

# MEMOIRE

*Présenté à*

L'Institut Supérieur d'Informatique  
et de Multimédia de Sfax

*en vue de l'obtention du diplôme de*

LICENCE FONDAMENTALE

*en Informatique et Multimédia*

*intitulé*

Conception et développement d'une application  
Web/Mobile de réclamations municipales  
en ligne

*Par*

**Fares Ben Slama  
Wajdi Gabsi**

*Soutenu le 7 Juin 2019, devant le jury composé de :*

**M. Mohamed Tmar**

*Président*

**Mme Raida Ktari**

*Rapporteur*

**Mme Rim Walha**

*Encadreur*

# ***Dédicace***

Je dédie ce mémoire

À mon père & ma mère, vous êtes pour moi une source de vie car sans vos sacrifices, votre tendresse et votre affection je ne pourrais arriver jusqu'au bout.

Que Dieu vous garde afin que votre regard puisse suivre ma destinée.

Mes sœurs qui ont été toujours présents pour moi.

À mes amis avec qui j'ai partagé des moments les plus agréables.

À tous ceux qui sont chers, proches de mon cœur, et à tous ceux qui m'aiment et qui aurait voulu partager ma joie...

***Wajdi***

# ***Dédicace***

Je dédie ce mémoire

À mes parents qui m'ont soutenu et encouragé durant ces années d'études.

À mon frère, ma grand-mère, ma famille et tous ceux qui ont partagé avec moi  
tous les moments d'émotion lors de la réalisation de ce travail.

Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

À tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de  
Succès.

À tous ceux que j'aime  
Merci

***Fares***



# Remerciements

Nous exprimons nos profondes gratitude et respectueuse reconnaissance à  
notre encadreur :

**Mme Rim Walha**

Pour sa bonne volonté d'accepter de nous encadrer, pour tout le temps qu'elle  
nous a encadré et pour tous les conseils qu'elle nous a prodigué.

Nous remercions aussi nos encadreurs de la société Tritux **Mme Salma**

**Abdelmoumen et Mme Nouha Hammami**

Pour leurs directives précieuses, et pour la qualité de leurs suivis durant toute la  
période de notre projet.

Nous tenons à remercier **Mr. Mounir Khalifa**

Qui a nous a donner l'honneur de nous accepter dans la société Tritux.

Nous remercions sincèrement **les membres du jury** qui ont accepté d'honorer  
cette soutenance, pour leurs présences et leurs savoirs.

Nos vifs remerciements s'adressent également à nos enseignants et à nos amis,  
et familles pour leur présence chaleureuse et leur encouragement. Nous  
remercions sincèrement.



# Table des matières

<b>Remerciements</b>	<b>v</b>
<b>Table des matières</b>	<b>vii</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Table des figures</b>	<b>xiii</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Etude Préalable</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Présentation de l'organisme d'accueil . . . . .	3
1.3 Cadre du travail . . . . .	4
1.4 Présentation du projet . . . . .	4
1.5 Etude de l'existant . . . . .	5
1.5.1 Description des solutions existantes . . . . .	5
1.5.2 Critique de l'existant . . . . .	9
1.5.3 Solution proposée . . . . .	10
1.6 Planning prévisionnel . . . . .	10
1.7 Conclusion . . . . .	11
<b>2 Analyse et spécification des besoins</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Identification des acteurs et des cas d'utilisation . . . . .	13
2.2.1 Identification des acteurs . . . . .	13
2.2.2 Identification des cas d'utilisation . . . . .	14
2.3 Les besoins . . . . .	15
2.3.1 Les besoins fonctionnels . . . . .	15
2.3.2 Les besoins non fonctionnels . . . . .	17
2.4 Diagramme de cas d'utilisation . . . . .	17

2.4.1	Affectation des priorités aux cas d'utilisation . . . . .	19
2.4.2	Raffinement des cas d'utilisation . . . . .	19
2.5	Conclusion . . . . .	29
<b>3</b>	<b>Analyse</b>	<b>31</b>
3.1	Introduction . . . . .	31
3.2	Analyse des cas d'utilisation . . . . .	31
3.2.1	Analyse du cas d'utilisation s'inscrire . . . . .	31
3.2.2	Analyse du cas d'utilisation s'authentifier . . . . .	33
3.2.3	Analyse du cas d'utilisation « Effectuer réclamation » . . . . .	34
3.2.4	Analyse du cas d'utilisation « Suivre réclamation » . . . . .	36
3.2.5	Analyse du cas d'utilisation « gérer réclamation » . . . . .	37
3.2.6	Analyse du cas d'utilisation « gérer Agent » . . . . .	39
3.2.7	Analyse du cas d'utilisation « gérer tâches » . . . . .	39
3.2.8	Analyse du cas d'utilisation « Valider tâche » . . . . .	40
3.3	Conclusion . . . . .	41
<b>4</b>	<b>Conception</b>	<b>43</b>
4.1	Introduction . . . . .	43
4.2	Conception architecturale . . . . .	43
4.3	Conception des cas d'utilisation . . . . .	44
4.3.1	Conception du cas d'utilisation «S'authentifier » . . . . .	44
4.3.2	Conception du cas d'utilisation « effectuer réclamation » . . . . .	46
4.3.3	Conception du cas d'utilisation «suivre réclamation» . . . . .	48
4.3.4	Conception du cas d'utilisation «gérer réclamation» . . . . .	50
4.3.5	Conception du cas d'utilisation «Gérer agent» . . . . .	50
4.3.6	Conception du cas d'utilisation «Gérer tâche» . . . . .	51
4.4	Conception des classes . . . . .	52
4.4.1	Diagramme classes entités . . . . .	52
4.4.2	Schéma de la base de données . . . . .	54
4.5	Conclusion . . . . .	55
<b>5</b>	<b>Réalisation</b>	<b>57</b>
5.1	Introduction . . . . .	57
5.2	Environnement de travail . . . . .	57
5.2.1	Environnement matériel . . . . .	57
5.2.2	Environnement logiciel . . . . .	58
5.3	Scénario d'exécution . . . . .	63
5.3.1	Scénario d'exécution de la partie web . . . . .	64
5.3.2	Scénario d'exécution de la partie mobile . . . . .	66
5.4	Conclusion . . . . .	69



*TABLE DES MATIÈRES*

ix

**Conclusion générale** 71

**Bibliographie** 73

**Bibliographie** 73

**Webographie** 75

**Webographie** 75



# Liste des tableaux

2.1	table d'affectation des priorités aux cas d'utilisation . . . . .	19
2.2	Description textuelle du cas d'utilisation « S'inscrire » . . . . .	20
2.3	Description textuelle du cas d'utilisation « S'authentifier » . . .	21
2.4	Description textuelle du cas d'utilisation « Effectuer réclamation »	22
2.5	Description textuelle du cas d'utilisation «Suivre réclamation. »	23
2.6	Description textuelle du cas d'utilisation « consulter réclamation»	24
2.7	Description textuelle du cas d'utilisation « Accepter réclamation »	24
2.8	Description textuelle du cas d'utilisation « Supprimer réclamation » . . . . .	25
2.9	Description textuelle du cas d'utilisation «Ajouter agent » . . .	26
2.10	Description textuelle du cas d'utilisation «Affecter tâche». . .	27
2.11	Description textuelle du cas d'utilisation «Valider tâche » . . .	28
2.12	Description textuelle du cas d'utilisation « Modifier ses informa- tions » . . . . .	29



# Table des figures

1.1	Logo de TRITUX[Ref.w5] . . . . .	4
1.2	objectif de l'application . . . . .	5
1.3	Illustration d'espace de la réclamation dans le site de lacommune Marsa[Ref.w2] . . . . .	6
1.4	Illustration du formulaire de dépôt de réclamation dans le site de lacommune Marsa [Ref.w2] . . . . .	7
1.5	Illustration du formulaire de dépôt de réclamation dans le site de lacommune Gabes [Ref.w4] . . . . .	7
1.6	Illustration du formulaire de dépôt de réclamation dans le site de la commune Sfax[Ref.w3] . . . . .	8
1.7	Illustration d'espace réclamation dans le site commune sfax[Ref.w3] . . . . .	9
1.8	Diagramme de gaint . . . . .	11
2.1	Diagramme de cas d'utilisation . . . . .	18
3.1	Traçabilité entre le Model CU et le Model Analyse du cas d'utilisation «S'inscrire» . . . . .	32
3.2	Diagramme de classes d'analyse du cas d'utilisation « S'inscrire » . . . . .	32
3.3	Diagramme de collaboration relatif au cas d'utilisation « S'inscrire » . . . . .	33
3.4	Traçabilité entre le Model CU et le Model Analyse du cas d'utilisation «S'authentifier» . . . . .	33
3.5	Diagramme de classes d'analyse du cas d'utilisation « S'authentifier » . . . . .	34
3.6	Diagramme de collaboration relatif au cas d'utilisation « S'authentifier » . . . . .	34
3.7	Traçabilité entre le Model CU et le Model Analyse du cas d'utilisation «effectuer réclamation» . . . . .	35
3.8	Diagramme de classes d'analyse du cas d'utilisation «effectuer réclamation » . . . . .	35

3.9	Diagramme de collaboration relatif au cas d'utilisation « effectuer réclamation » . . . . .	36
3.10	Diagramme de collaboration relatif au cas d'utilisation « suivre réclamation » . . . . .	36
3.11	Diagramme de collaboration relatif au cas d'utilisation « Accepter réclamation » . . . . .	37
3.12	Diagramme de collaboration relatif au cas d'utilisation « Supprimer réclamation » . . . . .	38
3.13	Diagramme de collaboration relatif au cas d'utilisation « Consulter réclamation » . . . . .	38
3.14	Diagramme de collaboration relatif au cas d'utilisation « Ajouter agent » . . . . .	39
3.15	Traçabilité entre le modèle du cas d'utilisation et le modèle d'analyse . . . . .	39
3.16	Diagramme de classes d'analyse relatif au cas d'utilisation « Affecter tâche » . . . . .	40
3.17	Diagramme de collaboration relatif au cas d'utilisation « Affecter tâche » . . . . .	40
3.18	Diagramme de collaboration relatif au cas d'utilisation « Valider tâche » . . . . .	41
4.1	Illustration de l'architecture MVC [Ref.w6] . . . . .	44
4.2	Traçabilité entre le modèle d'analyse et le modèle de conception pour le cas d'utilisation « S'authentifier ». . . . .	45
4.3	Diagramme de classes de conception relatif au cas d'utilisation « S'authentifier » . . . . .	45
4.4	Diagramme de séquence relatif au cas d'utilisation « S'authentifier » . . . . .	46
4.5	Diagramme de classes de conception relatif au cas d'utilisation « effectuer réclamation ». . . . .	47
4.6	Diagramme de séquence relatif au cas d'utilisation « Effectuer réclamation » . . . . .	48
4.7	Diagramme de classes de conception relatif au cas d'utilisation « suivre réclamation ». . . . .	49
4.8	Diagramme de séquence relatif au cas d'utilisation « suivre réclamation ». . . . .	49
4.9	Diagramme de séquence relatif au cas d'utilisation « Consulter réclamation » . . . . .	50
4.10	Diagramme de séquence relatif au cas d'utilisation « Ajouter agent » . . . . .	51
4.11	Diagramme de séquence relatif au cas d'utilisation « Affecter tâche » . . . . .	52
4.12	Diagramme de classes . . . . .	53

5.1	Logo de Ionic . . . . .	58
5.2	Logo Angular[réf.w7] . . . . .	59
5.3	Logo Nodejs[Ref.w9] . . . . .	60
5.4	Logo NPM [Ref.w8] . . . . .	60
5.5	Logo Xamp . . . . .	61
5.6	Logo Mysql [Ref.w11] . . . . .	61
5.7	Logo Uml [Ref.w12] . . . . .	62
5.8	Logo PHP . . . . .	62
5.9	Logo Visual Studio Code[Ref.w14] . . . . .	63
5.10	Logo Overleaf . . . . .	63
5.11	Interface inscription Admin . . . . .	64
5.12	Interface authentication . . . . .	64
5.13	Interface réclamation en cours . . . . .	65
5.14	Interface affecter tache . . . . .	65
5.15	Interface ajouter agent . . . . .	66
5.16	Interface authentication . . . . .	66
5.17	Interface de l'espace citoyen . . . . .	67
5.18	Interface effectuer réclamaion automatique . . . . .	67
5.20	Interface valider tache . . . . .	68
5.19	Interface de l'espace agents . . . . .	68





# Introduction générale

Ces dernières décennies ont connu une révolution innovante sans précédent dans les technologies de l'information et de la communication. Ces technologies sont de plus en plus orientées vers le confort et la réponse aux nouveaux besoins de l'être humain, surtout dans des délais raisonnables et à distance. De nouveaux concepts ont ainsi émergé tels que la portabilité, la mobilité et les flux continus d'informations. Parmi les principales innovations, dont l'homme ne peut plus se séparer, nous citons l'apparition d'appareils de haute technologie tels que les Smartphones et les tablettes de plus en plus dotés d'applications pratiques.

Dans le cadre de l'utilité de l'informatique qui doit être au service de l'homme, ce projet de fin d'étude s'intéresse à concevoir et développer une application web/mobile de réclamations municipales en ligne. Le choix du sujet répond à un réel besoin ressenti par les citoyens, notamment suite aux dernières élections municipales. En effet, la majorité des électeurs a voté pour des représentants municipaux qu'ils estiment capables d'améliorer la qualité de leur vie dans leur ville. Ainsi, avec l'adoption de la notion de gouvernance locale dans la nouvelle constitution, les municipalités sont dotées d'un plus grand pouvoir et doivent être de plus en plus performantes et réactives par rapport aux exigences des citoyens. Cependant, nous avons remarqué que la relation entre la municipalité et le citoyen présentent des limites comme par exemple : l'absence d'applications mobiles spécifiques aux réclamations, la plupart du temps on ne remarque qu'une interface sur le site web de la commune, pas de détermination automatique de la position exacte de la réclamation, pas de communication entre l'administrateur et les citoyens, etc.

Pour tout cela nous avons pensé à concevoir notre application intitulée « E-Réclamation Municipale ». Nos objectifs à travers cette application consistent

à aider à répondre aux besoins des citoyens en garantissant la performance, la transparence, et une meilleure communication de l'administrateur avec ses agents. Ce type de services ne peut que renforcer la relation entre le citoyen et la commune.

Notre manuscrit se compose de cinq chapitres.

Le premier chapitre a pour but en premier lieu de présenter la société TRITUX qui nous a accueilli durant ce stage, En second lieu, il s'agit de présenter l'idée générale de notre projet, les différents objectifs à atteindre accompagnés par les solutions proposées et pour finir par l'élaboration de l'étude de l'existant.

Notre deuxième chapitre intitulé analyse des besoins, nous avons essayé de définir les acteurs de l'application à développer, les besoins fonctionnels, les besoins non fonctionnels et les besoins techniques de notre projet. Il vise aussi à identifier les principaux cas d'utilisation associés aux différents acteurs et de les projeter dans un diagramme de cas d'utilisation structuré.

Le troisième chapitre présente l'analyse qui consiste à procéder à une répartition du comportement du système entre les différentes classes d'analyses. Il s'agit, ainsi, de déterminer les différentes classes d'analyse qui participent à la réalisation de chaque cas d'utilisation et de modéliser les messages échangés pour chaque scénario de réalisation.

Le quatrième chapitre présente une conception globale de notre solution suivie d'une explication plus détaillée de ses modules à travers des diagrammes de classes et des diagrammes de séquence.

Le dernier chapitre « Réalisation » contient le dernier volet de ce rapport ayant pour objectif la description de l'aspect implémentation et développement de l'application déjà conçue et modélisée dans les chapitres précédents. Dans ce chapitre, nous présenterons l'environnement matériel, logiciel et les différents outils de développement utilisés.

# Etude Préalable

## 1.1 Introduction

Ce chapitre a pour but de se placer dans le contexte du stage. Il s'agit en premier lieu de présenter la société TRITUX qui nous a accueilli durant ce stage, En second lieu, il s'agit de présenter l'idée générale de notre projet, les différents objectifs à atteindre accompagnés par les solutions proposées et nous finissons par l'élaboration de l'étude de l'existant.

## 1.2 Présentation de l'organisme d'accueil

Notre Stage est réalisé au sein de l'entreprise « TRITUX » qui nous a proposé de développer une application Web/mobile de réclamation municipale en ligne. TRITUX est une société internationale spécialisée dans l'ingénierie informatique, le conseil en informatique et l'externalisation. Avec plus de 13 ans d'expérience, TRITUX est l'un des principaux fournisseurs de solutions informatiques de télécommunication.

Grâce à une infrastructure solide et à de multiples certifications dans les nouvelles technologies, cette entreprise fournit des services d'ingénierie et de conseil en innovation et en agilité commerciale. L'entreprise percevaient l'environnement commercial et la rentabilité de leurs clients en proposant des solutions adaptées et personnalisées, avec un ensemble complet de services,

de la conception à la mise en œuvre. TRITUX est un partenaire stratégique dans : (IoT, Roaming, Solution informatique, Secteurs d'activité Télécommunication, Bancaire, Industriel, Secteur public).



FIGURE 1.1 – Logo de TRITUX[Ref.w5]

### 1.3 Cadre du travail

Ce stage a été effectué dans le cadre d'un projet de fin d'études pour l'obtention du diplôme « Licence fondamentale en science d'informatique et multimédia » de l'institut supérieur d'informatique et de multimédia de Sfax. L'estage est réalisé au sein de l'entreprise « TRITUX » qui nous a proposés de concevoir une Application web /mobile de réclamation municipale en ligne.

### 1.4 Présentation du projet

Notre projet de fin d'études consiste à développer une application web/mobile de réclamations municipales en ligne qui permet aux citoyens d'effectuer des réclamations municipales à la commune automatiquement avec géo localisation ou manuellement et de les suivre. L'administrateur reçoit les réclamations qui correspondent à sa commune et les gère (consulter, accepter, supprimer). De plus l'administrateur gère les tâches (consulter, affecter) aux agents pour les réclamations acceptées, aussi il gère les agents (ajouter, consulter, supprimer). L'agent reçoit les tâches affectées et les consulte, après la fin des travaux d'une certaine tâche il la valide avec un justificatif (photo).

Enfin l'administrateur peut voir les changements au niveau de l'état de la tâche, le même pour le citoyen qui peut voir le justificatif de fin de tâches depuis son

espace.

Comme nous le présente la figure 1.2, l'idée de ce projet est de réaliser une application composée d'une partie mobile qui satisfait les besoins des citoyens et lui fait gagner du temps afin ne pas se présenter à la commune pour déposer une réclamation. Par conséquent, l'objectif consiste à développer une application conviviale et moderne qui propose le dépôt d'une réclamation en ligne et la suivre jusqu'à son état final.



FIGURE 1.2 – objectif de l'application

## 1.5 Etude de l'existant

La phase d'étude de l'existant est très importante pour transformer une idée à une réelle réalisation. Cette phase nous permet de mieux comprendre notre mission et d'enrichir les fonctionnalités à réaliser. Dans cette section, nous allons étudier les différentes possibilités de dépôt d'une réclamation municipale en ligne pour bien prendre les décisions techniques. L'objectif de cette section aussi est de tenir compte des aspects présentés et de pallier aux insuffisances et difficultés déjà rencontrées. Ensuite, on déduira les avantages et les inconvénients des solutions existantes pour dégager enfin solutions à proposer.

### 1.5.1 Description des solutions existantes

#### □ Commune de Marsa :

La commune de la Marsa présente une interface de disposition des réclama-

tions municipale en ligne pour les citoyens sur le site web de la commune. Le formulaire d'une réclamation est comme suit :



FIGURE 1.3 – Illustration d'espace de la réclamation dans le site de lacommune Marsa[Ref.w2]

□ **Commune de Gabes :**

La commune de Gabes met au citoyens un espace dans le site web de la commune ou ils peuvent passer des réclamations avec le formulaire suivant :

Recherche...

Nom et Prénom \*

Email \*

N° Cell \*

Services  
Type de réclamation

Adresse \*

Le sujet de réclamation \*

Envoyer

Accès rapide

- e-Reclamations
- e-Documents
- Appel d'offre
- Sondages

Aujourd'hui 22°  
Peu nuageux  
Vent: 25 km/h  
Est. Sud Est

Samedi 20°  
Peu nuageux  
Vent: 25 km/h  
Est.

FIGURE 1.4 – Illustration du formulaire de dépôt de réclamation dans le site de lacommune Marsa [Ref.w2]

تشكايات

الاسم واللقب \*

البريد الإلكتروني\*

رقم بطاقة تعريف الوطنية\*

نوعية الشكاوى

العنوان\*

موضوع الشكاوى\*

الصورة

Choisir un fichier

طلبات العروض

الشكاوى

رخص البناء

إجراءات

خدمات الترميمات

الخطط الموضوعة

فضاء المواطن Espace Citoyen

لمتابعة مشاكلكم زر موقع فضاء المواطن

FIGURE 1.5 – Illustration du formulaire de dépôt de réclamation dans le site de lacommune Gabes [Ref.w4]

### □ Commune de Sfax :

Dans le site web de la commune de Sfax il y a un formulaire de disposition d'une réclamation municipale en ligne dans l'espace citoyen. Avec détermination de localisation de la réclamation aussi avec justification média (image).

#### Soumettre un Nouveau Rapport

**Titre du Rapport \***

**Description \***  
 Les balises HTML autorisées sont : b, p, img, br, b, u, strong, em, i.  
 Les iframes ne sont autorisés qu'à partir de www.youtube.com/embed/, player.vimeo.com/video/,  
 www.soundcloud.com/player.

**Date & Heure:** Aujourd'hui à 04:05 pm (Africa/Tunis) + [Changer Date](#)

**Catégories \***

- ☐ Défense
- ☐ Lutte
- ☐ Développement
- ☐ Environnement
- ☐ Santé
- ☐ Culture
- ☐ Éducation
- ☐ Économie
- ☐ Société
- ☐ Autres

**Informations Facultatives**

**Prénom**

**Nom de famille**

**E-mail**

**Map**

**Actions**

[SUPPRIMER LA DERNIÈRE ENTRÉE](#) [SUPPRIMER LA SÉLECTION](#)

[EFFACER LA CARTE](#)

[Trouver l'emplacement](#)

Rechercher l'emplacement en utilisant un nom de lieu ou les coordonnées de latitude et de longitude (format: 38,19, 85,61), ou cliquer sur la carte pour trouver l'emplacement.

**Nom de votre lieu \***  
 Exemple: Avenue du Martyr, Bangui

**Lien de sources de presse**

**Lien vidéo externe**

**Transférer les Photos**

[Choisir un fichier](#) [Aucun fichier choisi](#)

[Soumettre](#)

FIGURE 1.6 – Illustration du formulaire de dépôt de réclamation dans le site de la commune Sfax[Ref.w3]



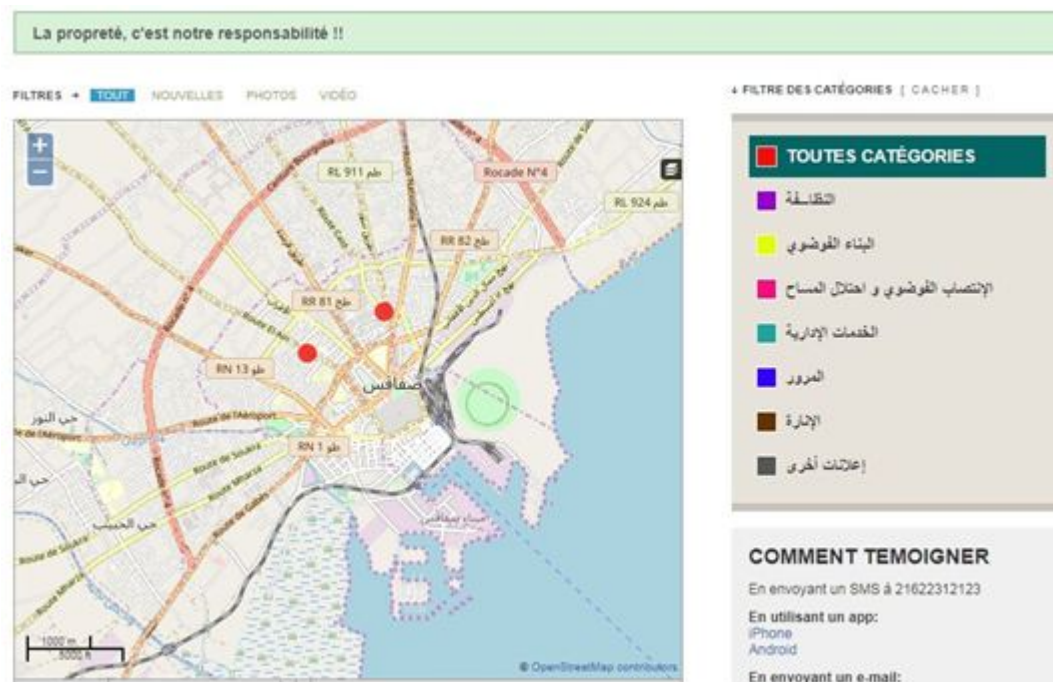


FIGURE 1.7 – Illustration d'espace réclamation dans le site commune sfax[Ref.w3]

### 1.5.2 Critique de l'existant

Les solutions existantes fournissent plusieurs avantages :

- Organisation des réclamations avec des catégories.
- Quelques communes offrent un service de géo localisation .
- Ergonomie claire et facile à comprendre.

Cependant, les solutions existantes présentent des limites :

- L'absence d'une application mobile spécifique pour la réclamation municipale en ligne.
- Les applications existantes ne présentent pas d'espace membres : les sites web ne présentent ni un espace pour les citoyens ni pour les agents.
- L'envoi de réclamation n'est pas toujours disponible sur les sites web de toutes les communes.
- Dans certains cas comme la commune de Gabès, il n'y a qu'un simple formulaire intégré dans le site web de la commune, le suivi de la réclamation ne s'effectue pas ni par les agents ni par les citoyens.
- L'absence du Justificatif (image) de la

réclamation.

- Pas de détermination automatique de la position exacte de la réclamation .
- Pas de communication entre l'administrateur et les citoyens.

### 1.5.3 Solution proposée

Pour répondre aux défaillances des systèmes disponibles étudiés précédemment, on souhaite développer dans le cadre de notre projet de fin d'étude un outil efficace pour améliorer ce système de réclamation municipale. En particulier, les solutions que nous envisageons dans notre projet sont les suivantes :

- Développer une application mobile pour la soumission et le suivi de la réclamation municipale en ligne.
- Ajouter un justificatif pour la réclamation( comme photo).
- Calculer automatiquement la position exacte de la réclamation où la commune doit s'intervenir .
- Le citoyen peut suivre la réclamation qu'il a envoyée de la soumission jusqu'à la fin des travaux.
- L'agent reçoit ses tâches en ligne.
- L'agent valide ses tâches .
- L'application met en communication le citoyen, l'administrateur et les agents de la commune.

## 1.6 Planning prévisionnel

Après cette étude profonde sur les différentes méthodologies existantes, nous allons tracer et suivre un plan de travail minutieux qui va nous permettre de mieux gérer toutes les phases de réalisation de l'application, commençant par les besoins fonctionnels attendus du système jusqu'à la conception et le codage final.

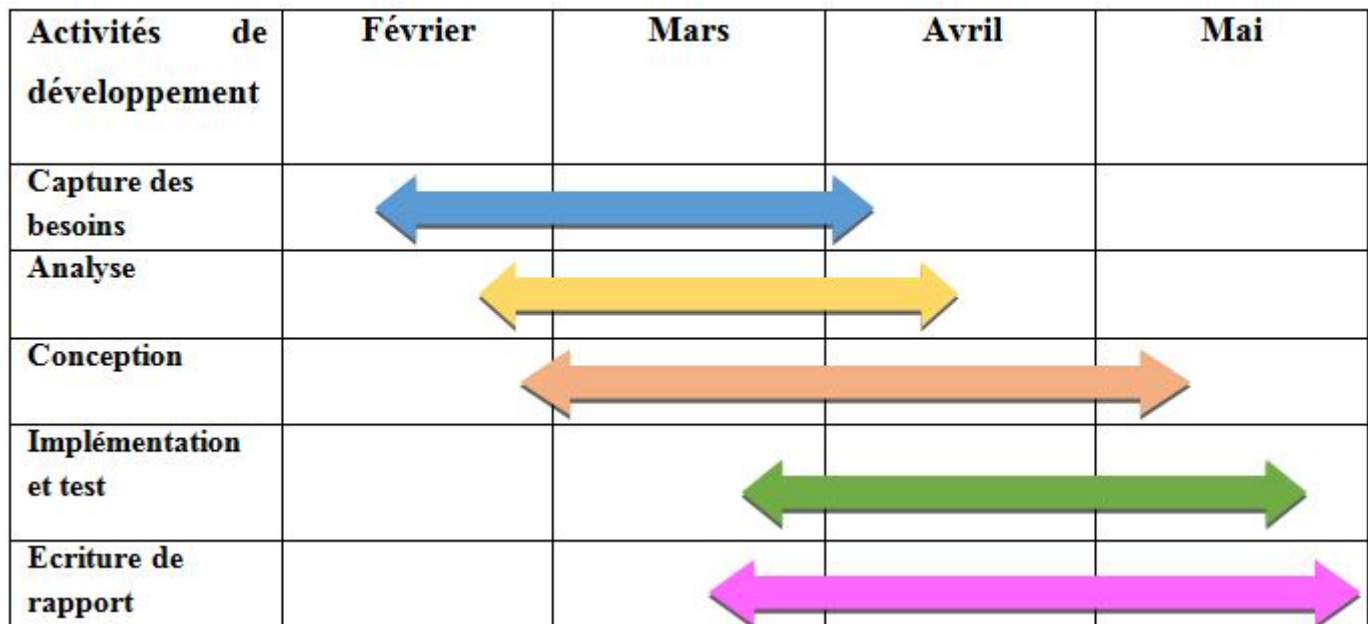


FIGURE 1.8 – Diagramme de gaint

## 1.7 Conclusion

Ce chapitre nous a permis d'exposer le contexte de projet de fin d'étude et de représenter la problématique générale. Ensuite, grâce à l'étude de l'existant, nous avons pu étudier quelques systèmes de réclamation municipales en ligne qui existent déjà et nous avons dégagé quelques lacunes et insuffisances afin de proposer une meilleure solution.



## Chapitre 2

# Analyse et spécification des besoins

## 2.1 Introduction

L'objectif de ce chapitre est de définir les acteurs de l'application à développer, les besoins fonctionnels et les besoins non fonctionnels de notre projet. Il vise aussi à identifier les principaux cas d'utilisation associés aux différents acteurs et de les projeter dans un diagramme de cas d'utilisation structuré.

## 2.2 Identification des acteurs et des cas d'utilisation

### 2.2.1 Identification des acteurs

Nous allons maintenant énumérer les acteurs susceptibles d'interagir avec notre application. En effet, les acteurs d'un système sont les entités externes à ce système qui interagissent directement avec lui, les acteurs qui interagissent avec notre application sont :

**Acteur :** le citoyen .

**Rôle :** c'est un utilisateur qui possède un compte à travers lequel il peut faire une réclamation à la municipalité et il peut suivre l'état de sa réclamation.

**Acteur :** l'administrateur

**Rôle :** C'est l'acteur qui s'occupe de la gestion des réclamations reçues et qui fait la gestion des agents ,et aussieur affecte tâches .

**Acteur :** l'agent.

**Rôle :** c'est l'acteur qui reçoit des tâches de son administrateur pour effectuer les travaux nécessaires , et aussi il les valide.

### 2.2.2 Identification des cas d'utilisation

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Les cas d'utilisations servent à structurer les besoins des utilisateurs et les objectifs correspondants du système. Les cas d'utilisation de notre application sont :

#### — Cas d'utilisation pour le citoyen

- **S'inscrire :**Le citoyen remplit un formulaire et saisit ses informations pour avoir un compte personnel dans l'application.
- **S'authentifier :** : Le citoyen doit avoir un cin et un mot de passe pour accéder à l'application.
- **Effectuer réclamation :** : Cette tache permet au citoyen de passer une réclamation municipale à travers un formulaire qu'il doit remplir attentivement.
- **Suivre réclamation :** : Cette tache permet au citoyen de suivre l'état d'avancement de la réclamation déjà envoyé.
- **Modifier ces informations :** Le citoyen utilise un formulaire pour modifier son email ou bien mot de passe par exemple.

#### — Cas d'utilisation pour l'admin

- **S'inscrire :** :Cette tache d'inscription est spécifique pour l'administrateur qui fait une inscription pour avoir un compte administrateur.

- **S'authentifier** : L'administrateur doit avoir un cin et un mot de passe pour accéder à son compte.
- **Gérer réclamation** : Cette tâche permet de gérer les réclamations reçus (accepter, supprimer, consulter).
- **Gérer agent** : : Cette tâche permet de gérer les agents (ajouter, supprimer, consulter) les agents.
- **gérer tâche** : Permet de consulter ou bien d'affecter une tâche à un agent pour réaliser les fins travaux nécessaires sur terrain.
- **Modifier ces informations** : L'administrateur utilise un formulaire pour modifier son email ou bien mot de passe par exemple..

#### — Cas d'utilisation pour l'agent

- **S'authentifier** : L'agent doit avoir un cin et un mot de passe pour accéder à son compte dans l'application.
- **Consulter tâche** : Permet de consulter les tâches affectées par son administrateur.
- **Valider tâche** : Permet à l'agent de valider ses tâches effectuées.
- **Modifier ces informations** : : L'agent utilise un formulaire pour modifier son email ou bien mot de passe par exemple.

## 2.3 Les besoins

### 2.3.1 Les besoins fonctionnels

Les besoins fonctionnels représentent les attentes de chaque acteur de l'application à développer.

L'application à développer doit être composée d'une partie web et une autre partie mobile et doit satisfaire les besoins fonctionnels suivants .

□ **Partie Web** :

destinée pour l'administrateur

- **Inscription** : L'administrateur doit s'inscrire dans l'application.
- **Authentification** : L'admin doit s'authentifier pour accéder à l'application.
- **Gérer réclamation** : l'administrateur gère les réclamations reçues (consulter, accepter, supprimer).
- **Gérer les agents** : L'administrateur gère ses agents (ajouter, supprimer, consulter).
- **Gérer tâche** : l'administrateur gère les tâches (consulter, affecter)
- **Modifier ses informations** : L'administrateur utilise un formulaire pour modifier son email ou bien mot de passe par exemple.

□ **Partie Mobile :**

Destinée pour le citoyen et pour l'agent

● **Besoins pour le citoyen :**

- **Inscription** : Le citoyen doit s'inscrire dans l'application.
- **Authentification** : Le citoyen doit s'authentifier pour effectuer ou suivre ses réclamations.
- **Effectuer réclamation** : Le citoyen remplit un formulaire pour effectuer une réclamation municipale.
- **Suivre réclamation** : Le citoyen peut suivre ses réclamations.
- **Modifier ses informations** : Le citoyen utilise un formulaire pour modifier son email ou bien mot de passe par exemple.

● **Besoins pour l'agent.**

- **Authentification** : L'agent doit s'authentifier pour consulter ou suivre ses tâches.
- **Consulter tâche** : L'agent peut suivre ses tâches affectées par son administrateur.
- **Valider tâche** : Après la finalisation des travaux sur terrain, l'agent va valider la tâche par l'envoi de l'état final de la réclamation.
- **Modifier ses informations** : L'agent utilise un formulaire pour modifier ses informations.



### 2.3.2 Les besoins non fonctionnels

Ce sont les besoins qui permettraient d'améliorer la qualité des services de l'application comme la convivialité et l'ergonomie des interfaces et l'amélioration du temps de réponse. Parmi ces besoins on cite :

**La sécurité :** Besoins d'établissement de la connexion, il faut avoir une interface d'authentification qui permet à chaque utilisateur de se connecter pour consulter son profil.

**Ergonomies :** L'application doit respecter les standards ergonomiques tels que les densités d'éléments sur l'écran, les couleurs de l'interface. Elle doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur, simple et significative qui facilite leur usage par les utilisateurs.

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

**La portabilité :** L'application doit être portable pour tous les terminaux mobiles et web.

**Efficacité :** le système doit fournir les bons résultats qui répondent exactement aux besoins de l'application.

## 2.4 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases).



FIGURE 2.1 – Diagramme de cas d'utilisation

### 2.4.1 Affectation des priorités aux cas d'utilisation

Dans le tableau ci-dessous, nous présentons les priorités que nous avons affectées aux différents cas d'utilisation :

Cas d'utilisation	Acteurs	Priorité
<b>S'inscrire</b>	-Admin -Citoyen	1
<b>S'authentifier</b>	-Admin - Citoyen -Agent	1
<b>Effectuer réclamations</b>	- Citoyen	1
<b>Suivre réclamations</b>	- Citoyen	2
<b>Modifier ces information</b>	-Admin -Citoyen -Agent	3
<b>Gérer réclamations</b>	- Admin	1
<b>Gérer agents</b>	- Admin	2
<b>gérer taches</b>	- Admin	1
<b>consulter taches</b>	- Agent	1
<b>valider taches</b>	- Agent	1

TABLEAU 2.1 – table d'affectation des priorités aux cas d'utilisation

### 2.4.2 Raffinement des cas d'utilisation

Dans cette partie, nous allons décrire textuellement les pré-conditions, les post-conditions, le scénario nominal et le scénario alternatif de chaque cas d'utilisation.

#### Raffinement du cas d'utilisation « S'inscrire »

Il s'agit de faire une inscription dans l'application pour avoir les droits de bénéficier des différentes fonctionnalités de l'application.

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

<b>Cas d'utilisation</b>	<b>S'inscrire</b>
<b>Acteur(s)</b>	- Admin - Citoyen
<b>Objectif</b>	L'acteur fait l'inscription a l'application
<b>Pré-condition</b>	être un responsable d'une commune
<b>Post-condition</b>	L'acteur est inscrit
<b>Scénario principale</b>	1) L'acteur demande le formulaire d'inscription. 2) Le système affiche le formulaire d'inscription. 3) L'acteur remplir le formulaire. 4) Le système vérifie les données saisie. 5) L'acteur valide le formulaire d'inscription. 6)l'acteur est redirigé vers la section qui lui convient.
<b>Scénario alternatif</b>	<b>A/Champs obligatoire non valide ou vide.</b> Cet enchainement démarre au point 4. 5. Le système affiche un message d'erreur. scénario reprend au point 2. <b>B/Inscription déjà existe.</b> Cet enchainement démarre au point 4. 5.Le système affiche un message d'erreur. scénario reprend au point 2.

TABLEAU 2.2 – Description textuelle du cas d'utilisation « S'inscrire »

**Raffinement du cas d'utilisation « S'authentifier »****Description textuelle du cas d'utilisation «S'authentifier».**

Cas d'utilisation	S'authentifier
Acteur(s)	<ul style="list-style-type: none"> <li>- Admin</li> <li>- Agent</li> <li>- Citoyen</li> </ul>
Objectif	L'authentification
Pré-condition	L'acteur est inscrit
Post-condition	Authentification réussie
Scénario principal	1) L'acteur demande le formulaire d'authentification. 2) Le système affiche la page d'authentification. 3) L'acteur saisit son login et son mot de passe. 4) Le système vérifie les données saisies.
Scénario alternatif	<b>A/ Erreur d'authentification</b> : login ou mot de passe non valide Cet enchainement démarre au point 4. 5. Le système affiche un message d'erreur. Le scénario reprend au point 2. <b>B/Champs obligatoire non valide ou vide.</b> Cet enchainement démarre au point 3. 5. Le système affiche un message d'erreur. Le scénario reprend au point 2.

TABLEAU 2.3 – Description textuelle du cas d'utilisation « S'authentifier »

### Raffinement du cas d'utilisation « effectuer réclamation »

Il s'agit de remplir un formulaire de soumission d'une réclamation municipale en ligne.

### Description textuelle du cas d'utilisation « effectuer réclamation »

<b>Cas d'utilisation</b>	<b>Effectuer réclamation</b>
<b>Acteur</b>	- Citoyen
<b>Objectif</b>	Le citoyen veut passer une réclamation.
<b>Pré-condition</b>	Le citoyen doit être authentifié.
<b>Post-condition</b>	Les réclamations doivent être enregistrées dans une base de données.
<b>Scénario principal</b>	1) Le citoyen choisit la rubrique d'effectuer réclamation. 2) Le système affiche le contenu de la rubrique. 3) Le citoyen choisit entre localisation manuelle ou automatique. 4) Le système affiche le formulaire de la réclamation. 5) Le citoyen remplit le formulaire. 6) Le système localise la réclamation si le citoyen a choisi localisation automatique. 7) Le citoyen envoie les informations remplies dans le formulaire. 8) Le système vérifie les données. 9) Le système affiche un message d'envoi / échec de la réclamation.
<b>Scénario alternatif</b>	<b>A/Champs obligatoire non valide ou vide.</b> Cet enchainement démarre au point 5. 4. Le système affiche un message d'erreur. Le scénario reprend au point 4. <b>B/Problème de déterminer la localisation.</b> Cet enchainement démarre au point 6. 5. Le système affiche un message d'erreur. Le scénario reprend au point 4.

TABLEAU 2.4 – Description textuelle du cas d'utilisation « Effectuer réclamation »

### Raffinement du cas d'utilisation « suivre réclamation ».

#### Description textuelle du cas d'utilisation « suivre réclamation ».

<b>Cas d'utilisation</b>	<b>Suivre réclamation</b>
<b>Acteur</b>	-Citoyen
<b>Objectif</b>	Le citoyen suit ses réclamation.
<b>Pré-condition</b>	Le citoyen doit être authentifié
<b>Post-condition</b>	Les information des réclamations effectuées par le citoyen doivent être affichées.
<b>Scénario principal</b>	1) Le citoyen choisit la rubrique suivre réclamation. 2) Le système affiche les informations des réclamations affectées par le citoyen.
<b>Scénario alternatif</b>	- En cas d'échec de connexion à la BD, le système affiche un message d'erreur.

TABLEAU 2.5 – Description textuelle du cas d'utilisation «Suivre réclamation. »

**Raffinement du cas d'utilisation « Gérer réclamation ».**

Il s'agit de gérer les réclamations reçues à la commune

- Consulter réclamation
- Accepter réclamation
- Supprimer réclamation

**Raffinement du cas d'utilisation «consulter réclamation ».**

**Description textuelle du cas d'utilisation « Consulter réclamations »**

<b>Cas d'utilisation</b>	<b>Consulter réclamation</b>
<b>Acteur</b>	-Administrateur
<b>Objectif</b>	L'administrateur consulter les réclamation.
<b>Pré-condition</b>	L'administrateur doit être authentifié.
<b>Post-condition</b>	Affichage de les réclamations
<b>Scénario principal</b>	1) L'administrateur choisit la rubrique consulter réclamation. 2) Le système affiche les réclamations reçues a la commune
<b>Scénario alternatif</b>	- En cas d'échec de connexion à la BD, le système affiche un message d'erreur.

TABLEAU 2.6 – Description textuelle du cas d'utilisation « consulter réclamation»

**Raffinement du cas d'utilisation « accepter réclamation».****Description textuelle du cas d'utilisation « accepter réclamations »**

<b>Cas d'utilisation</b>	<b>Accepter réclamation</b>
<b>Acteur</b>	-Administrateur
<b>Objectif</b>	L'administrateur accepter le réclamation.
<b>Pré-condition</b>	L'administrateur doit être authentifié.
<b>Post-condition</b>	La réclamation doit être accepter.
<b>Scénario principal</b>	1) Le administrateur choisit la rubrique consulter réclamation. 2) Le système affiche les réclamations reçues a la commune 3) L'administrateur sélectionne une réclamation et clique-btn- Accepter. 4)Le système enregistre le réclamation 5) Le système affiche un message d'acceptation la réclamation à l'administrateur .
<b>Scénario alternatif</b>	- En cas d'échec de connexion à la BD, le système affiche un message d'erreur.

TABLEAU 2.7 – Description textuelle du cas d'utilisation « Accepter réclamation »



**Raffinement du cas d'utilisation « Supprimer réclamation »****Description textuelle du cas d'utilisation « Supprimer réclamation »**

<b>Cas d'utilisation</b>	<b>Supprimer réclamation</b>
<b>Acteur</b>	-Administrateur
<b>Objectif</b>	L'administrateur supprimer le réclamation.
<b>Pré-condition</b>	L'administrateur doit être authentifier.
<b>Post-condition</b>	La réclamation doit être supprimer.
<b>Scénario principal</b>	1) Le administrateur choisit la rubrique consulter réclamation. 2) Le système affiche les réclamations reçues a la commune 3) L'administrateur sélectionne une réclamation et clique-btn- Supprimer. 4)Le système effacer le réclamation 5) Le système affiche un message de Suppression la réclamation .
<b>Scénario alternatif</b>	- En cas d'échec de connexion à la BD, le système affiche un message d'erreur.

TABLEAU 2.8 – Description textuelle du cas d'utilisation « Supprimer réclamation »

**Raffinement du cas d'utilisation « gérer Agents »**

L'administrateur gère la liste des agents, ce cas d'utilisation comprend les sous-cas suivants :

- Consulter Agents
- Ajouter Agents
- Supprimer Agents

**Raffinement du cas d'utilisation « Ajouter agent »****Description textuelle du cas d'utilisation « Ajouter agent ».**

<b>Cas d'utilisation</b>	<b>Ajouter Agent</b>
<b>Acteur</b>	-Administrateur
<b>Objectif</b>	L'administrateur ajoute un agent.
<b>Pré-condition</b>	L'administrateur doit être authentifié.
<b>Post-condition</b>	Un nouveau agent ajouter.
<b>Scénario principal</b>	1) Le administrateur choisit la rubrique ajouter agents. 2) Le système affiche un formulaire d'ajout un agents 3) L'administrateur remplit le formulaire 4) Le système vérifie les données 5) L'administrateur valider le formulaire 6) Le système vérifie l'existence du agent 7) Le système ajouter un nouveau agent 8)Le système affiche un message de succès d'ajouter un agent .
<b>Scénario alternatif</b>	<b>A/Champs obligatoire non valide ou vide.</b> Cet enchainement démarre au point 4. 5. Le système affiche un message d'erreur. Le scénario reprend au point 2. <b>B/ Agent déjà existe.</b> Cet enchainement démarre au point 6. Le scénario reprend au point 2.

TABLEAU 2.9 – Description textuelle du cas d'utilisation «Ajouter agent »

**Raffinement du cas d'utilisation « gérer Tâches » :**

Il s'agit de consulter et d'affecter une tâche à un agent de la commune pour se déplacer sur terrain et traiter la réclamation

- Consulter tâche.
- Affecter tâche.

**Raffinement du cas d'utilisation « Affecter Tâche» :****Description textuelle du cas d'utilisation « Affecter tâche »**

Cas d'utilisation	Affecter tâche
Acteur	- Administrateur
Objectif	L'administrateur affecter le tâche à un agent
Pré-condition	L'administrateur doit être authentifier
Post-condition	La réclamation doit être affecter.
Scénario principal	1) L'administrateur choisit la rubrique affecter tâche. 2) Le système affiche le formulaire . 3) L'administrateur remplit le formulaire et valide. 4) Le système vérifie les données. 5) Le système enregistre les informations et affiche un message de réussite.
Scénario alternatif	<b>A/Champs obligatoire non valide ou vide.</b> Cet enchainement démarre au point 3. 4. Le système affiche un message d'erreur. Le scénario reprend au point 2.

TABLEAU 2.10 – Description textuelle du cas d'utilisation «Affecter tâche».

### Raffinement du cas d'utilisation « Valider tâche »

Il s'agit de valider la tâche, lorsque l'agent termine son travail sur terrain il doit valider la terminaison de la tâche sur l'application. La validation doit être accessible pour l'administrateur et aussi pour le citoyen.

### Description textuelle du cas d'utilisation « Valider tâche ».

<b>Cas d'utilisation</b>	<b>Valider tâche</b>
<b>Acteur</b>	- Citoyen
<b>Objectif</b>	L'agent valide les tâches affectées par l'administrateur.
<b>Pré-condition</b>	- L'agent doit être authentifié. - La tâche doit être terminée.
<b>Post-condition</b>	La validation envoyée à l'administrateur.
<b>Scénario principal</b>	1) L'agent choisit la rubrique valider tâches. 2) Le système affiche un formulaire et charge la liste des tâches. 3) L'agent remplit le formulaire, sélectionne la tâche et valide 4) Le système vérifie les données. 5) Le système enregistre la validation dans la base de données. 6) Le système affiche un message de réussite de validation.
<b>Scénario alternatif</b>	<b>A/Champs obligatoire non valide ou vide.</b> Cet enchaînement démarre au point 3. 4. Le système affiche un message d'erreur. Le scénario reprend au point 2. <b>B/ Pas de sélection d'une tâche.</b> Cet enchaînement démarre au point 4. 5. Le système affiche un message d'erreur. Le scénario reprend au point 2

TABLEAU 2.11 – Description textuelle du cas d'utilisation «Valider tâche »

**Raffinement du cas d'utilisation « Modifier ses informations »**

Cas d'utilisation	Modifier ses informations
Acteur	<ul style="list-style-type: none"> <li>- Administrateur</li> <li>- Citoyen</li> <li>- Agent</li> </ul>
Objectif	Modifier les informations du profil
Pré-condition	L'acteur doit être authentifié
Post-condition	Les informations du profil sont mises à jour.
Scénario principal	1) L'acteur demande le formulaire de paramètres. 2) le système affiche le formulaire. 3) L'acteur saisie les informations. 4) Le système vérifie les données saisies. 5) L'acteur est redirigé vers la section qui lui convient.
Scénario alternatif	<b>A/Champs obligatoire non valide ou vide.</b> Cet enchainement démarre au point 3. 4. Le système affiche un message d'erreur. Le scénario reprend au point 2.

TABLEAU 2.12 – Description textuelle du cas d'utilisation « Modifier ses informations »

## 2.5 Conclusion

Dans ce chapitre, nous avons cerné les différentes fonctionnalités que notre application devra offrir ainsi que les contraintes qu'elle doit satisfaire. De plus, nous avons présenté le diagramme de cas d'utilisation structuré et la description textuelle de la plupart des cas d'utilisation identifiés. L'objectif du prochain chapitre « Analyse » est de mieux comprendre les besoins et les exigences des utilisateurs. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.



# Chapitre 3

## Analyse

### 3.1 Introduction

L'objectif de l'activité d'analyse est de procéder à une répartition du comportement du système entre les différentes classes d'analyse. Il s'agit, ainsi, de déterminer les différentes classes d'analyse qui participent à la réalisation de chaque cas d'utilisation et de modéliser les messages échangés pour chaque scénario de réalisation.

### 3.2 Analyse des cas d'utilisation

#### 3.2.1 Analyse du cas d'utilisation s'inscrire

- Traçabilité entre le modèle du cas d'utilisation et le modèle d'analyse

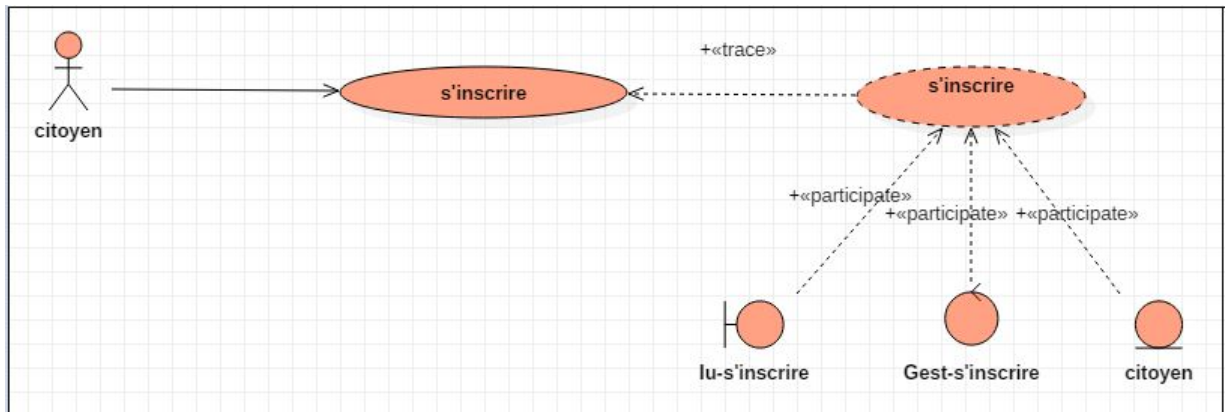


FIGURE 3.1 – Traçabilité entre le Model CU et le Model Analyse du cas d'utilisation «S'inscrire»

#### - Diagramme de classes d'analyse relatif au cas d'utilisation « S'inscrire »

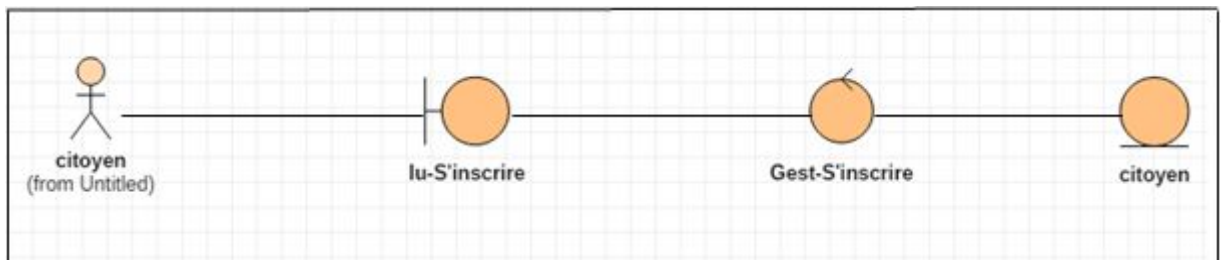


FIGURE 3.2 – Diagramme de classes d'analyse du cas d'utilisation « S'inscrire »

#### Diagramme de collaboration relatif au cas d'utilisation « S'inscrire »



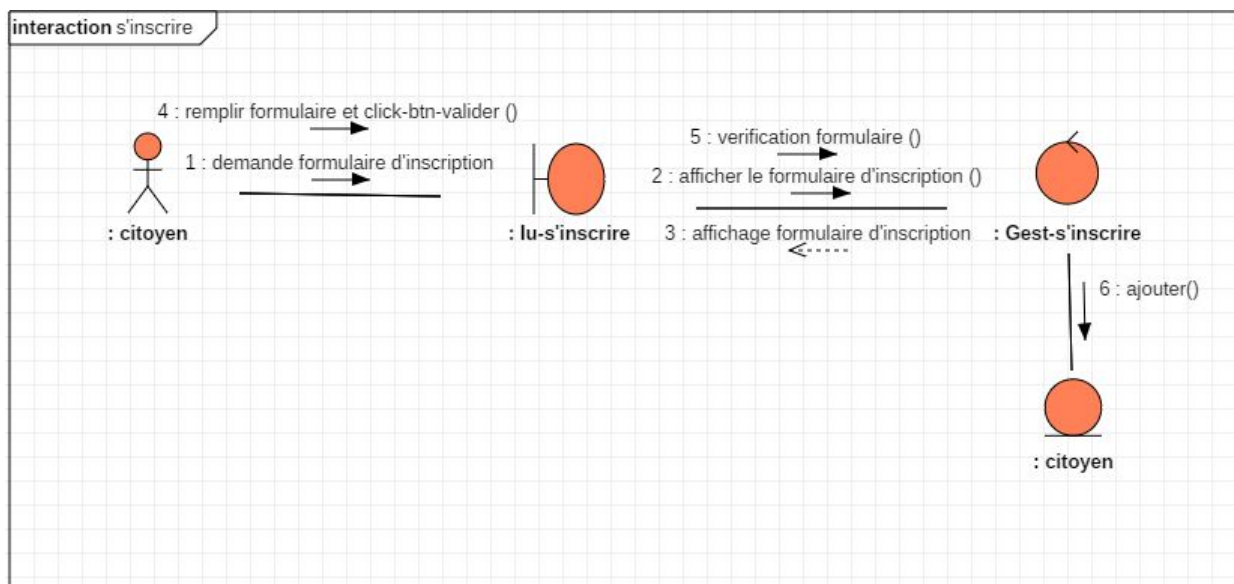


FIGURE 3.3 – Diagramme de collaboration relatif au cas d'utilisation « S'inscrire »

### 3.2.2 Analyse du cas d'utilisation s'authentifier

- Traçabilité entre le modèle du cas d'utilisation et le modèle d'analyse

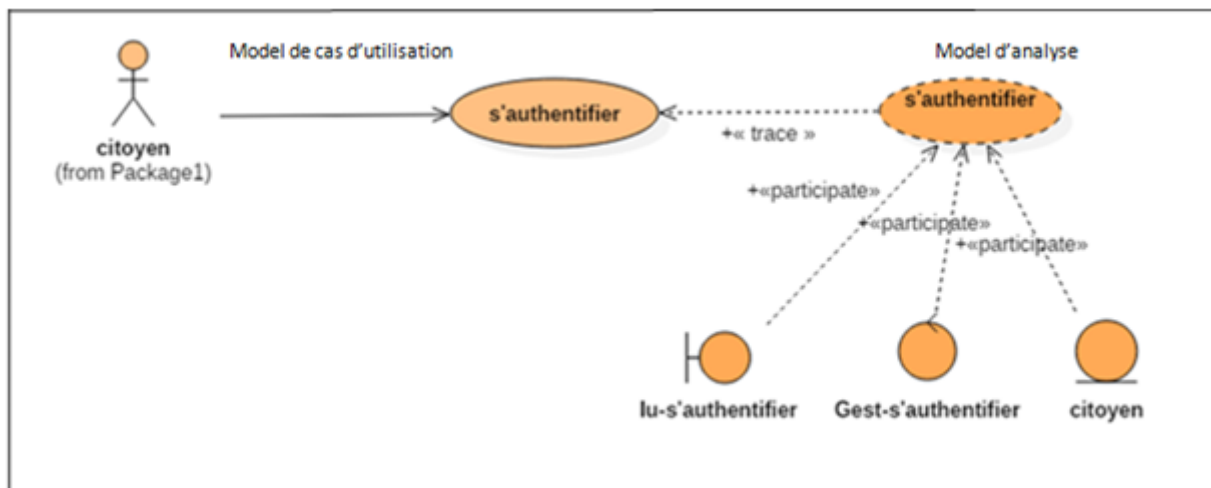


FIGURE 3.4 – Traçabilité entre le Model CU et le Model Analyse du cas d'utilisation « S'authentifier »

- Diagramme de classes d'analyse relatif au cas d'utilisation « S'authenti-

fier »

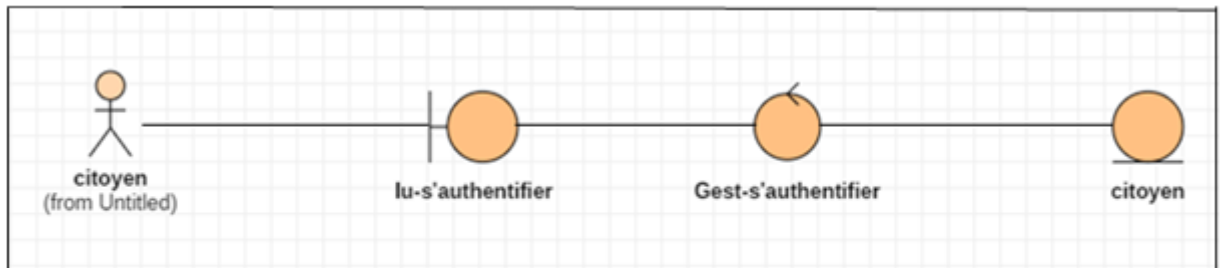


FIGURE 3.5 – Diagramme de classes d'analyse du cas d'utilisation « S'authentifier »

- Diagramme de collaboration relatif au cas d'utilisation « S'identifier »

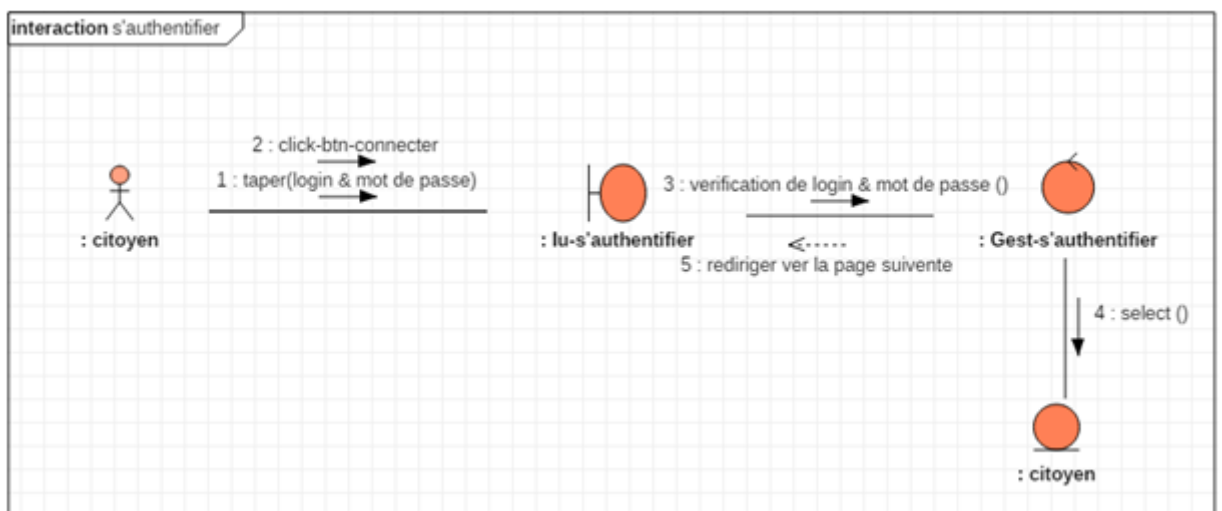


FIGURE 3.6 – Diagramme de collaboration relatif au cas d'utilisation « S'authentifier »

### 3.2.3 Analyse du cas d'utilisation « Effectuer réclamation »

- Traçabilité entre le modèle du cas d'utilisation et le modèle d'analyse

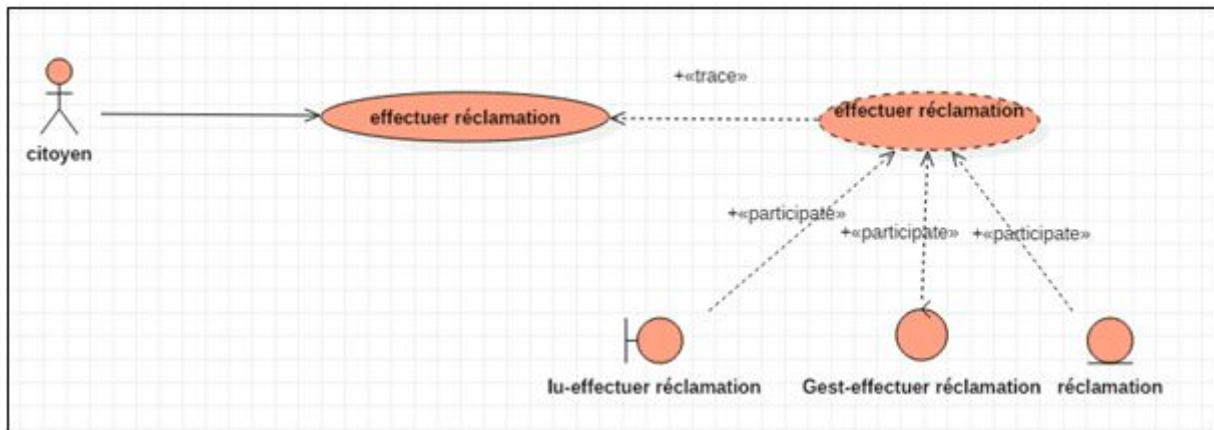


FIGURE 3.7 – Traçabilité entre le Model CU et le Model Analyse du cas d'utilisation «effectuer réclamation»

- Diagramme de classes d'analyse relatif au cas d'utilisation « effectuer réclamation »



FIGURE 3.8 – Diagramme de classes d'analyse du cas d'utilisation «effectuer réclamation »

- Diagramme de collaboration relatif au cas d'utilisation « effectuer réclamation »

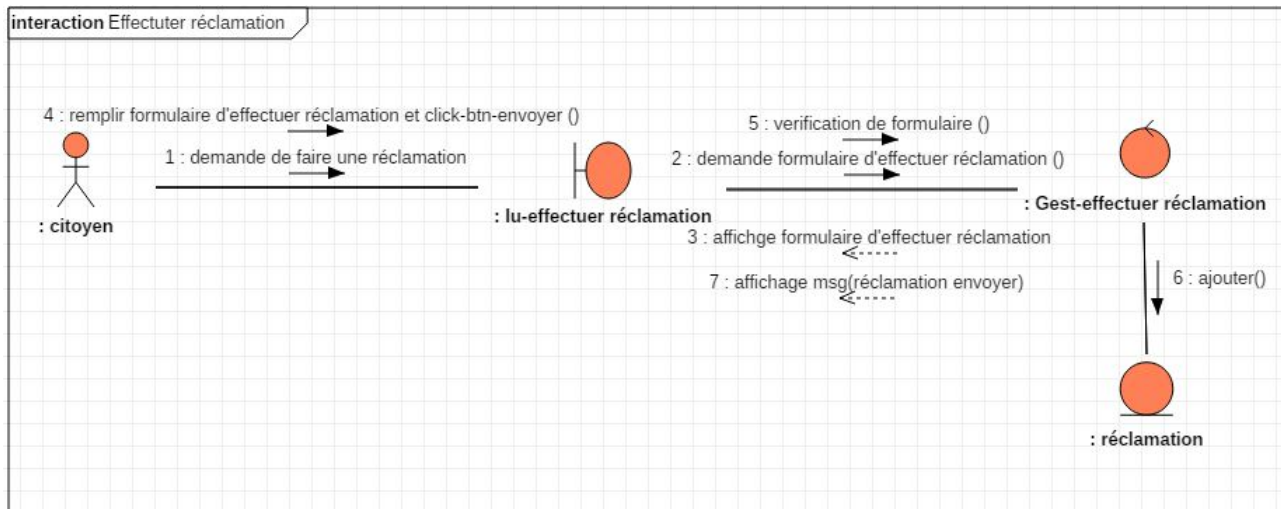


FIGURE 3.9 – Diagramme de collaboration relatif au cas d'utilisation « effectuer réclamation »

### 3.2.4 Analyse du cas d'utilisation « Suivre réclamation »

- Diagramme de collaboration relatif au cas d'utilisation « Suivre réclamation »

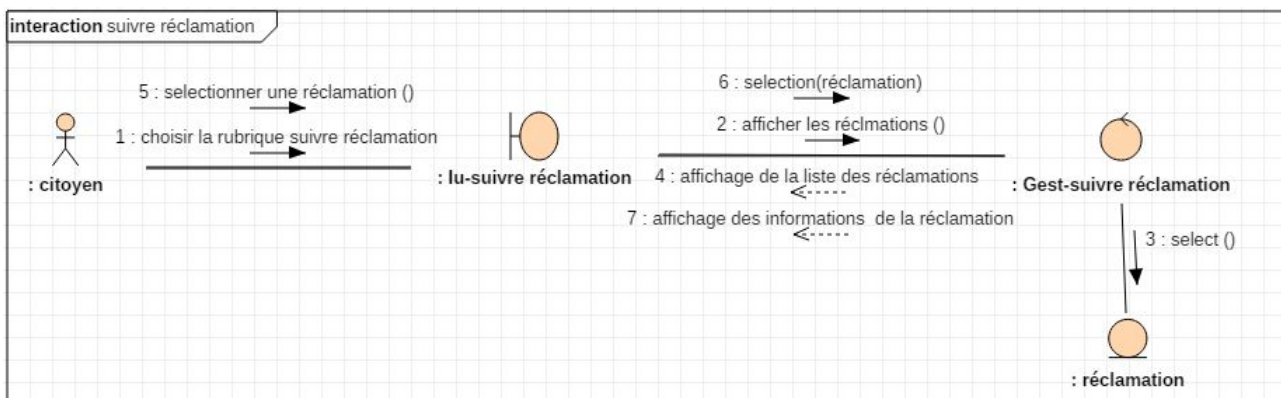


FIGURE 3.10 – Diagramme de collaboration relatif au cas d'utilisation« suivre réclamation »

### 3.2.5 Analyse du cas d'utilisation « gérer réclamation »

#### Analyse du cas d'utilisation « Accepter réclamation »

- Diagramme de collaboration relatif au cas d'utilisation « Accepter réclamation »

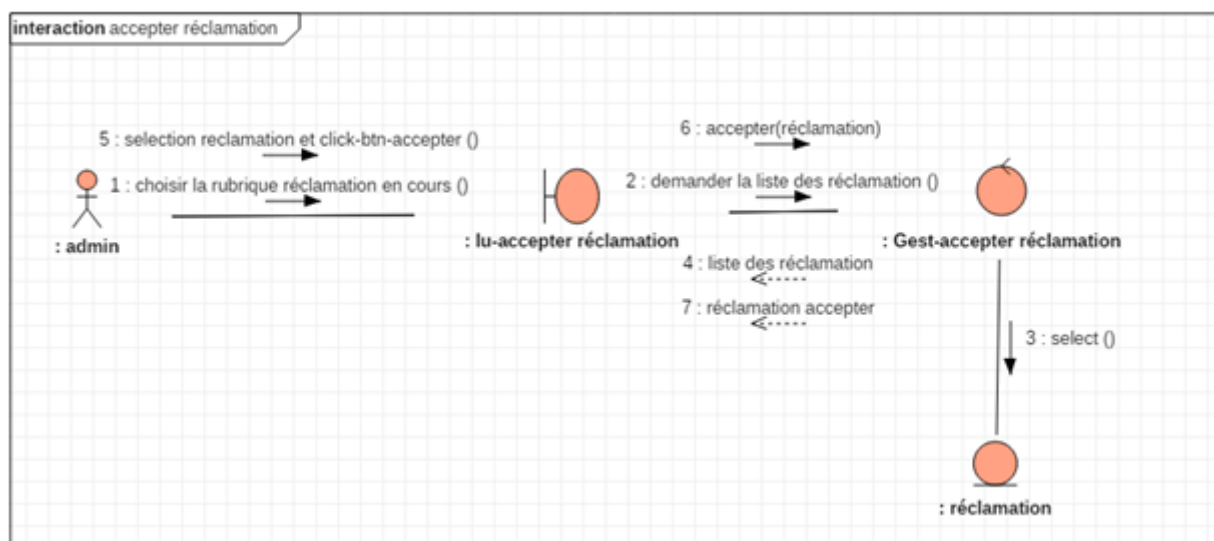


FIGURE 3.11 – Diagramme de collaboration relatif au cas d'utilisation « Accepter réclamation »

#### Analyse du cas d'utilisation « Supprimer réclamation »

- Diagramme de collaboration relatif au cas d'utilisation « Supprimer réclamation »

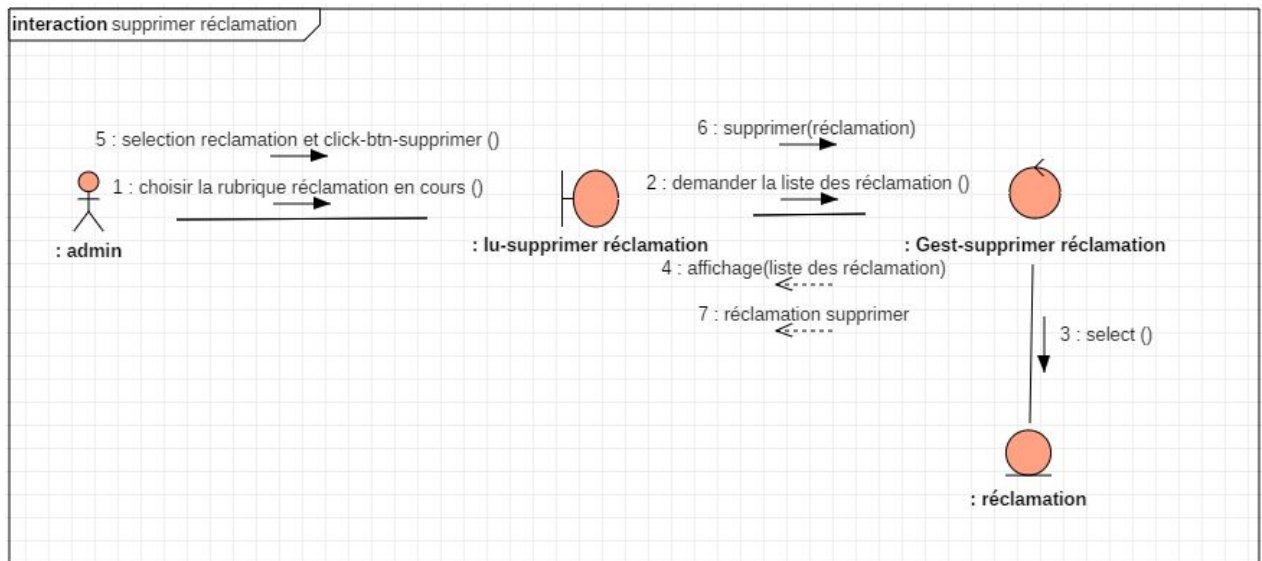


FIGURE 3.12 – Diagramme de collaboration relatif au cas d'utilisation « Supprimer réclamation »

### Analyse du cas d'utilisation « Consulter réclamation »

#### - Diagramme de collaboration relatif au cas d'utilisation « Consulter réclamation »

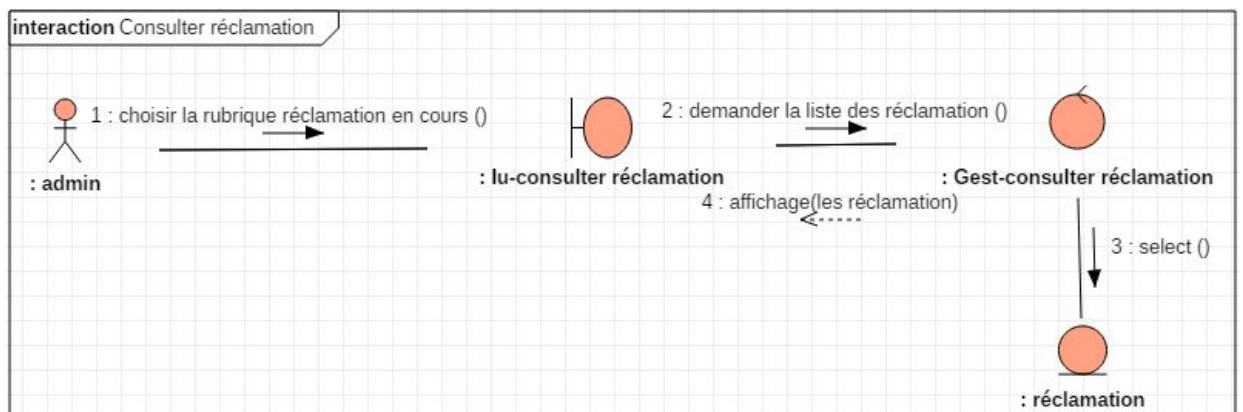


FIGURE 3.13 – Diagramme de collaboration relatif au cas d'utilisation « Consulter réclamation »

### 3.2.6 Analyse du cas d'utilisation « gérer Agent »

#### Analyse du cas d'utilisation « Ajouter Agent »

##### - Diagramme de collaboration relatif au cas d'utilisation « Ajouter agent»

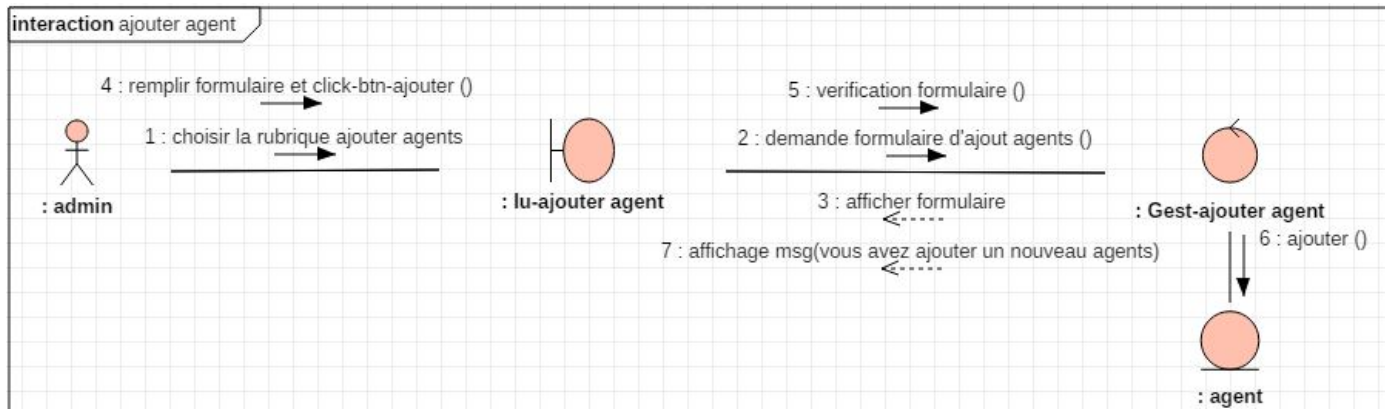


FIGURE 3.14 – Diagramme de collaboration relatif au cas d'utilisation « Ajouter agent»

### 3.2.7 Analyse du cas d'utilisation « gérer taches »

#### Analyse du cas d'utilisation « Affecter tache»

##### - Traçabilité entre le modèle du cas d'utilisation et le modèle d'analyse

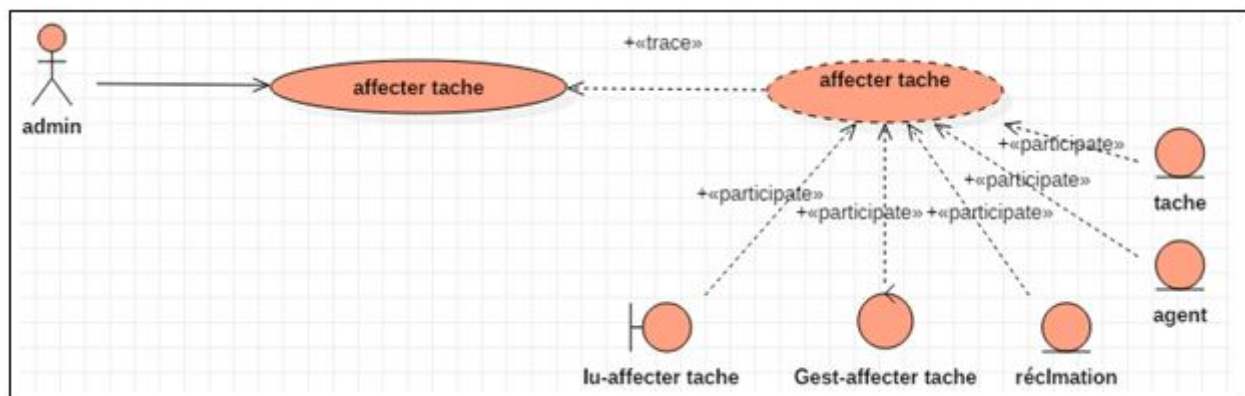


FIGURE 3.15 – Traçabilité entre le modèle du cas d'utilisation et le modèle d'analyse

- Diagramme de classes d'analyse relatif au cas d'utilisation « Affecter tâche »

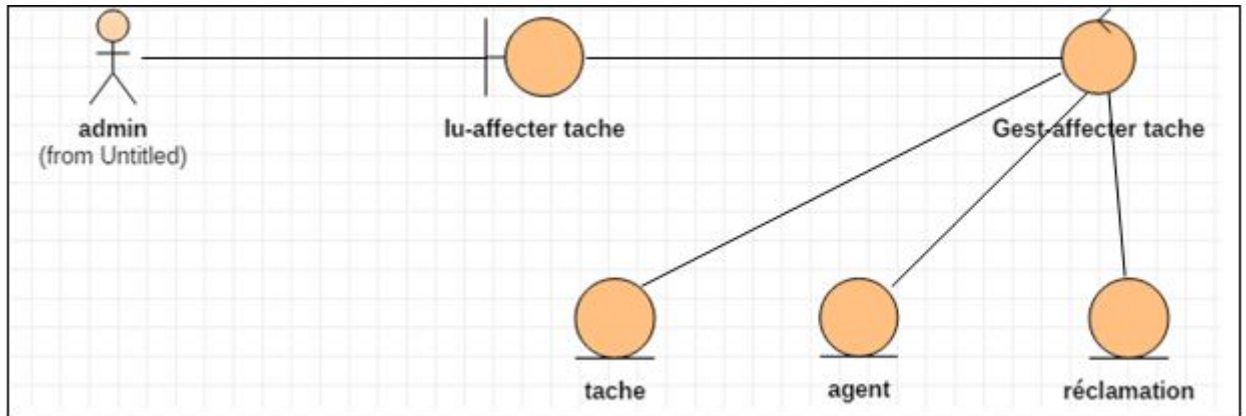


FIGURE 3.16 – Diagramme de classes d'analyse relatif au cas d'utilisation « Affecter tâche »

- Diagramme de collaboration relatif au cas d'utilisation «Affecter tâche»

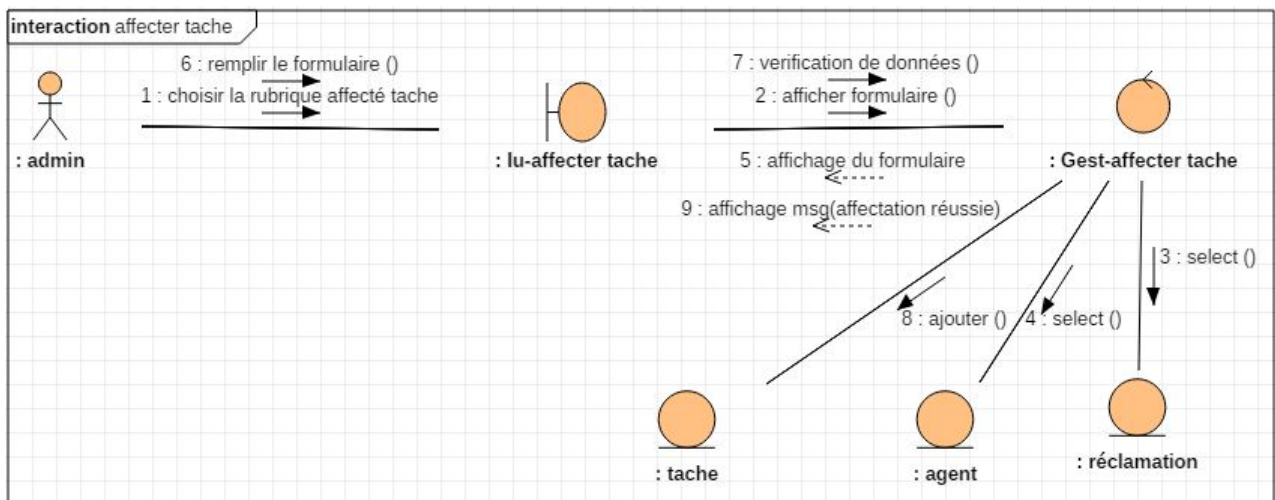


FIGURE 3.17 – Diagramme de collaboration relatif au cas d'utilisation « Affecter tâche »

### 3.2.8 Analyse du cas d'utilisation « Valider tâche »

- Diagramme de collaboration relatif au cas d'utilisation «Valider tâche»



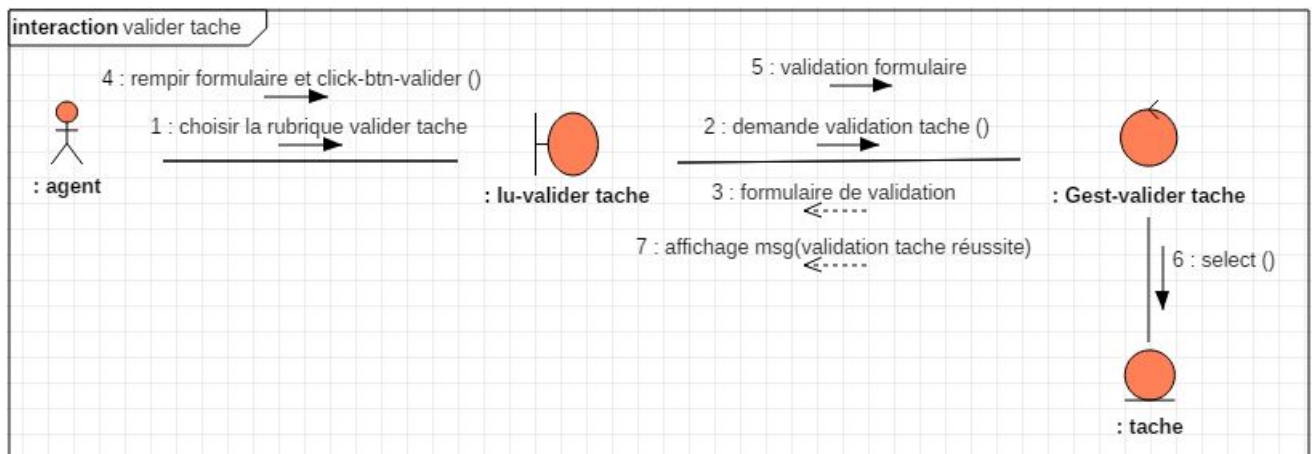


FIGURE 3.18 – Diagramme de collaboration relatif au cas d'utilisation « Valider tache »

### 3.3 Conclusion

Dans ce chapitre, nous avons pu analyser les cas d'utilisation présentés dans le diagramme de cas d'utilisation global de notre application. En particulier, pour chaque cas d'utilisation, nous avons commencé par élaborer la traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse et identifier les classes d'analyse participantes à la réalisation de ce cas. Ensuite, nous avons réalisé le diagramme de classes d'analyse pour illustrer les relations qui existent entre ces classes. Finalement, nous avons élaboré le diagramme de collaboration pour illustrer les interactions entre les objets concernés. Le prochain chapitre s'intéresse à l'activité de conception qui vise à affiner la spécification faite durant l'analyse et évoluer ainsi dans le niveau de détail.



# Conception

## 4.1 Introduction

Dans ce chapitre, nous présentons une conception globale de notre solution suivie d'une explication plus détaillée de ses modules à travers des diagrammes de classes et des diagrammes de séquences.

## 4.2 Conception architecturale

L'architecture d'un système informatique est la façon dont les fonctions ou traitements du système sont distribués entre ses divers composants matériels et logiciels. Pour notre cas, nous avons choisi d'utiliser l'architecture **MVC (Model-View-Controller)** .

Afin d'organiser notre travail, nous avons eu recours au modèle de conception MVC qui est une architecture et une méthode de conception qui organise l'interface Homme-Machine autrement dit Utilisateur-Système de notre application mobile.

Ce paradigme divise cette interface en un modèle (modèle des données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion d'événement, synchronisation), chacun ayant un rôle précis dans l'interface. Ce modèle d'architecture impose la séparation entre les données, la présentation et traitement, ce qui donne trois parties fondamentales dans l'application finale : le

modèle, la vue et le contrôleur comme le montre la figure suivant :

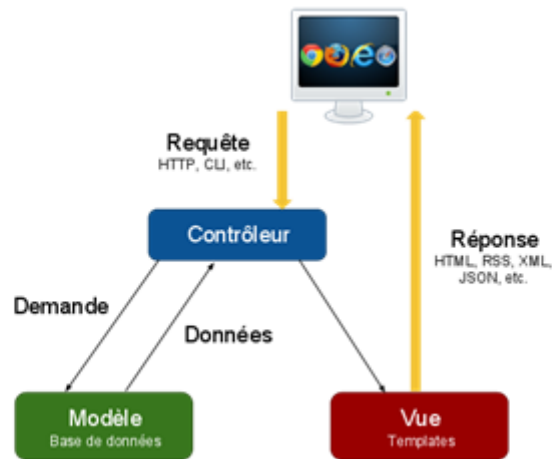


FIGURE 4.1 – Illustration de l'architecture MVC[Ref.w6]

## 4.3 Conception des cas d'utilisation

Pour la conception des cas d'utilisation, nous élaborons :

- La Traçabilité entre le modèle d'analyse et le modèle de conception.
- Le diagramme de classes de conception.
- Le diagramme de séquence.

### 4.3.1 Conception du cas d'utilisation «S'authentifier »

- La figure suivante présente la Traçabilité entre le modèle d'analyse et le modèle de conception pour le cas d'utilisation «S'authentifier».

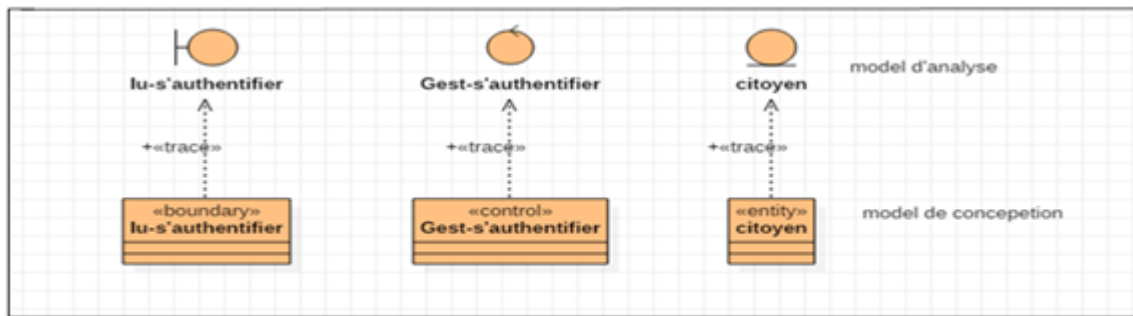


FIGURE 4.2 – Traçabilité entre le modèle d'analyse et le modèle de conception pour le cas d'utilisation «S'authentifier».

- Diagramme de classes de conception relatif au cas d'utilisation « S'authentifier »

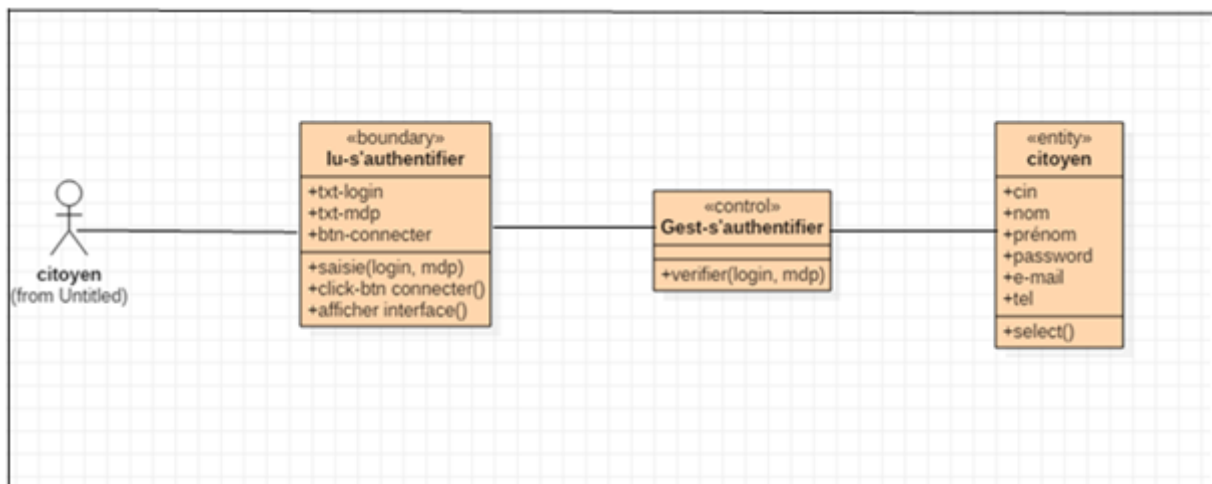


FIGURE 4.3 – Diagramme de classes de conception relatif au cas d'utilisation « S'authentifier »

- Diagramme de séquence relatif au cas d'utilisation « S'authentifier ».

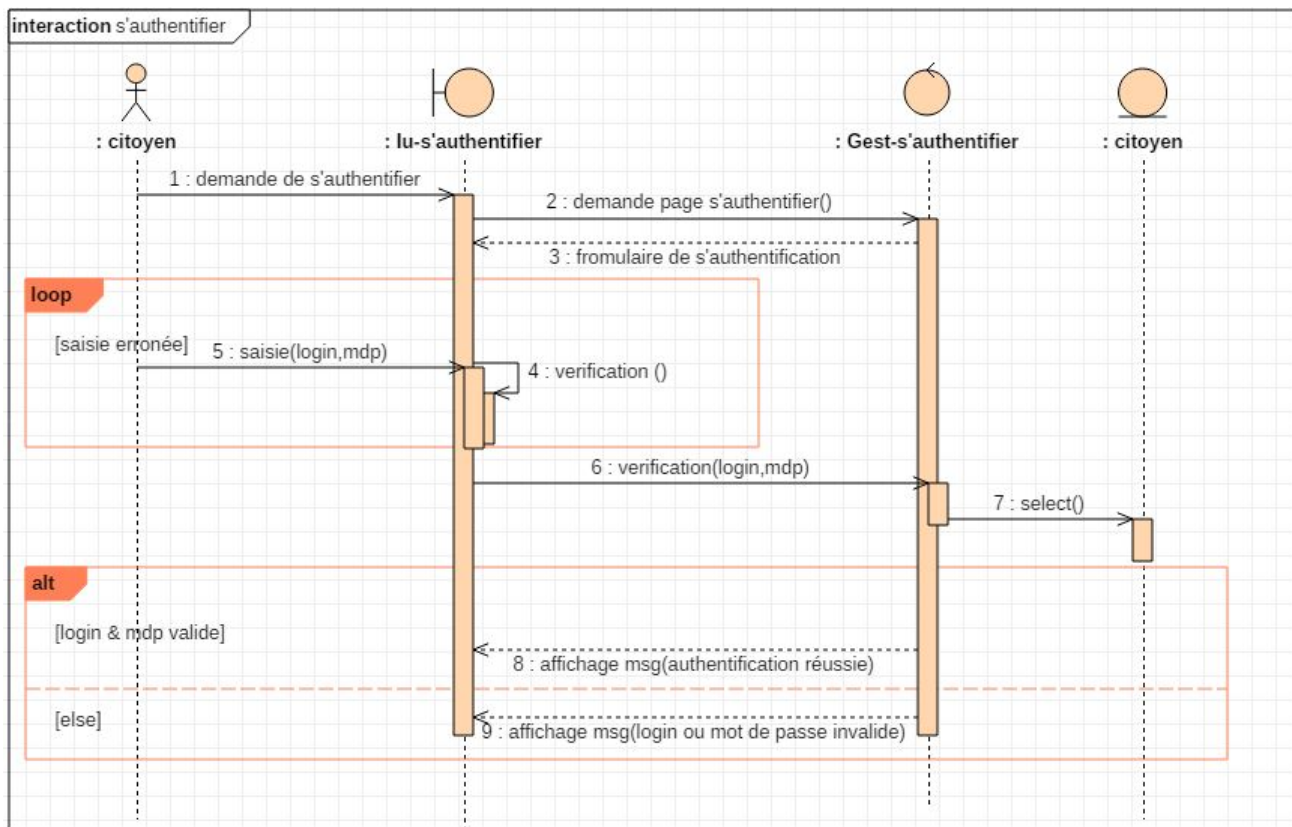


FIGURE 4.4 – Diagramme de séquence relatif au cas d'utilisation « S'authentifier »

### 4.3.2 Conception du cas d'utilisation « effectuer réclamation »

- Diagramme de classes de conception relatif au cas d'utilisation « effectuer réclamation ».

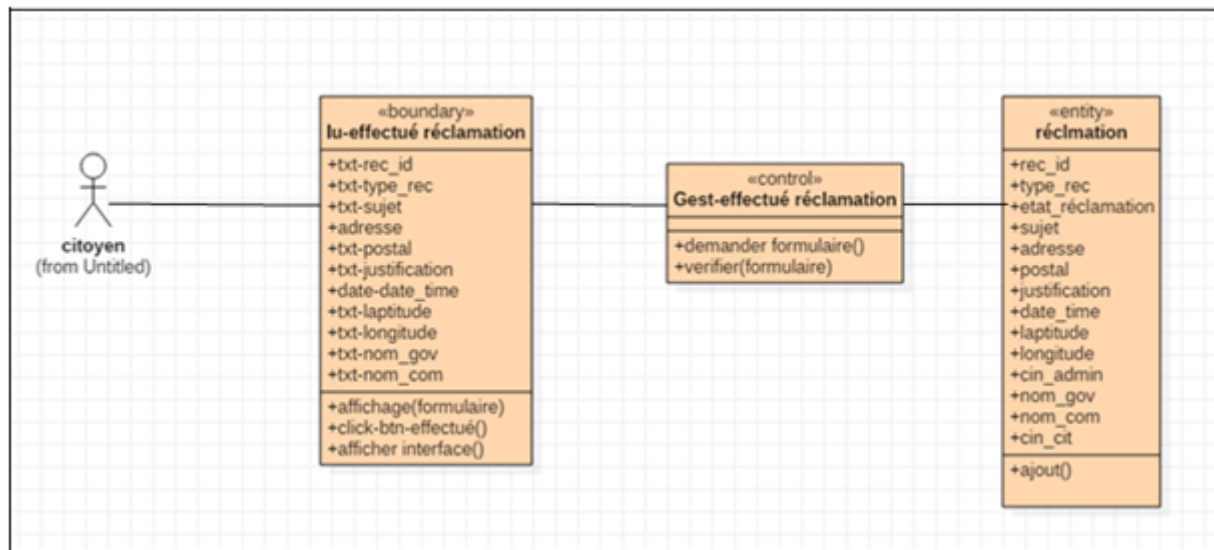


FIGURE 4.5 – Diagramme de classes de conception relatif au cas d'utilisation «effectuer réclamation».

- Diagramme de séquence relatif au cas d'utilisation «Effectuer réclamation»

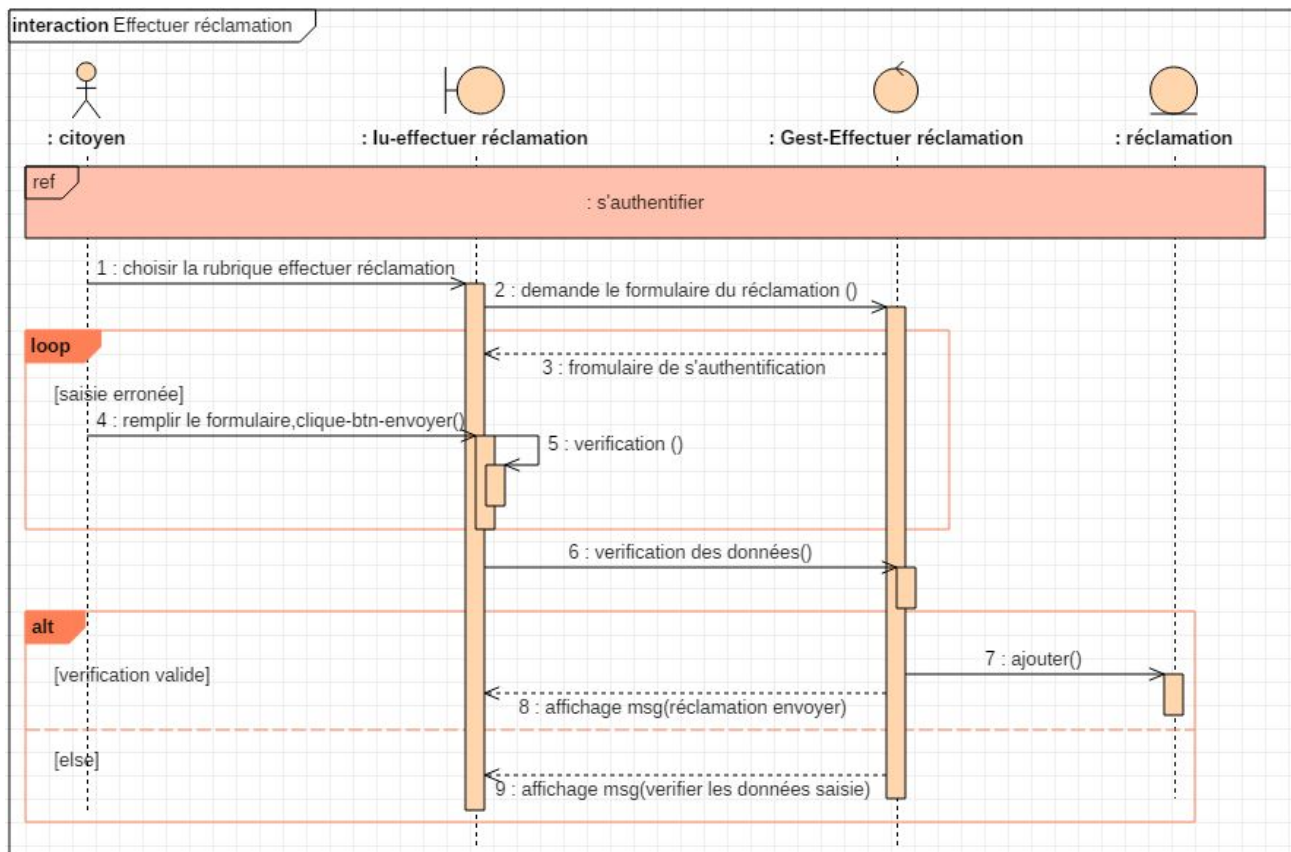


FIGURE 4.6 – Diagramme de séquence relatif au cas d'utilisation « Effectuer réclamation »

### 4.3.3 Conception du cas d'utilisation «suivre réclamation»

Diagramme de classes de conception relatif au cas d'utilisation «suivre réclamation».



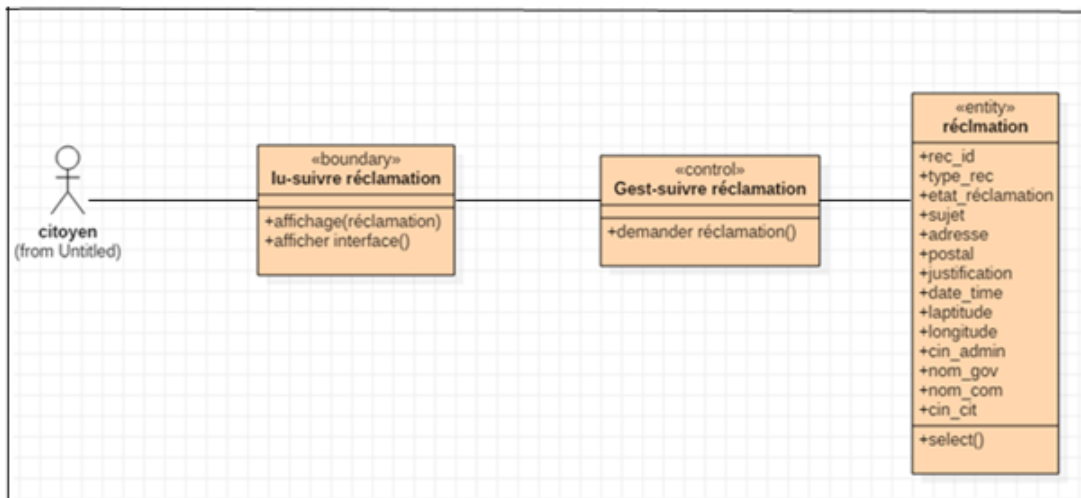


FIGURE 4.7 – Diagramme de classes de conception relatif au cas d'utilisation «suivre réclamation».

- Diagramme de séquence relatif au cas d'utilisation «suivre réclamation».

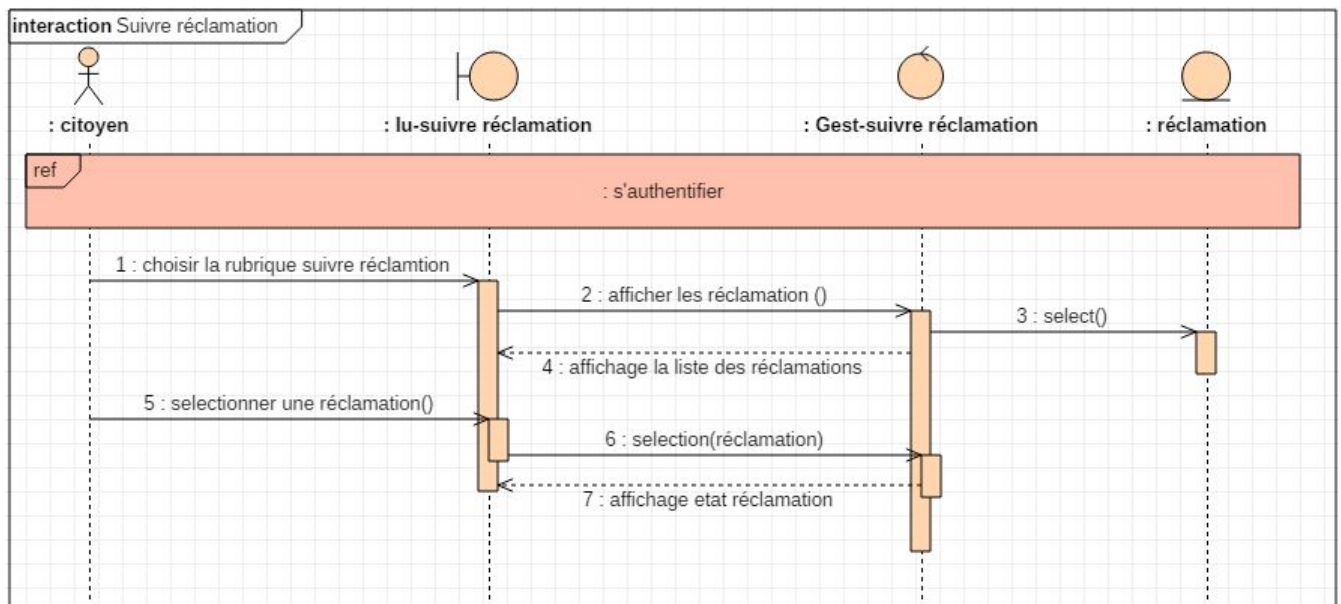


FIGURE 4.8 – Diagramme de séquence relatif au cas d'utilisation «suivre réclamation».

### 4.3.4 Conception du cas d'utilisation «gérer réclamation»

#### Conception du cas d'utilisation «Consulter réclamation»

- Diagramme de séquence relatif au cas d'utilisation «Consulter réclamation».

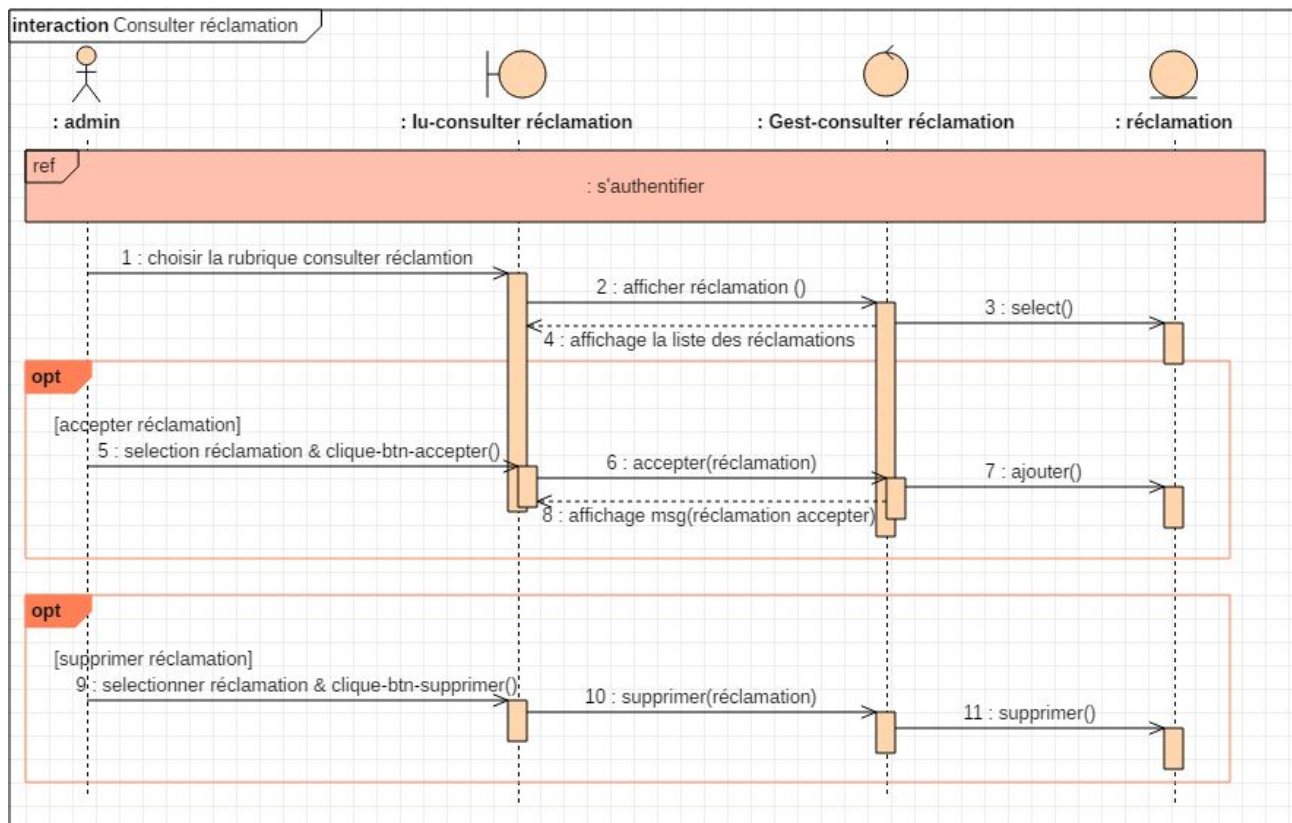


FIGURE 4.9 – Diagramme de séquence relatif au cas d'utilisation «Consulter réclamation»

### 4.3.5 Conception du cas d'utilisation «Gérer agent»

#### Conception du cas d'utilisation «Ajouter agent»

- Diagramme de séquence relatif au cas d'utilisation «Ajouter agent»

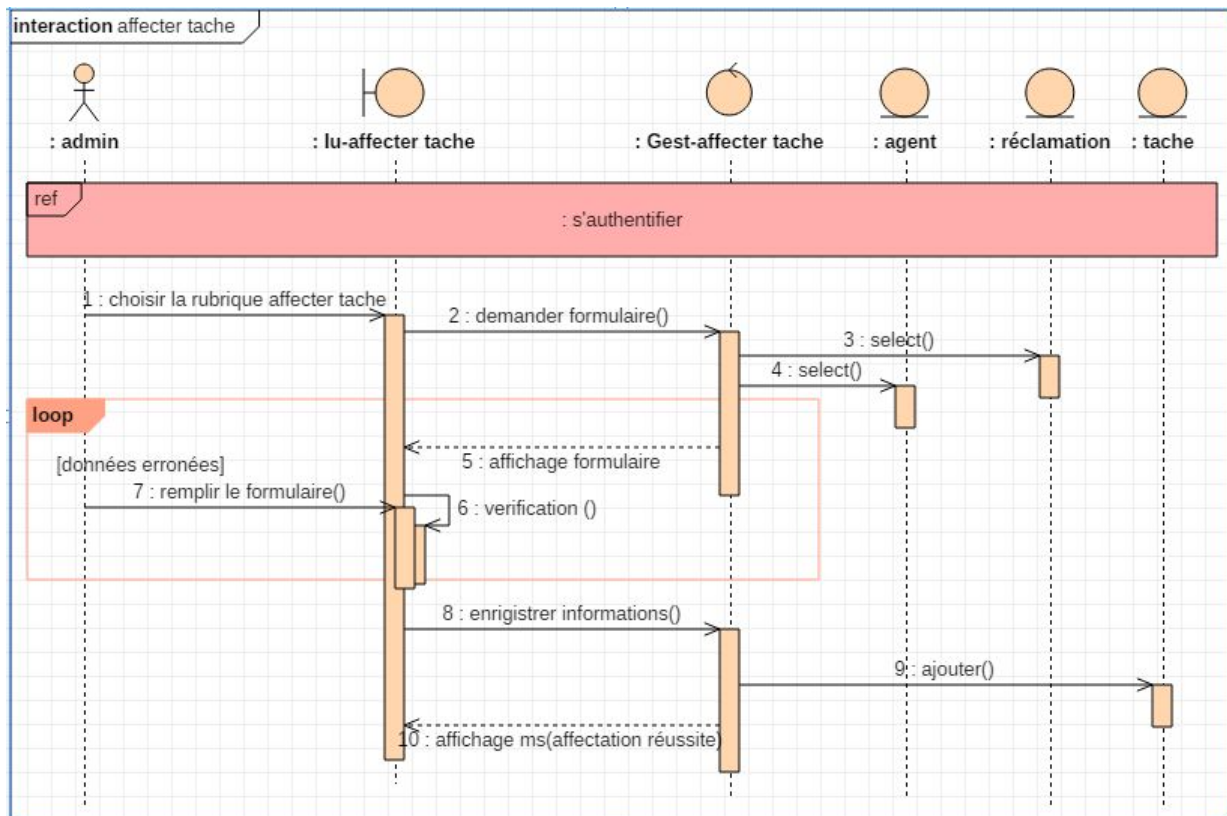


FIGURE 4.10 – Diagramme de séquence relatif au cas d'utilisation «Ajouter agent»

### 4.3.6 Conception du cas d'utilisation «Gérer tâche»

#### Conception du cas d'utilisation «Affecter tâche»

-Diagramme de séquence relatif au cas d'utilisation «Affecter tâche»

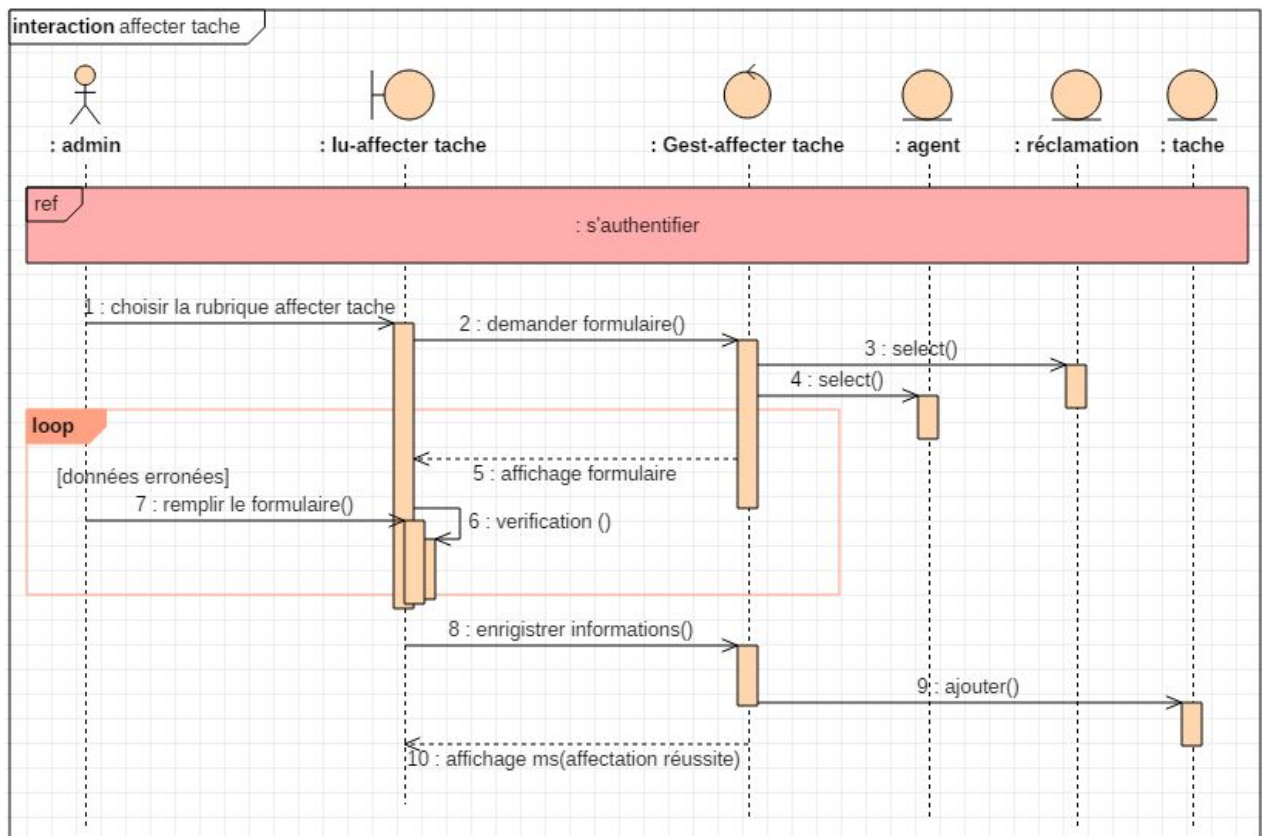


FIGURE 4.11 – Diagramme de séquence relatif au cas d'utilisation «Affecter tache»

## 4.4 Conception des classes

### 4.4.1 Diagramme classes entités

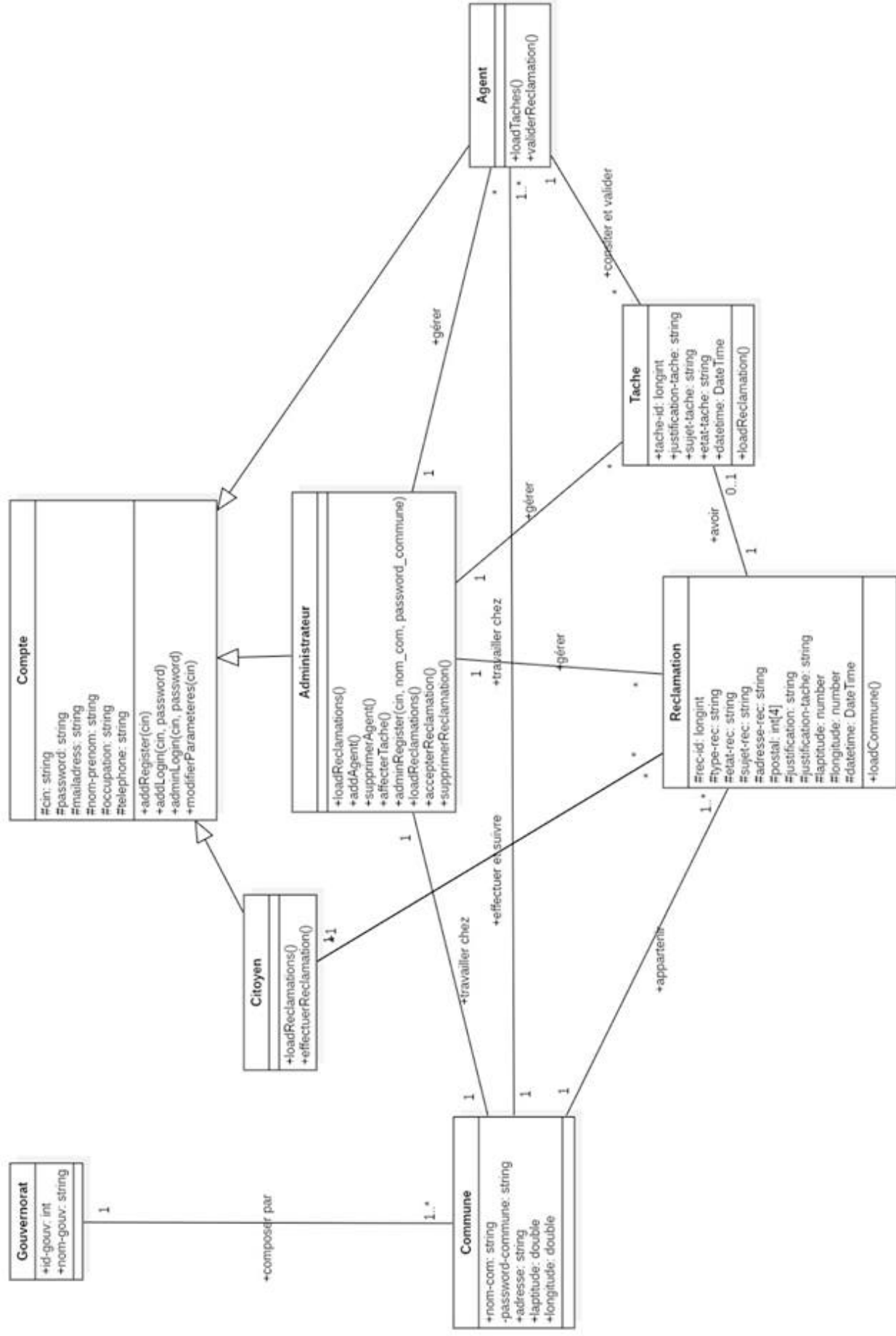


FIGURE 4.12 – Diagramme de classes

### 4.4.2 Schéma de la base de données

#### Règles de passage du diagramme de classes vers le modèle relationnel

Le schéma de la base de données est déduit du diagramme de classes entités présenté ci-dessus. Pour le passage du diagramme de classes de conception vers la base de données relationnelle, nous avons respecté les règles suivantes :

- **Règle N1 :**

- Chaque classe entité se transforme en une table.
- Chaque attribut de classe se transforme en un champ de table.
- L'identifiant de la classe devient la clé primaire de la table.

- **Règle N2 :** Association binaire (1..1) - (0..n) ou (1..1) - (1..n) se traduit en :

- Ajoutant une clé étrangère (identifiant de la classe de cardinalité (1..1)) à la table provenant de la classe dont la cardinalité est (0..N) ou (1..N).

- **Règle N3 :** Association binaire (0..1) - (1..1) se traduit en :

- Ajoutant une clé étrangère (identifiant de la classe de cardinalité (1..1)) à la table provenant de la classe dont la cardinalité est (0..1).

- **Règle N4 :** Association binaire (0..N) ou (1..N) - (0..N) ou (1..N) se traduit par :

- Une nouvelle table dont la clé primaire est composée des identifiants des deux classes. Les éventuelles propriétés de l'association deviennent les attributs de cette table.

- **Règle N5 :** Présence d'une généralisation :

- Créer une table avec tous les attributs des classes.
- Ajouter un attribut pour distinguer les types des objets.

**OU**

- Créer une table pour chaque sous type, chaque table se compose des attributs génériques et des attributs spécifiques.

**OU**

- Créer une table par classe et des associations.

Après avoir appliqué les règles de passage citées ci-dessus, nous avons obtenu le modèle relationnel suivant :

```
compte(cin, password, mailadress, nom_prenom, occupation, telephone)
administrateur(cin_admin #nom_com)
agent(#cin_agent, #cin_admin, #nom_com)
citoyen(cin_cit)
gouvernorat(nom_gouv)
commune(nom_com, adresse, password_dmin, #nom_gouv)
reclamation(rec_id, type_rec, etat_reclamation, sujet, adresse, postal, datetimes,
laptitude, longitude, photo, #cin_admin, #nom_gouv, #nom_com, #cin_cit)
tache(tache_id, justification, sujet_tache, adresse, etat_tache, datetimes,
#cin_agent, #cin_admin, #rec_id, #nom_gouv)
```

## 4.5 Conclusion

À la fin de cette activité, nous avons réussi à réaliser la conception des cas d'utilisation, présenter le diagramme de classes global de notre application, et générer le schéma final de notre base de données.





# Réalisation

## 5.1 Introduction

Dans ce qui précède, nous avons essayé de suivre un enchaînement logique qui nous a permis de bâtir la conception de notre projet. Ainsi, cette partie contient le dernier volet de ce rapport ayant pour objectif la description de l'aspect implémentation et développement de l'application déjà conçue et modélisée dans les chapitres précédents. Dans ce chapitre, nous présenterons l'environnement matériel, logiciel et les différents outils de développement utilisés.

## 5.2 Environnement de travail

### 5.2.1 Environnement matériel

Pour la réalisation du projet, nous avons utilisé deux ordinateurs portables et un smartphone Nokia 2.1 dont les caractéristiques sont les suivantes :

**Premier ordinateur :**

- Processeur Intel CORE i7 7ème génération 2.7GHz.
- 12 Go DDR4 mémoire vive.
- 1000 Go disque dur.
- Système d'exploitation Microsoft Windows 10 pro x64.

**Deuxième ordinateur :**

- Processeur Intel CORE i5 2.5GHz.
- 4 Go DDR3 mémoire vive.
- 500 Go disque dur.
- Système d'exploitation Microsoft Windows 7pro x64.

**Smart Phone :**

- Quad-core 1.4 GHz Cortex-A53
- 1 GB RAM.
- 8 GB stockage interne.
- Système d'exploitation Android 9 Pie.

### 5.2.2 Environnement logiciel

□ **Ionic 3 :**

Ionic Framework est un mélange d'outils et de technos pour développer des applications mobiles hybrides rapidement et facilement. Il s'appuie sur Angular pour la partie application web du Framework et sur Cordova pour la partie construction des applications natives. Ce Framework open source permet de développer une application déployable sur plusieurs environnements tels qu'un site web ou une application mobile pour des systèmes tels qu'Android ou Ios ou Windows Phone par exemple. [Ref.w1]



FIGURE 5.1 – Logo de Ionic

- **Angular7 :** Angular est un Framework JavaScript Open Source développé par Google. Il utilise l'architecture MVM (Modèle Vue Modèle), proche

du modèle MVC. Cela va permettre de structurer son code et bien séparer la vue (l'interface) des modèles (fonctionnement). Jusqu'à maintenant, Angular a eu droit à 2 versions majeures. Angular 1, Appelé AngularJS et Angular 2+ et spécifiquement la version,7, appelé simplement Angular. Il est considéré comme un langage « côté client », ceux-ci permettent de gérer l'interface utilisateur de chaque page (affichage, interactions...) de façon dynamique et viennent en complément aux langages côté serveur.[réf.w7]



FIGURE 5.2 – Logo Angular[réf.w7]

- **Node.js :** Node.js est une plateforme de développement JavaScript. Ce n'est pas un serveur, ce n'est pas un Framework, c'est juste le langage JavaScript avec des bibliothèques permettant de réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connections réseau ou encore créer un fichier.

Il est souvent confondu avec un serveur car c'est son origine : Node.js a été créé par Ryan Dahl dans le but de pouvoir créer des applications temps réel ou le serveur est capable de pousser de l'information au client.[Ref.w9]



FIGURE 5.3 – Logo Nodejs[Ref.w9]

- **NPM :** Npm est le gestionnaire de paquets officiel de Node.js. Depuis la version 0.6.3 de Node.js, npm fait partie de l'environnement et est donc automatiquement installé par défaut<sup>3</sup>. Npm fonctionne avec un terminal et gère les dépendances pour une application. Il permet également d'installer des applications Node.js disponibles sur le dépôt npm.[Ref.w8]



FIGURE 5.4 – Logo NPM [Ref.w8]

- **Xampp :** Possédant deux serveur (Apache et MySQL), nous avons opté pour XAMPP afin de gérer plus facilement la base de données. XAMPP est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache<sup>2</sup>, du langage de scripts PHP et d'une base de données MySQL.[Ref.w10]



FIGURE 5.5 – Logo Xamp

- **Système de gestion de base de données MySQL** MySQL [Oracle Corporation, 2017] est un système de gestion de base de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle et Microsoft SQL Server. MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service des données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multi-utilisateur.[Ref.w11]



FIGURE 5.6 – Logo Mysql [Ref.w11]

- **StarUML** : StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux. Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation. Cependant, seule une version Windows est disponible.[Ref.w12]



FIGURE 5.7 – Logo Uml [Ref.w12]

- **PHP 7 :** PHP est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes. Dans le contexte de notre projet, nous avons utilisé un web service REST qui contient spécifiquement l'extension MySQLI pour la partie mobile et l'extension PDO sur la partie web.[Ref w.13]



FIGURE 5.8 – Logo PHP

- **Visual Studio Code :** Visual Studio Code (VSC par la suite) est un éditeur de code open-source, gratuit et multiplateforme (Windows, Mac et Linux), développé par Microsoft, à ne pas confondre avec Visual Studio, l'IDE propriétaire de Microsoft. VSC est développé avec Electron et exploite des fonctionnalités d'édition avancées du projet Monaco Editor. Principalement conçu pour le développement d'application avec JavaScript, Type Script et Node.js, l'éditeur peut s'adapter à d'autres types de langages grâce à un système d'extension bien fourni.[Ref.w14]



FIGURE 5.9 – Logo Visual Studio Code[Ref.w14]

- **Overleaf** : Overleaf est une plateforme en ligne gratuite permettant d'éditer du texte en LATEX sans aucun téléchargement d'application. En outre, elle offre la possibilité de rédiger des documents de manière collaborative, de proposer ses documents directement à différents éditeurs (IEEE Journal, Springer, etc.) ou plateformes d'archives ouvertes (arXiv, engrxiv, etc.) pour une éventuelle publication.

Cette plateforme est très compatible avec différents supports tels que tablettes et smartphones.

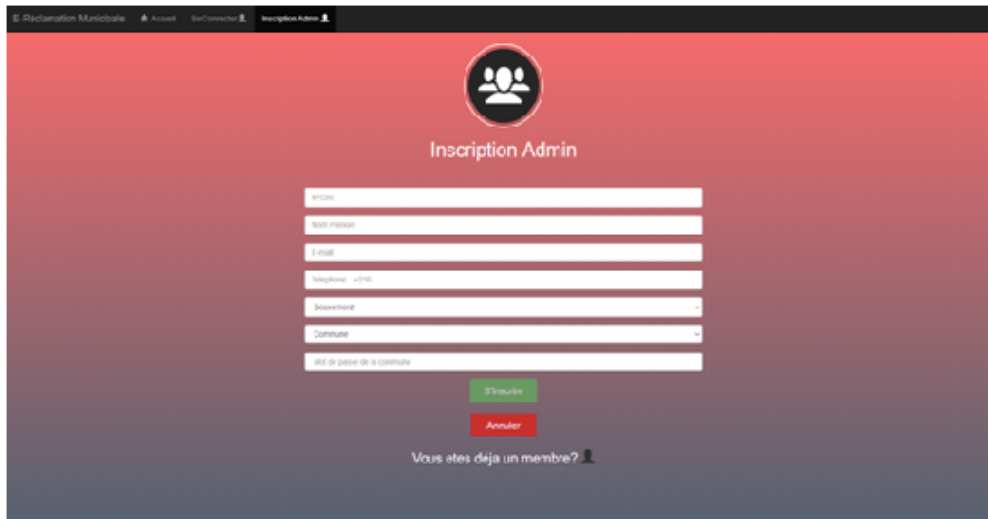


FIGURE 5.10 – Logo Overleaf

## 5.3 Scénario d'exécution

A ce stade nous présentons notre application à travers les divers imprimés écrans réalisés.

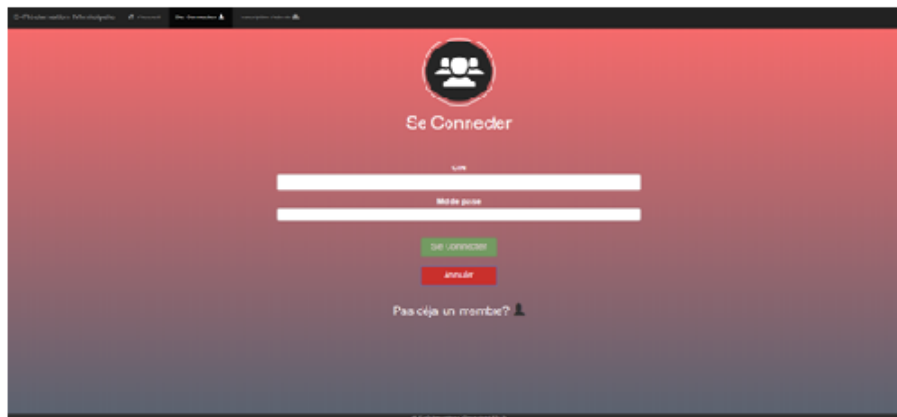
### 5.3.1 Scénario d'exécution de la partie web



The screenshot shows the 'Inscription Admin' page of a web application. The page has a dark red header with navigation links: 'Reclamation Municipale', 'Accueil', 'Se Connecter', and 'Inscription Admin'. Below the header is a circular icon with three stylized figures. The main title 'Inscription Admin' is centered. Below it is a form with the following fields: 'Nom', 'Nom complet', 'Email', 'Telephone +216', 'Sexe' (a dropdown menu), 'Commune' (a dropdown menu), and 'Mot de passe de la commune'. There are two buttons: a green 'S'inscrire' button and a red 'Annuler' button. At the bottom, there is a link 'Vous êtes déjà un membre?' with a user icon.

FIGURE 5.11 – Interface inscription Admin

Cette figure présente l'interface inscription d'admin. En effet pour s'inscrire l'administrateur doit remplir le formulaire.



The screenshot shows the 'Se Connecter' page of the same web application. The header is identical to the previous page. Below the header is the same circular icon. The main title 'Se Connecter' is centered. Below it is a form with two fields: 'User' and 'Mot de passe'. There are two buttons: a green 'Se connecter' button and a red 'Annuler' button. At the bottom, there is a link 'Pas déjà un membre?' with a user icon.

FIGURE 5.12 – Interface authentication



Cette figure présente l'interface authentification pour l'administrateur.

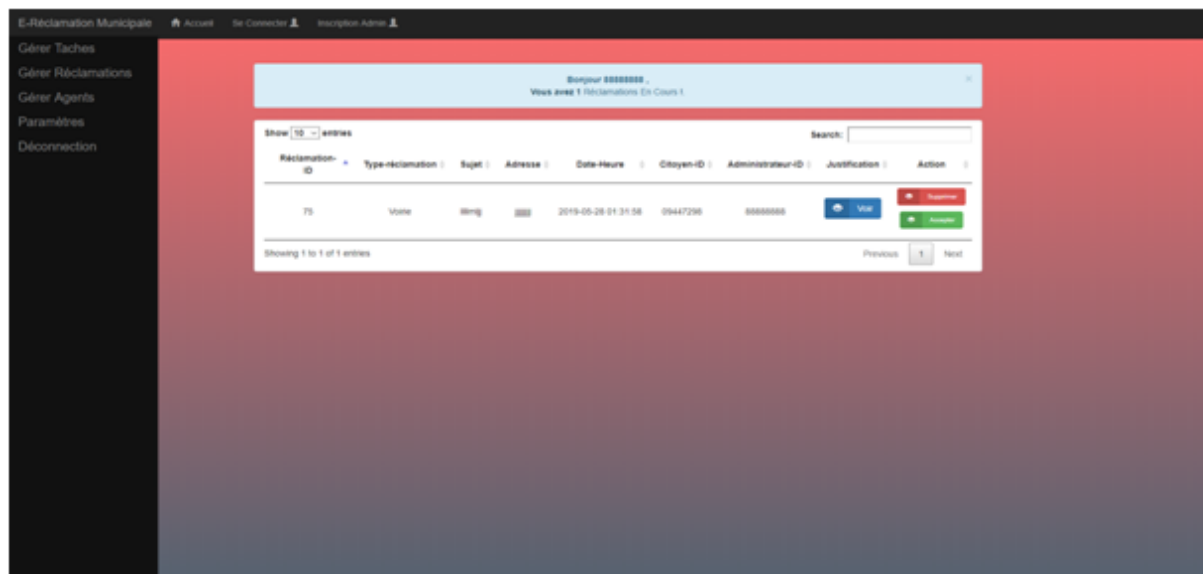


FIGURE 5.13 – Interface réclamation en cours

Cette figure présente l'interface de les réclamations en cours.



FIGURE 5.14 – Interface affecter tache

Cette figure présente l'interface affecter tache, l'administrateur va affecter une réclamation à un agent.

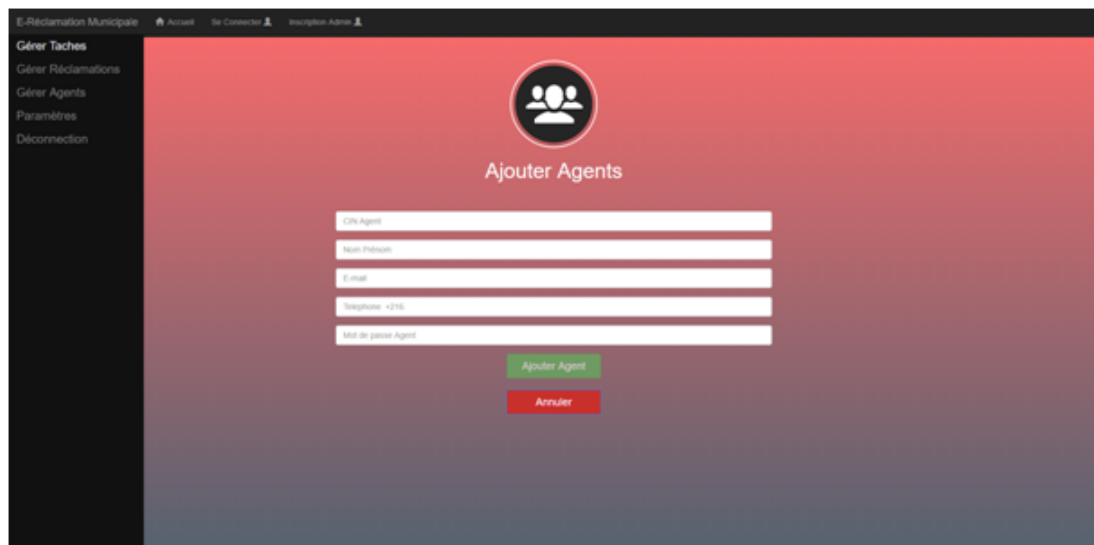
The screenshot shows a web application interface for adding agents. On the left is a dark sidebar with a menu containing 'Gérer Tâches', 'Gérer Réclamations', 'Gérer Agents', 'Paramètres', and 'Déconnection'. The main content area has a red-to-purple gradient background. At the top center is a circular icon with three stylized figures. Below it, the text 'Ajouter Agents' is displayed. A form with five white input fields is centered: 'CIN Agent', 'Nom Prénom', 'E-mail', 'Téléphone +216', and 'Mot de passe Agent'. Below the form are two buttons: a green 'Ajouter Agent' button and a red 'Annuler' button. At the top of the page, a navigation bar includes the text 'E-Réclamation Municipale' and links for 'Accueil', 'Se Connecter', and 'Inscription Admin'.

FIGURE 5.15 – Interface ajouter agent

Pour ajouter un agent l'administrateur doit remplir ce formulaire.

### 5.3.2 Scénario d'exécution de la partie mobile

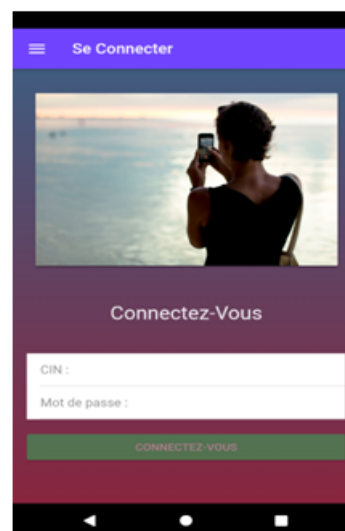
The screenshot shows a mobile application interface for authentication. At the top is a purple header bar with a hamburger menu icon and the text 'Se Connecter'. Below the header is a large image of a person taking a photo of a sunset over water. Underneath the image, the text 'Connectez-Vous' is displayed. There are two white input fields: 'CIN :' and 'Mot de passe :'. Below these fields is a green button labeled 'CONNECTEZ-VOUS'. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a circle, and a square.

FIGURE 5.16 – Interface authentication

Cette figure présente l'interface authentication.

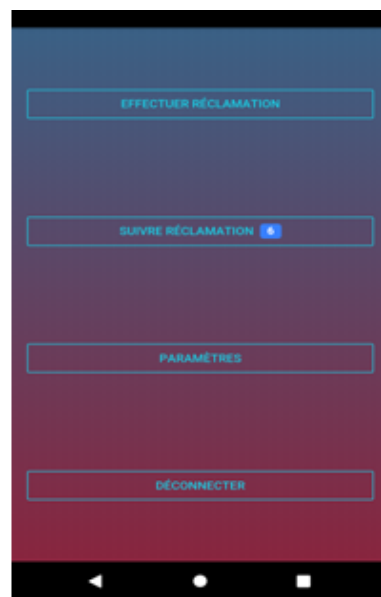


FIGURE 5.17 – Interface de l'espace citoyen

Cette figure présente l'espace du citoyen dans notre application ou il peut effectuer une réclamation, suivre une réclamation et modifier ses informations.

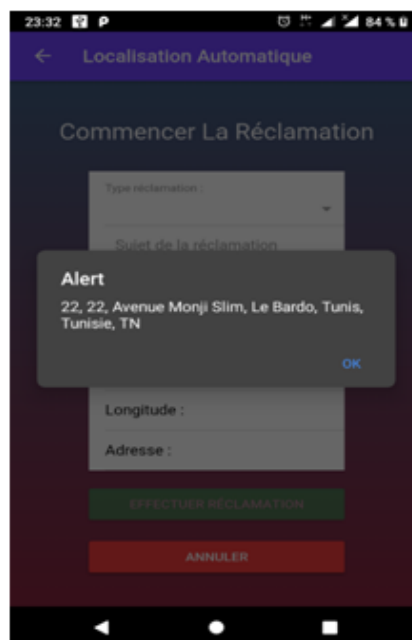


FIGURE 5.18 – Interface effectuer réclamation automatique

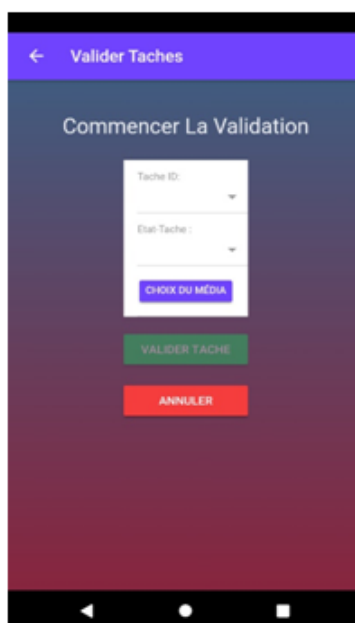


FIGURE 5.20 – Interface valider tache

Cette figure présente l'interface de effectuer réclamation avec calcule de l'adresse exacte de la réclamation automatiquement.



FIGURE 5.19 – Interface de l'espace agents

Cette figure présente l'interface de validation de tâche après la fin des travaux sur terrain.

## 5.4 Conclusion

Dans ce dernier chapitre, nous avons traité les détails de la réalisation de notre application en présentant les différents outils et les technologies utilisées. De plus, nous avons montré les différentes interfaces de l'application.



# Conclusion générale

Suite aux dernières élections municipales, la majorité des électeurs a voté pour des représentants municipaux qu'ils estiment capables d'améliorer la qualité de leurs vies dans leurs villes. Ainsi les municipalités sont dotées d'un plus grand pouvoir et doivent être de plus en plus performantes et réactives par rapport aux exigences des citoyens. Notre projet intitulé « Conception et développement d'une application web/mobile de réclamations municipales en ligne » consiste à la conception et la réalisation d'une plateforme web et mobile destinée pour l'effectuation et la consultation des réclamations municipales en ligne. Contrairement à la majorité des applications existantes qui offrent des fonctionnalités limitées et nécessitent un effort d'optimisation, nous avons réalisé un système qui permet à la fois de gérer les réclamations, les tâches, donner la possibilité aux citoyens de choisir d'effectuer des réclamations municipales manuellement ou avec géo localisation et de les suivre à distance et depuis leurs smartphones. En effet, donner la possibilité aux citoyens d'avoir une relation plus forte avec toutes les municipalités et garantir un accès facile aux services de réclamations municipales. En plus, nous avons proposé la relation administrateur-agent pour améliorer l'architecture des réclamations municipales en ligne en donnant à chaque commune le choix de choisir son administrateur, et pour l'administrateur le choix de choisir ses agents, leur affecter des tâches pour réaliser les travaux sur terrain et les valider par une photo après avoir été accomplis. Ce processus entraîne plus de transparence dans la relation des citoyens avec leurs communes.

Au long de ce mémoire nous avons décrit toutes les fonctionnalités de notre application et les avons expliquées en détail. Néanmoins, notre projet pourra

être amélioré par l'ajout d'autres fonctionnalités tels que :

- la capture d'une vidéo comme un justificatif d'effectuation de réclamations municipales pour les citoyens.
- la capture d'une vidéo comme un justificatif de validation des tâches pour les agents.
- la recherche des communes adéquates en utilisant les services de géolocalisation.
- Les notifications pour tous les acteurs s'il y'a de nouveaux changements.

Il est important à noter que la réalisation de ce projet nous a été très utile, avantageuse et bienfaisante sur tous les plans pédagogiques et techniques. Sur le plan technique, ce projet nous a été une bonne occasion pour découvrir et maîtriser les technologies Angular 7, Ionic 3 et d'approfondir nos connaissances et capacités sur le plan de nouvelles technologies de création d'applications web et d'applications mobiles hybrides, modernes et fabuleuses qui répondent aux besoins du marché, aux besoins des citoyens et des communes. Sur le plan humain, ce projet a été une véritable occasion de vivre l'expérience professionnelle du travail au sein d'une société, ce qui nous a permis d'améliorer nos capacités de communication et de nous initier à la vie professionnelle.



# Bibliographie

.... Auteur, titre du livre, éditeur, année

....

....



# Webographie

1. <https://ionicframework.com/docs/v3>
2. <http://www.communemarsa.tn>
3. <http://www.commune-sfax.gov.tn>
4. <http://www.commune-gabes.gov.tn/ar/>
5. <http://www.tritux.com>
6. <https://iutdoua-web.univ-lyon1.fr/isabelle.goncalves/programmation-web/mvc.html>
7. <https://www.peerbits.com/blog/angular-7-features-and-updates.html>
8. <http://elisabeth.pointal.org/doc/code/javascript/outils/npm>
9. <https://fr.wikipedia.org/wiki/Node.js>
10. <https://desgeeksetdeslettres.com/programmation-java/xampp-plateforme-pour-heberger-son-propre-site-web>
11. <https://www.mysql.com/fr/>
12. <https://fr.wikipedia.org/wiki/StarUML>
13. <http://www.acseo.fr/php7-est-arrive/>
14. [https://fr.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://fr.wikipedia.org/wiki/Visual_Studio_Code)
- 15.

---

# Conception et développement d'une application Web/Mobile de réclamations municipales en ligne

---

Fares Ben Slama  
Wajdi Gabsi

---

## الخلاصة:

الهدف الرئيسي من مشروع نهاية الدراسة هذا هو تصميم وتطبيق موقع ويب / تطبيق للهاتف المحمول لإدارة المطالبات عبر الإنترنت البلدية للعديد من البلديات على منصة واحدة. ويستند تحقيق هذا التطبيق بشكل رئيسي على تقنيات الويب وتقنيات الهاتف المحمول

## Résumé :

L'objectif principal de ce travail de fin d'étude consiste à concevoir et implémenter une application web/mobile pour la gestion des réclamations municipales en ligne de plusieurs communes sur une seule plateforme. La réalisation de cette application est basée principalement sur les technologies Angular et Ionic.

## Abstract:

The main goal of this end-of-course project is to design and implement a web/mobile application for the online management of municipal claims of several municipalities on a single platform. The realization of this application is based primarily on Angular and Ionic technologies.

المفاتيح: تقنيات الهاتف المحمول, تقنيات الويب, مطالبات, موقع ويب / تطبيق للهاتف المحمول

Mots clés: application web/mobile, réclamation municipale, Angular, Ionic.

Key-words: web/mobile application, municipal claims, Angular, Ionic.