

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



N° Réf :.....

## Centre Universitaire de Mila

Institut des Sciences et de la Technologie

Département de Mathématiques et Informatique

**Mémoire préparé En vue de l'obtention du diplôme de licence**

**En : -Filière informatique général**

*Proposition d'une infrastructure pour  
l'e-administration à base de service web*

**Préparé par:** AGGOUN SALIM

Boukezzoula Aicha  
Bouchloukh Meryem

**Encadré par :** Attia mourad

**Année universitaire : 2013/2014**

# Remerciement

*Tout d'abord, nous tenons à remercier le bon Dieu pour nous avoir illuminés et menés jusqu'ici.*

*Nombreux sont ceux qui nous ont aidé, encouragé, réconforté, soulagé, tout au long de ces années et nous ne saurions leur exprimer notre remerciements autant que nous le souhaiterions ; leurs discussions, leurs conseils, leur présence, nous furent éminemment précieux.*

*Nous tenons à remercier chaleureusement à notre encadreur M. Attia Mourad nous lui somme particulièrement reconnaissant pour sa disponibilité, sa compétence, son soutien, ses conseils judicieux et la confiance dont il nous a fait part lors de la réalisation de ce travail. Merci, car il ne serait pas ce qu'il est sans vous.*

## *DEDICACE SALIM*

Je dédie mon petit Projet de fin d'étude à mon Dieu Allah qui m'a  
Donner la patience et l'encouragement pour finalisé ce travail .

---

A Mon Père, Rachid ,En vous, je voie un père dévoué à sa famille.

Ta présence en toute circonstance m'a maintes fois rappelé

Le sens de la responsabilité .

A Ma Mère, Baya, En vous, je voie la maman parfaite, toujours prête  
a se sacrifier pour le bonheur de ses enfants.

---

A mon frère Ramzi et sa femme faten .

A ma sœur radia et son marie Ibrahim.

A mes oncles, tantes et leurs enfants .

A tous la famille.

Qui je le sais ma réussite est très importante pour vous.

J'exprime ma gratitude à mon encadreur Attia Mourad .

Que Dieu Vous paye pour tous vos bienfaits.

---

Je ne saurais terminer ces dédicaces sans penser Aux mes amis .

Merci pour tous

## *DEDICACE MERJEM*

Au nom de Dieu le Miséricordieux

(Say travail Allah va faire venir observer votre travail et à Son messager et les croyants)  
Dieu Tout-Puissant Nuit divine ne parfumé pas Sauf remercier et à jour ne parfumé pas Sauf  
Subordination. les moments ne parfumé pas Sauf de vous voir

Dieu Tout-Puissant

Pour frapper le message de Le Secrétariat a informé la nation et pour le Prophète de la  
miséricorde et de la lumière des mondes.

Prophète Muhammad paix soit sur lui

Pour le prestige et la dignité à ceux qui m'ont enseigné tendre sans attendre à ceux qui  
portent son nom fièrement Dieu. Je demande à Dieu de donner votre âge pour voir le fruit  
est la cueillette venu après une longue attente et restera votre étoile les guidait aujourd'hui  
et de demain et pour toujours.

Cher papa

Pour mon ange dans la vie au sens de l'amour et le sens de la compassion et de dévouement  
à sourire. vie mystère de l'existence

Pour son invocation du secret de mon succès et chirurgiens affection Baumedouloureuses à  
plus cher ma mère bien-aimée Pour voir d'un certain optimisme et de bonheur dans  
Hilarité, à Brûler des bougies pour éclairer les ténèbres de ma vie, j'ai regardé à ma réussite  
regarde fixement espoir.

Mon frères aux garçons et aux filles de mes oncles et mes collègues oncles les chemins cette  
vie sans eux rien avec eux et je suis sans eux, j'aime tout.

Pour l'optimisme planté dans notre chemin et nous a donné une aide, et les installations, des  
idées et des informations, sans peut-être le sentiment qu'ils ont, à leur tour, a remercié tout  
le monde, et en particulier de leur mon cousin.

Pour le plus grand plaisir de mon cœur leur revoir «3.A.F.Y my best friend ». à tous famille et  
les amis, à de la route pavée en face de moi pour obtenir au science." Soyez un scientifique,  
Si vous ne pouvez pas être éduqués, ne peux pas, je voudrais scientifiques, ne peux pas, ne  
pas leur animosité "

\*\*\*\*Et merci mille fois, ô Dieu\*\*\*\*

## *DEDICACE AICHA*

*A mon dieu qui nous a donné le courage pour terminer notre projet.*

*A ma très chère patrie l'Algérie ;*

*A mon très cher grand père : mouhamed*

*A mes très cher grandes mères : khadouja et zohra*

*A celle pour qui je dois toute réussite, a mon adorable mère saliha la plus chère au monde ;*

*A mon très cher père : abdlmoumen pour son patience, son soutient et son confiance.*

*A ma merveilleuse sœur chaïma*

*A mon sympathique frère ziad et la joie de notre maison abdrahman « oppa »*

*A tous mes oncles badis salim salih toufik kamal riad said mokhtar tahar et leur familles*

*A mes tantes nacira loubna hasina et surtout ma tante karima.*

*A tous mes gentils cousins et chaleureusement à la princesse hadjer*

*A tous mes enseignants et surtout mon encadreur M :Attia Mourad*

*A tous mes amis de la « klïka » pour votre sincère amitié, votre soutien permanent me remonte le moral et vos conseils m'incitent à relever les défis.  
et tous les amis de centre universitaire de mila*

*A ratiba et sonia*

*A mon trinôme salim et meryem*

*A tous ceux que j'aime tant et que je n'ai pas cités ;*

# Résumé

*Faire interagir des programmes différents en réseau a toujours été une affaire complexe, notamment si on souhaite standardiser certains aspects de cette interaction, les web service sont un ensemble de protocoles qui permettent de faire communiquer des programmes tournant sur des machines différentes et écrits dans des langages de programmation différents. Il le font en reprenant certains principes du web (transport sur **HTTP**, formatage sur **XML**), représente un nouveau paradigme prometteur pour le développement, le déploiement et l'intégration d'applications internet. Ces technologies fournissent une infrastructure pour décrire (**WSDL**), découvrir (**UDDI**) et invoquer (**SOAP**) des services web.*

*Notre projet consiste à créer un environnement administratif par la réalisation d'une administration électronique qui est basée sur les services web. Ces services permettent l'interopérabilité entre différentes applications.*

## *Mots clés :*

**E-ADMINISTRATION, SERVICE WEB, PROCEDURE ADMINISTRATIVE (PERMIS DE CONSTRUIRE), HTTP, XML, WSDL, UDDI, SOAP.**

# TABLE DES MATIÈRES

INTRODUCTION GENERAL.....	1
<b>CHAPITR 1 : LES SERVICE WEB</b>	
1.1 Introduction.....	3
1.2 Définitions et concepts	
1.2.1 Origine.....	3
1.2.2 Définition.....	4
1.2.3 Caractéristique des services web.....	4
1.3 Les avantages et les inconvénients des services web	
1.3.1 Les avantages.....	4
1.3.2 Les inconvénient.....	4
1.4 Architecture des services web	
1.4.1 une architecture oriente SOA.....	5
1.5 Model fonctionnellement.....	5
1.6 Infrastructure des services web	
1.6.1 La communication SOAP.....	7
1.6.2 Le protocole de communication http.....	8
1.6.3 La description du service : WSDL .....	10
1.6.4 Service de recherche et publication UDDI.....	12
1.7 Conclusion sur les standards.....	15
1.8 Les Services web pour la e-administration.....	15
1.9 Conclusion.....	16
<b>CHAPITRE 2 : E_ADMINISTRATION</b>	
2.1 Introduction.....	17
2.2 Définition.....	17
2.3 Objectif.....	18

2.4 Les obstacles les plus importants au développement de l'e_administration.....	18
2.5 Le permis de construire.....	19
2.5.1 Définition du permis de construire.....	20
2.5.2 La demande du permis de construire.....	20
2.5.3 Dépôt de la demande de permis de construire.....	21
2.5.4 Instruction de la demande du permis de construire.....	21
2.6 Conclusion .....	21

## CHAPITRE 3 : UML

3.1 Introduction.....	22
3.2 Définition.....	22
3.3 Les diagramme d'UML.....	22
3.4 Description textuel	
3.4.1 Le cas d'utilisation <<s'authentifier>>.....	23
3.4.2 Le cas d'utilisation supprimer citoyen.....	24
3.4.3 Le cas d'utilisation archiver citoyen.....	24
3.4.4 Le cas d'utilisation gérer les dossier-citoy.....	25
3.4.5 Le cas d'utilisation archiver dossier.....	27
3.4.6 Le cas d'utilisation contacter service .....	27
3.4.7 le cas d'utilisation remplir le formulaire.....	28
3.6 Diagramme de cas d'utilisation.....	29
3.7 Diagramme de classe.....	30
3.8 Diagramme de séquence.....	31
3.9 Diagramme d'activité.....	41
3.10 Conclusion.....	50

## CHAPITRE 4 : réalisation du système

4.1 Introduction.....	51
4.2 Définition du Visual studio .....	51
4.3 Visual studio 2010.....	51
4.4 Objectif de la formation Visual studio 2010.....	51
4.5 Objectif de l'utilisation de C# sous Visual studio 2010.....	52
4.6 ASP.NET.....	52
4.6.1 Principes.....	52
4.7 Les langages de programmation	
4.7.1 HTML.....	53
4.7.2 CSS.....	54
4.7.3 SQL.....	54
4.8 Base de données Microsoft Access.....	54
4.9 Page de démarrage.....	54
4.10 Présentation de l'implémentation des web services .....	55
4.11 Présentation du portail web.....	57
4.12 Conclusion.....	59
Conclusion générale.....	60
Bibliographie.....	61

# LISTE DES FIGURES

<b>Figure 1.1</b> : évolution des architectures logicielles .....	5
<b>Figure 1.2</b> : les interactions entre les services web.....	6
<b>Figure 1.3</b> : schéma de fonctionnement du système Unidirectionnel SOAP.....	9
<b>Figure 1.4</b> : la structure des messages SOAP.....	10
<b>Figure 1.5</b> : structure d'une interface WSDL.....	11
<b>Figure 1.6</b> : architecture d'UDD.....	14
<b>Figure 1.7</b> : interaction entre les web service administratif .....	16
<b>Figure 3.1</b> : Les diagrammes UML.....	22
<b>Figure 3.2</b> : diagramme de cas d'utilisation.....	29
<b>Figure 3.3</b> :diagramme de classe.....	30
<b>Figure 3.4</b> : Diagramme de séquence « s'authentifie ».....	31
<b>Figure 3.5</b> : Diagramme de séquence « ajouter dossier » .....	32
<b>Figure 3.6</b> : Diagramme de séquence « supprimer citoyen ».....	33
<b>Figure 3.7</b> : Diagramme de séquence « rechercher citoyen ».....	34
<b>Figure 3.8</b> : Diagramme de séquence « ajouter dossier ».....	35
<b>Figure 3.9</b> : Diagramme de séquence « supprimer dossier ».....	36
<b>Figure 3.10</b> : Diagramme de séquence « rechercher dossier ».....	37
<b>Figure 3.11</b> : Diagramme de séquence « archiver dossier ».....	38
<b>Figure 3.12</b> : Diagramme de séquence « contacter dossier » .....	39
<b>Figure 3.13</b> : Diagramme de séquence « remplir le formulaire de permis de construire ».....	40
<b>Figure 3.14</b> : Diagramme d'activité « s'authentifie ».....	41
<b>Figure 3.15</b> : Diagramme d'activité « ajouter citoyen ».....	42
<b>Figure 3.16</b> : Diagramme d'activité « supprimer citoyen ».....	43
<b>Figure 3.17</b> : Diagramme d'activité « rechercher citoyen ».....	44

<b>Figure 3.18</b> : Diagramme d'activité « ajouter dossier » .....	45
<b>Figure 3.19</b> : Diagramme d'activité « supprimer dossier » .....	46
<b>Figure 3.20</b> : Diagramme d'activité « rechercher dossier ».....	47
<b>Figure 3.21</b> : Diagramme d'activité « archiver dossier » .....	48
<b>Figure 3.22</b> : Diagramme d'activité « contacter dossier » .....	49
<b>Figure 3.23</b> : Diagramme d'activité « remplir le formulaire de permis de construire ».....	50
<b>Figure 4.1</b> : le serveur de base de données Microsoft Access.....	54
<b>Figure 4.2</b> : la page de démarrage de Microsoft Visual studio 2010.....	55
<b>Figure 4.3</b> : les applications web.....	55
<b>Figure 4.4</b> : les pages web.....	56
<b>Figure 4.5</b> : les web services.....	56
<b>Figure 4.6</b> : service web.....	57
<b>Figure 4.7</b> : la page d'accueil.....	57
<b>Figure 4.8</b> : la page de formulaire de demande de permis de construire.....	58
<b>Figure 4.9</b> : la page de contact.....	58
<b>Figure 4.10</b> : la page d'annuaire.....	58
<b>Figure 4.11</b> : la page de connecté de l'administrateur.....	59

# GLOSSAIRE

HTTP: Hypertext Transfer Protocol.

HTML: Hypertext Markup Language.

SOAP: Simple Object Access Protocol.

TIC: Technologie de l'information et de la communication.

UML: Unified Modeling language.

UDDI: Universal Description Language.

WSDL: Web Service Description Language.

W3C: World Wide Web Consortium.

XML: extensible Markup Language.

# INTRODUCTION

Les nouvelles technologies de l'information et de la communication (TIC) ont un impact important dans de nombreux domaines de notre société et provoquent des changements sociaux auxquels doivent s'adapter les institutions publiques. Depuis quelques années, l'e-administration a pris de l'ampleur dans la plupart des pays de l'Union européenne. L'e-administration concerne tous les niveaux de pouvoir: les administrations nationales, mais également les organisations locales, régionales voire même paneuropéennes.

L'utilisation de l'e-administration comme outil d'organisation offre d'importants avantages pour l'administration. Il mène essentiellement à une amélioration de l'efficacité et à des économies liées à :

- \* la simplification des démarches administratives,
- \* l'accélération des transactions à moindre coût,
- \* des économies de papier,
- \* la réallocation de l'espace ainsi gagné,
- \* la réorientation du personnel vers d'autres projets.

L'e-administration améliore le flux d'information entre les citoyens, les associations et les autres institutions administratives.

En outre, l'utilisation de cette technologie de pointe dans les services administratifs confère une image dynamique à une ville ou à un pays.

Et pour la réalisation d'une administration électronique réussit, on utilise la technologie service web. Dans ce mémoire on présente cette amélioration dans quatre chapitres :

**1. Chapitre 1 :** On présente les services web avec leur architecture

**2. Chapitre2 :** On présente E-administration et leur objectif et l'exemple de travaille qu'est le permis de construire

**3. Chapitre 3 :** On définit l'UML comme un langage de modélisation de notre projet

**4. Chapitre 4 :** La réalisation du projet avec le Visual studio 2010

A decorative scroll graphic with a light pink-to-white gradient, rounded corners, and a vertical tab on the left side. The scroll is set against a white background with a faint, dashed border. The text is centered on the scroll.

**Partie I :**

**Cahier de  
charge**

The background features a white page with several thin red lines. One line starts from the top left and extends towards the center. Another line starts from the top center and extends towards the right. A third line starts from the top right and extends towards the bottom right. There are three sets of concentric circles in a reddish-pink color: one large set in the top right, one medium set in the center right, and one large set in the bottom right.

**CHAPITRE 1 :**  
**LES SERVICES WEB**

## CHAPITRE 1: Services WEB

### 1.1 Introduction

Selon la définition du W3C (World Wide Web Consortium), un Web service (ou service Web) est une application callable via Internet - par une autre application d'un autre site Internet Permettant l'échange de données (de manière textuelle) afin que l'application appelante puisse Intégrer le résultat de l'échange à ses propres analyses. Les requêtes et les réponses sont soumises à des standards et normalisés. [1]

### 1.2. Définition et Concepts

#### 1.2.1 Origines[1]

Les Services Web sont considérées comme étant l'évolution naturelle du Web

Et on peut distinguer trois phases de développement dans l'histoire du Web :

*1.2.1.1 Le Web du Document*, le phénomène Internet originel, utilisé principalement par des organisations et des particuliers pour publier des informations sur leur travail, leurs produits, etc.

*1.2.1.2 Le Web Applicatif*, le progrès grâce auquel les entreprises ont commencé à utiliser le Web à des fins commerciales. Les sites Internet sont devenus plus interactifs, et plus complexes, gérés par des serveurs d'applications, capables de distribuer leur charge avec d'autres serveurs, en fonction des besoins.

*1.2.1.3 Le Web des Services*, est la phase émergente, dans laquelle les serveurs d'application précédents communiquent désormais entre eux. Cette évolution a été poussée par le désir de pouvoir réaliser des échanges interentreprises dans un environnement automatisé et ouvert tel qu'Internet. L'échange de données informatisées entre deux applications nécessite une normalisation des messages échangés.

Une approche Web Services du système d'information vise à transformer chaque composant, base de données, applicatif métier, application de bureautique, en nœud s'exposant sur des standards de l'Internet, soit pour consommer des Web Services, soit pour en fournir.

Ainsi, on passe d'une interopérabilité applicative à une interopérabilité entre services.

Cette approche propose une API (Application Programming Interface) universelle qui ne requiert pas d'autres protocoles que ceux d'Internet : HTTP (HyperText Transfer Protocol)

sur TCP/IP1 (Transmission Control Protocol / Internet Protocol) principalement. De plus, ils normalisent l'appel, l'échange et l'organisation de services applicatifs

# Chapitre 1 : service web

---

## 1.2.2. Définition

Un service web (ou service de la toile) est un programme informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone. Le protocole de communication est défini dans le cadre de la norme SOAP dans la signature du service exposé (WSDL). Actuellement le protocole de transports est essentiellement HTTP(S).[2]

## 1.2.3 Caractéristique des services web

Pour atteindre une réutilisation maximale, les services doivent être interopérable. Pour atteindre cette interopérable, la définition des services doit avoir un certain nombre de caractéristiques. En voici quelques-unes :

- L'échange d'information entre services doit être standardisé.
- La manière d'utiliser le service est définie par une interface.
- Un service web est une application logicielle qui est reconnue par un URI.

## 1.3 Les avantages et les inconvénients des services web

### 1.3.1 Les avantages

De nos jours, la technologie des services web est populaire et couramment utilisée car elle offre des avantages intéressants pour les utilisateurs des systèmes distribués :

- ✓ Les services web facilitent l'intégration et la réutilisation d'applications et de service.
- ✓ L'hétérogénéité des application disparaît : accès transparent a tout type de plateforme.
- ✓ Réduction des couts et efficacité : plus rapide a crêt et plus facile de maintenir les applications.
- ✓ Les services web utilise des normes et protocole ouverts.

### 1.3.2 Les inconvénients

La technologie des services web comporte plusieurs inconvénients dont :

- Problèmes de sécurité : Il est facile contourner les mesures de sécurité mises en place par les pare-feu l'utilisation du protocole http.

# Chapitre 1 : service web

---

- Problèmes de performance : Les service web sont encore relativement fiables par rapport à d'autres approches de l'informatique répartie telles que CORBA ou RMI.
- Fiabilité : Il est difficile de s'assurer de la fiabilité d'un service car on ne peut garantir que ses fournisseurs ainsi que les personnes qui l'invoquent travaillent d'une façon fiable.

## 1.4 Architecture des services Web.

### 1.4.1 Une architecture orientée services SOA : (Service Oriented Architecture) :

L'architecture des services Web comme l'architecture du Web sont des instances d'architectures orientées services (SOA) [03]. Elle propose une perspective globale sur le développement, la gestion et le fonctionnement des services Web [04]. L'architecture SOA est un modèle (abstrait) qui définit un système par un ensemble d'agents logiciels distribués qui fonctionnent de concert afin de réaliser une fonctionnalité globale préalablement établie [05]

Elle consiste à diviser le logiciel répondant à un problème, en un ensemble d'entités proposant des services. Chacune de ces entités peut utiliser les services proposés par d'autres entités. On obtient ainsi un réseau de services interagissant entre eux. Cette architecture s'appuie sur une architecture à composants (implémentation « réelle » des services), et suit l'évolution logique des architectures logicielles [END et al 04] (figure 1) :



Figure 1 - Evolution des architectures logicielles.

Le choix d'une architecture SOA entre dans la perspective de transformer le Web en une énorme plate-forme de composants faiblement couplés et automatiquement intégrables. L'architecture SOA vise trois objectifs importants : (i) identification des composants fonctionnels, (ii) définition des relations entre ces composants et (iii) établissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture.

## 1.5 Modèle de fonctionnement

## Chapitre 1 : service web

Les interactions entre les services Web impliquent trois participants (acteurs) [06]: le fournisseur de services (prestataire), l'annuaire de services et le client (utilisateur du service) (vo- L'annuaire des services figure 2 ).

*Le fournisseur de services* correspond au propriétaire du service (i.e. entité responsable du service Web). D'un point de vue technique, il est constitué par la plate-forme d'accueil du service.

*Le client* correspond au demandeur du service. D'un point de vue technique, il est constitué par l'application de recherche et d'invocation d'un service. L'application client peut être elle même un service Web.

représente l'entité logicielle qui joue le rôle de l'intermédiaire entre les clients et les fournisseurs de services, il correspond à un registre de descriptions de services offrant des facilités de publication de services pour les fournisseurs ainsi que des facilités de recherche pour les clients et le moyen de localiser leurs besoin en terme de services.

*La dynamique de l'architecture se décompose ainsi* : Le fournisseur de services définit la description de son service et la publie dans un annuaire de service (dans des registres) en vue d'être localisé par des clients; Le client utilise les facilités de recherche disponibles au niveau de l'annuaire pour retrouver et sélectionner un service donné.

Il examine ensuite la description du service sélectionné (extrait sa description du registre) pour récupérer les informations nécessaires lui permettant de se connecter au fournisseur du service et d'interagir avec l'implémentation du service considéré (entreprend une interaction).

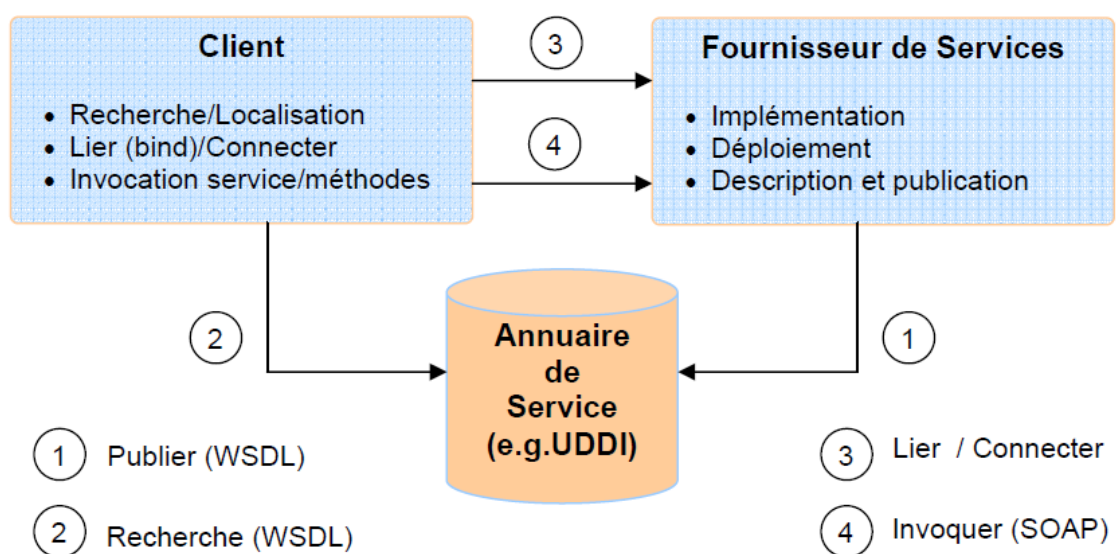


Figure 2 - Les interactions entre les services Web.

# Chapitre 1 : service web

---

Les interactions entre les services Web impliquent trois intervenants: le fournisseur de services, l'annuaire de services et le client.

Le modèle de fonctionnement de l'architecture des services Web se base sur un cadre technologique qui constitue l'infrastructure de l'architecture par composants des services Web.

Cette infrastructure offre les services nécessaires pour la réalisation des différentes étapes du cycle de vie d'un service Web.

## **1.6 L'infrastructure des services Web : les standards mis en jeu dans l'architecture**

### **1.6.1 La communication : SOAP**

Le formalisme XML étant basé sur du texte, il est tout à fait adapté pour être véhiculé par le protocole de communication HTTP. Pour assurer la communication entre les participants d'un service Web, le couple HTTP/XML est donc naturellement utilisé. Les messages exprimés à l'aide de XML respectent un format standard définis par le W3C que l'on appelle le protocole SOAP (Simple Object Access Protocol).

Ainsi, dans le cadre des services Web, une application communique avec une autre application par l'intermédiaire du protocole SOAP/HTTP. Cela signifie qu'une application envoie un message SOAP (donc un message exprimé en XML) vers une autre application, que celle-ci traite la demande effectuée par ce message et renvoie un message de réponse à l'application appelante, basant sur le mécanisme RPC (Remote Procedure Call: appel de procédures localisées sur des machines distantes) qui simplifie la coopération entre applications en fournissant un mécanisme pour transmettre un appel de procédure vers une application distante offertes sur le Web. [07]

#### *1.6.2.1 Les messages SOAP*

La communication par message constitue un point crucial dans toute architecture SOA. SOAP est au cœur de l'échange des messages entre applications sur le Web. Du fait qu'il est basé sur XML, il permet l'échange de données structurées indépendamment des langages de programmation ou des systèmes d'exploitation. Le protocole SOAP est une spécification XML qui définit un protocole léger d'échange de données structurées entre services Web dans un environnement totalement distribué et hétérogène. Il est indépendant du contenu du

## Chapitre 1 : service web

---

message, à proprement parler, il laisse la responsabilité de l'interprétation aux couches de communication supérieures [08].

Il se contente d'offrir la possibilité de structurer des messages destinés à des objectifs particuliers allant d'un simple échange de données jusqu'à l'appel de procédures à distance.

N'imposant aucun modèle de programmation spécifique, SOAP peut être employé dans tous les styles de communication : synchrone (s'appuyer sur HTTP) ou asynchrone (s'appuyer sur SMTP1), point à point ou multipoint, intranet ou Internet.

### *1.6.2.2 Modèle d'échange de messages en SOAP*

Les messages SOAP sont des transmissions fondamentalement à sens unique (unidirectionnel) d'un expéditeur à un récepteur (figure3). Lorsqu'une transmission d'un message commence (ex. invocation d'un service web), un message SOAP (ou document XML) est généré.

Ce message est envoyé à partir d'une entité appelée le SOAP Sender, localisé dans un SOAP nœud, celui-ci permet d'appeler une ou plusieurs fonctions du service. Les paramètres des fonctions invoquées se situent à l'intérieur du message. Le message est transité par zéro ou plusieurs nœuds intermédiaires (SOAP intermediates) et le processus fini lorsque le message arrive au SOAP receiver.

Le message est alors traité par le service Web avec les paramètres correspondants. Un message SOAP, de même structure, est retourné.

Mais les paramètres transportés représentent les résultats de la fonction invoquée.

Le chemin suivi par un message SOAP est nommé message path. .

### **1.6.2 Le protocole de communication HTTP**

Le choix de HTTP est un élément très important car il illustre parfaitement la capacité de SOAP à s'adapter aux échanges sur Internet réutilisant une technologie largement déployée et acceptée. Le protocole HTTP est un protocole simple capable de s'accommoder à la fois de la qualité de service et des temps de latence très variables de l'Internet; ce que n'est pas le cas par exemple des protocoles d'environnements répartis tels que DCOM ou CORBA.

De plus, en s'appuyant sur HTTP, on ne se heurte pas aux problèmes de pare-feu (firewall) ou de configuration de réseau IP qui rendent parfois difficile le déploiement d'applications à

# Chapitre 1 : service web

objets répartis au-delà du périmètre du réseau d'entreprise. [09]

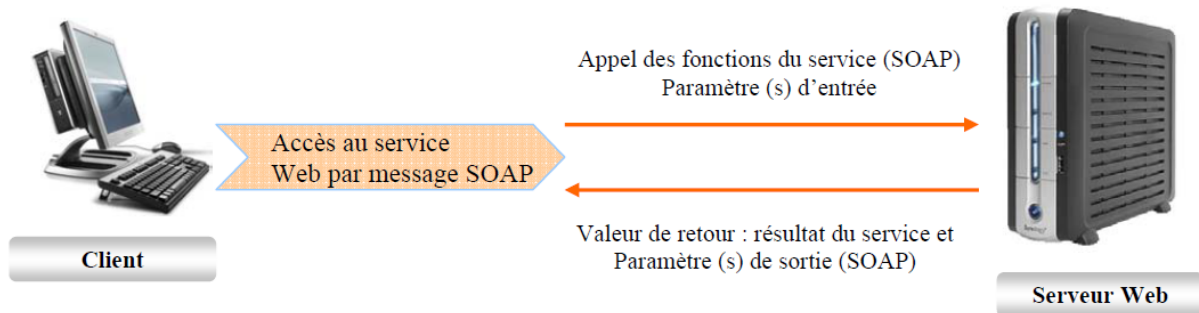


Figure 3 - Schéma de fonctionnement du système unidirectionnel SOAP

**Enveloppe SOAP:** Un message SOAP est un document XML composé d'un élément enveloppe qui est appelé "enveloppe SOAP", contenant un en-tête (header) facultatif et un corps du message obligatoire

**SOAP Header :** l'en-tête de message SOAP, est le premier fils de l'élément enveloppe.

Même s'il peut être vide, il doit impérativement être écrit. Cet élément apporte des données supplémentaires à SOAP. Ces données peuvent être des informations d'authentification, de gestion de transactions, de paiement, etc.

**SOAP Body :** l'élément Body contient l'information destinée au receveur. Il faut que cet élément encadre une balise contenant le nom de la méthode invoquée (pour une requête), ou le nom de la méthode suivie de Response (pour la réponse). Cette balise doit aussi contenir l'espace de noms correspondant au nom de service.

Un message SOAP peut aussi contenir un ou plusieurs attachements (document, images, etc.) à transmettre avec le message. Ces données ne sont pas représentables en XML.

De ce fait, SOAP utilise un mécanisme d'inclusion appelé MIME (Multipurpose Internet Mail Extensions), cette méthode est répandue pour transmettre des documents autres que du texte dans des courriers électroniques.

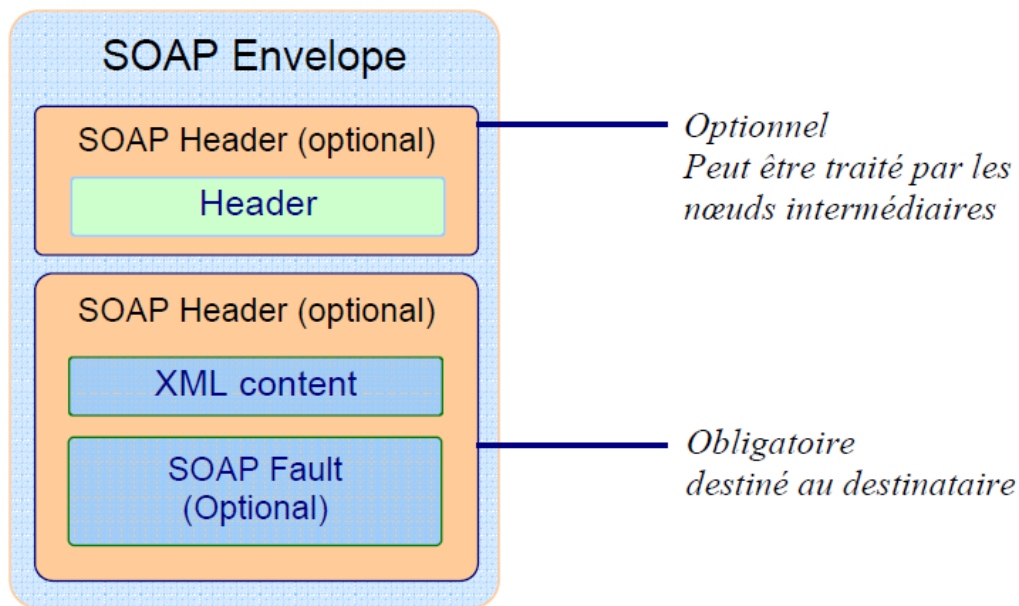


Figure 4 - La structure des messages SOAP

Pour résumer, un message SOAP en cours de transit est composé de deux parties indépendantes :

- une structure XML qui constitue le message SOAP;
- un en-tête du protocole de transport où le message SOAP est encapsulé en vue d'être livré à l'application destination;

Nous pouvons cependant ajouter que le protocole SOAP n'intègre pas la sémantique du message. Pour remédier à cela, le langage RDF technologies du Web sémantique, permet de supporter les échanges de messages.

### 1.6.3 Description de service : WSDL

Le protocole SOAP met à la disposition des services Web un moyen standard de structuration et d'échange de messages XML. Il ne fournit aucune indication sur la structure que le message doit respecter vis à vis du service Web sollicité. C'est toujours dans le but de rendre les services Web faiblement couplés et autonomes, que la spécification WSDL a vu le jour. Contrairement aux architectures monolithiques où la description des composants ainsi que les moyens de les invoquer dépendent fortement de l'infrastructure utilisée, la spécification WSDL offre une grammaire qui décrit l'interface des composants services Web de telle façon qu'ils se suffisent à eux-mêmes.

# Chapitre 1 : service web

WSDL est un langage de description des capacités de services Web basé sur XML. Un document WSDL décrit essentiellement le nom de la méthode utilisé, son nombre de paramètres, et leur type, ce qu'un service Web offre, où il réside et comment on peut l'invoquer.

Nous pouvons dire que les services Web sont auto descriptifs.

La spécification du service est composée de deux parties : une définition abstraite des services en terme de messages échangés entre les différents types de port et la définition des mécanismes de liaison entre les définitions abstraites et un ensemble de techniques de déploiement (généralement des protocoles Internet).

Un document WSDL est devisé en sept sections distinctes (figure 5) :

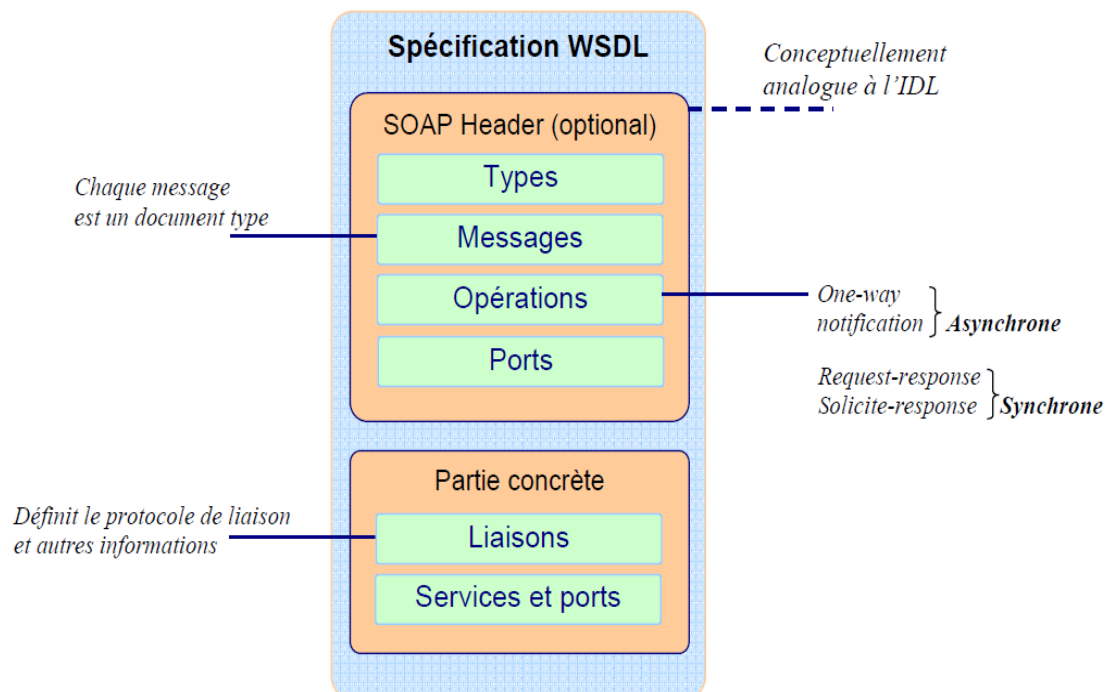


Figure 5 - Structure d'une interface WSDL

- types : Fournissent des définitions de types de donnée afin de décrire les messages échangés.
- message : Par l'intermédiaire de cette section, on définit le format des messages échangés, Un message correspond aux données qui seront véhiculées selon les méthodes invoquées. Chaque méthode du service possède deux éléments message, le premier correspond à la requête et le second correspond à la réponse. Chacun est constitué d'une ou plusieurs parties (sous éléments part) décrivant un élément du

## Chapitre 1 : service web

---

message. La description contient le nom de l'élément en paramètre d'entrée ou de sortie selon le message et son type.

- type de port (portType) : Ensemble d'opérations abstraites. Chaque opération se réfère à un message entrant et à des messages sortants. Chaque opération est identifiée dans le document WSDL par l'élément operation déclarant le nom de la méthode et les données passées en paramètres. Ces opérations sont regroupées dans un élément portType. Le document WSDL est constitué d'autant d'élément portType que le service contient de groupe de méthodes.
- opération : Décrit les opérations invoquées à l'aide des messages reçus, émit par le service et éventuellement des messages d'erreur.
- port : Ce qui spécifie une adresse pour un rattachement, déterminant ainsi un seul nœud branché, ou point terminal 1 (endpoint).
- rattachement (binding) : Ce qui spécifie les aspects concrets de protocole de communication et le format des données pour les opérations et messages définis par un type de port particulier.
- service : Représente une collection de points d'entrée (endpoint) relatifs, il sert à regrouper un ensemble de ports.

La spécification WSDL joue un rôle important dans l'interopérabilité des composants services Web. Moyennant un schéma uniforme obéissant à une sémantique bien définie, elle permet aux composants de définir ce qui est nécessaire à leur invocation. La spécification WSDL est définie selon une sémantique totalement indépendante du modèle de programmation de l'application. Elle sépare clairement la définition abstraite du service (échange de messages) de ses mécanismes de liaison (définition des protocoles applicatifs). Cette dernière caractéristique permet au composant d'interagir même si l'application a été modifiée ce qui est un point important pour assurer l'interopérabilité des services.

La complétude de la spécification WSDL permet l'automatisation du processus d'invocation.

**1.6.4 Service recherche et publication : UDDI** (Universal Description, Discovery and Integration)

Les deux standards exposés précédemment définissent ensemble l'aspect le plus basique de développement de l'infrastructure des services Web. Toutefois, dans un environnement ouvert comme Internet, le modèle de description des services Web n'est d'aucune utilité s'il n'existe pas un moyen de localiser aussi bien les services que leurs descriptions WSDL.

## Chapitre 1 : service web

---

Un troisième standard a été conçu pour réduire l'écart entre les applications clientes et les services Web, appelé UDDI.

UDDI est un annuaire mondial d'entreprises s'appuyant sur le réseau Internet. Il permet d'automatiser les communications entre prestataires, clients, etc. Dans ce but, celui-ci propose plusieurs entrées : nom, carte d'identité des sociétés, description des produits et services. Et, si ces derniers sont des applicatifs invocables à distance, il fournit les références des connexions permettant de communiquer avec eux.

UDDI est une spécification qui définit les mécanismes qui permettent aux entreprises de publier leurs services et de découvrir et interagir avec d'autres services via le Web. UDDI se base sur SOAP et suppose que les requêtes et les réponses sont des objets UDDI envoyés comme des messages SOAP.

L'enregistrement des services Web dans un annuaire UDDI s'effectue auprès d'un opérateur en accédant au site Web de ce dernier à partir d'un navigateur ou d'un outil intégré à un environnement de développement. Des recherches précises peuvent s'effectuer dans l'annuaire par catégories d'entreprise en utilisant des standards taxinomiques d'identification d'entreprise et par mots clés.

La spécification UDDI constitue une norme pour les annuaires des services Web. Les fournisseurs disposent d'un schéma de description permettant de publier des données concernant leurs activités, la liste des services qu'ils offrent et les détails techniques sur chaque service. Elle offre aussi une API aux applications clientes, pour consulter et extraire des données concernant un service et/ou son fournisseur (Interrogation de service).

Les registres UDDI sont des services Web qui offrent deux fonctionnalités de base : la publication des différents types d'information sur les services et leurs fournisseurs selon un schéma de description, et la consultation du contenu des registres.

La publication d'un service chez un opérateur, donne lieu automatiquement à un processus de propagation des informations aux différents registres UDDI. L'accès à l'ensemble d'informations des registres peut se faire de n'importe quel opérateur UDDI. La recherche se fait grâce à un moteur de recherche intégré au site de l'opérateur UDDI choisi. Ce moteur de recherche permettra d'affiner la recherche selon plusieurs critères : Nom de l'entreprise, la localisation de l'entreprise, identifiant de l'entreprise, le nom du service Web ...etc.

Chaque registre UDDI stocke trois sortes de données : des données concernant les fournisseurs de services telles que le nom de l'entreprise, ses coordonnées et des descriptions de l'entreprise consultable par l'utilisateur appelées pages blanches, des données concernant

## Chapitre 1 : service web

l'activité ou le service métier des fournisseurs, la description des services Web déployés par les entreprises appelées pages jaunes et les données techniques de chaque service publié qui constituent les pages vertes. La spécification UDDI offre un schéma qui structure d'une manière uniforme les différents types concernant les trois niveaux de description.

Le fonctionnement d'un UDDI est décrit par les modèles suivants:

*Modèle de données (figure 6):* Des différents composants sont des documents XML :

Publisher Assertion, Business Entity, Business services, Binding Template, et Tmodel.

**Publisher Assertion :** Cette partie est facultative. Elle permet de décrire l'organisation dans son intégrité si elle est divisée en plusieurs divisions. [7]

**Business Entity:** Décrivent les organisations ayant publié des services dans le répertoire

**Business Key; Business Service:** Décrivent de manière non technique les services; **Binding**

**Templates** Spécifient les coordonnées des services; **Tmodel :** Décrivent de manière technique les services;

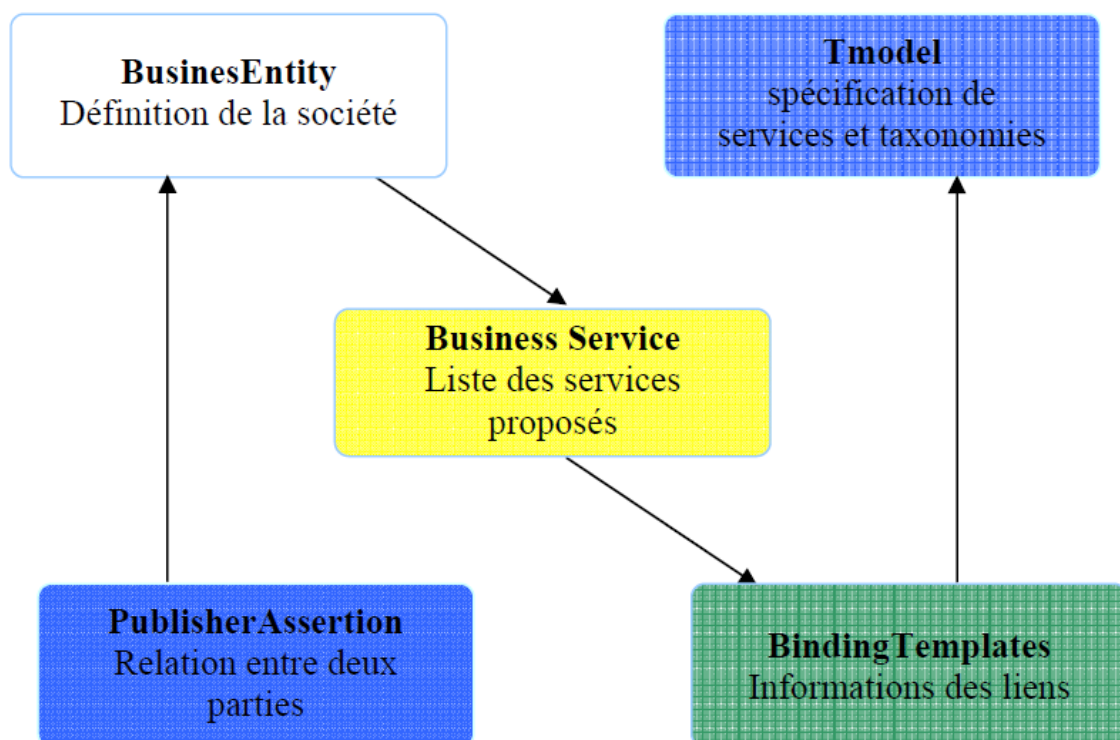


Figure 6 Architecture de UDDI (Entités composant un annuaire).

## Chapitre 1 : service web

□ Modèle de programmation: L'API UDDI est divisée en une interface de programmation pour l'enregistrement de services Web dans un annuaire UDDI et une interface de programmation pour la recherche d'informations. Cette API est composée de 2 grandes bibliothèques :

- API de requête (interrogation du Registre);
- API de publication;

□ Modèle d'usage: Publication de service, Découverte de service, Description d'utilisation de service.

### 1.7. Conclusion sur les standards

En conclusion, les standards proposés sont de bas niveau sans véritable sémantique.

Cependant, ces standards permettent un certain nombre de tâches automatiques qui induisent une sémantique: diagnostic et découverte de protocoles pour SOAP et composition automatique pour WSDL. Comme il n'y a pas de véritable interaction entre ces trois standards, des couches ont été ajoutées afin de remédier à ce problème. Si ces couches sont bien utilisées elles peuvent contribuer à ajouter de la sémantique en permettant d'effectuer des tâches automatiques. En pratique WSDL et SOAP sont seulement utilisés. En pratique aussi d'autres outils correspondant à des couches de plus haut niveau sont utilisés même s'ils ne sont pas considérés comme des standards.

### 1.7 Les Services web pour la e-administration:

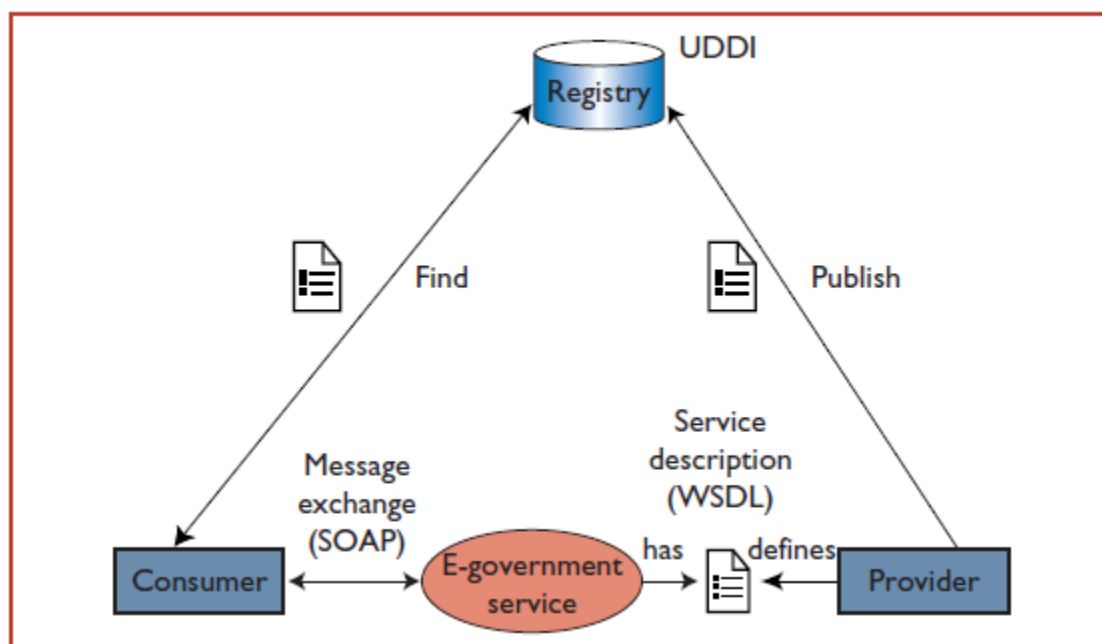


Figure7 : interaction entre les web service administratif [10]

### **1.8 Conclusion**

Dans ce chapitre on a présenté les concepts de base des services web, et on a basé sur les éléments de leur infrastructure comme le SOAP, UDDI, WSDL... et aussi le langage utilisé dans ce domaine qui est le XML.

The background features a large, stylized red 'V' shape formed by two thin lines meeting at the top. Inside the 'V' are two sets of concentric circles, one larger than the other, both with a red-to-white gradient. At the bottom right, there is a large, rounded square shape with a red-to-white gradient, partially overlapping the 'V' lines.

**CHAPITRE 2 :**  
**E\_ADMINISTRATION**

### CHAPITRE 2 : E-ADMINISTRATION

#### 2.1 Introduction

Depuis l'avènement des ordinateurs, les administrations publiques sont de grandes consommatrices de technologies de l'information et de la communication (TIC). A l'instar des entreprises privées, les premières applications informatiques ont été l'automatisation des tâches de traitement de masse utilisant ainsi la puissance de calcul des gros ordinateurs centraux. Puis l'évolution technologique a permis de passer de l'informatique de gestion aux systèmes d'information, véritable support aux activités des métiers. Désormais, depuis une décennie, les administrations ont recours aux technologies basées sur l'Internet.



#### 2.2 Définition :

- ❖ l'administration électronique est un grand domaine de L'e-gouvernement, L'e-gouvernement se décompose en 3 grands domaines
  - l'e-administration
  - l'e-démocratie (synonymes : *démocratie en ligne, cyber démocratie*) .
  - l'e-société
- ❖ (synonymes : *administration en ligne, administration électronique cyber administration*) peut se développer dans tout type d'administration ou de service public, en contact avec le public (front-office) ou non (back-office). Elle se caractérise par l'emploi de technologies de l'information et de la communication visant à améliorer les processus, la communication entre usagers et administrations ou

## Chapitre 2: e\_administration

---

entre administrations et l'efficacité de l'administration, que ce soit en termes de délais, de qualité, ou de productivité des agents publics.[11]

### 2.3 OBJECTIFS

- Accroître la qualité de vie du citoyen
- Garantir l'efficacité des services de l'administration
- Valoriser l'agent de l'administration
- Faire de l'administration un levier majeur dans la modernisation de l'économie
- Promouvoir la connaissance
- L'amélioration de la relation avec les citoyens.
- Des services accessibles à tous
- Des services faciles d'emploi
- Des démarches simplifiées
- Des services permettant de transférer la complexité réglementaire et administrative derrière les guichets
- Des services plus rapides : meilleure circulation de l'information par la voie électronique et délais d'instruction raccourcis.
- Des services plus efficaces
- Des services personnalisés
- Simplifier la vie des entreprises

### 2.4 Les obstacles les plus importants au développement de l'administration électronique

Le cadre législatif et réglementaire: l'adoption de solutions en ligne nécessite des modifications au niveau légal, car elles doivent désormais correspondre aux procédures papiers ; la «signature électronique» devient dès lors un pré-requis. De plus, les interactions entre services administratifs engendreront de nouvelles structures organisationnelles. Enfin, la modification du cadre légal prend beaucoup de temps puisqu'elle dépend de décisions politiques.

Le cadre budgétaire: les mécanismes de financement sont cloisonnés par département et des enveloppes budgétaires transversales sont difficiles à obtenir. Les coûts associés à la mise en

## Chapitre 2: e\_administration

---

œuvre de l'administration électronique sont souvent élevés, avec un risque d'échec non négligeable.

La résistance au changement: l'administration électronique exige une analyse des processus, mais également une très grande transparence, ce qui n'est pas forcément apprécié par certains fonctionnaires. Il s'agit de travailler étroitement avec tous les acteurs concernés et de les convaincre du bien-fondé des réformes, notamment en les incitant au changement plutôt qu'en l'imposant, ceci par divers moyens comme des séminaires d'échange d'informations, des palmarès des meilleurs sites, des communiqués aux médias sur les nouveautés, des bonus aux usagers lors d'utilisation en ligne, etc.

Enfin, mentionnons les obstacles techniques (exemple: la sécurité des échanges) et le fossé numérique (taux de pénétration d'Internet parmi la population) qui peuvent dans une moindre mesure enrayer le développement des TIC.

### **2.5 Le permis de construire**

Un permis de construire est obligatoire avant tout début de travaux, il atteste la conformité du projet par rapport à des dispositions techniques, réglementaires et législatives. Il donne les informations nécessaires à l'administration pour vérifier si le projet respecte les règles d'urbanisme.

Il existe plusieurs manuels pratique qui explique les démarches à suivre pour avoir un permis de construire, les règles à respecter et faciliter en principe les démarches des demandeurs au permis de construire.



## Chapitre 2: e\_administration

---

### **2.5.1 Définition permis de construire:**

Le permis de construire c'est un acte administratif délivrée par le maire d'une commune pour la construction d'un ouvrage ou pour la modification de l'architecture extérieur d'un ouvrage existant.

### **2.5.2 La demande du permis de construire**

En règle générale, c'est le propriétaire du terrain ou son mandataire qui formule la demande de permis. la demande peut également être faite par une personne attestant être autorisée par le propriétaire du terrain à exécuter les travaux.

#### *2.5.2.1 qui peuvent demander un permis de construire :*

soit le ou les propriétaires du ou des terrains, soit leur mandataire ou une ou plusieurs personnes attestant être autorisées par eux exécuter les travaux.

-soit en cas d'indivision, un ou plusieurs Co-indivisaires ou leur mandataire.

-soit une personne ayant qualité pour bénéficier de l'expropriation pour cause d'utilité publique.

#### *2.5.2.2 Où déposer le dossier :*

Le dossier peut être déposé à la mairie du lieu des travaux ou adressé à la mairie sous pli recommandé avec accusé de réception.

Le permis de construire est gratuit.

#### *2.5.2.3 Enregistrement du dossier :*

-dès réception du dossier, le maire délivre un récépissé qui précise le n<sup>o</sup> d'enregistrement de la demande et le délai d'instruction de droit commun.

-Si le dossier est incomplet, la demande de pièce complémentaire doit être adressée dans le délai d'un mois suivant le dépôt de la demande.

## Chapitre 2: e\_administration

---

-seules les demandes de pièces prévues par le code de l'urbanisme et notifiées dans le délai d'un mois suspendent les délais d'instruction.

### **2.5.3 Dépôt de la demande du permis de construire :**

- Au siège de l'APC (commune), de cinq (05) exemplaires du dossier contre récépissé de dépôt.

Le dossier est transmis au service de l'état chargé de l'urbanisme dans un délai de huit (8) jours tude d'impact pour les projets industriels

### **2.5.4 Instruction de la demande du permis de construire :**

Par les services compétents chargés de l'urbanisme en concertation avec les services des autres secteurs concernés.

Vérification de la conformité du projet avec les prescriptions d'urbanisme applicables au site (plans d'urbanisme : PDAU POS ou RGAU) : par la DUC (Direction de l'urbanisme et de la construction de la Wilaya) en concertation avec les services des autres secteurs concernés le cas échéant.

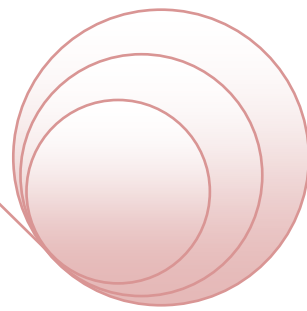
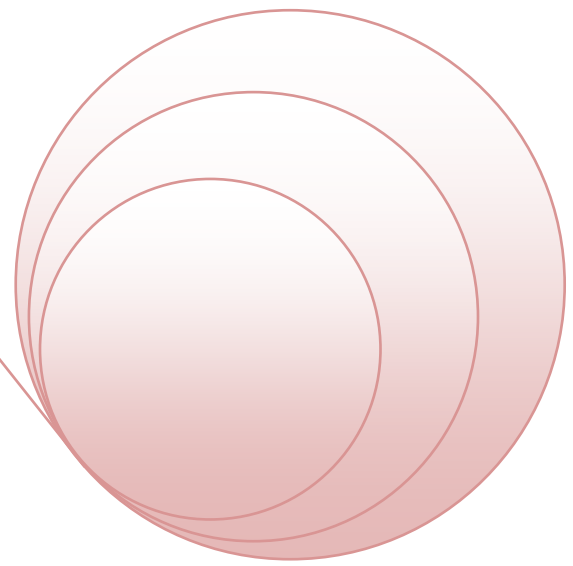
## **2.6 Conclusion**

Dans ce chapitre nous avons donné des concepts sur l'administration électronique et nous avons pris la demande du permis de construire de la mairie comme un exemple de travail.



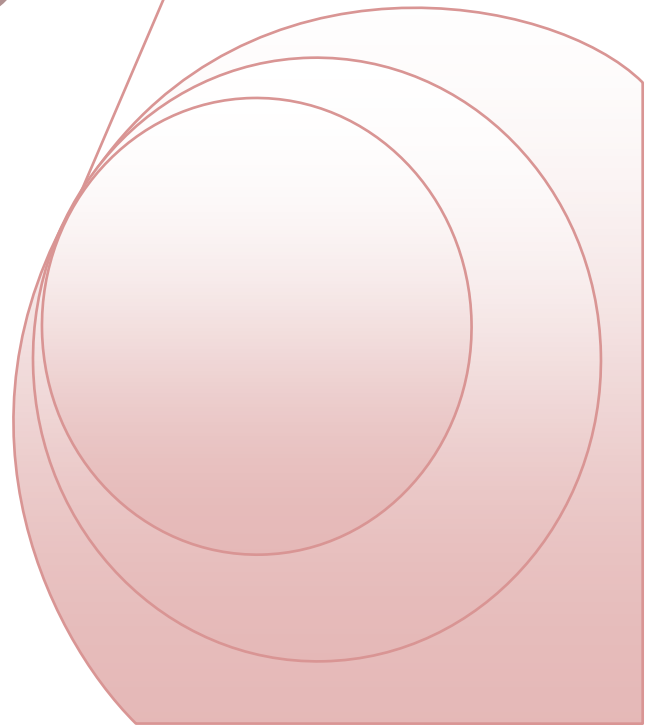
**Partie II :**

**Conception**



## **CHAPITRE 3 :**

### **CONCEPTION UML**



## CHAPITRE 3: Conception UML

### 3.1. Introduction

Ce chapitre présente les concepts de base du langage de modélisation UML, pour permettre une bonne lecture de ce qui va suivre et exprimer de manière uniforme l'analyse, la conception et la réalisation d'une application informatique. Par la suite nous abordons les notions d'UML (Unified Modeling Languages).

### 3.2. Définition

UML (*unifies Mödling langage*) est un langage de modélisation qui se base sur une sémantique précise et sur des notations graphiques, il est indépendant du domaine d'application et du langage d'implémentation. UML est ensemble de graphismes définis pour représenter les objets, leurs types et leurs interactions, leurs évolutions et leurs assemblages. Donc il représente une grande richesse et diversité. Ce qui facilite la comparaison et l'évolution des solutions.

### 3.3. Les diagrammes UML

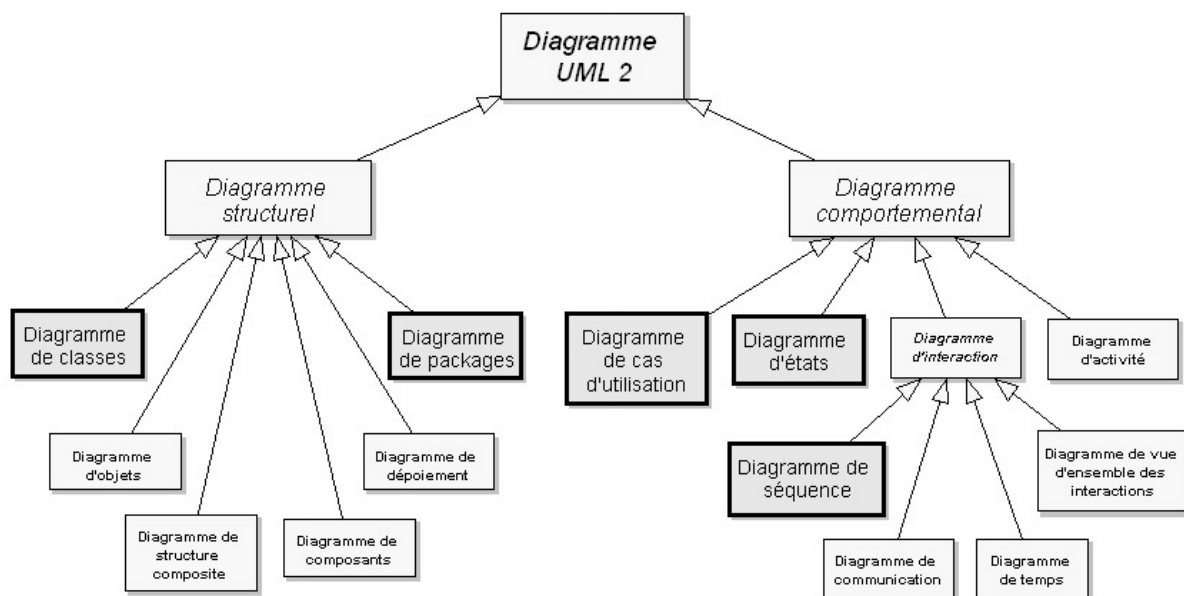


Figure 1: Les diagrammes UML.

✚ Nous signalons ici que dans un projet, il n'est pas nécessaire d'utiliser tous ces diagrammes. Le concepteur à l'intérêt de choisir parmi ces diagrammes ceux qui jugent utiles à son projet.

### 3.4 La description textuelle des cas d'utilisation :

#### 3.4.1 cas d'utilisation « s'authentifier »

Cas : authentification.

Acteur : l'administrateur.

But : ce cas permet au administrateur d'accéder à l'application.

Pré condition : l'administrateur possède un compte.

Post condition : le menu de l'application est accessible.

Scenario nominal :

L'administrateur demande ouvrir l'application.

Le système affiche une fenêtre pour qu'il s'identifie.

L'administrateur saisit son login et mot de passe.

Le système vérifie et demande la validation.

L'administrateur valide leur saisie.

Le système lance la menu principal d'application.

Scenario alternatif:

Le nom de l'administrateur et/ou le mot de passe incomplète ou erroné :

Le système affiche un message d'erreur.

Scenario exception:

l'administrateur ne saisit pas le bon mot de passe 3 fois. L'application se ferme.

#### 2/-cas d'utilisation « gérer les demande des citoyens »

##### 2.1/-cas d'utilisation « ajouter citoyen »

Cas : ajouter citoyen.

acteur : l'administrateur.

But : ca cas permet d'ajouter un citoyen.

Pré condition : l'administrateur possède s'authentifie.

Post condition : un nouveau citoyen est ajouté.

Scenario nominal :

L'administrateur demande le formulaire «ajouter citoyen ».

## Chapitre 3: Conception UML

---

Le système affiche le formulaire.

L'administrateur saisit les informations d'un citoyen.

Le système vérifie la saisie des informations et demande de validation.

L'administrateur valide la saisie.

Le système ajoute le nouveau citoyen.

Scenario alternatif:

Les informations sont incomplète ou erroné :

Le système affiche un message d'erreur.

Scenario exception:

L'administrateur ne valide pas l'ajout donc l'opération est annulée.

### **3.4.2 cas d'utilisation « supprimer citoyen »**

Cas : supprimer citoyen.

Acteur : l'administrateur.

But : ce cas permet de supprimer un citoyen existant.

Pré condition: l'administrateur s'authentifie.

Post condition : le citoyen est supprimé.

Scenario nominal :

L'administrateur demande le formulaire «supprimer citoyen ».

Le système affiche le formulaire.

L'administrateur sélectionne le citoyen qu'il veut supprimer.

Le système demande au administrateur de confirmer la suppression.

L'administrateur valide la suppression.

Le système supprime le citoyen et affiche au administrateur « citoyen supprimé ».

Scenario exception:

L'administrateur ne valide pas la suppression. alors l'opération est annulée.

### **3.4.3 cas d'utilisation « rechercher citoyen »**

Cas : rechercher citoyen.

Acteur : l'administrateur.

## Chapitre 3: Conception UML

---

But : ce cas permet de rechercher un citoyen .

Pré condition: l'administrateur s'authentifie.

Post condition : les informations de citoyen sont affichées.

Scenario nominal :

L'administrateur demande le formulaire «rechercher citoyen ».

Le système affiche le formulaire.

L'administrateur saisit les informations nécessaires d'un citoyen.

L'administrateur lance la recherche.

Le système affiche les informations de citoyen concerné.

Scenario alternatif:

Les informations sont incomplètes ou erronées :

Le système affiche un message d'erreur.

### 3.4.4 cas d'utilisation « gérer le dossier\_citoyen »

#### 3.4.4.1 cas d'utilisation « ajouter dossier »

Cas : ajouter dossier.

Acteur : l'administrateur.

But : ce cas permet d'ajouter un dossier.

Pré condition : l'administrateur s'authentifie.

Post condition : un nouveau dossier est ajouté.

Scenario nominal :

1. l'administrateur demande le formulaire «ajouter dossier ».
2. Le système affiche le formulaire.
3. l'administrateur saisit les informations de dossier.
4. Le système vérifie la saisie des informations et demande de validation.
5. l'administrateur valide la saisie.
6. Le système ajoute le nouveau dossier.

Scenario alternatif:

## Chapitre 3: Conception UML

---

1. les informations sont incomplètes ou erroné :

Le système affiche un message d'erreur.

Scenario exception:

L'administrateur ne valide pas l'ajout donc l'opération est annulée.

### 3.4.4.2 cas d'utilisation «supprimer dossier»

Cas : supprimer dossier.

Acteur : l'administrateur.

But : ce cas permet de supprimer un dossier existant.

Pré condition: l'administrateur s'authentifie.

Post condition : le dossier est supprimé.

Scenario nominal :

1. l'administrateur demande le formulaire «supprimer dossier ».
2. Le système affiche le formulaire.
3. l'administrateur sélectionne le dossier qu'il veut supprimer.
4. Le système demande à l'administrateur de confirmer la suppression.
5. l'administrateur valide la suppression.
6. Le système supprime le dossier et affiche à l'administrateur « dossier supprimé ».

Scenario exception:

L'administrateur ne valide pas la suppression. Alors l'opération est annulée.

### 3.4.4.3 cas d'utilisation « rechercher dossier »

Cas : rechercher dossier.

Acteur : l'administrateur.

But : ce cas permet de rechercher le dossier d'un citoyen .

Pré condition: l'administrateur s'authentifie.

Post condition : le dossier d'un citoyen est affiché.

Scenario nominal :

1. l'administrateur demande le formulaire «rechercher dossier\_citoyen ».

## Chapitre 3: Conception UML

---

2. Le système affiche le formulaire.
3. l'administrateur saisit le code (identifie, numéro\_dossier , date de création, ...).
4. l'administrateur lance la recherche.
5. Le système affiche les données de dossier d'un citoyen.

Scenario alternatif:

1. Le code saisi est incomplète ou erroné :

Le système affiche un message d'erreur.

### **3.4.5 cas d'utilisation « archiver dossier»**

Cas: rechercher dossier.

Acteur : l'administrateur.

But : ce cas permet l'archivage électronique d'un dossier.

Pré condition: l'administrateur s'authentifie.

Post condition : le dossier d'un citoyen est archivé.

Scenario nominal :

1. l'administrateur demande le formulaire «archiver dossier».
2. Le système affiche le formulaire.
3. l'administrateur saisit les renseignements concernant le dossier.
4. Le système vérifie la saisie des renseignements et demande de validation.
5. l'administrateur valide la saisie.
6. Le système archive le dossier d'un citoyen.

Scenario alternatif:

Le code saisi est incomplète ou erroné :

Le système affiche un message d'erreur.

Scenario exception:

L'administrateur ne valide pas l'archivage. Alors l'opération est annulée.

### **3.4.6 cas d'utilisation « contacter service»**

Cas: contacter service.

Acteur : l'administrateur.

## Chapitre 3: Conception UML

---

But : ca cas permet au administrateur contacte avec las services.

Pré condition: l'administrateur s'authentifie.

Post condition : l'administrateur est contacté.

Scenario nominal :

1. l'administrateur demande le formulaire «contacter service».
2. Le système affiche le formulaire.
3. l'administrateur saisit les informations de service (nom, code, spécialité ;..).
4. l'administrateur envoie son formulaire.
5. Le système vérifie les informations saisie et demande de validation.
6. Le système affiche un message envoyé.

Scenario alternatif:

Les informations saisie est incomplètes ou erroné :

Le système affiche un message d'erreur.

Scenario exception:

L'administrateur ne valide pas l'envoi. Alors l'opération est annulée.

### **3.4.7 cas d'utilisation « remplir le formulaire de permis de construire»**

Cas : remplir le formulaire de permis de construire.

Acteur : citoyen.

But : ca cas permet le citoyen demande le permis de construire.

Pré condition: /

Post condition : le citoyen est prend le permis de construire.

Scenario nominal :

Le citoyen rempli le formulaire de permis de construire.

Le citoyen envoie le formulaire remplie.

Le système vérifie la saisie et enregistre le formulaire dans la BDD.

Scenario alternatif:

Les informations sont incomplètes ou erronées :

## Chapitre 3: Conception UML

Le système affiche un message d'erreur.

Scenario exception:

L'administrateur ne valide pas la demande. Alors l'opération est annulée.

### 3.5 Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.



Figure2 : diagramme de cas d'utilisation.

# Chapitre 3: Conception UML

## 3.6 Diagramme de classe

Le diagramme de classes a pour objectif de décrire la structure des entités manipulées par les utilisateurs, le diagramme de classe décrit la structure interne du système. On définit dans ce diagramme les classes du système et les relations entre ces classes.

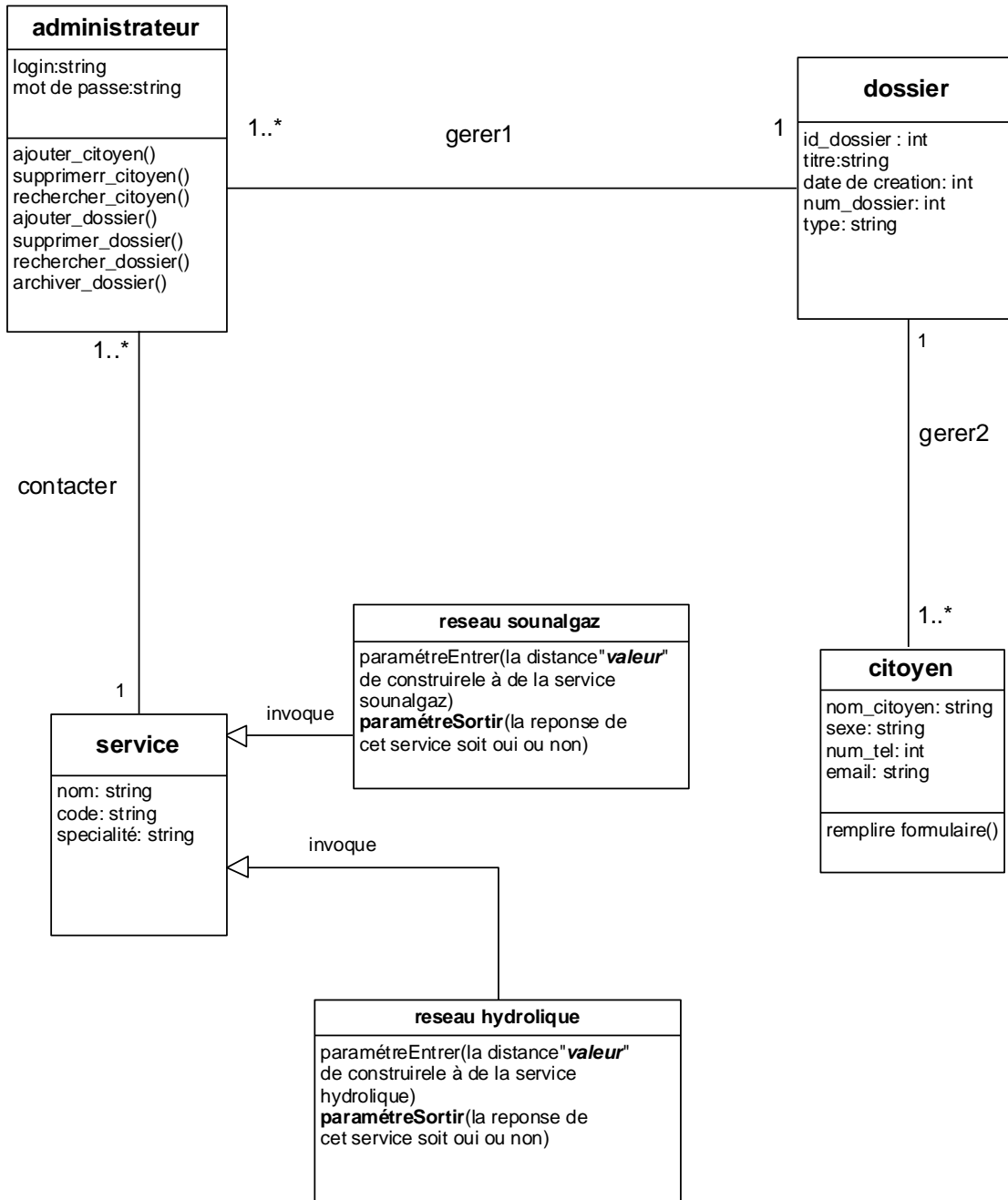


Figure3 : diagramme de classe de cas d'utilisation .

## 3.7 Diagramme de séquence

## Chapitre 3: Conception UML

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et les objets du système selon un ordre chronologique. 1/-« s'authentifie »

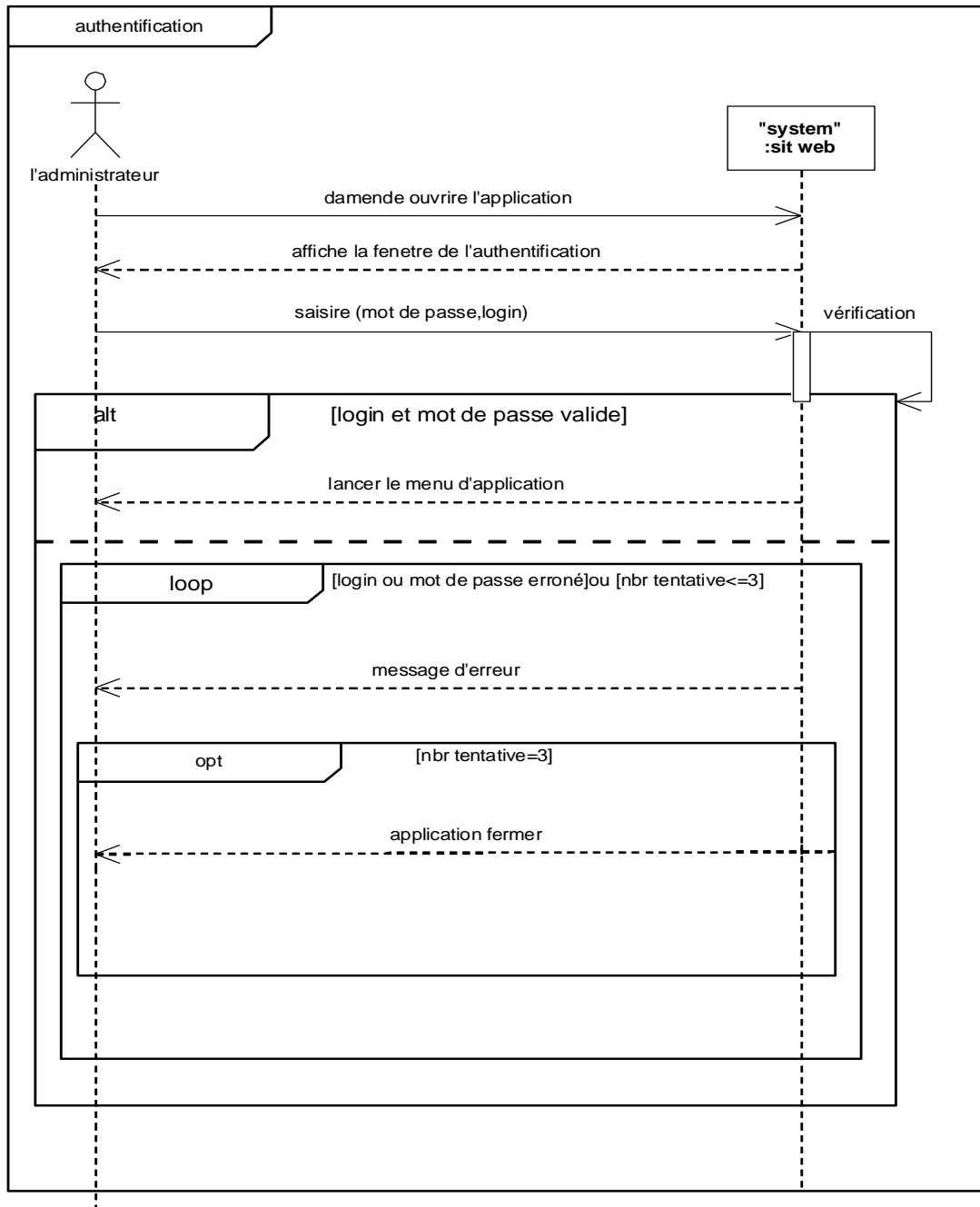
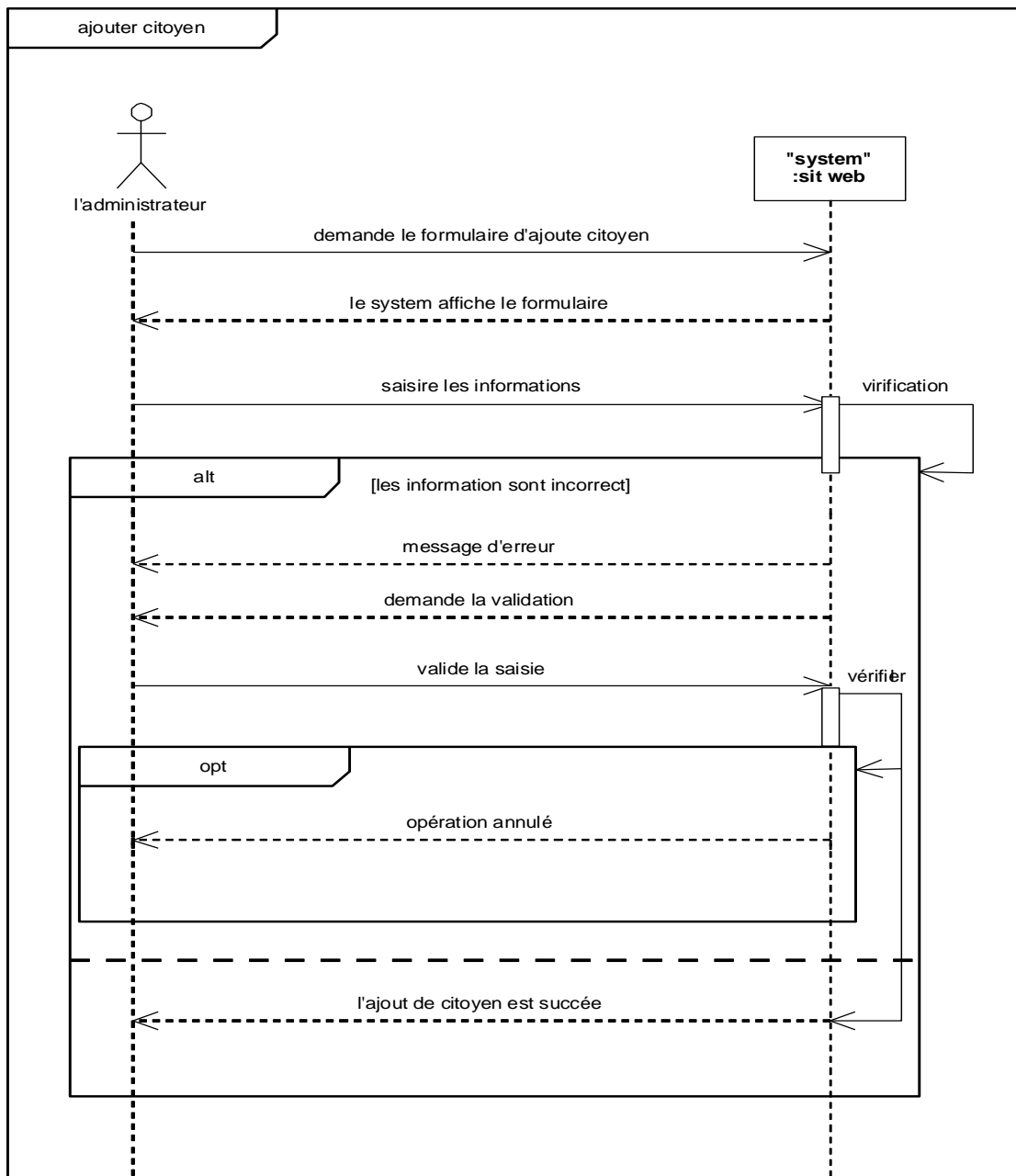


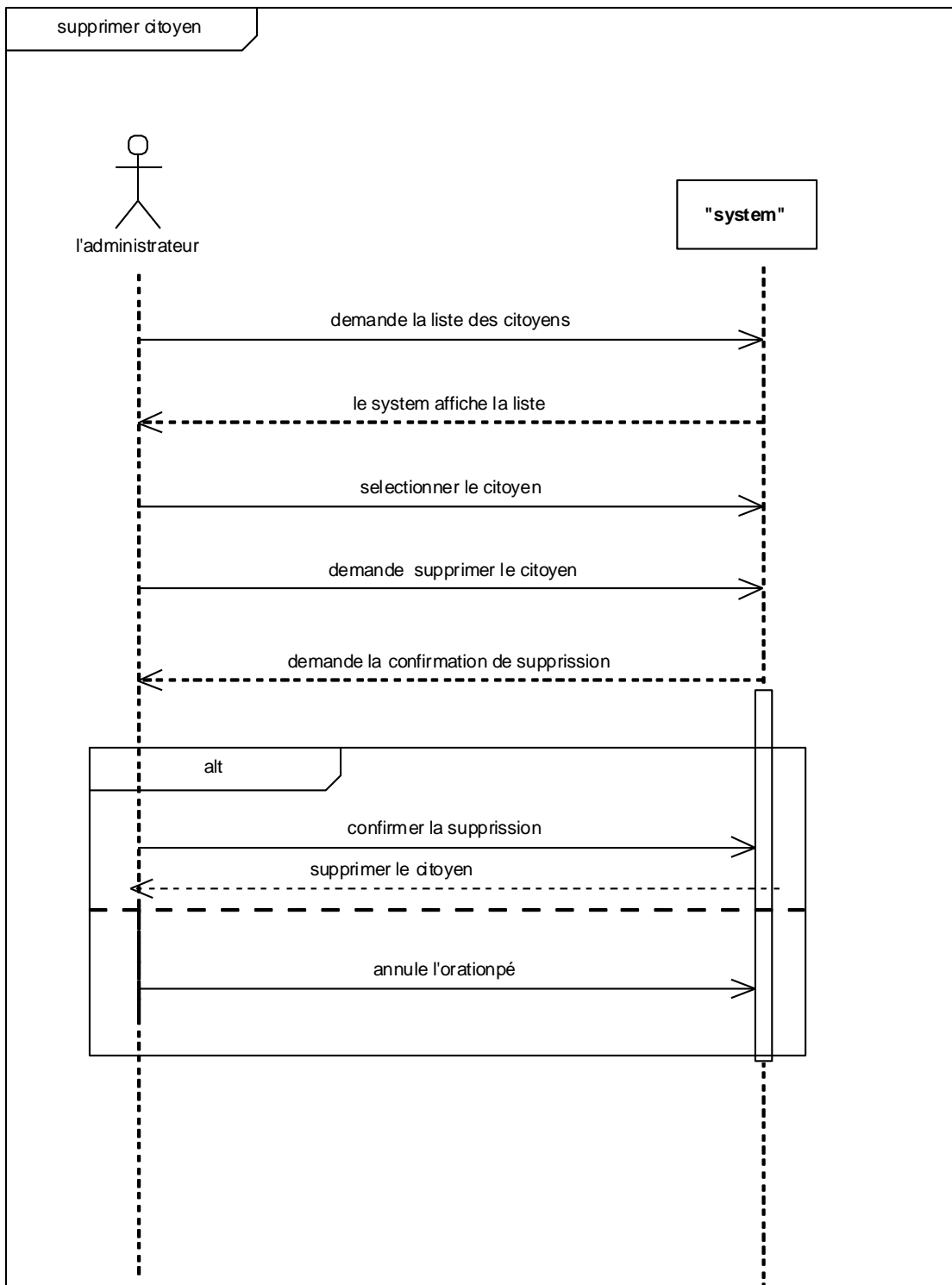
Figure 4 : diagramme de séquence de cas d'utilisation « s'authentifie ».

2/-« gérer les demandes des citoyens

2.1/-« ajouter citoyen »



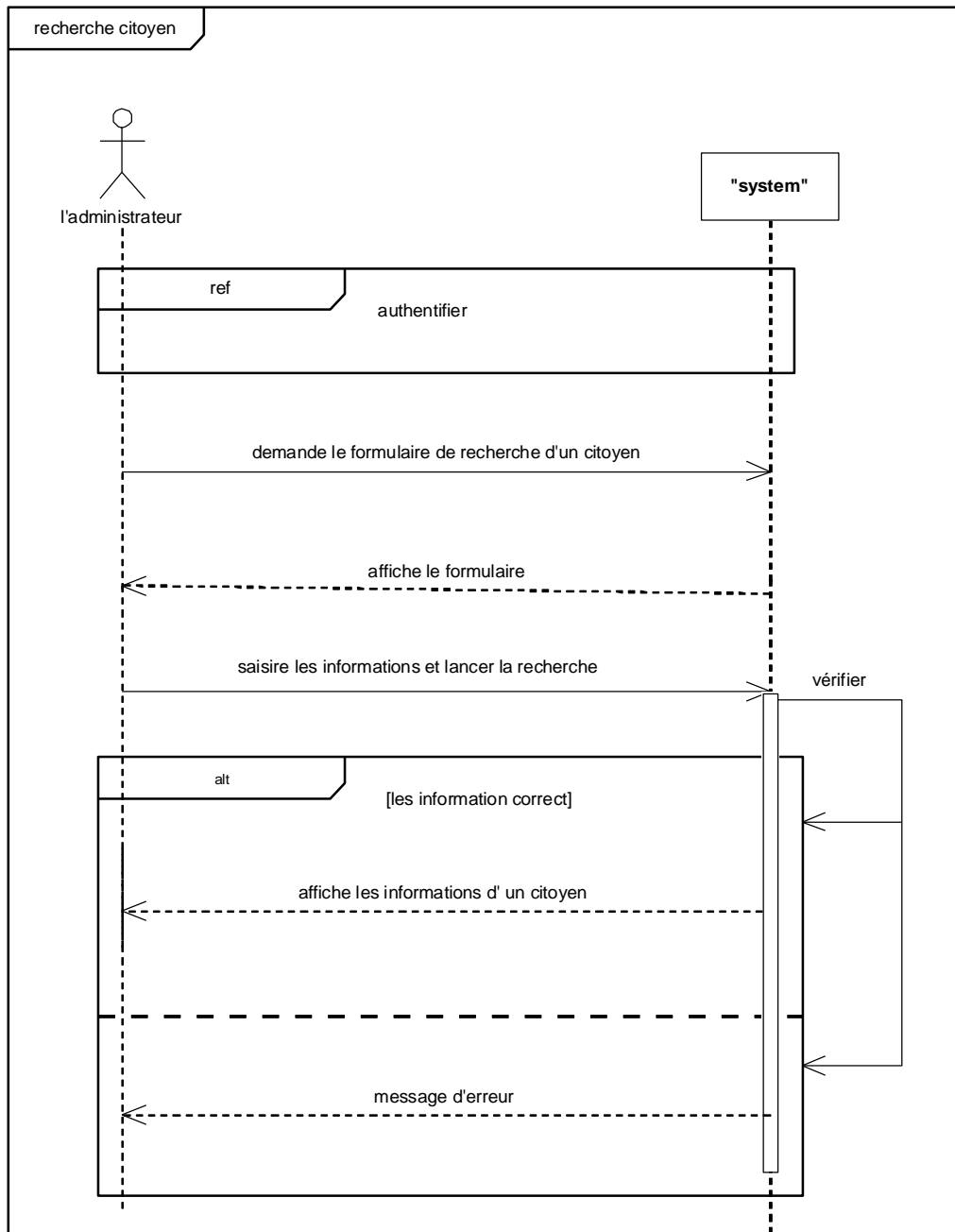
**Figure5** : diagramme de séquence de cas d'utilisation « ajouter citoyen ».



**Figure6** : diagramme de séquence de cas d'utilisation « supprimer citoyen ».

## Chapitre 3: Conception UML

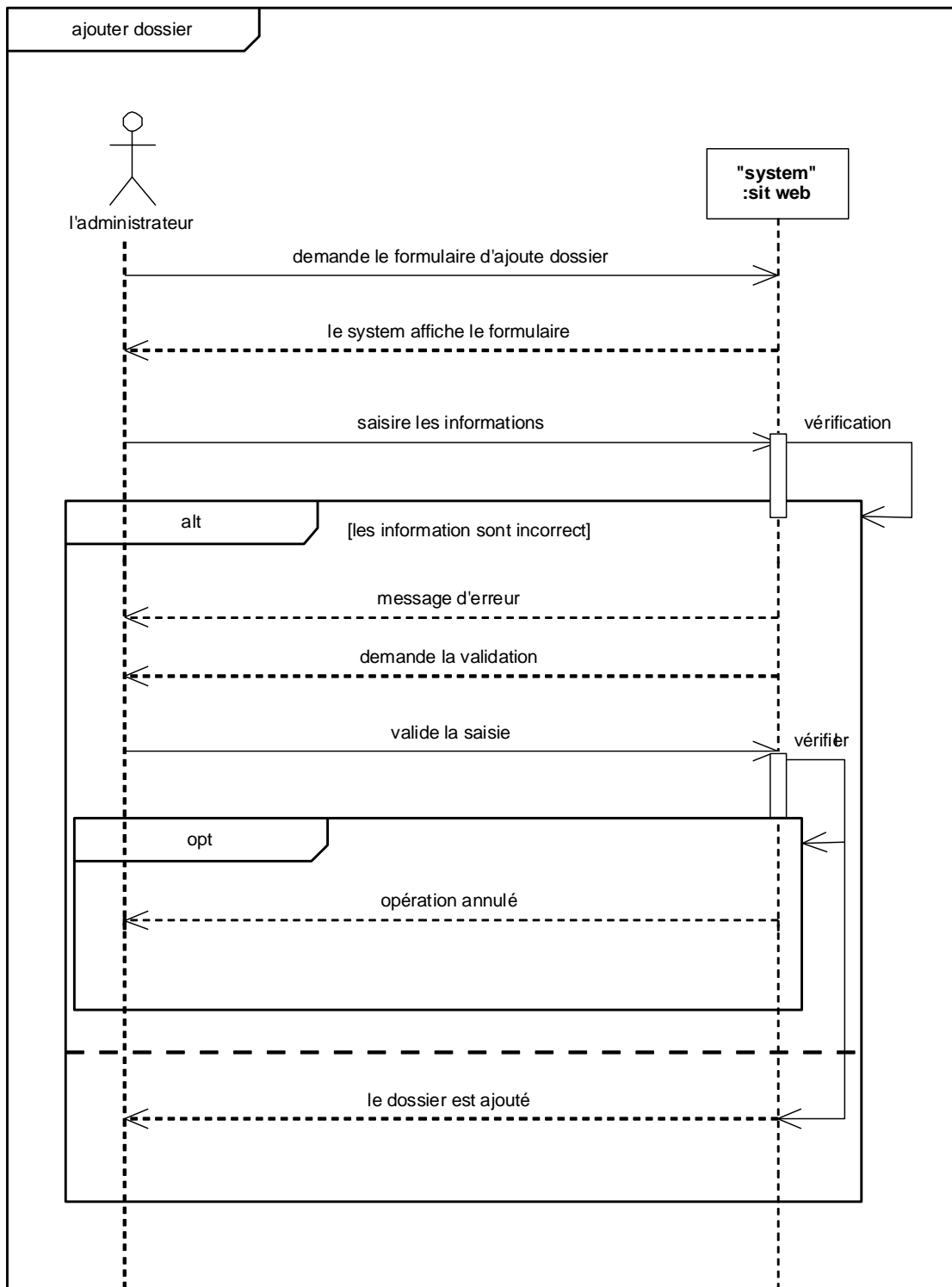
### 2.3/-« rechercher citoyen »



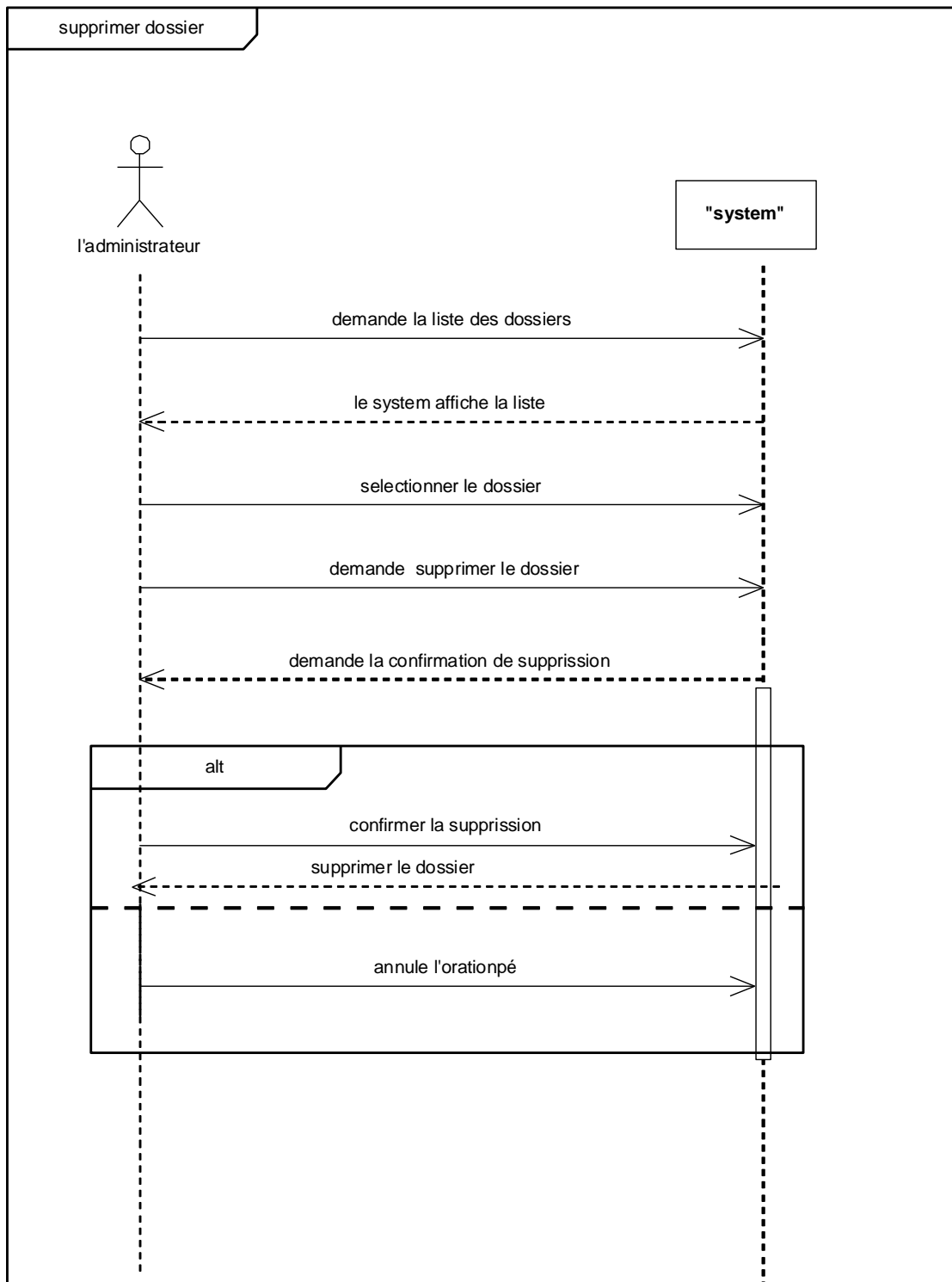
**Figure 7** : diagramme de séquence de cas d'utilisation « rechercher citoyen ».

### 3/-« gérer le dossier\_citoyen »

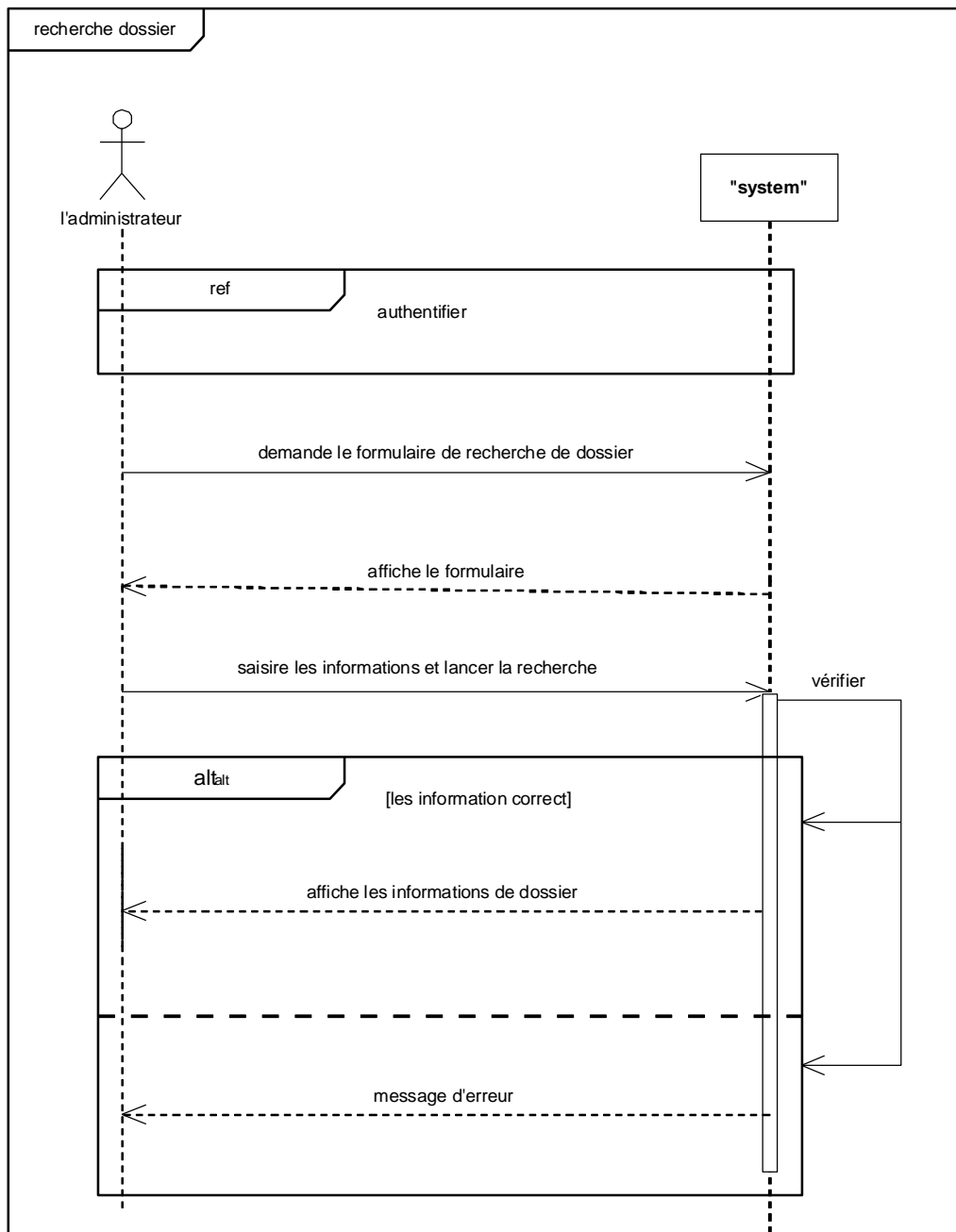
#### 3.1/-« ajouter dossier »



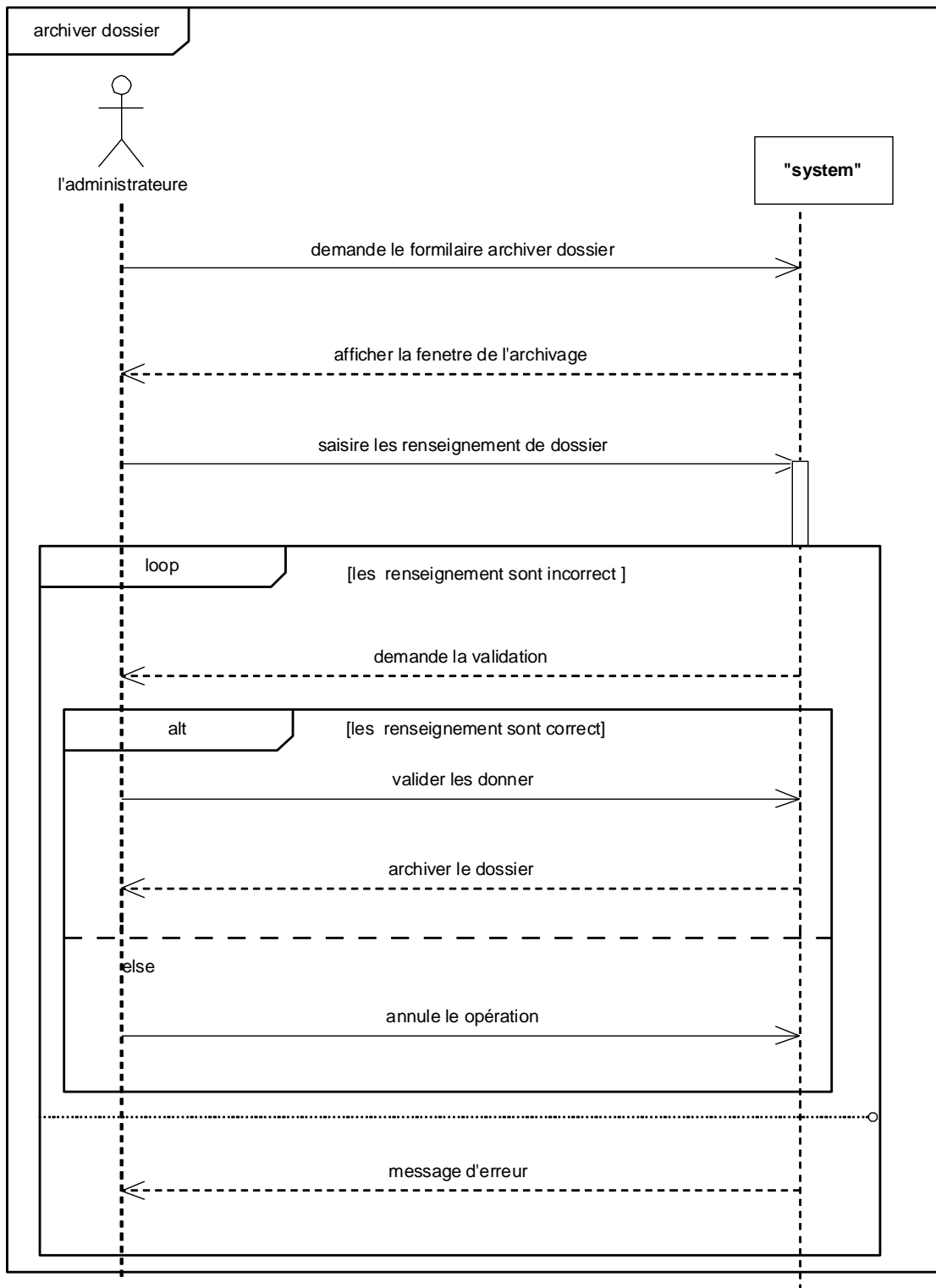
**Figure 8** : diagramme de séquence de cas d'utilisation « ajouter dossier ».



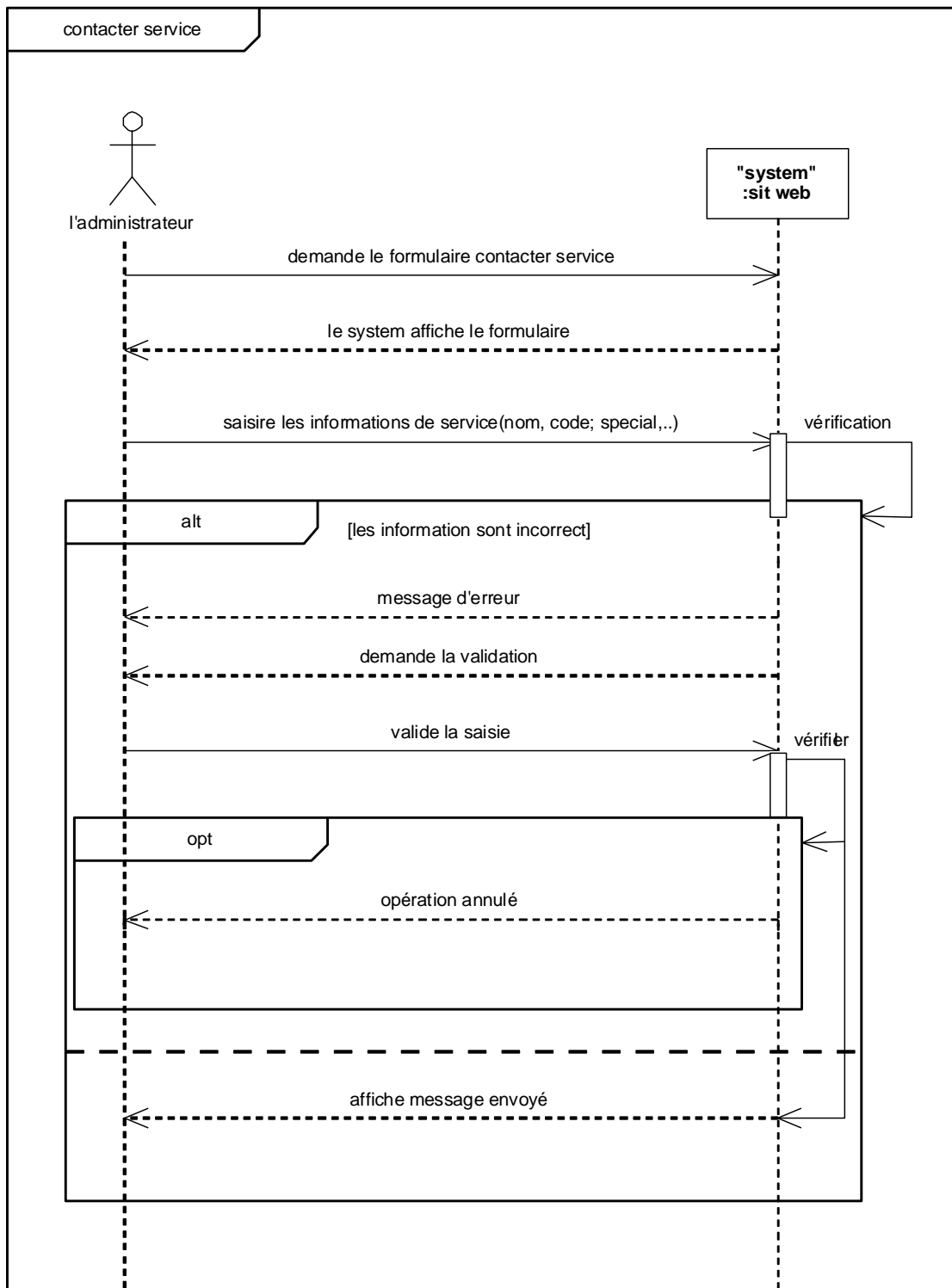
**Figure 9:** diagramme de séquence de cas d'utilisation « supprimer dossier ».



**Figure 10:** diagramme de séquence de cas d'utilisation « rechercher dossier ».

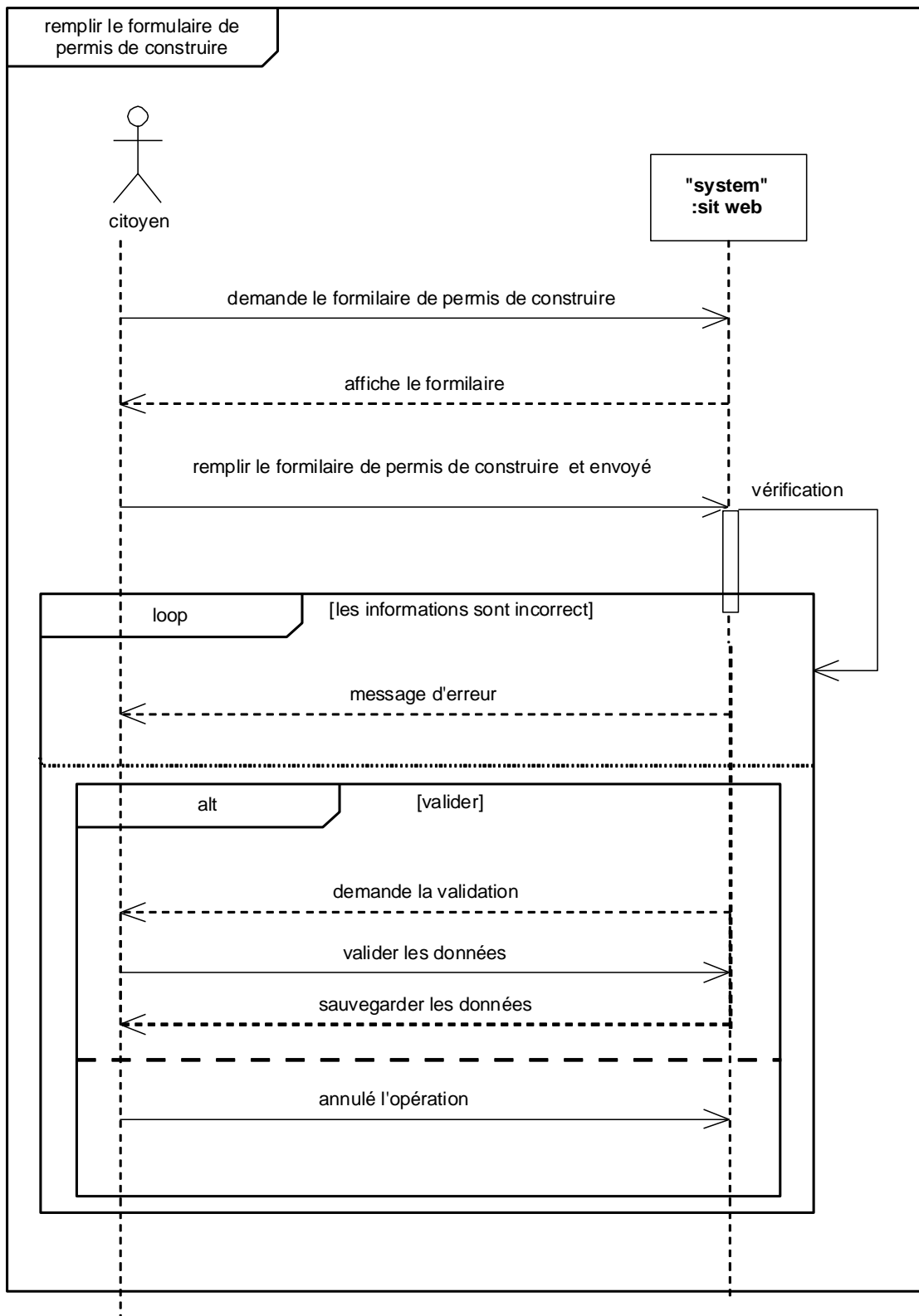


**Figure 11:** diagramme de séquence de cas d'utilisation « archiver dossier ».



**Figure 12:** diagramme de séquence de cas d'utilisation « contacter service ».

6/-« remplir le formulaire de permis de construire »



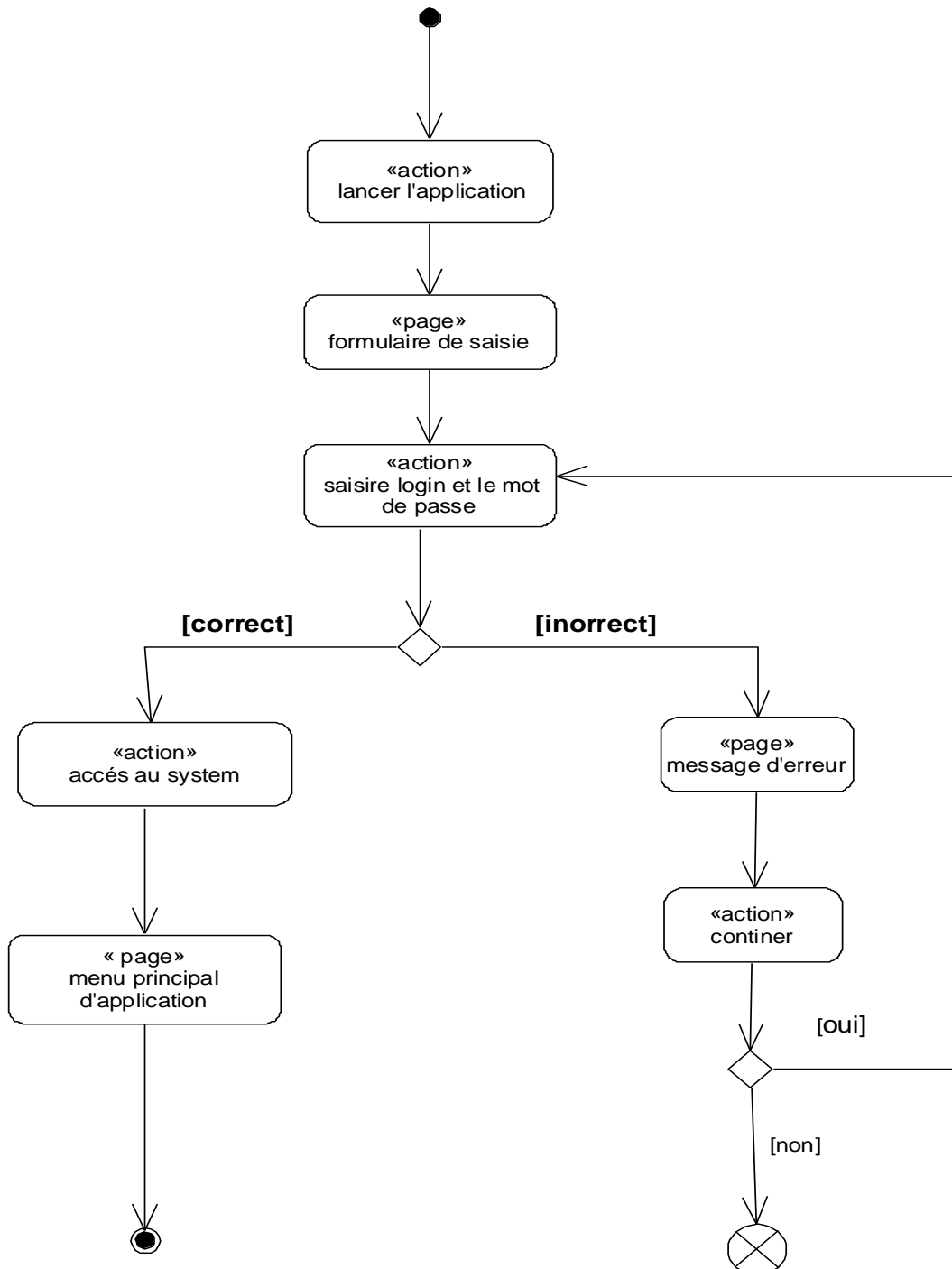
**Figure 13:** diagramme de séquence de cas d'utilisation «remplir le formulaire de permis de construire »

## 3.8 Diagramme d'activité

## Chapitre 3: Conception UML

Un diagramme d'activité permet de représenter le comportement interne d'une méthode (la réalisation d'une opération) ou d'un cas d'utilisation.

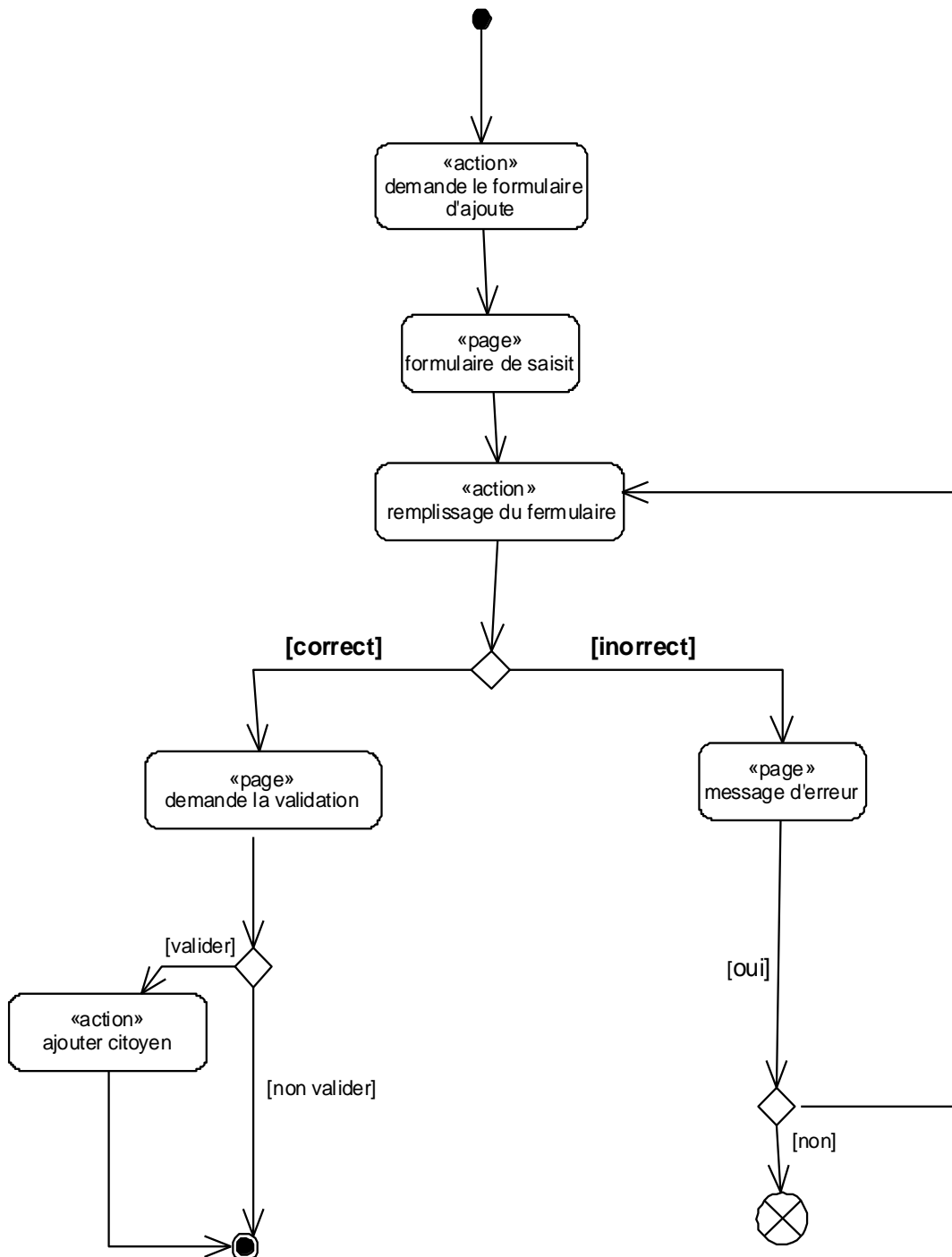
1/-« s'authentifier »



**Figure 14:** diagramme d'activité de cas d'utilisation « s'authentifier ».

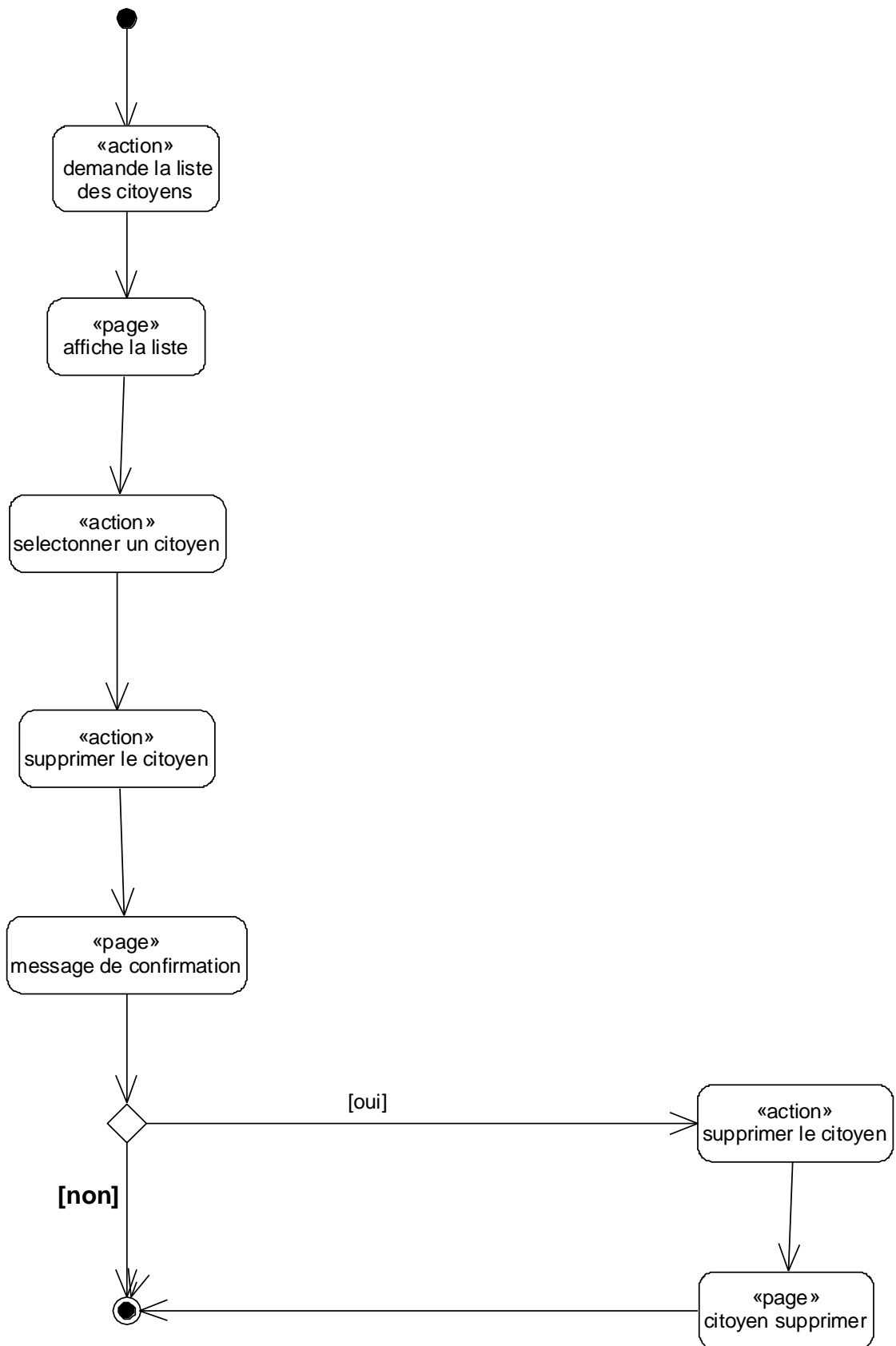
2/-« gérer les demande des citoyens »

## 2.1/-« ajouter citoyen »



**Figure 15:** diagramme d'activité de cas d'utilisation « ajouter citoyen ».

## 2.2/-« supprimer citoyen »



**Figure 16:** diagramme d'activité de cas d'utilisation «supprimer citoyen».

## 2.3/-« rechercher citoyen »

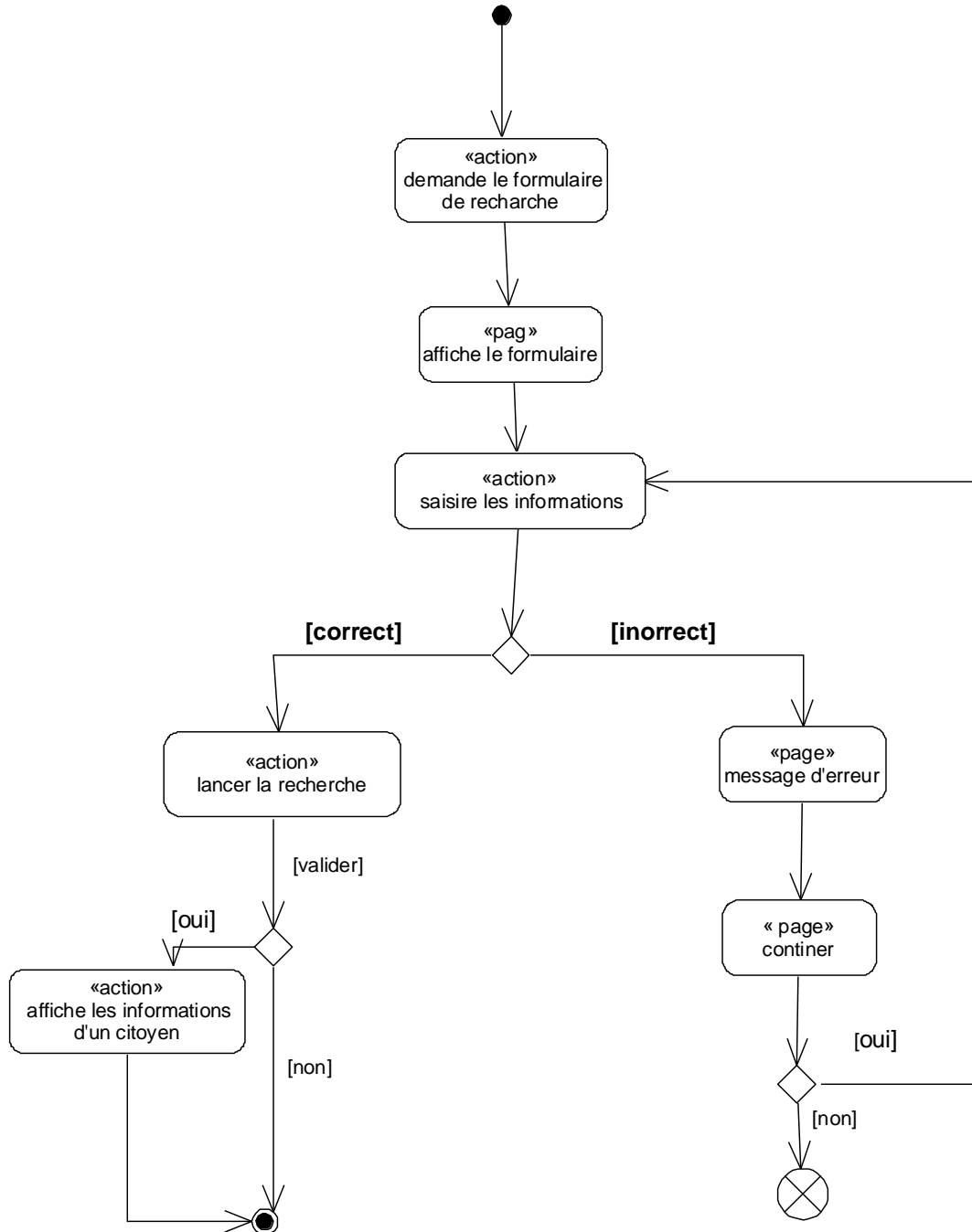
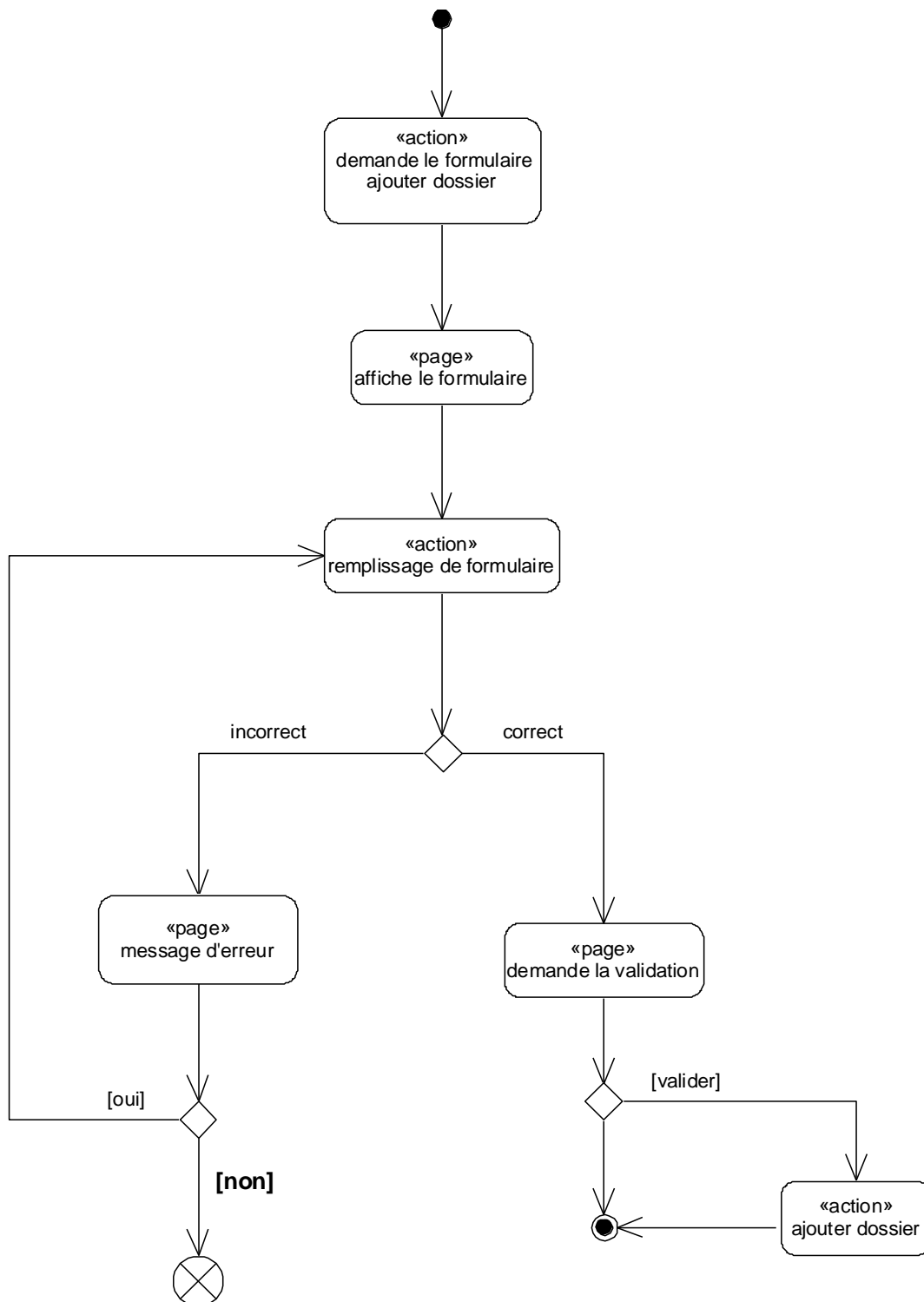


Figure 17: diagramme d'activité de cas d'utilisation «rechercher citoyen».

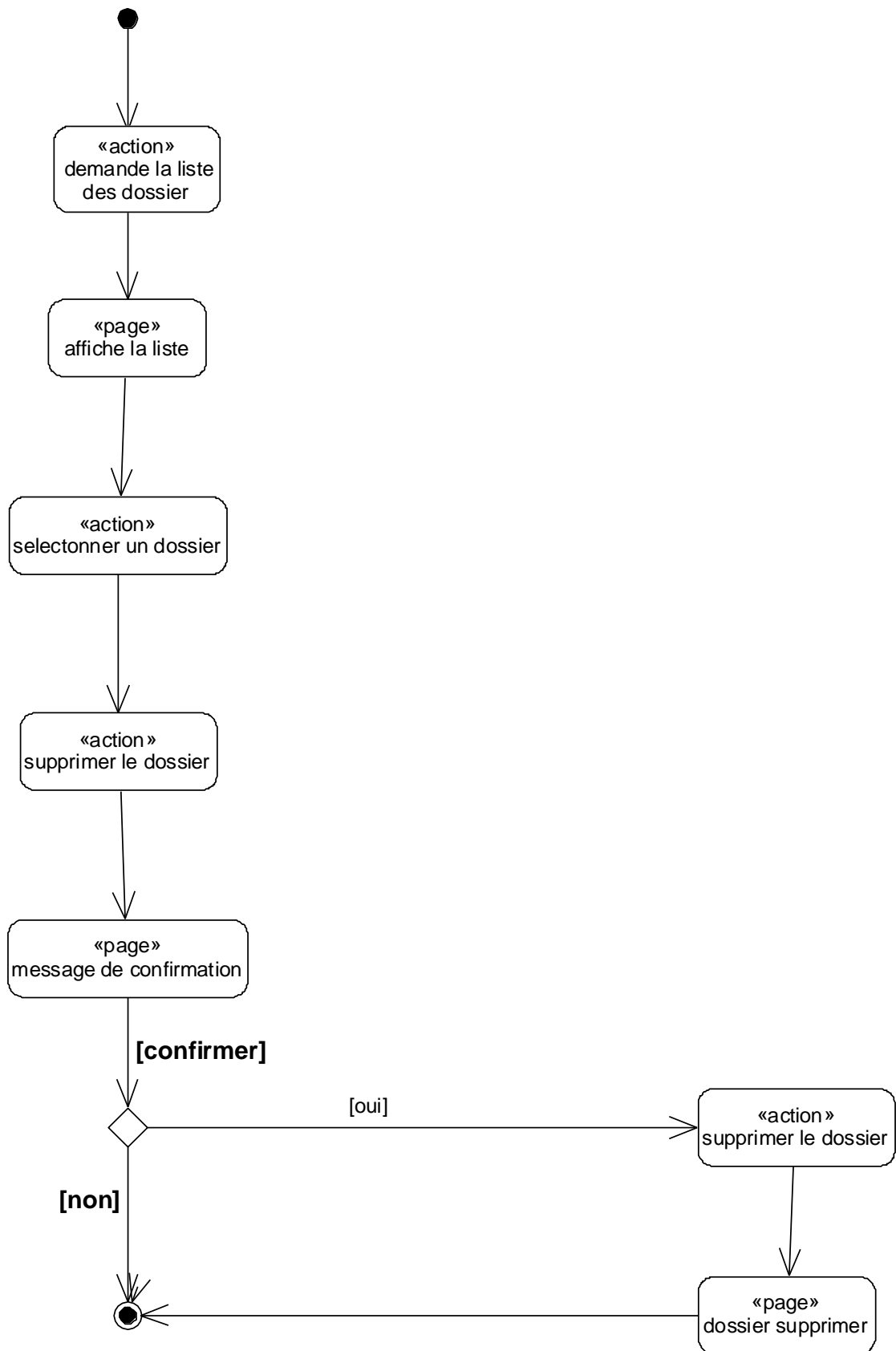
## 3/-« gérer les dossiers\_citoyens »

## 3.1/-«ajouter dossier »



**Figure 18:** diagramme d'activité de cas d'utilisation «ajouter dossier».

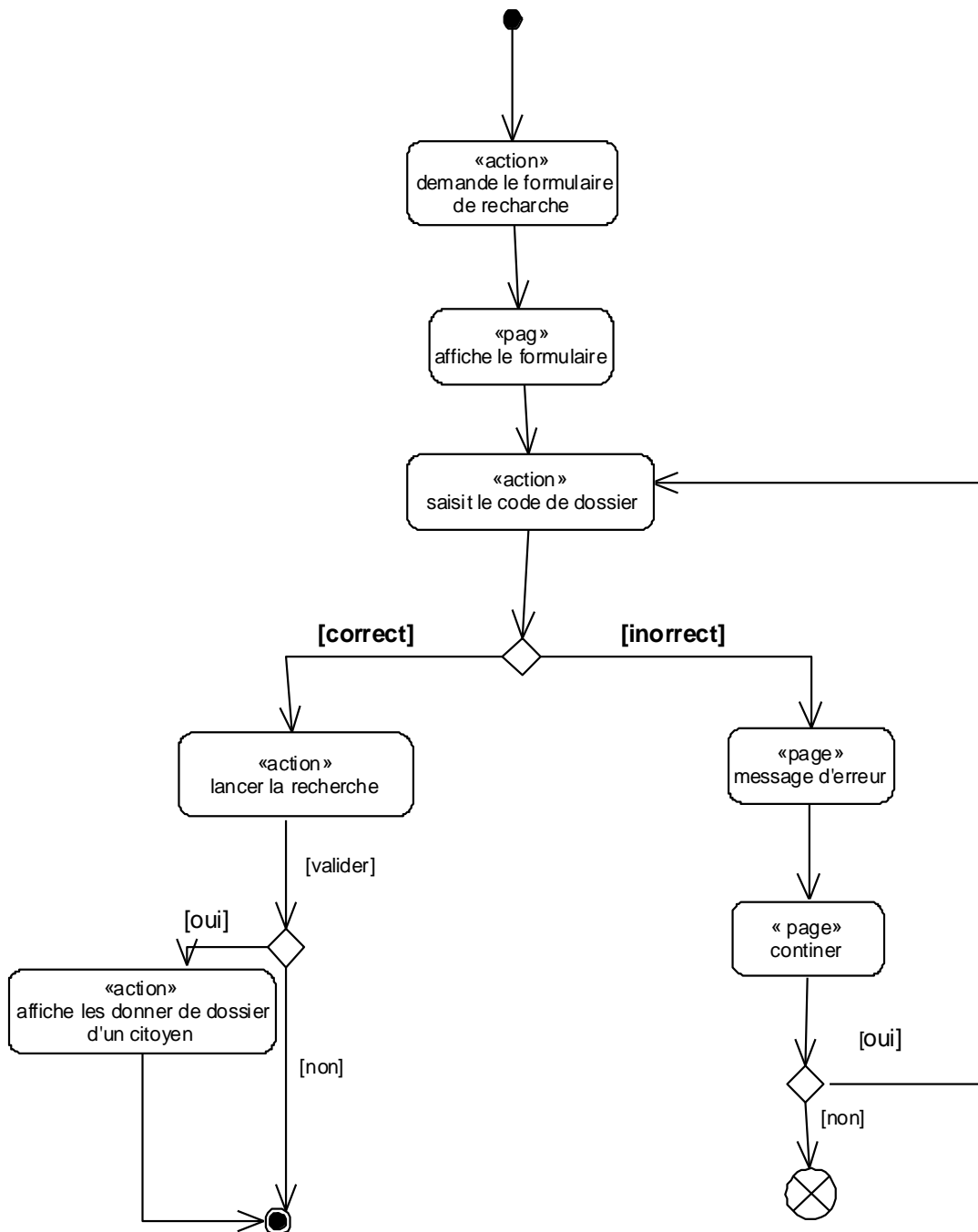
## 3.2/-«supprimer dossier »



**Figure 19:** diagramme d'activité de cas d'utilisation «supprimer dossier».

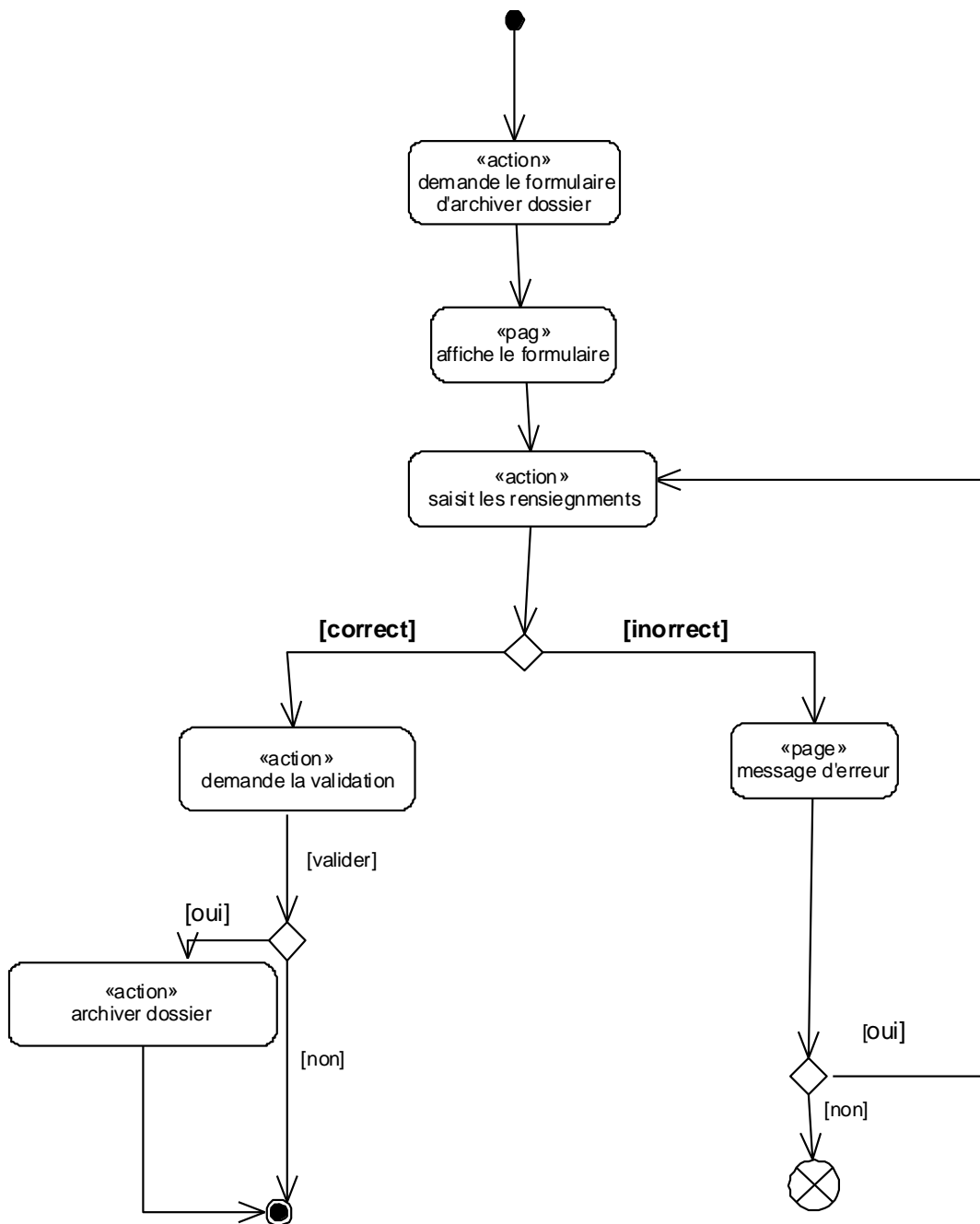
## Chapitre 3: Conception UML

### 3.3/-« rechercher dossier »



**Figure 20:** diagramme d'activité de cas d'utilisation «rechercher dossier».

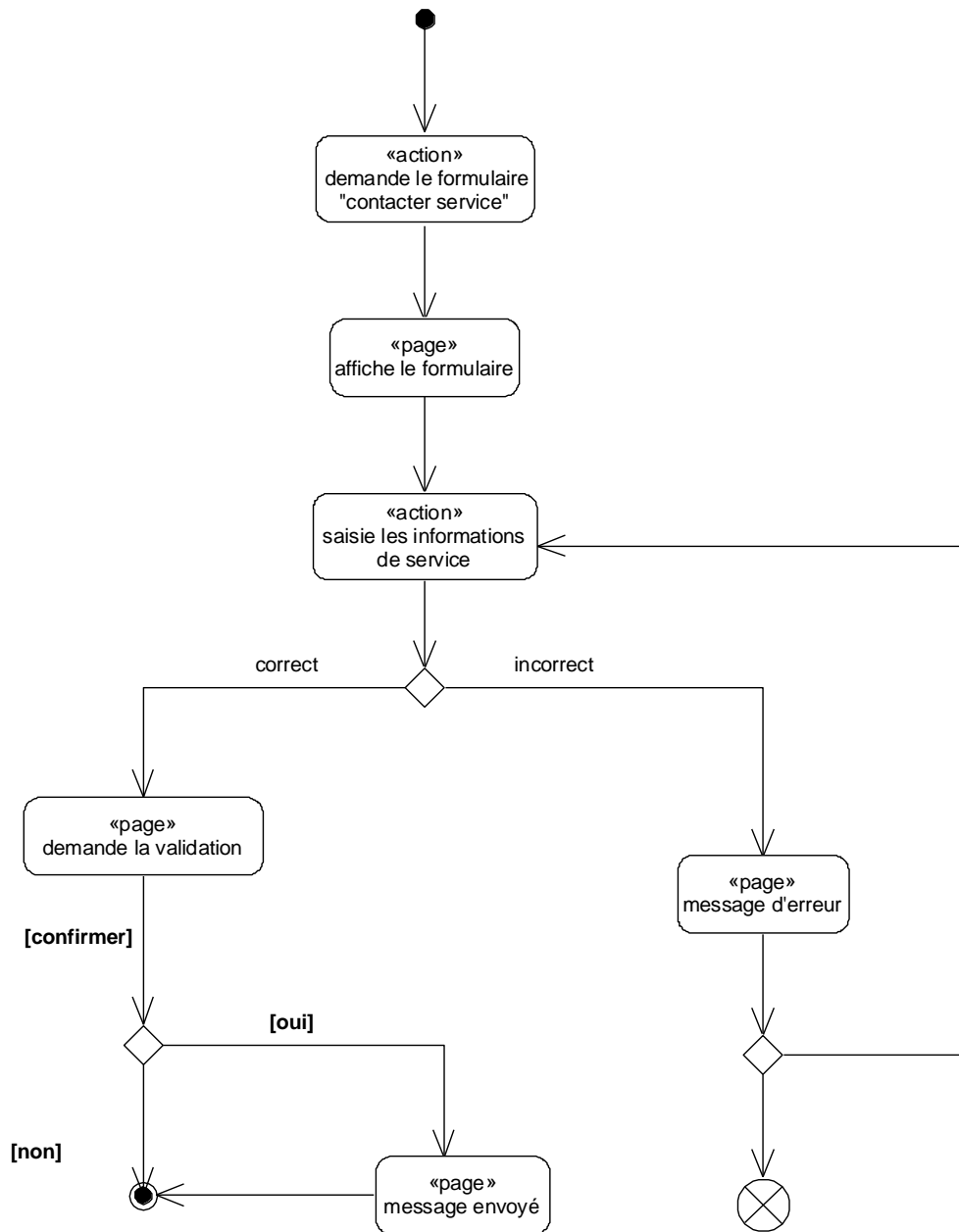
## 4/-« archiver dossier »



**Figure 21:** Diagramme d'activité de cas d'utilisation «archiver dossier».

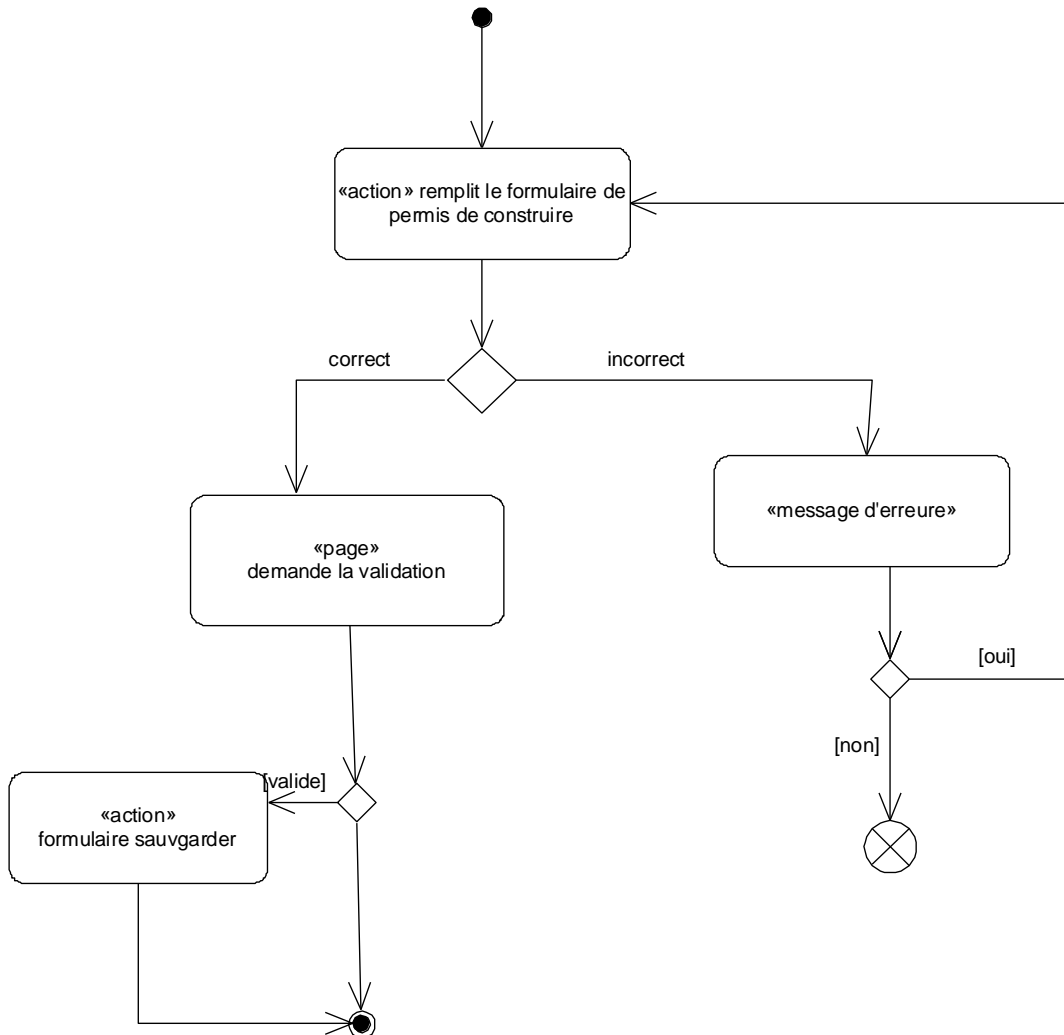
## Chapitre 3: Conception UML

5/-« contacter service »



**Figure 22:** diagramme d'activité de cas d'utilisation «contacter service».

6/-« remplir le formulaire de permis de construire »



**Figure 23:** diagramme d'activité de cas d'utilisation «remplir le formulaire de permis de construire».

### 3.9 Conclusion :

Dans le cadre de ce chapitre, nous allons définir quelques généralités portant sur la méthode et les outils utilisés pour réaliser notre projet. Nous allons présenter le langage de modélisation unifié UML (Unifie Mödling Lagunage)



**Partie III :**

**Réalisation**

The background features a large, faint red 'V' shape. Inside the upper right arm of the 'V', there are two sets of concentric circles. The upper set is larger, and the lower set is smaller. In the bottom right corner, there is a large, partially visible set of concentric circles. The text is centered within a rounded rectangular box in the middle-left area.

**CHAPITRE 4 :**  
**RÉALISATION DU SYSTÈME**

### CHAPITRE 4 : REALISATION DU SYSTEM

#### 4.1. Introduction

Ce chapitre consiste à représenter brièvement la structure, les langages de programmation et les outils utilisés pour la réalisation de notre application.

#### 4.2. Définition du Visual studio

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C# et Visual C++ utilisent tous le même environnement de développement intégré (IDE), qui permet le partage d'outils et facilite la création de solutions à plusieurs langages. Par ailleurs, ces langages utilisent les fonctionnalités du .NET Framework, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML[13].



#### 4.3 Visual studio 2010

Cette version est sortie en version finale le 12 avril 2010. Cette nouvelle version utilise la version 4.0 du .NET Framework dans le logiciel ainsi qu'une refonte de l'interface graphique propulsée par WPF<sup>3</sup>.

À cette occasion, le logo a changé, pour une version plus 'moderne'. Les différentes déclinaisons de Visual Studio ont elles aussi changé de nom par souci de clarification de l'offre.

Le numéro de version interne de Visual Studio 2010 est 10.0 (le symbole \_MSC\_VER étant défini comme 1600). [14]

#### 4.4 Objectif de la formation Visual studio 2010

Cette formation Visual studio 2010 vous permettra de voir « tout .NET » en 5 jours.

L'ensemble des connaissances en matière de conception d'applications distribuées avec .NET 4.0 sont traitées vous communiquant un panorama complet sur la question.

Le cours ne se contente pas d'effleurer les sujets en abordant de façon systématique l'essentiel de chaque technologie correspondant à un véritable usage sur le « terrain ».

À la fin de cette formation Visual Studio 2010, vous saurez entre autres :

- Utiliser au quotidien Visual Studio 2010.
- Programmer en C# 4 et/ou VB 10.
- Développer vos premiers clients riches avec Windows Forms.
- Accéder aux données avec ADO.NET.
- Développer vos premières applications Web en ASP.NET. [14]

### **4.5 Objectif d'utilisation de C# sous Microsoft Visual Studio 2010**

Cette formation aborde la structure des programmes, la syntaxe du langage et le détail des implémentations avec le framework .Net 4.0. Les améliorations du langage C# 4.0 avec Visual Studio 2010 sont présentées. Les développeurs intermédiaires pourront acquérir les connaissances et les compétences nécessaires pour développer des applications en C# avec les Framework .Net 4.0[14]

### **4.6 ASP.NET**

ASP.NET est un ensemble de technologies de programmation Web créé par Microsoft. Les programmeurs peuvent utiliser ASP.NET pour créer des sites Web dynamiques, des applications Web ou des Web services XML. La technologie est accessible grâce à une des options au choix :

- l'installation d'un serveur Web compatible ASP (IIS)
- l'intérieur de Visual Web Developer Express Edition
- ASPNET Web Matrix

ASP.NET fait partie de la plateforme Microsoft .NET et est le successeur de la technologie Active Server Pages (ASP).[16]

#### **4.6.1 Principe**

Bien qu'ASP.NET tienne son nom de l'ancienne technologie de développement de Microsoft, "ASP", les deux sont assez différentes. Microsoft a complètement repensé ASP.NET, en se basant sur le Common Language Runtime (CLR) partagé par tous les logiciels Microsoft .NET. Les programmeurs peuvent écrire du code ASP.NET en utilisant n'importe lequel des

langages de programmation supportés par le Framework .NET, généralement C#, Visual Basic.NET, Delphi.NET ou JScript .NET, mais aussi d'autres langages "indépendants" de Microsoft tels que Perl et Python. ASP.NET est, d'après Microsoft, censé avoir de meilleures performances que d'autres technologies basées sur des scripts<sup>[réf. souhaitée]</sup> car le code côté serveur est compilé en quelques simples DLL sur le serveur Web. Lors du développement, quand le code source est finalisé, la solution est precompilée avant d'être placée sur le serveur d'hébergement (publication).ASP.NET permet aux développeurs de passer plus facilement du développement classique d'applications Windows au développement d'applications Web en offrant la possibilité de créer des pages Web composées de *Widget* (ou zone de contrôle) similaires à celles des interfaces d'applications Windows habituelles.[16]

### 4.7 Les langages de programmation

#### 4.7.1 HTML

HTML « Hyper Text Markup Langage » est un langage permettant de décrire les différents composants d'un document : Définir des titres, construire des tableaux, mettre en formes des textes. Le HTML est un langage simple rédigé sous forme de texte et qui demande un simple éditeur de texte [15].

Le lien Hypertexte est un des principaux atouts du langage HTML, un lien permettant à l'utilisateur d'accéder rapidement à un autre emplacement du document ou à un autre page sur internet. Voici un exemple sur un code HTML :

```
<Html>

<Head>

<Title>Le titre de la page</title>

</Head>

<Body>

<h1>Mon premier titre</h1>

</Body>

</Html>
```

### 4.7.2 Css

Le langage css décrit la présentation des documents HTML et XML les standards définissant CSS sont publiés par (W3C)

CSS devient couramment utilisé dans la conception de sites web et bien supportés par les navigateurs web dans les années 2000. L'un des objectifs majeurs de CSS est de permettre la stylisation hors de document. Il est par exemple possible de ne décrire que la structure d'un document en HTML, et de décrire la présentation dans une feuille de style CSS séparée.

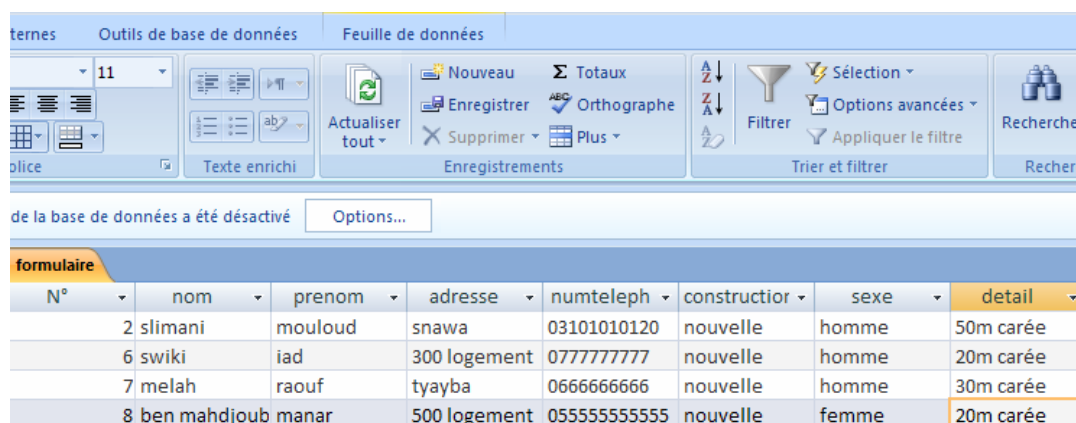
Les styles sont appliqués au dernier moment, dans le navigateur web des visiteurs qui consultent le document. Cette séparation fournit un certain nombre de bénéfices, permettant d'améliorer l'accessibilité, de changer plus facilement de structure de présentation, et de réduire la complexité de l'architecture d'un document.

### 4.7.3 SQL

Le langage SQL « Structure Query Language » est un langage normalisé d'intégration de base de données. Puis qu'il est normalisé, est-il indépendant du type des bases de données les mêmes commandes peuvent donc être exploitées quelle que soit la base utilisée « SQL client ». Les commandes SQL peuvent ainsi gérer tout type d'action sur le serveur de base de données.

## 4.8 base des données Microsoft Access

Microsoft Access (officiellement Microsoft Office Access) est un SGBD relationnel édité par Microsoft. Il fait partie de la suite bureautique MS Office Pro et le langage de programmation Visual C# for Application.

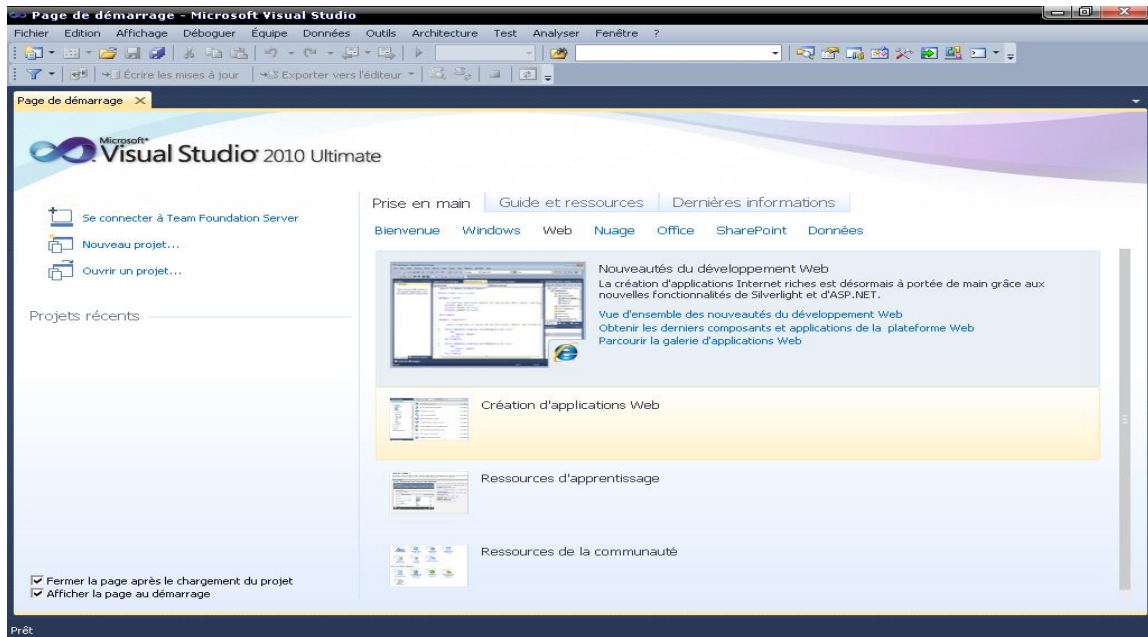


N°	nom	prenom	adresse	numteleph	constructeur	sexe	detail
2	slimani	mouloud	snawa	03101010120	nouvelle	homme	50m carée
6	swiki	iad	300 logement	0777777777	nouvelle	homme	20m carée
7	melah	raouf	tyayba	0666666666	nouvelle	homme	30m carée
8	ben mahdjoub	manar	500 logement	0555555555	nouvelle	femme	20m carée

Figure 1 base de données Microsoft Access

## 4. Page de démarrage

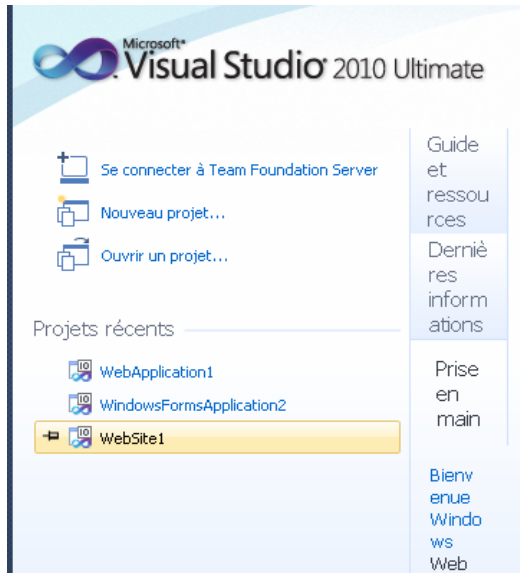
# Chapitre 5 : Réalisation du system



**Figure 2** la page de démarrage de Microsoft Visual studio 2010

## 4.10 Présentation de l'implémentation des web services

### Présentation des web services sous Microsoft Visual studio 2010



**Figure 3** les applications web.

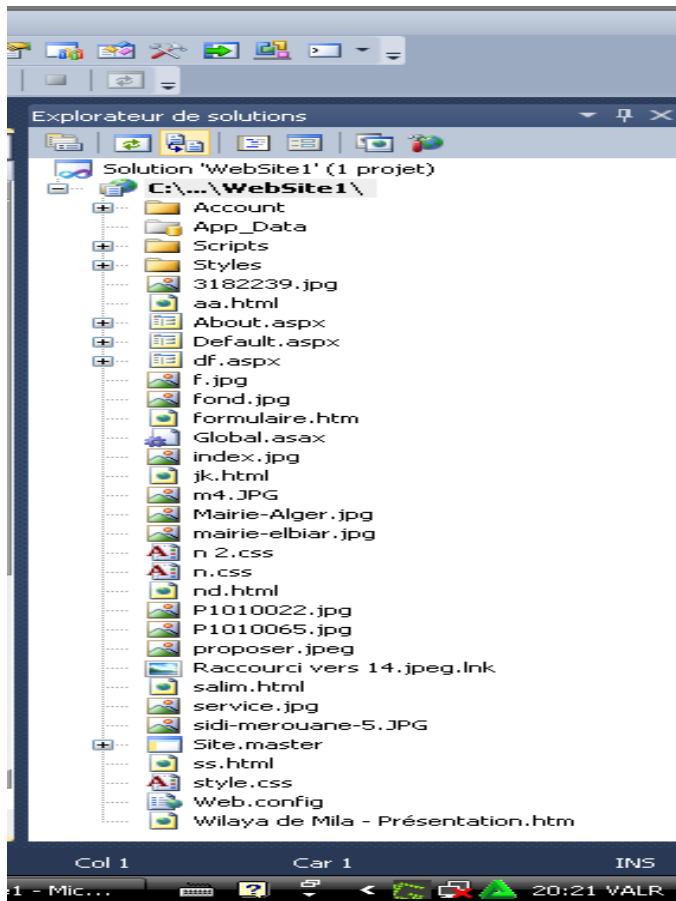


Figure 4 les pages web

Les services associés à le permis de construire

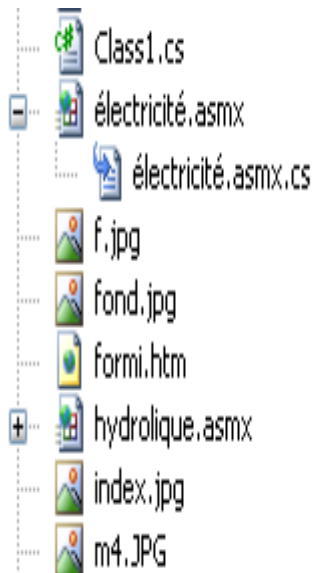


Figure 5 les web services

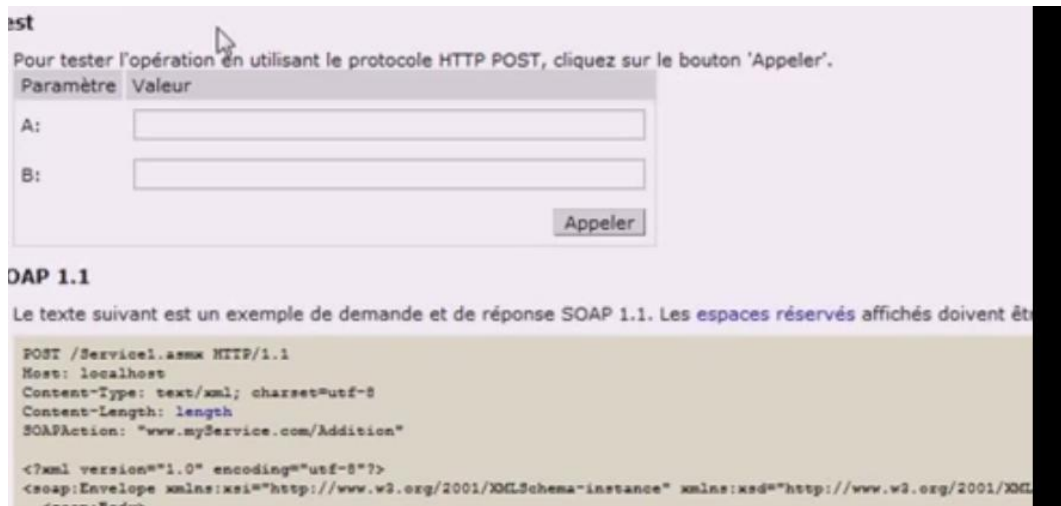


Figure 6 service web

### 4.11 Présentation du portail web

Les images suivantes illustrent le portail web associé à notre E-administration avec ses différentes pages :



Figure 7 la page d'accueil.

## Chapitre 5 : Réalisation du system

Formulaire de permis de construire

localhost:1027/formi.htm

**Vos information**

Nom :   
Prénom :   
eMail :   
num tel :   
Votre adresse :

**Vos construction**

constructions nouvelles  
 construction existents

hidrolique:   
sonalgaze :

Effacer Envoyer

Figure 8 la page de formulaire de demande de permis de construire

**Formulaire de contact**

Vous avez des services avec l'administration de la mairie de mila :  
Envoyez un message à : [mairie.mila@yahoo.fr](mailto:mairie.mila@yahoo.fr)  
Ou remplissez le formulaire de contact :

M.  Mme  Mlle

Nom\*   
Prénom\*   
Téléphone\*   
Mobile   
Fax   
Email\*   
Adresse postal   
Objet   
Message \*   
Envoyer

\* Champs obligatoires

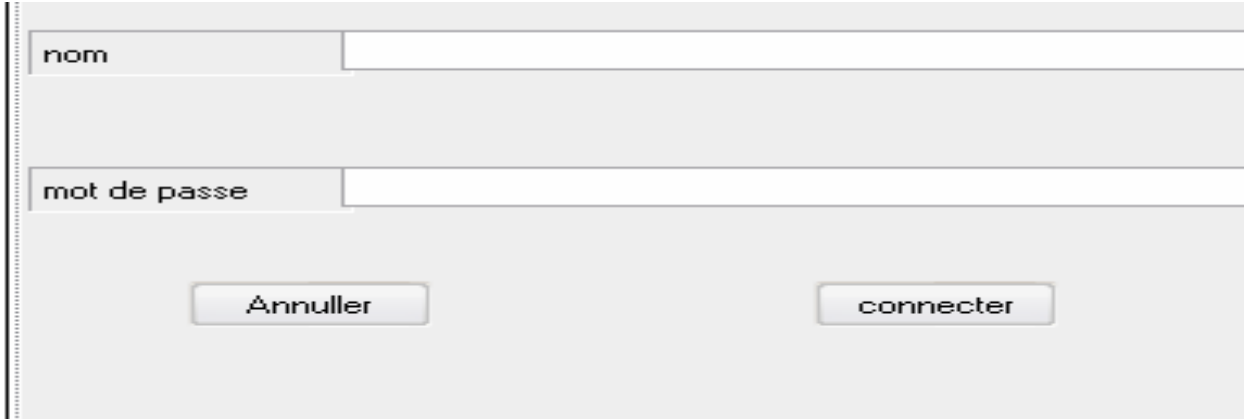
Figure 9 la page de contact

file:///D:/Nouveau...ier%20(13)/11.html

file:///D:/Nouveau dossier (13)/11.html

num\_tel:0310021210  
adresse:rue boussouf mila  
num\_tel\_ser\_permis de construire:0312020202  
adr\_email:mairie-mila yahoo.fr

Figure 10 la page d'annuaire



The image shows a web form for administrator login. It features two input fields: the first is labeled 'nom' and the second is labeled 'mot de passe'. Below the fields are two buttons: 'Annuler' on the left and 'connecter' on the right. The form is enclosed in a light gray border.

Figure 11 la page de connecté de l'administrateur.

### 4.12 Conclusion

Dans ce chapitre nous avons présenté la réalisation de notre application en utilisant les outils de développement des services web.

## CONCLUSION GÉNÉRALE

Les Web Services (Services Web) sont un sujet complexe, intégrant plusieurs domaines informatiques (documents, sécurité, programmation, implémentation, etc.). Le nombre de normes est impressionnant, mais est dépassé par la multitude de solutions d'implémentation.

Dans notre mémoire, nous avons essayé d'améliorer notre compréhension des Web Services, ferons un survol des normes les plus importantes de l'industrie (WSDL, SOAP et UDDI) et regarderons rapidement quels sont les alternatives d'implémentations.

Et nous avons appliqué cette méthode sur les administrations électroniques

# BIBLIOGRAPHIE

- [1]chwirf zahira , Un système de programmation fonctionnelle pour la composition de services Web, MEMOIRE DE MAGISTER, 2007/2008.
- [2] [fr.wikipedia.org/wiki/Page\\_Web\\_service.htm](http://fr.wikipedia.org/wiki/Page_Web_service.htm)
- [3] D. Booth, H. Haas, F. McCabe, and E. Newcomer. "Web services architecture ", aout 2003. <http://www.w3.org/TR/2003/WD-ws-arch-20030808/>.
- [4] Douglas K. Barry. "Web Services and Service-Oriented Architectures: The Savvy Manager's Guide". Morgan Kaufmann, 2003.
- [5]K. Heather. "Web services conceptual architecture (wsca 1.0)",may 2001. <http://www-306.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>.
- [6] P. Kellert and F. Toumani. "Les web services sémantiques". In Web sémantique, Action spéci\_que 32 CNRS/STIC, October 2003.
- [7] Jérôme Daniel. Extrait de livre "SERVICES WEB Concepts, techniques et outils" Edition vuibert Informatique 2003.
- [8] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, and H. Nielsen. "Simple object access protocol (soap) 1.1". Technical report, World Wide Web Consortium, may 2000. <http://www.w3.org/TR/SOAP/>.
- [9 ] Extrait de l'ouvrage "Services Web avec SOAP, WSDL, UDDI, ebXML... " de Jean-Marie Chauvet, © Eyrolles, 2002.
- [10] : Medjahed.B, Benattalah.B, Ouzzani.M, « Infrastructure for e-government web services » Published by the IEEE Computer Society 1089-7801/03/\$17.00©2003 IEEE IEEE INTERNET COMPUTING
- [11] [europa.eu\\_administration\\_electronique .htm](http://europa.eu_administration_electronique.htm)
- [12] M.bnhamada sadak ,cour MACOO, centre universitaire de mila . 2013/2014.
- [13]Microsoft Visual studio \_ wikpidia.html.
- [14] Formation Visual studio 2010, Visual Studio 2010 Conception et développement avec .NET 4 , Formation .NET \_ PLB.html.

[15] Boufenneche Amine, Boudjadja Sara, La Conception et Réalisation D'un Site Web Dynamique pour l'achat et la vente des immobiliers en ligne, Licence académique, 2012/2013.

[16] Asp.net\_ wikipidia.html.