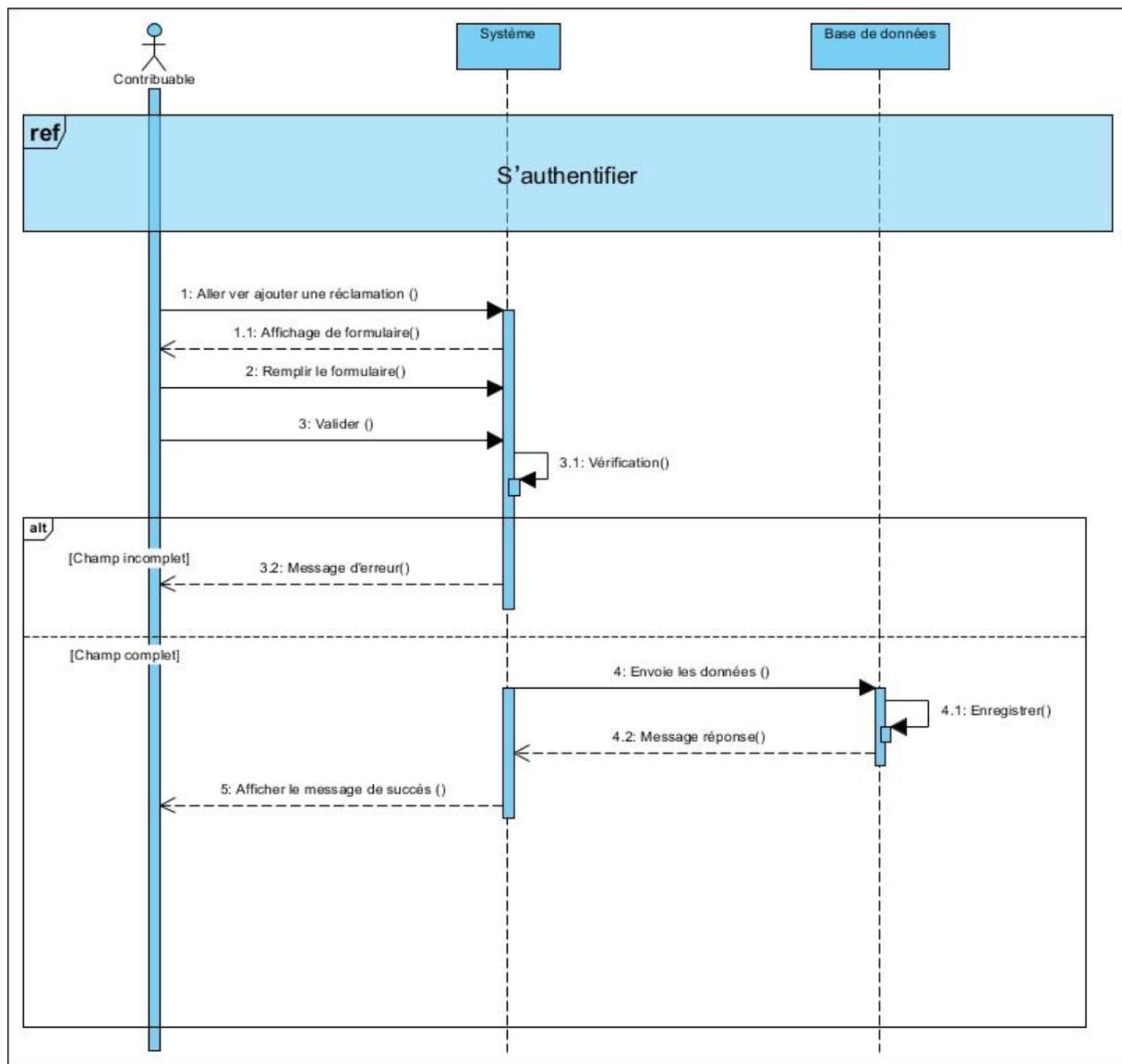


Figure III.9 : Diagramme de séquence de cas d’utilisation « ajouter une réclamation ».

III.2.7 Diagramme de séquence de cas d’utilisation « gérer recours »

➤ Description textuelle du scénario « ajouter un recours » :

Si le contribuable veut contester (faire un recours), quand il n’est pas convaincu par le résultat de la réclamation, il doit accéder à l’ajout d’un recours. Tout d’abord, il saisit les informations dans le formulaire, le système vérifie les informations et répond soit par un message d’erreur ou par un message de succès.

Les détails de diagramme de séquence sont donnés par la figure ci-dessus (Figure III.10) :

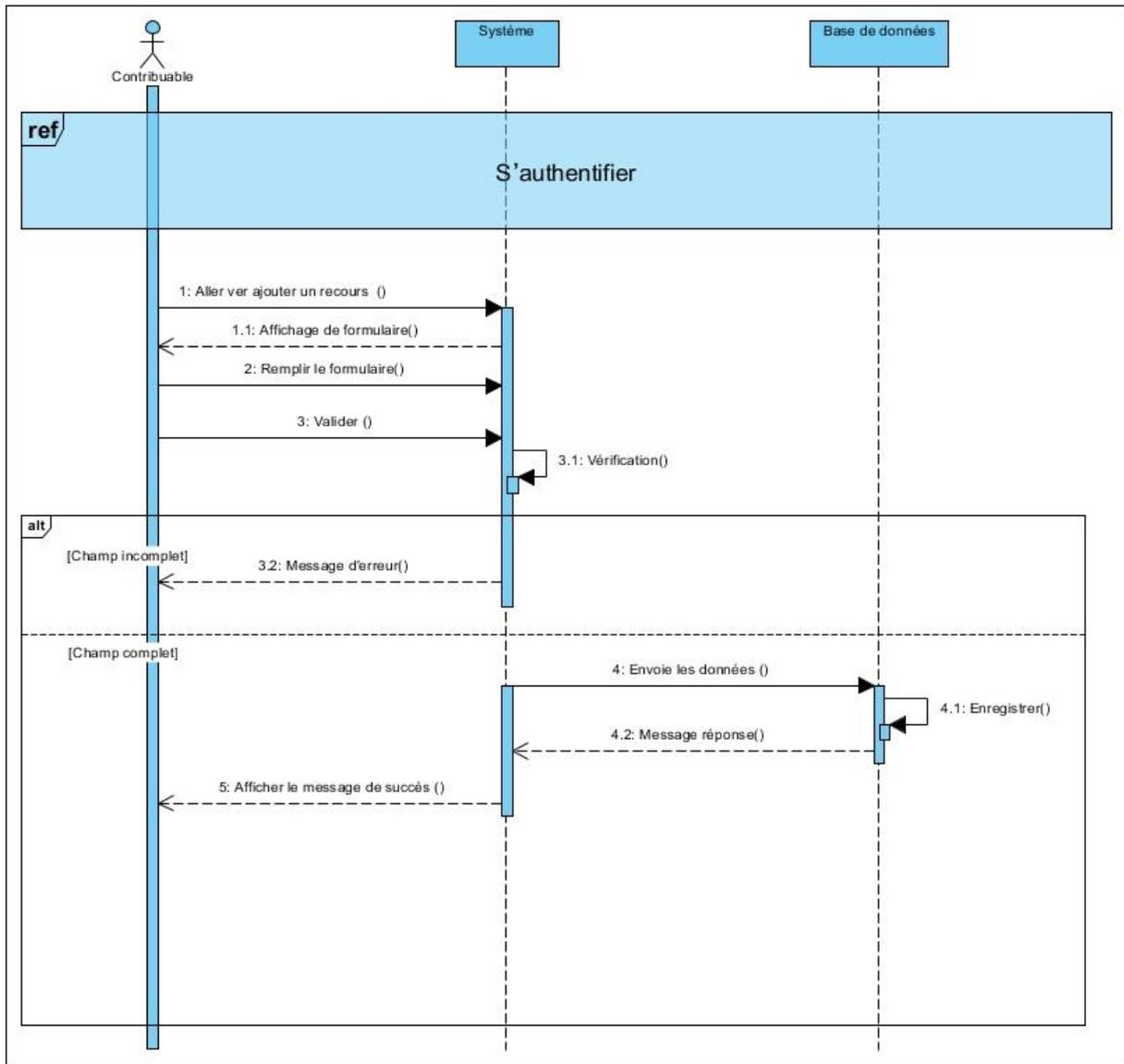


Figure III.10 : Diagramme de séquence de cas d’utilisation « ajouter recours ».

III.3 Rédaction du dictionnaire des données

Le recensement des informations et de leurs catégories permet de constituer le dictionnaire de données. Il s’agit d’un tableau recensant des informations rencontrées de l’analyse préalable.

Le tableau III.1 sise ci-dessous présente le dictionnaire des données de notre application :

Classes	Attributs	Description de l'attribut	Types	méthodes
utilisateur	<ul style="list-style-type: none"> ❖ Id_utilisateur ❖ nom ❖ prenom ❖ adr ❖ tel ❖ email ❖ login ❖ password 	<ul style="list-style-type: none"> -identificateur de l'utilisateur -Nom de l'utilisateur -Prénom de l'utilisateur - Adresse de l'utilisateur - Numéro de téléphone de l'utilisateur - Adresse email de l'utilisateur - Login de l'utilisateur - Mot de passe de l'utilisateur 	<ul style="list-style-type: none"> -INT(10) - VARCHAR (30) -VARCHAR (30) -VARCHAR (50) -VARCHAR (15) -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) 	<ul style="list-style-type: none"> Ajouter utilisateur Modifier utilisateur Supprimer utilisateur Rechercher utilisateur
Contribuable	<ul style="list-style-type: none"> ❖ nis ❖ nom ❖ prenom ❖ adr ❖ ville ❖ pays ❖ tel ❖ password ❖ date 	<ul style="list-style-type: none"> -Identificateur de contribuable -Nom de contribuable -Prénom de contribuable - Adresse de contribuable - ville de contribuable - pays de contribuable - Numéro de téléphone de contribuable - password de contribuable -date d'ajout de contribuable 	<ul style="list-style-type: none"> -Int (30) -VARCHAR (30) -VARCHAR (30) -VARCHAR (50) -VARCHAR (30) -VARCHAR (30) -VARCHAR (15) -VARCHAR (30) -datetime 	<ul style="list-style-type: none"> inscription

feuille_instruction	<ul style="list-style-type: none"> ❖ Numis ❖ Demande ❖ impottaxe ❖ numregistre ❖ numi ❖ mantantct ❖ mantantdt ❖ motifs ❖ date 	<ul style="list-style-type: none"> -Numéro d'identification statique de contribuable -le type de la demande -Alternative (impôt/taxe) -Numéro de registre de contribuable -numéro d'imposition -montant avant le dégrèvement - montant après le dégrèvement -motifs de dégrèvement -date de la rédaction de la feuille 	<ul style="list-style-type: none"> -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) -float -float -text -date time 	Ajouter
messages	<ul style="list-style-type: none"> ❖ Type ❖ NIS_m ❖ message ❖ date 	<ul style="list-style-type: none"> -Type de message -identificateur de message -Le contenu du message -Date d'envoi 	<ul style="list-style-type: none"> -int(2) -int(15) -text -datetime 	<ul style="list-style-type: none"> Ajouter message Supprimer message Afficher message
réclamation	<ul style="list-style-type: none"> ❖ NIS_recl ❖ Ai ❖ Nom ❖ Prenom ❖ Profession ❖ Adresse 	<ul style="list-style-type: none"> -identificateur de réclamation -Article d'imposition -le nom de contribuable -prénom de contribuable -la profession de contribuable -l'adresse de contribuable 	<ul style="list-style-type: none"> -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) 	Ajouter réclamation

	<ul style="list-style-type: none"> ❖ Lettre ❖ date 	<ul style="list-style-type: none"> -le contenu de la demande -date 	<ul style="list-style-type: none"> -VARCHAR (30) -datetime 	
images	<ul style="list-style-type: none"> ❖ Nis-Im ❖ img_nom ❖ img_taille ❖ img_type ❖ img_desc ❖ img_blob 	<ul style="list-style-type: none"> -Identificateur de l'image -Nom de l'image -Taille de l'image -Type de l'image -description sur l'image -Le contenu du l'image En binaire 	<ul style="list-style-type: none"> -Int(11) -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) -VARCHAR (30) -blob 	Ajouter image
recour	<ul style="list-style-type: none"> ❖ NIS_reco ❖ Ai ❖ Nom ❖ Prenom ❖ Profession ❖ Adresse ❖ Lettre ❖ Date 	<ul style="list-style-type: none"> -identificateur de recours -Article d'imposition -nom de contribuable -prénom de contribuable -Profession de contribuable -adresse de contribuable -le contenu de la demande -date 	<ul style="list-style-type: none"> -VARCHAR (30) -datetime 	Ajouter recours

Tableau III.1: Dictionnaire des données.

III.4 Réalisation des diagrammes de classes

III.4.1 Les règles de gestion entre les classes

- Un contribuable peut remplir une ou plusieurs réclamations.
- Une et une seule réclamation appartient à un et un seul contribuable.
- Un contribuable peut remplir un et un ou plusieurs recours.
- Un et un seul recours appartient à un et un seul contribuable.
- Un contribuable peut envoyer plusieurs messages.
- Un message peut être envoyé par un et un seul contribuable.
- Un ou plusieurs messages appartient à un et un seul contribuable.
- Un utilisateur peut envoyer un ou plusieurs messages.
- Un ou plusieurs messages appartient à un et un seul utilisateur.
- Un message peut être envoyé par un et un seul utilisateur.
- Un et un seul utilisateur peut gérer un ou plusieurs fiches.
- Une ou plusieurs fiches peut être gérées par un et un seul utilisateur.
- Une et une seule fiche appartient à un et un seul contribuable.

III.4.2 Principes et définitions de base

Dans ce qui suit, nous définissons les concepts utilisés dans un diagramme de classe :
[W15]

- ❖ **Objet** : Un objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer a priori les valeurs de ses propriétés.
- ❖ **Classe** : Une classe décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition).
- ❖ **Attribut** : Un attribut est une propriété élémentaire d'une classe. Pour chaque objet d'une classe, l'attribut prend une valeur (sauf cas d'attributs multi-values).
- ❖ **Opération** : Une opération est une fonction applicable aux objets d'une classe. Une opération permet de décrire le comportement d'un objet.
- ❖ **Association** : Une association entre classes représente les liens qui existent entre les instances de ces classes. Chaque association peut être identifiée par son nom.
- ❖ **Multiplicité** : La multiplicité indique un domaine de valeurs pour préciser le nombre d'instance d'une classe vis-à-vis d'une autre classe pour une association donnée.
- ❖ **Navigabilité** : la navigabilité si l'association de manière unidirectionnelle ou bidirectionnelle, elle est matérialisée par une ou deux extrémités fléchées. Le non-navigabilité se représente par "X".
- ❖ **Agrégation** : L'agrégation est une association qui permet de représenter un lien de type "ensemble" comprenant des "éléments". Il s'agit d'une relation entre une classe représentant le niveau "ensemble" et 1 à n classes de niveau "éléments". L'agrégation représente un lien structurel entre une classe et une ou plusieurs autres classes.
- ❖ **Composition** : La composition est une relation d'agrégation dans laquelle il existe une contrainte de durée de vie entre la classe "composé" implique la suppression de la ou des classes "composant".

- ❖ **Généralisation, spécialisation** : La généralisation est la relation entre une classe et deux autres classes ont plus partageant un sous-ensemble commun d'attributs et/ou d'opération. La classe qui est affinée s'appelle super classe, les classes partir de classes s'appelle la généralisation. Inversement la spécialisation consiste à créer des sous classes à partir d'une classe.
- ❖ **Classe abstraite** : Une classe abstraite est une classe qui n'a pas d'instance directe mais dont les classes descendantes ont des instances. Dans une relation d'héritage, la super classe est par définition une classe abstraite.

III.4.3 Diagramme de classe

Définition:

Les diagrammes de classe permettent de spécifier la structure et les liens entre les objets dont le système est composé. Il spécifie qui sera à l'œuvre dans le système pour réaliser les fonctionnalités décrites par les diagrammes de cas d'utilisation [W14].

La figure III.11 sise ci-dessous présente le diagramme de classe de notre application :

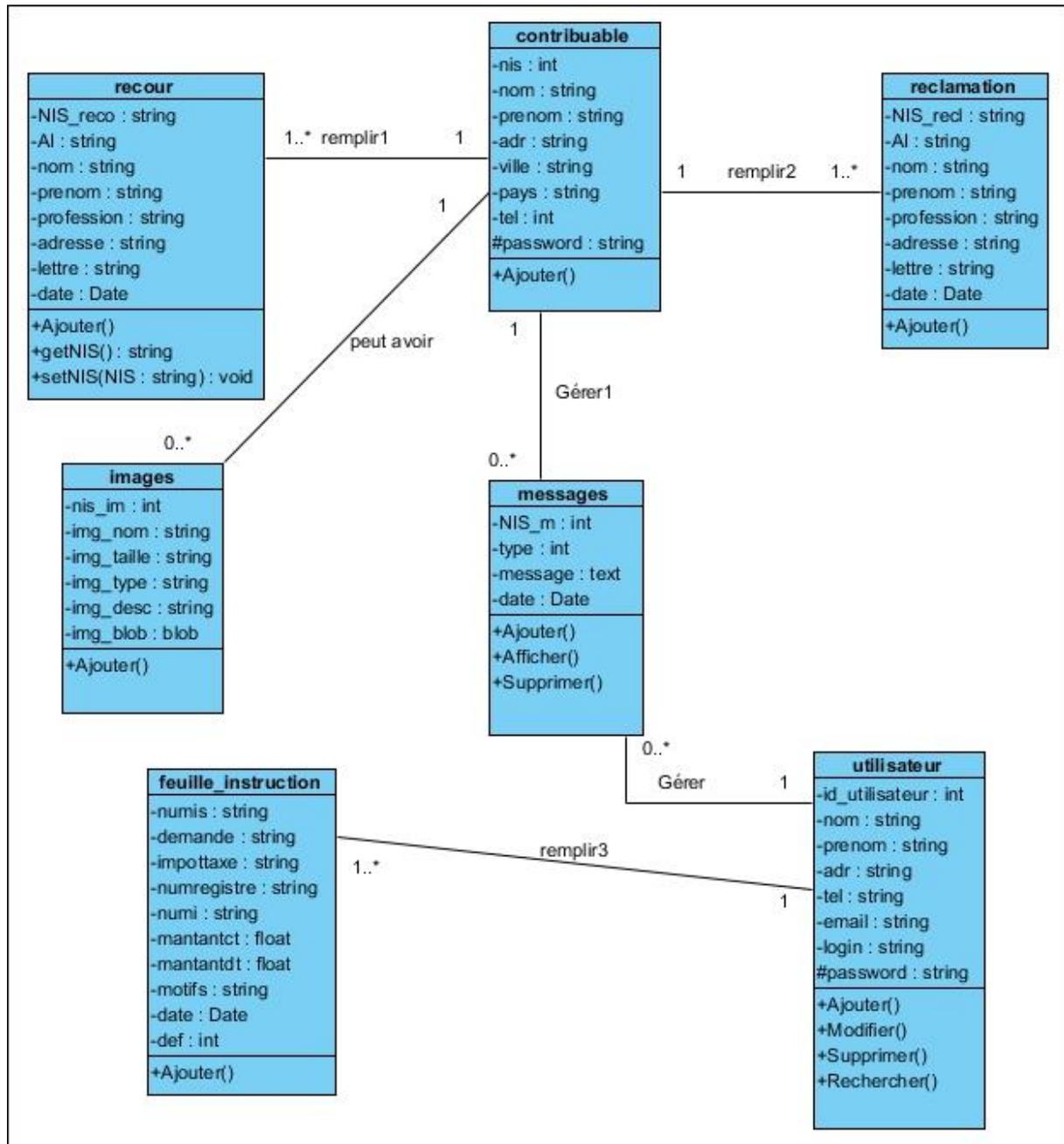


Figure III.11 : Diagramme de classe.

III.5 Le modèle relationnel

Le modèle relationnel doit refléter le diagramme de classes issu de l'analyse, et donc les éléments présents dans le diagramme des classes (classe, attribut, association, classe association) doivent se retrouver dans le modèle relationnel [B1].

III.5.1 Règles de passage au modèle relationnel

Règle 1 : Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant. Si aucun attribut ne convient en

tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire.

Règle 2 : Un-à-plusieurs. 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.

Règle 3 : Plusieurs-à-plusieurs ou classes-association de n-aires l'association devient une relation dont la clé primaire est composée par concaténation des identifiants des classes connectées à l'association. Les attributs de l'association doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

Règle 4 : Transformation de l'héritage il faut transformer chaque sous-classe en une relation. La clé primaire de la sous-classe migre dans la (les) relation(s) issue(s) de la (des) sous-classes et devient à la fois clé primaire à la clé étrangère.

Règle 5 : Transformation d'une composition la clé primaire des relations déduites des classes composantes doit contenir l'identifiant de la classe composite (quelles que soient les multiplicités) [B1].

III.5.2 Modèle relationnel pour notre cas

- ❖ **Utilisateur** (Id_utilisateur, nom, prenom, adr, tel, email, login, password).
- ❖ **Contribuale** (nis, nom, prenom, adr, ville, pays, tel, password, date, #id_utilisateur).
- ❖ **Feuille_instruction** (Numis, Demande, impottaxe, numregistre, numi, mantantct, mantantdt, motifs, date, #id_utilisateur).
- ❖ **Messages** (NIS_m, Type, message, date).
- ❖ **Réclamation** (NIS_recl, Ai, Nom, Prenom, Profession, Adresse, Lettre, date, #nis).
- ❖ **Recours** (NIS_reco, Ai, Nom, Prenom, Profession, Adresse, Lettre, Date, #nis).
- ❖ **Image** (Nis-Im, img_nom, img_taille, img_type, img_desc, img_blob).

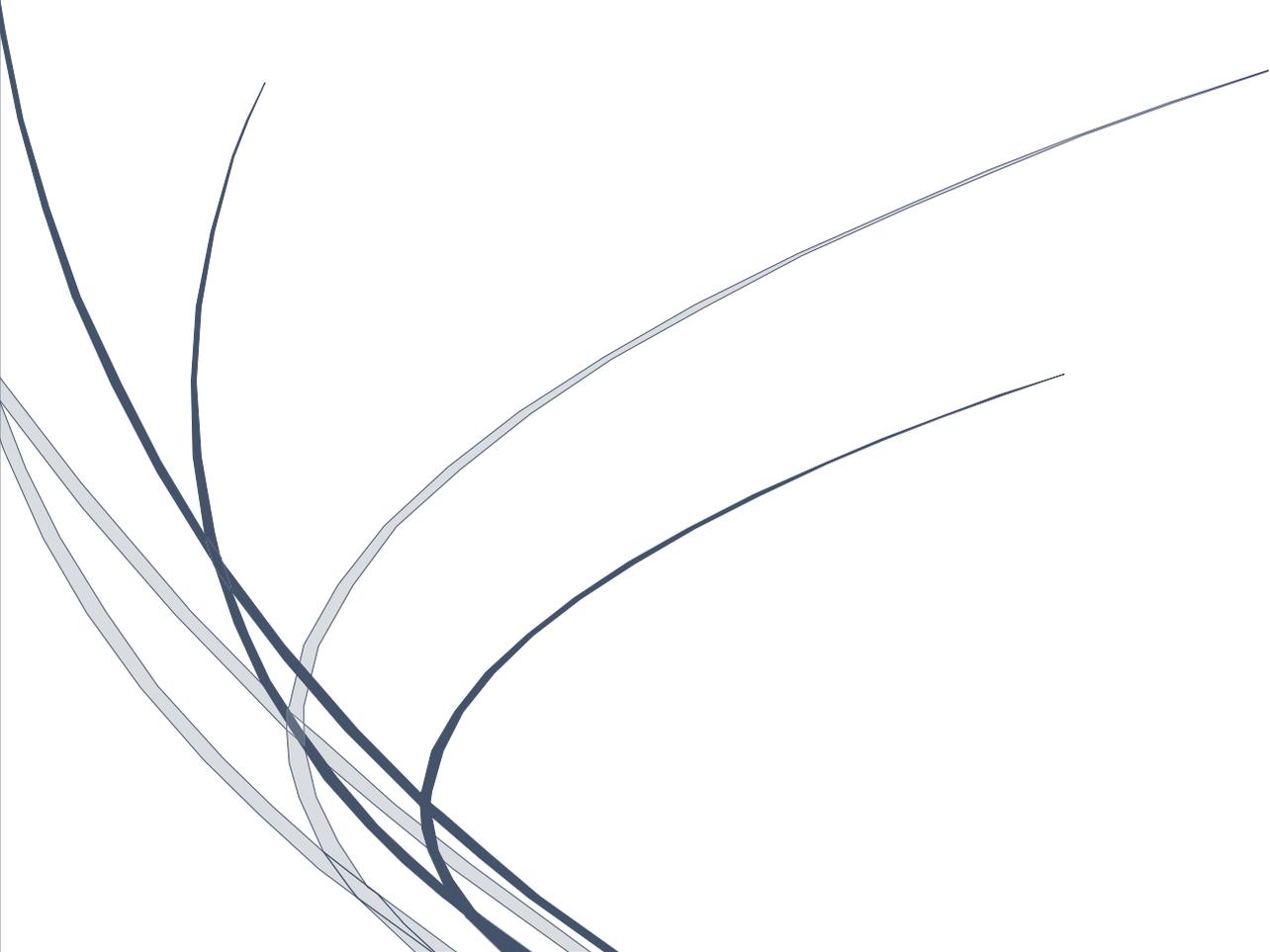
III.6 Conclusion

Ce chapitre a été consacré à la conception de notre application web. Nous avons exposé les différentes interactions entre les objets du système grâce aux diagrammes de séquence détaillés. Sur ce, nous sommes arrivés à l'obtention du modèle relationnel déduit à partir du diagramme de classe.

Désormais, la phase de conception est terminée, nous pouvons s'étaler alors sur l'activité finale de notre projet qui est la phase de réalisation et de mise en œuvre qui sera l'objectif de chapitre suivant.



Chapitre IV : Réalisation



IV.1 Introduction

Après avoir terminé la partie de conception, nous allons entamer la dernière phase de notre projet, une phase qui consiste à la mise en pratique de tous ce qui est déjà présenté dans le chapitre précédent.

Dans ce chapitre nous nous présenterons l'environnement et les outils de développement de notre application, nous expliquerons son fonctionnement et nous présenterons l'organigramme d'interaction est quelques interfaces illustratives.

IV.2 Outils et environnement de développement

IV.2.1 Code Visual Studio

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C ++, C #, Java, Python, PHP, Go) et des environnements d'exécution (tels que .NET et Unity) [W16].

Le logo du Visual Studio est donné par la figure ci-dessus (Figure IV.1) :



Figure IV.1 : Logo du Code Visual Studio.

IV. 2.2 WampServer

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL [W17]. Le logo du wampServer est donné par la figure ci-dessus (Figure IV.2) :



Figure IV.2 : Logo du wampServer.

IV. 2.3 Système d'exploitation Windows

Windows (littéralement « fenêtre » en anglais) est au départ une interface graphique unifiée produite par Microsoft, qui es devenue ensuite une gamme de systèmes d'exploitation à part entière, principalement destinés aux ordinateurs compatibles PC [W18].

Le logo du système d'exploitation Windows est donné par la figure ci-dessus (Figure IV.3) :



Figure IV.3 : Logo du système exploitation Windows.

IV.2.4 Serveur apache

Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde. Le nom officiel est Serveur Apache HTTP et il est maintenu et développé par Apache Software Fondation.

Il permet aux propriétaires de sites web de servir du contenu sur le web d'où le nom « serveur web ». C'est l'un des serveurs web les plus anciens et les plus fiables avec une première version sortie il y a plus de 20 ans, en 1995 [W19].

Le logo du Serveur apache est donné par la figure ci-dessus (Figure IV.4) :



Figure IV.4 : Logo du Serveur apache.

IV.2.5 MySQL

MySQL est un gestionnaire de base de données libre. Il est très utilisé dans les projets libres et dans le milieu industriel. En somme, grâce à ce logiciel, nous allons pouvoir créer notre propre base de données. Nous allons pouvoir classer tous ces éléments (patient médecin, message ...) et les rappeler quand bon nous semble. MySQL est très souvent utilisé avec PHP [W20].

Le logo du MySQL est donné par la figure ci-dessus (Figure IV.5) :



Figure IV.5 : Logo du MySQL.

IV.2.6 JavaScript

JavaScript (qui est souvent abrégé en « JS ») est un langage de script léger, orienté objet principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans

de nombreux environnements extérieurs aux navigateurs web tels que Node.js, Apache CouchDB voire Adobe Acrobat. Le code JavaScript est interprété ou compilé à la volée (JIT). C'est un langage à objets utilisant le concept de prototype, disposant d'un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation [W21].

Le logo du langage JavaScript est donné par la figure ci-dessus (Figure IV.6) :



Figure IV.6 : Logo du langage JavaScript.

IV.2.7 PHP

PHP est un langage de programmation informatique essentiellement utilisé pour produire à la volée des pages web dynamiques. Dans sa version 5 lancée en juillet 2004, PHP s'est imposé comme le langage de référence sur le web en raison de sa simplicité, de sa gratuité et de son origine de logiciel libre [W22].

Le logo du langage PHP est donné par la figure ci-dessus (Figure IV.7) :



Figure IV.7 : Logo du langage PHP.

IV.2.8 CSS3

CSS est l'acronyme de « Cascading Style Sheets » ce qui signifie « feuille de style en cascade ». Le CSS est un langage informatique permettant de mettre en forme des pages web (HTML ou XML).

Ce langage est donc composé des fameuses « feuilles de style en cascade » également appelées fichiers CSS (.css) et contient des éléments de codage [W23].

Le logo du langage CSS3 est donné par la figure ci-dessus (Figure IV.8) :



Figure IV.8 : logo du langage CSS3.

IV.2.9 HTML

HTML signifie « HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. D'autres technologies sont utilisées avec HTML pour décrire la présentation d'une page (CSS) et/ou ses fonctionnalités interactives (JavaScript) [W24].

Le logo du langage HTML est donné par la figure ci-dessus (Figure IV.9) :



Figure IV.9 : Logo du langage Html.

IV.2.10 Visual Paradigm pour uml

Visual Paradigm est un logiciel de création de diagrammes dans le cadre d'une programmation. Tout en un, il possède plusieurs options permettant une large possibilité de modélisation en ULM [W25].

Le logo du logiciel Visual Paradigm est donné par la figure ci-dessus (Figure IV.10) :



Figure IV.10 : Logo du logiciel Visual Paradigm.

IV.2.11 Bootstrap3

Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsif, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons des interfaces de navigation et bien d'autres encore. On appelle ce type de framework un "Front-End Framework"[W26].

Le logo du langage Bootstrap est donné par la figure ci-dessus (Figure IV.11) :



Figure IV.11 : Logo du langage Bootstrap.

IV.3 Organigramme de l’application

IV.3.1 Coté administrateur

Le schéma ci-après, présente dans la figure (IV.12) récapitule l'organisation des pages de L’application (Organigramme de l’application coté administrateur) :

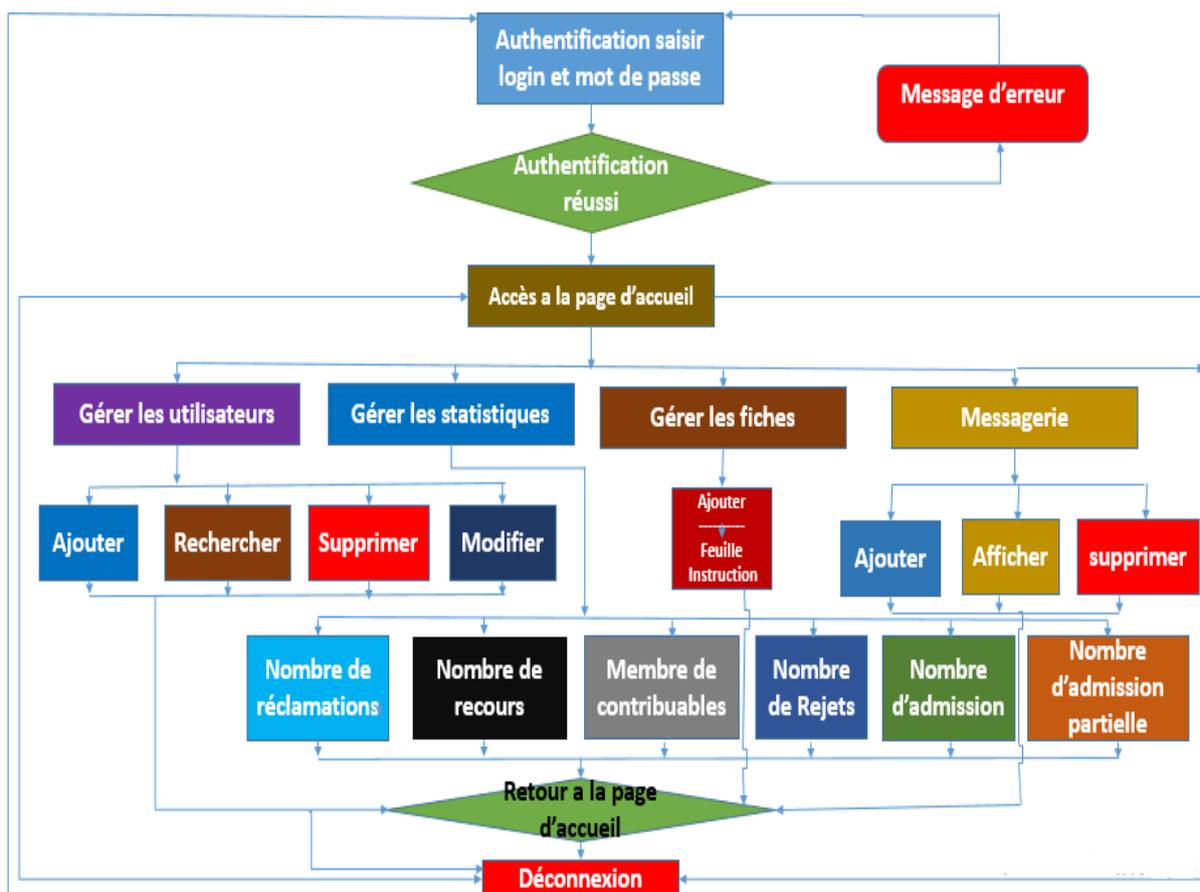


Figure IV.12 : Organigramme de l’application coté administrateur

IV.3.2 Coté Simple Employé

Le schéma ci-après, présente dans la figure (IV.13) récapitule l'organisation des pages de L’application (Organigramme de l’application coté simple employé) :

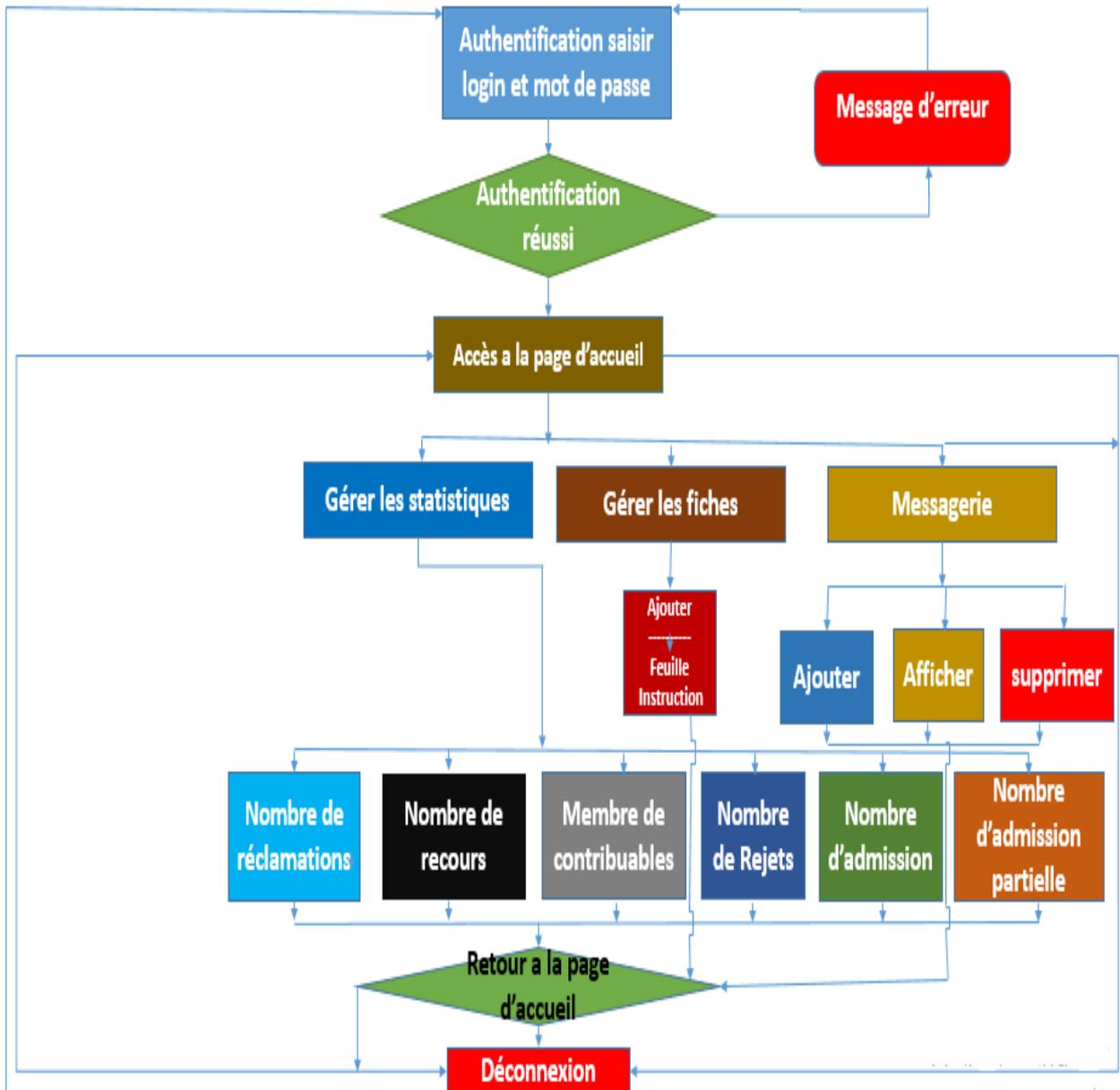


Figure IV.13 : Organigramme de l’application coté simple employé.

IV.3.3 Coté contribuable

Le schéma ci-après, présente dans la figure (IV.14) récapitule l'organisation des pages de l’application (Organigramme de l’application coté contribuable) :

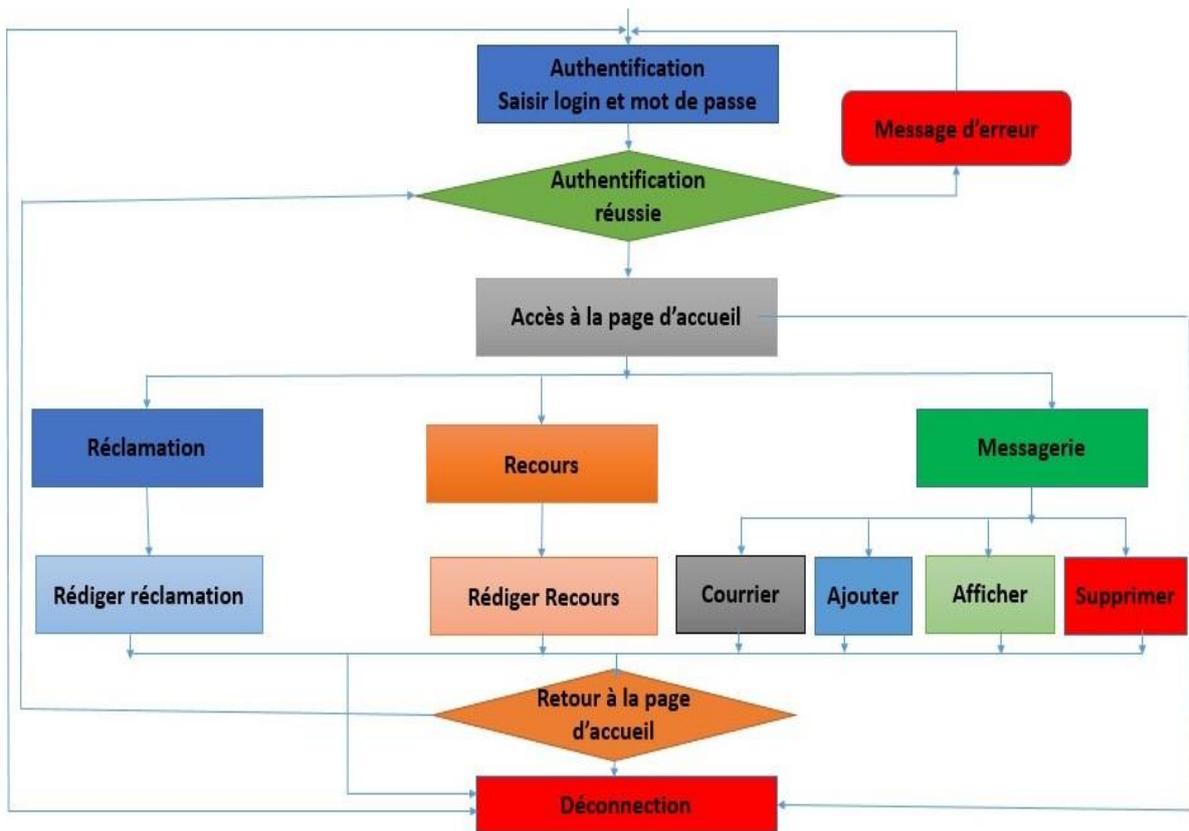


Figure IV.14 : Organigramme de l'application coté contribuable.

IV.4 Description des interfaces de l'application

La conception des interfaces de l'application est une étape très importante puisque toutes les interactions avec le cœur de l'application passent à travers ces interfaces. On doit alors guider l'utilisateur avec les messages d'erreurs et de notification si besoin et présenter ainsi un système complet.

IV.4.1 Espace Contribuable

IV.4.1.1 Interface d'authentification

Quand vous lancerez l'application, l'interface « authentification » (voir Figure IV.15) apparaîtra sur l'écran. Cette étape permet de sécuriser les données de contribuable et de protéger l'application. Le contribuable doit s'identifier par son login (Numéro d'identification statistique qui est unique) et son mot de passe pour pouvoir accéder à page d'accueil:

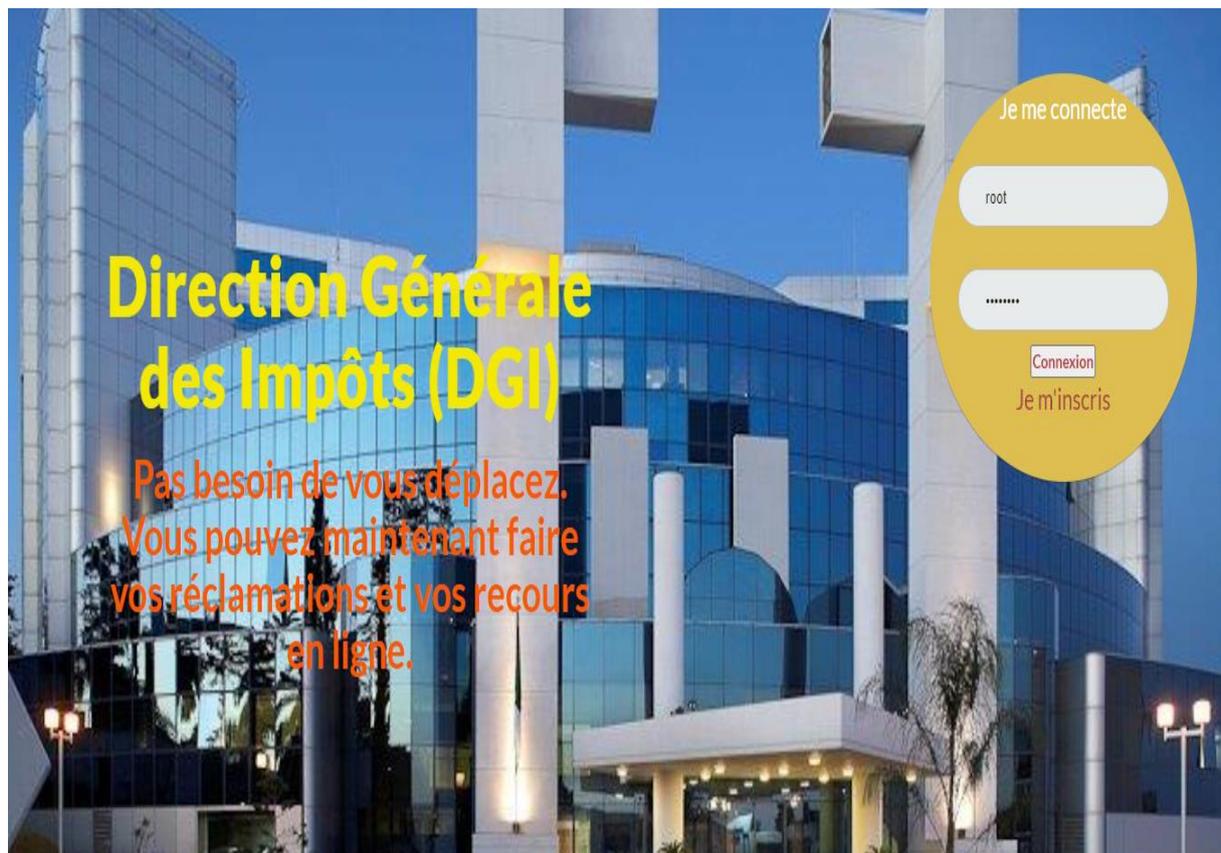


Figure IV.15 : Page authentification.

IV.4.1.2 Interface Inscription

L’interface illustrée par la figure (IV.16) permet au contribuable de s’inscrire à la base de données:

Ici vous pouvez vous inscrire

NIS:

Nom :

Prénom :

Adresse :

Ville :

Pays :

Telephone :

mots de passe :

confirmer votre mot de passe :




Figure IV.16 : Ajouter contribuable.

IV.4.1.3 Page d’accueil

La page d’accueil (voir Figure IV.17) représente l’espace réservé au contribuable, elle s’affiche une fois que ce dernier accède à l’application après authentification.



Figure IV.17 : Page d’accueil coté contribuable.

IV.4.1.4 Interface Réclamation

L'interface « réclamation » représentée par la figure (IV.17) permet au contribuable de rédiger une réclamation.

The screenshot shows a web interface for filing a complaint. At the top right, there are links for 'DÉCONNEXION' and 'ACCUEIL'. The main heading is 'Réclamations'. The form contains a label 'Article D'imposition:' followed by a text input field with the placeholder text 'Article D'imposition'. Below this is a label 'lettre:' followed by a large text area for writing the complaint. A blue button labeled 'envoyer' is positioned below the text area. At the bottom of the page, there is a label 'choisir un fichier', a 'Browse' button, and a 'Cancel' button.

Figure IV.18 : Interface réclamation.

IV.4.1.5 Interface messagerie

L'interface « messagerie » permet aux contribuables de voir les messages, comme l'illustre la figure (IV.19) :

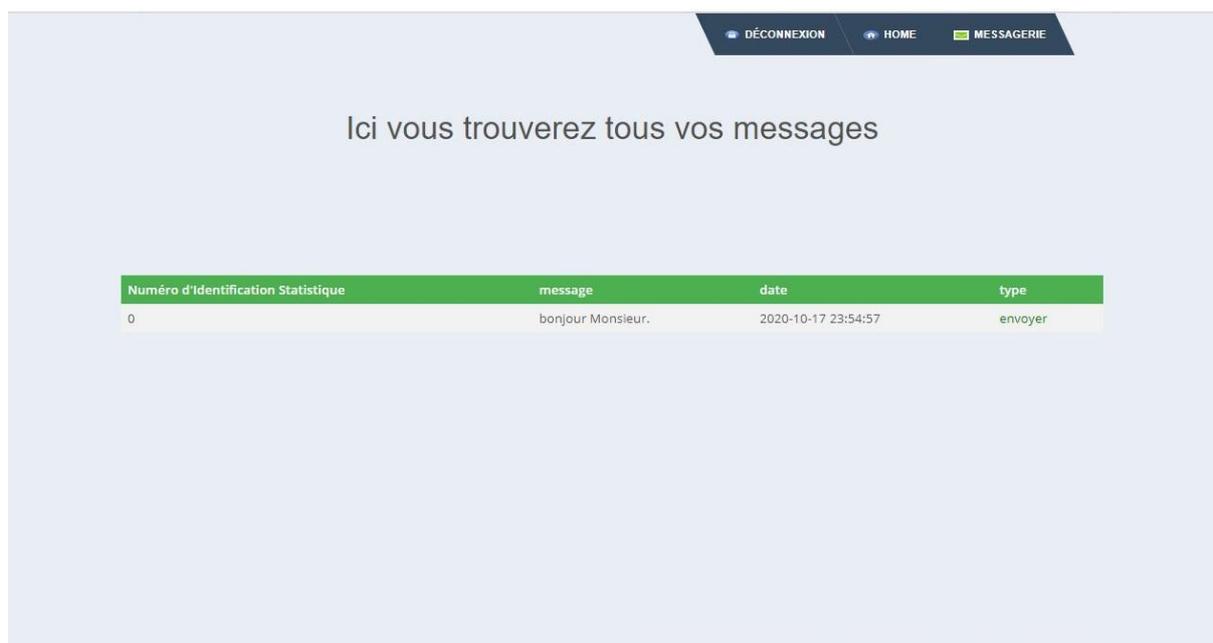


Figure IV.19 : Interface messagerie.

IV.4.1.6 Interface ajouter message

L'interface « ajouter message » permet aux contribuables d'envoyé les messages, comme l'illustre la figure (IV.20) :



Figure IV.20 : Interface ajouter message.

IV.4.2 Espace Admin, utilisateur (simple employé)

IV.4.2.1 Interface d'authentification

Quand vous lancerez l'application, l'interface « authentification » (illustré par la figure IV.21) apparaîtra sur l'écran. Cette étape permet de sécuriser les données de l'administrateur et de l'employé et protéger l'application.

L'employé doit s'identifier par son login et son mot de passe pour pouvoir accéder à l'interface « page d'accueil »:

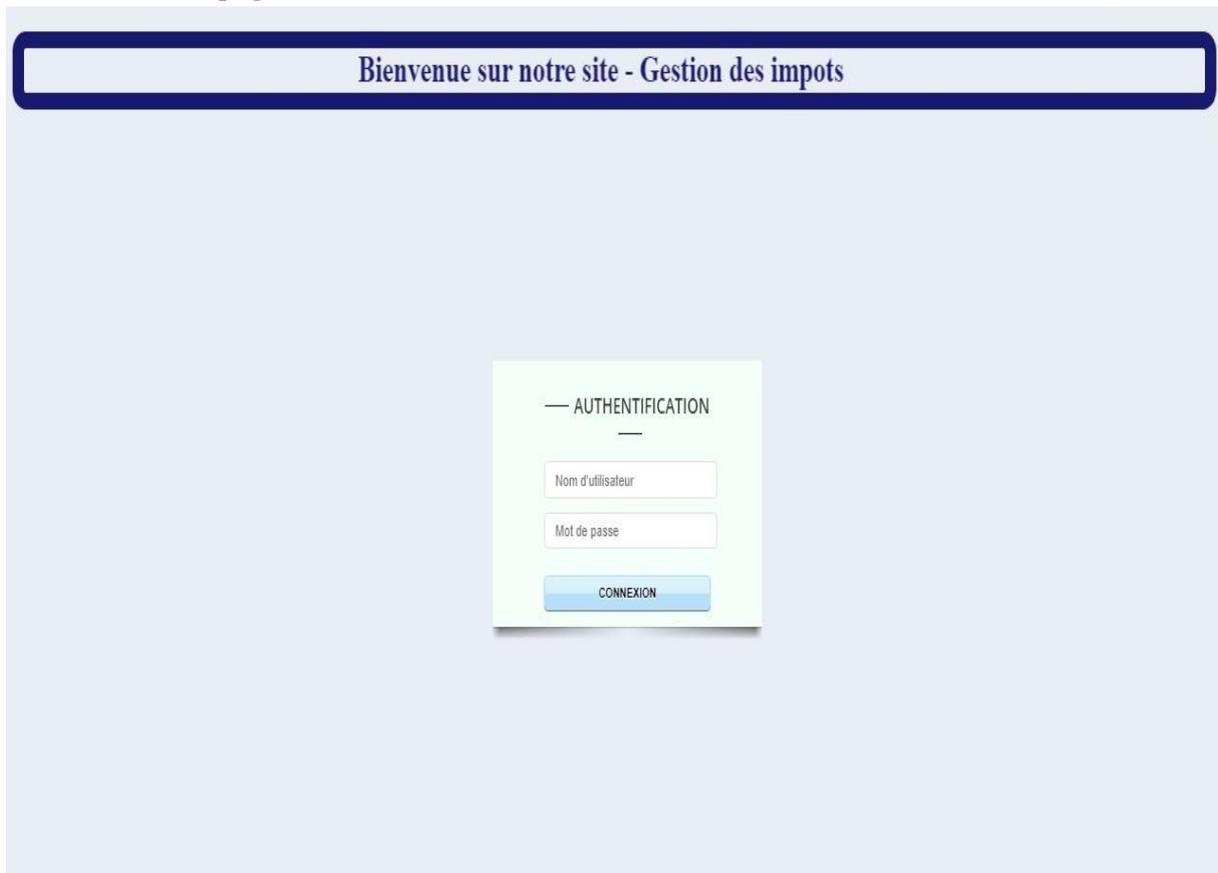


Figure IV.21 : Interface authentification Employés.

IV.4.2.2 Page d'accueil « coté administrateur »

Si le système valide les mots de passe de l'administrateur, l'interface principale (voir la figure IV.22) sera affichée, où le l'administrateur peut avoir toutes les fonctionnalités.



Figure IV.22 : Interface accueil « coté administrateur ».

IV.4.2.3 Page d'accueil « coté simple employé »

Si le système valide les mots de passe de l'employé, l'interface principale (voir la figure IV.23) sera affichée, où l'employé peut avoir ses fonctionnalités.



Figure IV.23 : Page d'accueil simple employé.

IV.4.2.4 Interface Statistiques

L'interface de la figure (IV.24) permet aux employés de voir toutes les statistiques.

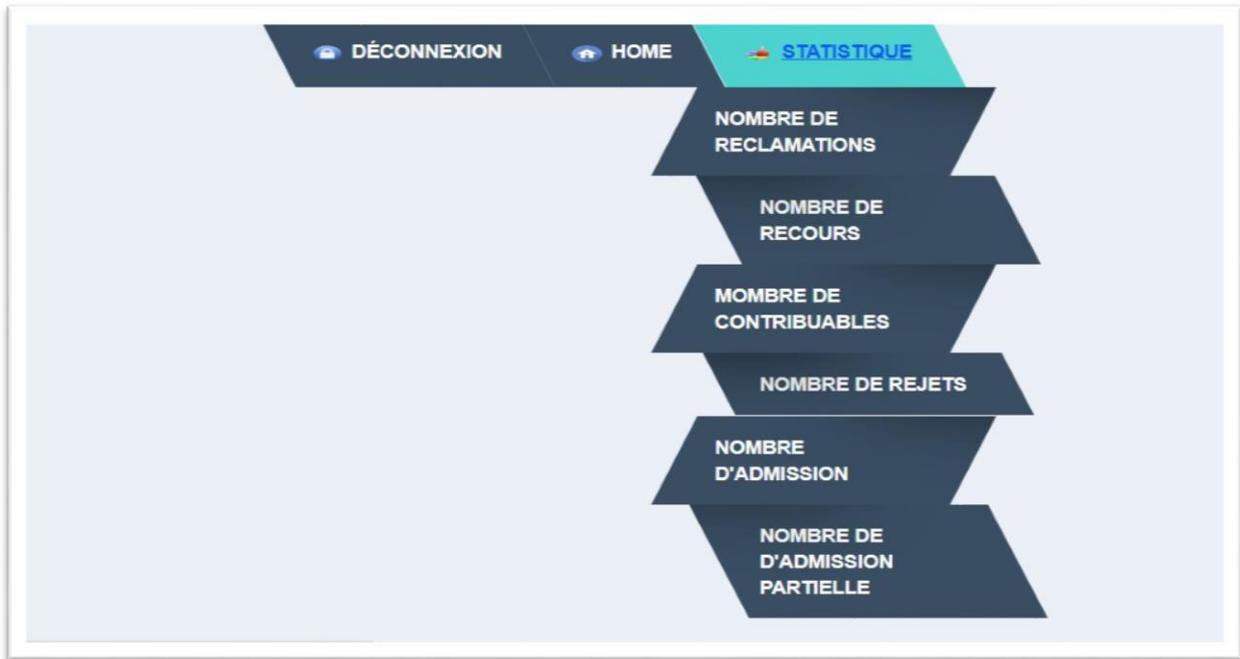


Figure IV.24: interface statistiques.

IV.4.2.5 Interface nombre de contribuables dans une date précise

L’interface présentée par la figure (Figure IV.25) permet à l’employé de consulter le nombre de contribuables dans un intervalle de temps :

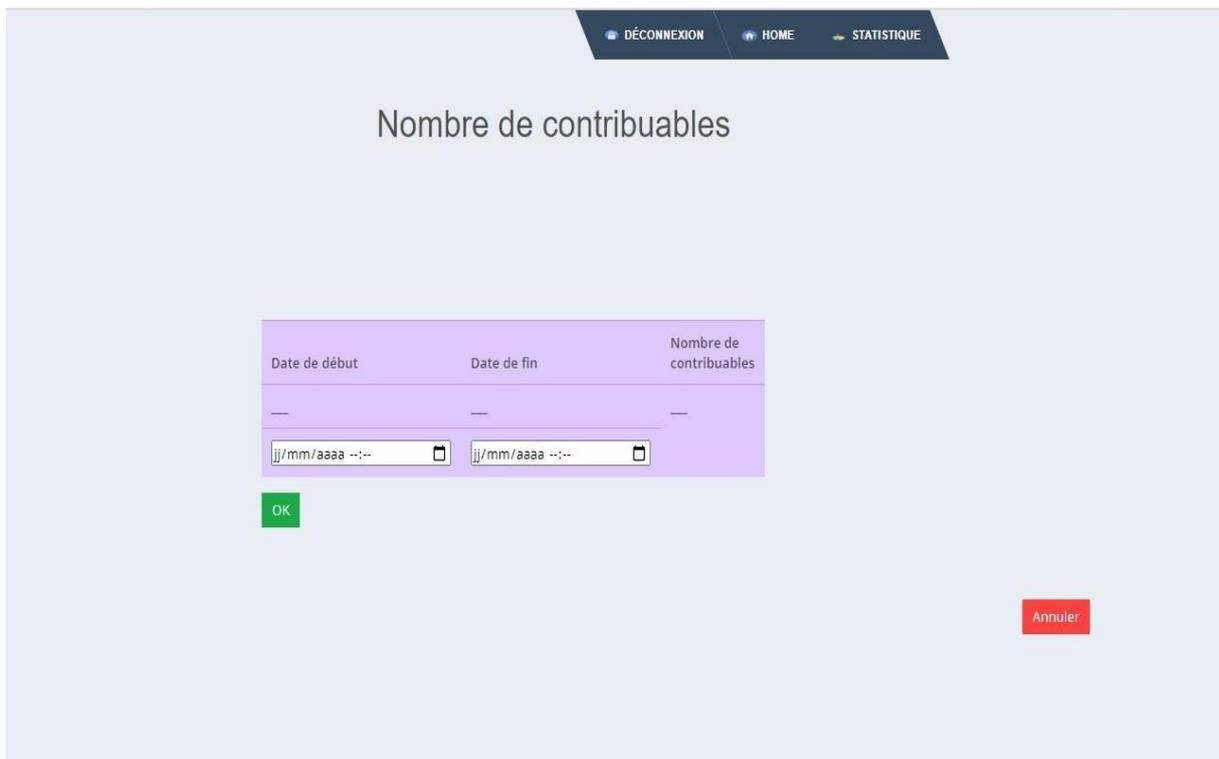


Figure IV.25 : interface nombre de contribuables.

IV.4.2.6 Interface de la feuille d'instruction

L'interface présentée par la figure (Figure IV.26) permet à l'employé de saisir la feuille d'instruction :

RECLAMATION ADRESSEE AU DIRECTEUR DES IMPOTS DE WILAYA
 (Impot Derects TVA Enregistrement Recouvrement)
FEUILLE D'INSTRUCTION
 CONTENTIEUX-RECOURS-GRACIEUX

Annulation ou Reduction:

Annulation
 Reduction

IMPOT OU TAXE:

IMPOT
 TAXE

Numéro de Registre:

valider
Annuler

Figure IV.26 : interface feuille d'instruction.

IV.4.2.7 Interface résultat sur la réclamation

L’interface présentée par la figure (Figure IV.27) permet au contribuable d’avoir la réponse sur sa réclamation (rejet totale, admission partielle, admission totale).

Exemple : Un Dégrèvement.

REPUBLIQUE ALGERIEN DEMOCRATIQUE ET POPULAIRE

MINISTRE DES FINANCE

DERECTION GENERALE DES IMPOTS

DERECTION IMPOTS DE WILAYA
DE: **bejaia**

Affair suivie par:
.....
.....
.....

A.R

AVIS DE DEGREVEMENT

Admission par le Directeur des impôts de wilaya d'une demande en décharge ou réduction

M: **fortas aziz**

NIF:**1234567**

A BEJAIA,le 2020-09-10 16:25:58

Madame, Monsieur,
vous m'avez adressé une réclamation concernant l'imposition visée ci-centre.
Après examen contentieux, j'ai décidé d'admettre votre réclamation dans son intégralité.
Les dégrèvements ou restitution prononcés sont indiqués au verso.
Si vous avez déjà payé l'impôt dont le dégrèvement vous est accordé et que vous n'etes redevable d'aucune somme auprès du receveur, le montant dégrèvé

(Role n°,art.....)

Avis de mise en recouvrement,

N°

Du.....

Demande en restitution

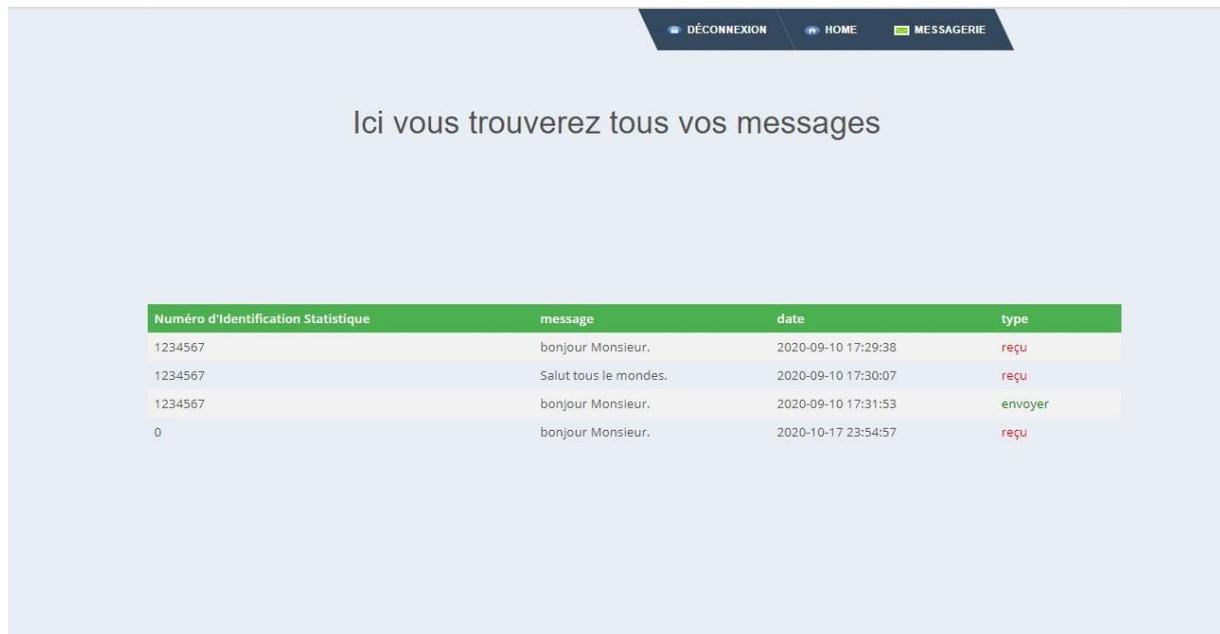
DEGREVEMENTS OU RESTITUTION ACCORDES:

Exercice	N° du rôle	impôts ou taxes	Mantant du dégrèvement ou restituion
---	---	TAXE	120

Figure IV.27: Interface de dégrèvement (admission totale).

IV.4.2.8 Interface messagerie

L'interface présentée par la figure (Figure IV.28) permet à l'employé de voir ou d'envoyer des messages :



Numéro d'Identification Statistique	message	date	type
1234567	bonjour Monsieur.	2020-09-10 17:29:38	reçu
1234567	Salut tous le mondes.	2020-09-10 17:30:07	reçu
1234567	bonjour Monsieur.	2020-09-10 17:31:53	envoyer
0	bonjour Monsieur.	2020-10-17 23:54:57	reçu

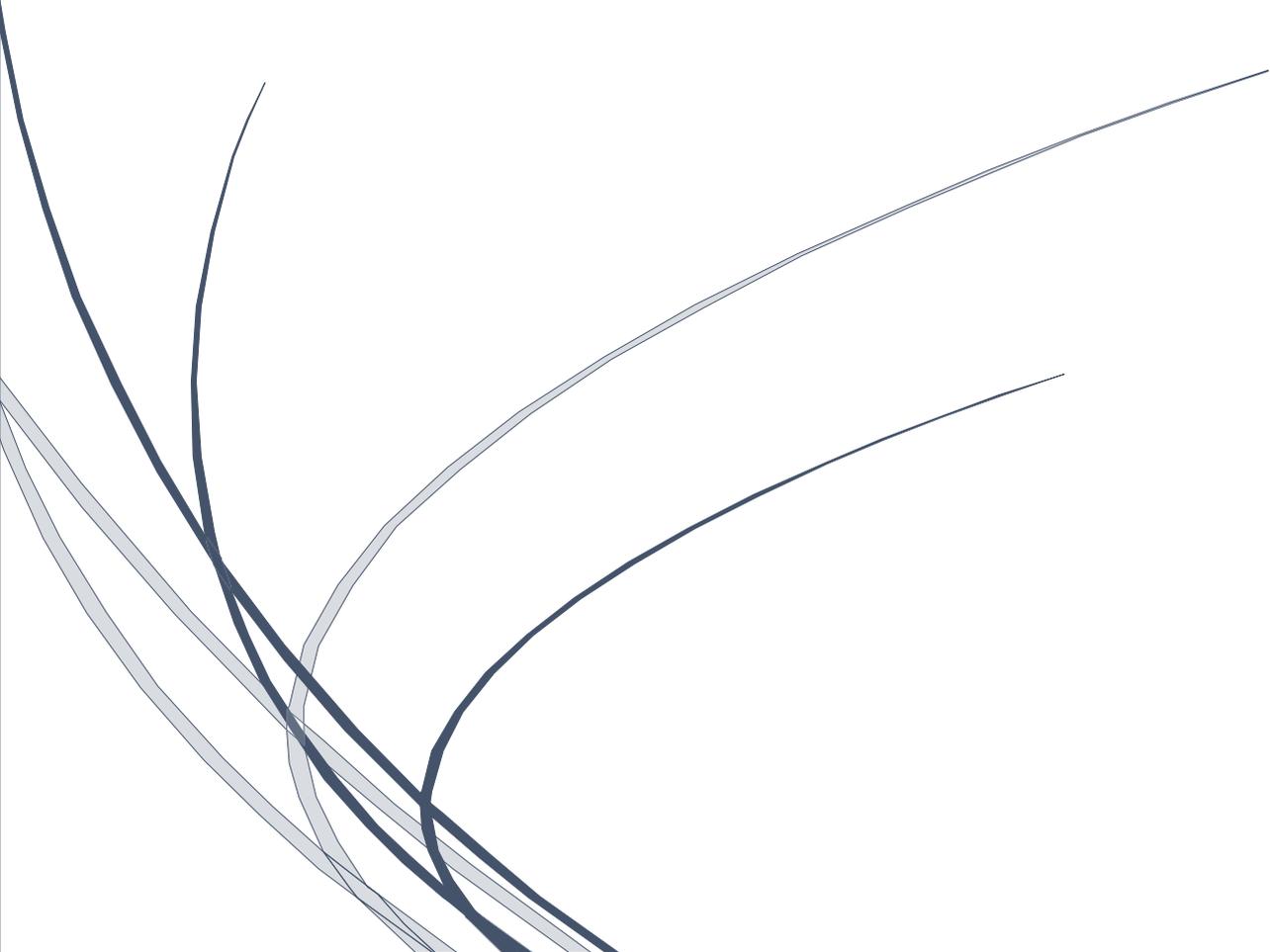
Figure IV.28 : Interface messagerie.

IV.5 Conclusion

La phase de réalisation est une étape très importante dans le cycle de vie d'une application. Dans ce dernier chapitre, nous avons illustré le déroulement de la réalisation de notre application, nous avons présenté les outils et le langage de programmation. En fin, on a présenté les différentes interfaces de notre application.



Conclusion Générale



Conclusion générale

Au cours de ce travail, nous avons présenté les différentes étapes ayant conduit à la mise en œuvre d'une application web dédiée à la gestion des contentieux au sein de la direction des impôts de Bejaia.

Nous avons commencé par recenser les difficultés que rencontre la sous-direction du contentieux afin d'apporter la solution adéquate et spécifier ainsi les besoins.

Pour réaliser l'application, nous avons utilisé le 'PhpMyAdmin MYSQL' pour implémenter la base de données et le langage PHP et JavaScript pour la programmation et HTML5, CSS3, bootstrap3 pour les interfaces.

Ce projet a fait l'objet d'une expérience intéressante, très bénéfique pour nous. En effet, il nous a permis d'enrichir nos connaissances théoriques et nos compétences dans le domaine de la conception et de la programmation. Ajoutant à ceci, la mise en application des connaissances acquises tout au long de nos études.

En plus, c'était une bonne occasion pour réaliser un travail concret avec des objectifs clairs et bien définis. Ce projet nous a également permis de nous familiariser avec l'environnement de travail et de la vie professionnelle.

Ainsi, le stage que nous avons effectué au sein des impôts a été pour nous une expérience très enrichissante et très intéressante, dans la mesure où il nous a permis de mettre en pratique et de développer les connaissances que nous avons acquises tout au long de notre cursus universitaire.

Enfin, nous espérons que nous avons pu construire une base de départ pour les améliorations futures.

La webographie

[W1] : <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/> /Date de consultation 14/7/2020.

[W2] : <https://www.cc-paysviganais.fr/contacts/impots/> / Date de consultation 1/8/2020.

[W3] : <https://www.mfdgi.gov.dz/index.php/com-smartslider3/2014-03-24-14-21-47/services-externes/113-direction-des-impots-de-wilaya> /Date de consultation 10/8/2020.

[W4] : <https://stph.scenari-community.org/bdd/lap2/co/webUC003archi.html> /Date de consultation 9/6/2020.

[W5] : <https://whatis.techtarget.com/fr/definition/Serveur-Web> /Date de consultation 9/7/2020.

[W6] : <https://archive.awt.be/web/res/index.aspx?page=res,fr,fic,040,002> /Date de consultation 4/6/2020.

[W7] : <https://www.protaiin.com/lhebergement-dun-site-web-quest-cest/> /Date de consultation 27/7/2020.

[W8] : <https://www.techno-science.net/definition/749.html> /Date de consultation 6/6/2020

[W9] : <https://support.google.com/analytics/answer/2731565?hl=fr> /Date de consultation 15/7/2020.

[W10] : <https://ensweb.users.info.unicaen.fr/pres/sessions/> /Date de consultation 4/7/2020.

[W11] : <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/> /Date de consultation 4/8/2020.

[W12] : <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation#L2-2-2/> /Date de consultation 28/8/2020.

[W13] : <https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2094981-la-description-textuelle-d-un-cas-d-utilisation/> /Date de consultation 28/8/2020.

[W14] : <https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours05.html/> /Date de consultation 9/7/2020.

[W15] : <https://ult.bi/v2/?q=student/chapitre-ii-presentation-du-langage-de-modelisation-uml/> /Date de consultation 28/8/2020.

[W16] : <https://code.visualstudio.com/docs/> /Date de consultation 28/8/2020.

[W17] : <https://fr.wikipedia.org/wiki/WampServer/> /Date de consultation 28/8/2020.

[W18] : https://fr.wikipedia.org/wiki/Microsoft_Windows / Date de consultation 28/8/2020.

[W19] : <https://www.hostinger.fr/tutoriels/quest-ce-quapache-serveur-web-apache/> Date de consultation 28/8/2020.

[W20] : <https://www.host4all.net/mutualise/mysql/> Date de consultation 28/8/2020.

[W21] : <https://developer.mozilla.org/fr/docs/Web/JavaScript/> Date de consultation 28/8/2020.

[W22] : <https://www.jobintree.com/dictionnaire/definition-php-309.html/> Date de consultation 28/8/2020.

[W23] : <https://www.atinternet.com/glossaire/> Date de consultation 28/8/2020.

[W24] : <https://developer.mozilla.org/fr/docs/Web/HTML/> Date de consultation 28/8/2020.

[W25] : <https://www.commentcamarche.net/download/telecharger-34058460-visual-paradigm/> Date de consultation 28/8/2020.

[W26] : <https://disciplines.ac-montpellier.fr/economie-gestion/bootstrap-creation-de-site-avec-un-design-responsive/> Date de consultation 28/8/2020.

La bibliographie

[B1] : SAICHE Cylia, OUYOUGOUTE Abdelatif. Conception et réalisation d'une application web pour la gestion des étudiants d'une école privée. Mémoire de fin de cycle 2014 université de Bejaia.

[B2] : FOTCHA Mouafo. Didacticiel multimédia pour le génie électrique accessible sur la plateforme CLAROLINE. Mémoire de fin d'étude 2007 université de douala.

[B3] : Eskandar Kouicem, Azza Dridi, Med Nadir Boukelal. Conception et réalisation d'un site web dynamique pour la prise de rendez-vous médicale en ligne. Licence en informatique 2016 Université de Constantine 2.

[B4] : MILLOGO Frédéric. Mise en place d'une application webmapping de géolocalisation des points d'intérêt de la ville de Ouagadougou. Master 2 Informatique Appliquée aux Systèmes d'Information Géographique 2012 Université de Douala et de l'AUF.

Résumé

L'utilisation des applications informatiques devient indispensable dans tous les domaines, notamment dans les secteurs qui rencontrent des difficultés pour gérer les différentes tâches et d'avoir un accès rapide aux informations et d'optimiser le temps et l'espace de stockage de données.

En effet, notre projet de fin de cycle consiste à la conception et la réalisation d'une application web pour la gestion des contentieux. Le cas d'étude est la direction des impôts de Bejaïa où les employés et les contribuables représentent les utilisateurs principaux de cette dernière.

Notre application permet de gérer plusieurs tâches : les contribuables, les réclamations, les recours, l'avis sur les réclamations ou recours, les statistiques, les messages.

Ce travail a été réalisé en utilisant le langage de modélisation Unified Modeling Language (UML) afin de schématiser la solution. Nous avons choisi de programmer l'application avec le langage PHP et MYSQL comme Système de Gestion de Base de données (SGBD).

Mots clés: Gestion du contentieux, Application web, UML, MySQL, WampServer, PHP, CSS3, HTML5, Bootstrap3, JavaScript.

Abstract

The use of computer applications is becoming essential in all areas, especially in industries that encounter difficulty in managing different tasks and having quick access to information and optimizing data storage time and space.

Indeed, our end-of-cycle project consists of the design and implementation of a web application for the management of disputes. The case study is the Bejaïa tax department where employees and taxpayers represent the main users of the latter.

Our application allows you to manage several tasks: taxpayers, complaints, and appeals, notice of complaints or appeals, statistics, messages.

This work was carried out using the Unified Modeling Language (UML) modeling language in order to model the solution. We chose to program the application with PHP and MYSQL as a Database Management System (DBMS).

Keywords: Litigation management, Web application, UML, MySQL, WampServer, PHP, CSS3, HTML5, Bootstrap3, JavaScript.