

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE



Université de Bejaia

Faculté des Sciences Exactes  
Département Informatique

## Mémoire de Fin de cycle

En vue de l'obtention du diplôme Master académique en Informatique  
Spécialité Génie logiciel

Thème

---

Proposition d'un cadre méthodologique agile-cloud pour le  
développement d'une application web et mobile. Cas d'étude: Gestion  
de crise

---

***Présenté par :***

*Abderrahmane Cylia*

***Encadré par :***

*Dr Karima Ait Abdelouhab*

***Devant le jury composé de :***

*Présidente: Mme. Ait Hacene Souhila*

*Mme. Nouicer Amina*

*Mr. Allem Khaled*

*Mr. Zerarga Loutfi*

Année universitaire 2024-2025

# Dédicaces

Je tiens à dédier ce travail à mes parents, pour leur amour, leurs sacrifices et leur soutien inconditionnel tout au long de mon parcours académique.

A mes sœurs, pour leur encouragement constant et appui infaillible. A mes amies, qui m'ont accompagnés durant cette période et qui ont su croire en moi.

Cylia

# Remerciements

Je tiens, tout d'abord, à remercier mon encadrante, Madame Karima AIT Abdelouhab, qui m'a accompagné tout au long de ce projet, pour ses précieux conseils, sa présence inébranlable et son soutien continu pendant la réalisation de ce mémoire. Je suis honoré d'avoir eu l'opportunité de travailler avec elle et de bénéficier de son expertise.

Je souhaite également exprimer ma plus profonde reconnaissance à mes chers parents qui ont fait de moi qui je suis et qui ont cru en moi depuis le début de cette aventure académique. Leurs mots d'encouragement, si petit soit-ils, ont été ma force et ma motivation.

Je remercie également, tous les enseignants qui ont contribué, de loin ou de près, à ma formation universitaire, sans laquelle ce projet de fin de cycle n'aurait pas vu le jour.

# Table des matières

Table des matières	i
Liste des figures	iii
Table des figures	iii
Liste des tableaux	v
Liste des tableaux	v

## Introduction générale 2

<b>1 Chapitre 1: Généralités sur le Cloud computing et les méthodes agiles</b>	<b>4</b>
1.1 Introduction	4
1.2 Méthodes Agiles	4
1.2.1 Avantages des méthodes agiles	5
1.2.2 Présentation des méthodes agiles	5
1.2.3 Limites du développement agile	8
1.3 Cloud Computing	8
1.3.1 Caractéristiques du Cloud Computing	9
1.3.2 Types de services du cloud computing	9
1.3.3 Modèles de déploiement du cloud computing	10
1.3.4 Avantages du Cloud Computing	11
1.3.5 Défis du cloud computing	11
1.4 Conclusion	12
<b>2 Chapitre 2: État de l'existant et Proposition d'un Framework</b>	<b>14</b>
2.1 Introduction	14
2.2 Étude critique sur les travaux existants	14
2.2.1 Développement logiciel agile utilisant le cloud computing: Une étude de cas	14
2.2.2 Méthodes agiles pour le cloud computing	15
2.2.3 Développement de produit logiciel basé sur l'agile utilisant les services du cloud computing: Résultats d'une étude de cas	16
2.2.4 Implémentation de Scrum dans le développement logiciel agile basé sur le cloud	17
2.2.5 Accentuation sur les tests d'assimilation en utilisant le Cloud	18
2.3 Étude Comparative	18
2.4 Proposition et description d'un cadre méthodologique (Framework) agile-cloud	19
2.4.1 Description globale du framework proposé	19
2.4.2 Présentation des acteurs du framework	20
2.4.3 Description détaillée des différentes phases du framework	21
2.5 Conclusion	21
<b>3 Chapitre 3: Validation de la proposition par une étude de cas</b>	<b>24</b>
3.1 Introduction	24
3.2 Définition de gestion de crise	24
3.3 Types de crise	26
3.4 Gestion de crise en Algérie	26
3.4.1 Scénario Typique d'une gestion de crise en Algérie	27
3.5 Outils de développement	28
3.6 Langages et frameworks de programmation	29
3.7 Mise en application du framework proposé	30
3.7.1 Pré-phase	31
3.7.2 Phase de développement	35

	Réalisation du sprint n°01 . . . . .	35
3.7.3	Post-phase . . . . .	48
	Réalisation du sprint n°02 . . . . .	48
3.7.4	Post-phase . . . . .	57
	Réalisation du sprint n°03 . . . . .	58
3.7.5	Post-phase . . . . .	65
	Réalisation du sprint n°04 . . . . .	65
3.7.6	Post-phase . . . . .	69
3.8	Conclusion . . . . .	70

**Conclusion générale et perspectives** **72**

**Références** **77**

## Table des figures

1	Processus Scrum . . . . .	6
2	Processus XP . . . . .	7
3	Kanban Board . . . . .	7
4	Définition du cloud computing du NIST . . . . .	8
5	Type de services du cloud computing . . . . .	10
6	Modèles de déploiement du cloud computing . . . . .	11
7	Procédure d’implémentation pour la conception du Framework . . . . .	14
8	Développement agile basé sur les services cloud . . . . .	16
9	Le Modele TAAS . . . . .	18
10	Framework SBCC . . . . .	20
11	Phases de crises . . . . .	25
12	Diagramme de cas d’utilisation globale . . . . .	32
13	Sprint 1 Backlog . . . . .	34
14	Sprint 2 Backlog . . . . .	34
15	Sprint 3 Backlog . . . . .	34
16	Sprint 4 Backlog . . . . .	35
17	Diagramme de cas d’utilisation du sprint 1 . . . . .	35
18	Diagramme de cas d’utilisation “Inscription” . . . . .	38
19	Diagramme de cas d’utilisation “Connexion” . . . . .	39
20	Diagramme de séquence “Signalement d’incident” . . . . .	40
21	Diagramme de séquence “Gestion des utilisateurs” . . . . .	41
22	Diagramme de séquence détaillé de “Inscription” . . . . .	42
23	Diagramme de séquence détaillé de “Connexion” . . . . .	42
24	Diagramme de séquence détaillé de “Signalement d’incidents” . . . . .	43
25	Diagramme de séquence détaillé de “Ajouter un utilisateur” . . . . .	43
26	Diagramme de séquence détaillé de “Modifier un utilisateur” . . . . .	44
27	Diagramme de séquence détaillé de “Supprimer un utilisateur” . . . . .	44
28	Diagramme de séquence détaillé de “Suspendre un utilisateur” . . . . .	45
29	Diagramme de classe du sprint 1 . . . . .	45
30	Interfaces de Connexion de l’administrateur . . . . .	46
31	Interfaces de Inscription des citoyens . . . . .	46
32	Interfaces de Connexion des utilisateurs de l’application mobile . . . . .	46
33	Interfaces de Signalement d’incident . . . . .	47
34	Interfaces de Gestion des utilisateurs citoyens . . . . .	47
35	Interfaces de Gestion des utilisateurs agents de terrain . . . . .	47
36	Interfaces de Gestion des utilisateurs responsables . . . . .	48
37	Interfaces de Gestion des administrateurs . . . . .	48
38	Diagramme de cas d’utilisation du sprint 2 . . . . .	49
39	Diagramme de séquence “Gestion des ressources” . . . . .	51
40	Diagramme de séquence “Notifier les utilisateurs” . . . . .	51
41	Diagramme de séquence “Valider le rapport d’incident” . . . . .	52
42	Diagramme de séquence détaillé de “Ajouter une ressource” . . . . .	53
43	Diagramme de séquence “Modifier une ressource” . . . . .	53
44	Diagramme de séquence “Supprimer unes ressource” . . . . .	54
45	Diagramme de séquence détaillé “Notifier les utilisateurs” . . . . .	54
46	Diagramme de séquence détaillé de “Valider le rapport d’incident” . . . . .	55
47	Diagramme de classe du sprint 2 . . . . .	55
48	Interfaces de Gestion des ressources agents . . . . .	56
49	Interfaces de Gestion des infrastructures . . . . .	56
50	Interfaces de Gestion des ressources responsables . . . . .	56
51	Interfaces de Gestion des ressources matérielles . . . . .	56
52	Interfaces de Notifications . . . . .	57
53	Interfaces de validation des rapports . . . . .	57

54	Diagramme de cas d'utilisation du sprint 3 . . . . .	58
55	Diagramme de séquence de "Localisation des agents" . . . . .	61
56	Diagramme de séquence de "Localisation des zones de crise" . . . . .	61
57	Diagramme de séquence de "Coordination des secours" . . . . .	61
58	Diagramme de séquence de "Gestion des missions" . . . . .	62
59	Diagramme de séquence détaillé de "Localisation des agents" . . . . .	62
60	Diagramme de séquence détaillé de "Localisation des zones de crises " . . . . .	62
61	Diagramme de séquence détaillé de "Coordination des secours" . . . . .	63
62	Diagramme de séquence détaillé de "Gestion des missions" . . . . .	63
63	Diagramme de classe du sprint 3 . . . . .	64
64	Système de géolocalisation . . . . .	64
65	Gestion des missions . . . . .	65
66	Coordination des secours . . . . .	65
67	Diagramme de cas d'utilisation du sprint 4 . . . . .	66
68	Diagramme de séquence de "Archivage des crises" . . . . .	67
69	Diagramme de séquence de "Publication d'annonces et de communiqués" . . . . .	67
70	Diagramme de séquence détaillé de "Archivage des crises" . . . . .	68
71	Diagramme de séquence détaillé de "Publication d'annonces et de communiqués" . . . . .	68
72	Diagramme de classe du sprint 4 . . . . .	69
73	Publication d'annonces . . . . .	70
74	Archivage des crises . . . . .	70
75	Diagramme de classe . . . . .	74

## Liste des tableaux

1	Scénarios de réalisation du projet . . . . .	15
2	Étude comparative des travaux . . . . .	19
3	Rôles des parties prenantes du projets . . . . .	31
4	Product Backlog . . . . .	33
5	Planification des sprints . . . . .	34
6	Description textuelle du cas d'utilisation "S'inscrire" . . . . .	36
7	Description textuelle du cas d'utilisation "Connexion" . . . . .	36
8	Description textuelle du cas d'utilisation "Signaler un incident" . . . . .	37
9	Sprint 4 Backlog . . . . .	37
10	Description textuelle du cas d'utilisation "Gestion des ressources" . . . . .	49
11	Description textuelle du cas d'utilisation "Notification des utilisateurs" . . . . .	50
12	Description textuelle du cas d'utilisation "Validation des rapports d'incident" . . . . .	50
13	Description textuelle du cas d'utilisation "Géolocalisation des agents." . . . . .	59
14	Description textuelle du cas d'utilisation "Géolocalisation des crises" . . . . .	59
15	Description textuelle du cas d'utilisation "Coordination des secours" . . . . .	60
16	Description textuelle du cas d'utilisation "Gestion des mission" . . . . .	60
17	Description textuelle du cas d'utilisation "Archiver une crise" . . . . .	66
18	Description textuelle du cas d'utilisation "Publier annonces" . . . . .	66

# Introduction générale

# Introduction générale

Dans un monde en constante évolution, le domaine de la technologie continue de progresser à un rythme fulgurant ne cessant de s'ancrer toujours plus profondément dans la vie quotidienne de l'être humain. Des avancées spectaculaires qui continuent de transformer notre manière de concevoir et de développer des solutions logicielles évolutives et réactives aux besoins et attentes des utilisateurs.

D'une part, la demande de développement de services et les exigences des clients qui continue de s'accroître et d'évoluer a conduit à l'émergence de l'approche agile. Des méthodologies de développement itératives et collaboratives assurant une adaptabilité continue aux exigences changeantes des clients.

D'autre part, ces dernières années ont vu émerger plusieurs technologies innovantes notamment le Cloud Computing. Une technologie qui a révolutionné l'accès aux infrastructures informatiques en offrant des ressources évolutives, accessibles en temps réel et sans contraintes matérielles majeures transformant la manière dont les entreprises et particuliers opèrent.

De plus, depuis toujours, le monde est confronté à une multitude de crises au quotidien, qu'elles soient sanitaires, environnementales ou sécuritaires. Parmi elles, les catastrophes naturelles se distinguent par leurs impacts souvent dévastateurs sur les populations et les infrastructures. Avec le progrès rapide de la société et des enjeux globaux, l'ampleur de ces crises ne cesse de croître particulièrement en Algérie qui n'est pas une exception. Cependant, cette même évolution a également permis à la technologie d'être un atout majeur dans la mise en œuvre de solutions logicielles permettant une meilleure coordination, réactivité et adaptabilité aux situations d'urgence.

Ainsi, le présent projet de fin de cycle a pour objectif de développer une application mobile et web permettant d'informatiser la gestion de crises en Algérie, spécifiquement les catastrophes naturelles, en faisant usage de la conjonction du développement agile avec le cloud computing.

Ce mémoire est organisé comme suit:

Le premier chapitre introduit les concepts fondamentaux des méthodes agiles et du cloud computing.

Le deuxième chapitre aborde les travaux existants qui traitent la notion du développement agile basé sur le cloud computing, soutenue sur une étude comparative de ces derniers dans le but de proposer notre propre cadre méthodologique agile-cloud à la fin du chapitre.

Le dernier chapitre se porte sur l'adoption et validation du Framework précédemment proposé par une étude de cas qui consiste à la mise en œuvre d'une application de gestion de crise.

# Chapitre 1: Généralités sur le Cloud computing et les méthodes agiles

# 1 Chapitre 1: Généralités sur le Cloud computing et les méthodes agiles

## 1.1 Introduction

Le développement logiciel a connu de nombreux changements au fil des années et reste, à ce jour, en constante évolution. Tout d'abord avec l'émergence de l'approche agile qui a su répondre aux nombreuses limitations des méthodes traditionnelles, grâce à ses cycles courts itératives et collaboratives permettant de livrer des produits de qualité et adaptés aux besoins changeants des clients et utilisateurs.

Plus récemment, le cloud computing, à son tour, s'impose en tant que technologie clé offrant un accès à des infrastructures évolutives et disponibles à la demande permettant de propulser le processus de développement logiciel.

Ce chapitre vise à introduire ces deux notions, en mettant l'accent, à la fois, sur leurs contributions au processus de développement logiciels et sur les limites qu'elles présentent.

## 1.2 Méthodes Agiles

Avec l'évolution de la technologie, le développement agile émerge dans les années 1990 comme une solution optimale aux limitations des approches de développement traditionnelles de gestion de projet liées à leurs rigidité et lourdeur d'exécution. De nombreux praticiens de ces méthodes classiques se sont retrouvés face à des étapes initiales de développement frustrantes, même, irréalisables, notamment lorsque les besoins n'étaient pas clairement définis dès le départ. Dans une industrie en constant progrès où les attentes des clients changent rapidement et sont parfois difficiles à exprimer avec précision, ces méthodes traditionnelles conduisaient fréquemment à l'insatisfaction des utilisateurs finaux. Ces difficultés ont poussé les développeurs à aborder une nouvelle approche de développement conçue pour répondre efficacement à ces problématiques.

Les méthodes agiles reposent sur une approche itérative et incrémentale mettant l'accent sur la flexibilité, la collaboration et la communication continue entre les membres de l'équipe de développement et les clients. D'après (Beck et al, 2001) Cette approche vise à valoriser les individus et leurs interaction plus que les processus et outils, les logiciels opérationnelles plus qu'une documentation compréhensive et exhaustif, la collaboration avec les clients plus que les négociations contractuelles et la réactivité aux changements plus que le suivi d'un plan. Les auteurs (Beck et al, 2001) propose 12 principes renforçant l'importance de l'agilité du développement logiciel:

- Prioriser la satisfaction du client
- Accepter les changements
- Livrer en permanence des versions opérationnelles du produit
- Assurer, le plus souvent possible, une coopération entre l'équipe du projet et les gens du métier
- Construire les projets autour de personnes motivées
- Favoriser le dialogue direct
- Mesurer l'avancement du projet en fonction de l'opérationnalité du produit
- Adopter un rythme de travail constant et soutenable par tous les intervenants du projet
- Contrôler continuellement l'excellence de la conception et la bonne qualité technique
- Privilégier la simplicité en évitant le travail inutile
- Auto-organiser et responsabiliser les équipes
- Améliorer régulièrement l'efficacité de l'équipe en ajustant son comportement

### 1.2.1 Avantages des méthodes agiles

L'approche agile rapporte une multitude d'avantages, tels que:

- Une meilleure adaptabilité aux changements des besoins des clients.
- Des retours et feedbacks rapide du client grâce à des livraisons fréquentes du produit.
- La livraison de produit de bonne qualité satisfaisant les exigences et attentes des clients.
- Une meilleure collaboration avec transparence au sein de l'équipe pour un meilleur flux de travail.

### 1.2.2 Présentation des méthodes agiles

Au fil des années, l'approche agile a connue l'émergence de plusieurs méthodes . Parmi les plus connues et utilisées, on retrouve:

#### • La méthode Scrum

Scrum [10] est un cadre méthodologique de travail agile principalement utilisé dans le développement logiciel. Il vise à améliorer la productivité des équipes de développement ainsi que la qualité des produits livrés, en se reposant sur un processus incrémental et itératif favorisant la transparence, la collaboration et des retours réguliers.

L'équipe Scrum repose sur ces rôles clés:

- Le product owner qui a pour responsabilité de définir les exigences du produit et leurs priorités.
- Le scrum master qui se charge de garantir et faciliter l'application des règles Scrum.
- L'équipe de développement auto-organisée et multidisciplinaire qui se charge de développer les fonctionnalités du produit et de livrer des incréments à la fin de chaque sprint.

Les artefacts sur lesquels repose le développement d'un logiciel sont les suivants:

- Le product backlog qui représente la liste des fonctionnalités proposées à développer pour le produit.
- Le sprint backlog qui constitue une liste des tâches à accomplir pendant le sprint en cours.
- L'incrément qui est une version fonctionnelle et opérationnelle du produit livrée à la fin de chaque sprint.

Le cycle de développement est, également, renforcé par des réunions régulières énumérés ci-dessous:

- Le sprint planning qui vise à planifier le flux de travail du sprint.
- Les daily stand ups qui s'agit d'une réunion quotidienne d'environ 15 minutes afin de suivre l'avancée du travail de chaque membre d'équipe.
- Le sprint review qui consiste en une démonstration du travail accomplie lors du sprint avec le client exprimant ses retours.
- Le sprint retrospective où l'équipe agile évalue le flux du travail pendant le sprint.

La figure 1 représente la méthode agile Scrum.

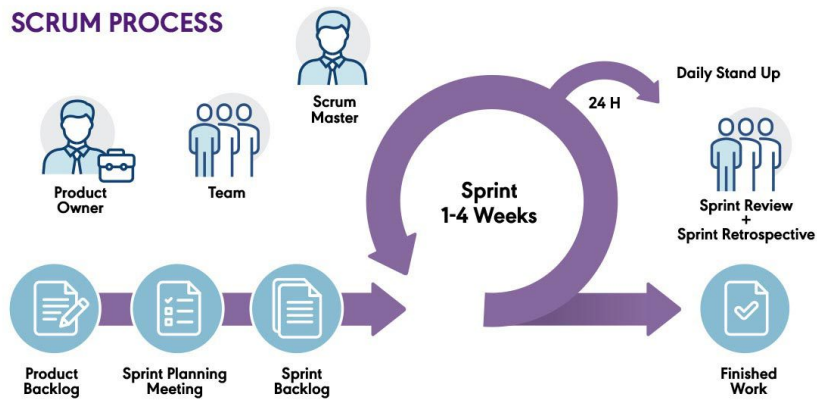


FIG. 1 – *Processus Scrum*

- **La méthode Extreme Programming (XP)**

XP [12] est une méthode agile de développement logiciel orientée objet qui pousse à l'extrême les principes et valeurs du développement agile. Un framework qui a pour objectif d'optimiser l'adaptabilité et réactivité à l'évolution dynamique des exigences, à travers une constante communication avec le client et des cycles de développement de courte durée afin de produire des logiciels de meilleure qualité et un meilleur flux de travail pour les équipes de développement.

Les auteurs de [25] (SHARP et al, 2004) ont repris la liste des douze pratiques annotées par Beck (2000, p. 54) comme suit:

- Le jeu de planification qui détermine la portée de la prochaine livraison.
- Livraison rapide et régulière de versions fonctionnelles du produit.
- Une métaphore qui sert de guide de développement sur le fonctionnement du système.
- Une conception design simple du système.
- Pair programming qui implique la programmation du code en binôme.
- Intégration continue du système quotidiennement à la fin de chaque tâche.
- Des tests unitaires qui doivent s'exécuter parfaitement pour une bonne continuation du développement.
- Refactoring qui consiste à, continuellement, améliorer le code sans changer son comportement.
- Propriété collective permettant à toute partie prenante dans le processus de développement d'apporter des changements dans le système, à volonté.
- Un période de travaille ne dépassant pas quarante heures par semaine.
- Communication constante avec le client
- Le respect des normes de programmation soulignant la communication à travers le code.

Le processus XP est représenté par la figure 2.

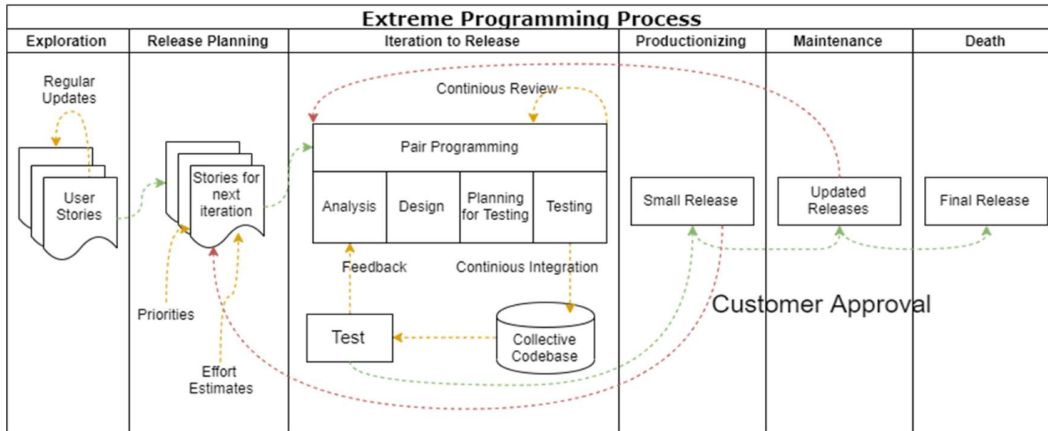


FIG. 2 – *Processus XP*

• **La méthode Kanban**

Kanban est un framework[11] de développement logiciel agile permettant de visualiser et gérer le flux de travail, à travers un tableau (Kanban board), suivre l'avancement des tâches en temps réel et identifier les blocages tout en limitant le travail en cours (Work In Progress Limit). Il repose sur un travail effectué en toute clarté et transparence, une communication en temps réel, une réactivité aux éventuels blocages, une flexibilité par rapport aux tâches et une amélioration continue du flux de production.

Le Kanban board est composé de trois colonnes principales:

- Colonne TO DO (à faire) qui consiste d'une liste de tâches à accomplir.
- Colonne In progress (en cours) qui liste les tâches en train d'être réalisées.
- Colonne Done (terminé) qui affiche les tâches achevées et prêtes à être livrées.

Des colonnes supplémentaires peuvent être ajoutées au Kanban Board pour un meilleur suivi du flux de travail, comme une colonne TEST qui affiche les tâches en attente de validation.

Cette méthode agile est illustré à travers la figure 3.

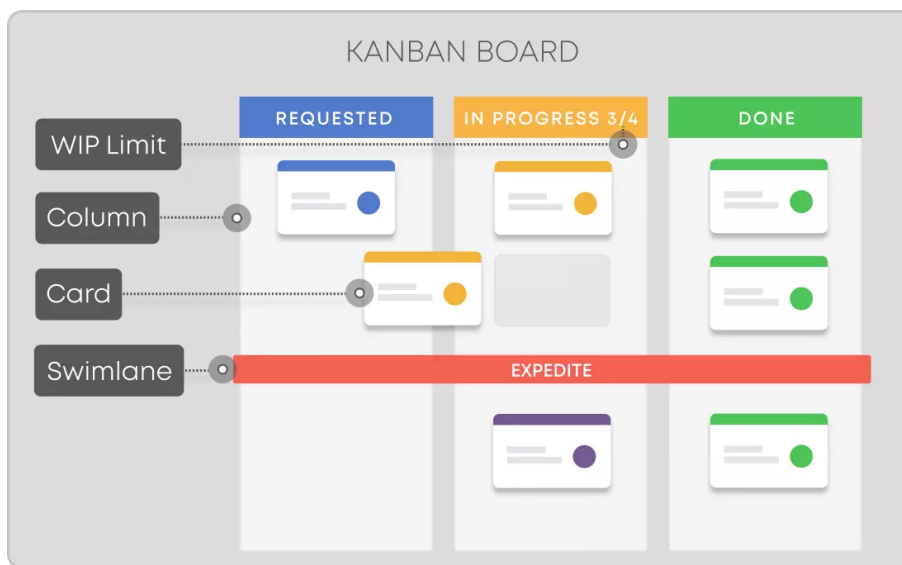


FIG. 3 – *Kanban Board*

### 1.2.3 Limites du développement agile

L'approche agile peut se heurter à certaines contraintes lors de sa mise en œuvre dans un contexte réel, notamment:

- La scalabilité qui rend compliqué l'applicabilité de l'agilité à grande échelle, en particulier dans des environnements complexes et distribués.
- Manque de transparence et de clarté pour les parties prenantes.
- Difficulté de maintenir une communication face-à-face avec des équipes réparties géographiquement.
- Capacité à gérer un projet à partir de localisations distribuées pouvant affecter la coordination et cohésion entre les membres de l'équipe.
- Documentation relative aux fonctionnalités du produit est souvent peu détaillée, ce qui peut être problématique pour la maintenance, les audits et l'intégration de nouveaux membres.
- Difficulté de prévoir correctement le coût, le temps et les ressources nécessaires à la réussite du projet.
- Nécessite une forte implication du client, ce qui n'est pas toujours faisable.

## 1.3 Cloud Computing

D'après l'institut national des normes et de la technology (NIST), le cloud computing est défini comme un modèle permettant un accès réseau omniprésent, pratique et à la demande, à un ensemble partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services), qui peuvent être rapidement provisionnées et libérées avec un minimum d'effort de gestion ou d'interaction avec le fournisseur de services comme illustré dans la figure 4. D'après (Mell et al, 2011):

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources... that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

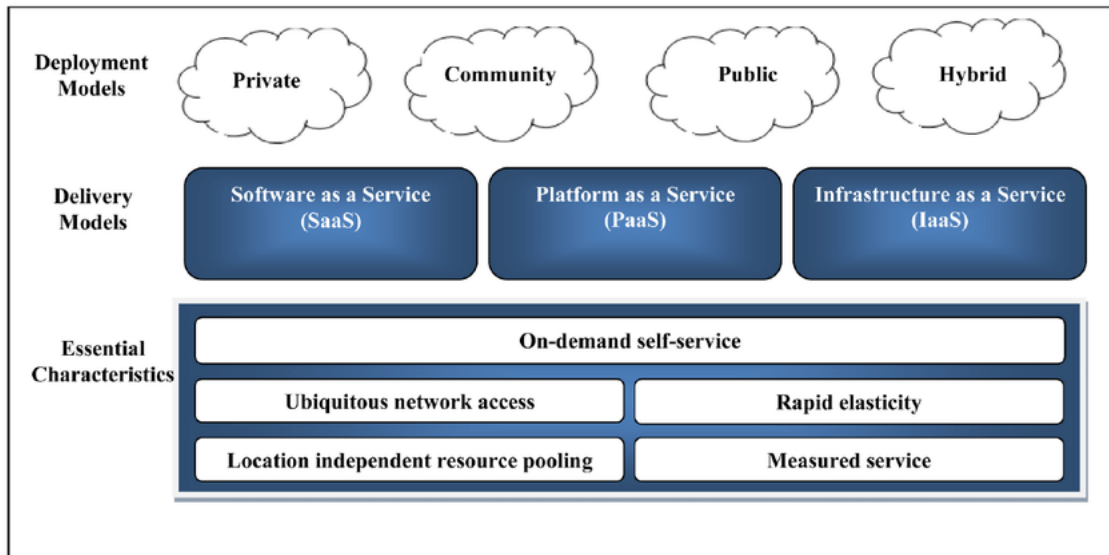


FIG. 4 – Définition du cloud computing du NIST

Le concept du cloud computing remonte aux années 1950 avec l'apparition des ordinateurs centraux à grande échelle. Pour une utilisation efficace de ces dernières, la pratique du partage du temps et de la mise en commun des ressources a évolué en utilisant les terminaux passifs pour faciliter l'accès aux ordinateurs centraux. Toutefois, l'émergence des systèmes d'exploitation appelé 'machine virtuelle' dans les années 1970 a été un tournant important

dans l'informatique. Cette avancée a permis aux ordinateurs centraux d'avoir plusieurs machines virtuelles sur un seul nœud physique permettant à plusieurs environnements informatiques distincts d'exister sur le même matériel physique.

Au fil des années, différentes technologies ont pu s'améliorer, se partager et livrer des ressources d'une manière fiable ce qui a poussé certaines entreprises à rendre les services cloud accessibles au public avec paiement à la demande.

Aujourd'hui, le cloud computing est devenu une technologie pionnière dans le domaine de l'informatique. Une suite de services pilotés par des APIs (application programming interface), accessible via un réseau de télécommunication, détenant une puissance de calcul capable de gérer tous les aspects des ressources informatiques (réseaux, serveurs, stockage, application et services) avec plus de flexibilité et d'évolutivité.

### 1.3.1 Caractéristiques du Cloud Computing

Le cloud computing repose sur cinq caractéristiques essentielles énumérées par (Mell et al, 2011):

- A la demande et libre-service (en anglais, 'On-demand self-service'): Les ressources cloud restent disponibles à tout moment à l'exception de pannes ou de failles de sécurité
- Large accès par réseau (en anglais, 'Broad network access'): Les ressources cloud sont accessibles à travers le réseau via une multitude d'appareils. L'accès Internet est suffisant pour les services cloud publics, toutefois, un accès Intranet est requis pour les services cloud privés.
- Mise en commun des ressources (en anglais, 'Resource pooling'): Le modèle cloud est partagé entre multiples clients qui sont assignés et affectés dynamiquement sans prendre en compte de la location.
- Élasticité rapide (en anglais, 'Rapid elasticity'): Les ressources cloud mises à disposition s'adaptent automatiquement et rapidement en fonction de la demande en temps réel grâce à une scalabilité verticale - qui consiste à augmenter ou diminuer la capacité d'un serveur en modifiant ses caractéristiques matérielles- et une scalabilité horizontale - qui consiste à ajouter ou retirer des serveurs pour répartir la charge de travail entre plusieurs machines.
- Service mesuré (en anglais, 'Measured service'): Les clients paient pour ce qu'ils utilisent ou réservent (paiement à l'usage). Cependant cela n'est pas applicable pour certains services tels que les services de messagerie (Gmail, hotmail,etc), les réseaux sociaux (Facebook, Twitter,etc) et les services cloud à titre d'essai (AWS, Azure,etc).

### 1.3.2 Types de services du cloud computing

Le cloud computing propose trois modèles de services décrits par (Mell et al, 2011):

- IaaS (Infrastructure as a service): Fournit aux utilisateurs des ressources informatiques (serveurs, stockage,etc) virtualisées et évolutives via Internet. Exemples: AWS EC2, Google Compute Engine.
- PaaS (Platform as a service): Offre un environnement de développement complet facilitant le développement, le test, le déploiement et la gestion des applications sans acquérir, gérer ou bien maintenir les architectures qui y sont liées. Exemples: Google App Engine, Microsoft Azure, Heroku.
- SaaS (Software as a service): Propose des services cloud permettant aux utilisateurs d'héberger et de gérer les applications logicielles sans avoir besoin de matérielle informatique. Exemples: Gmail, Microsoft 365, Salesforce,etc.

La figure 5 est une illustrations de ces modèles de services, mettant l'accent sur les aspects gérés par les utilisateurs et ceux gérés par les fournisseurs de cloud.

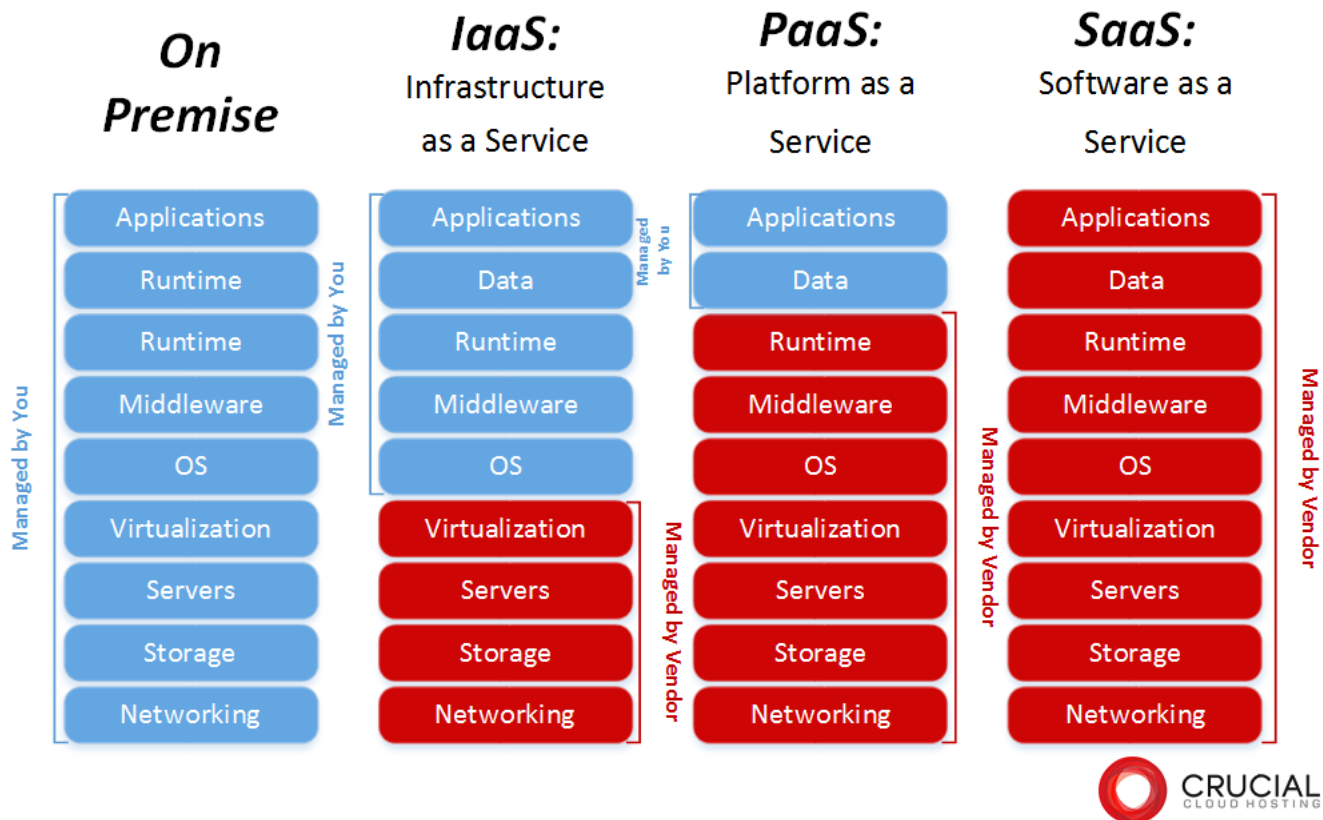


FIG. 5 – Type de services du cloud computing

### 1.3.3 Modèles de déploiement du cloud computing

Le cloud computing offre quatre modèles de déploiement comme cités par (Mell et al, 2011):

- Cloud public: Les infrastructures cloud - gérées par les fournisseurs de service cloud- sont accessibles pour le public via l'Internet avec une évolutivité élevée, de faibles coûts et un accès simple et rapide aux ressources.
- Cloud privé: L'infrastructure cloud est opérée par une seule organisation. Elle peut être gérée par cette organisation ou par une partie tiers assurant un contrôle total sur les données et la sécurité. En faisant, un modèle adapté aux entreprises avec des exigences strictes en matière de conformité.
- Cloud hybride: Une combinaison du cloud privé et du cloud public permettant de bénéficier à la fois de la flexibilité du cloud public et du contrôle du cloud privé. Ainsi, optimisant les coûts tout en conservant la sécurité pour les données sensibles.
- Cloud communautaire: Une infrastructure cloud mise en place pour une communauté spécifique d'organisations qui partagent des intérêts communs et qui peut être détenue, gérée et exploitée par une ou plusieurs organisations de la communauté.

La figure 6 représente ces quatre modèles de déploiement.

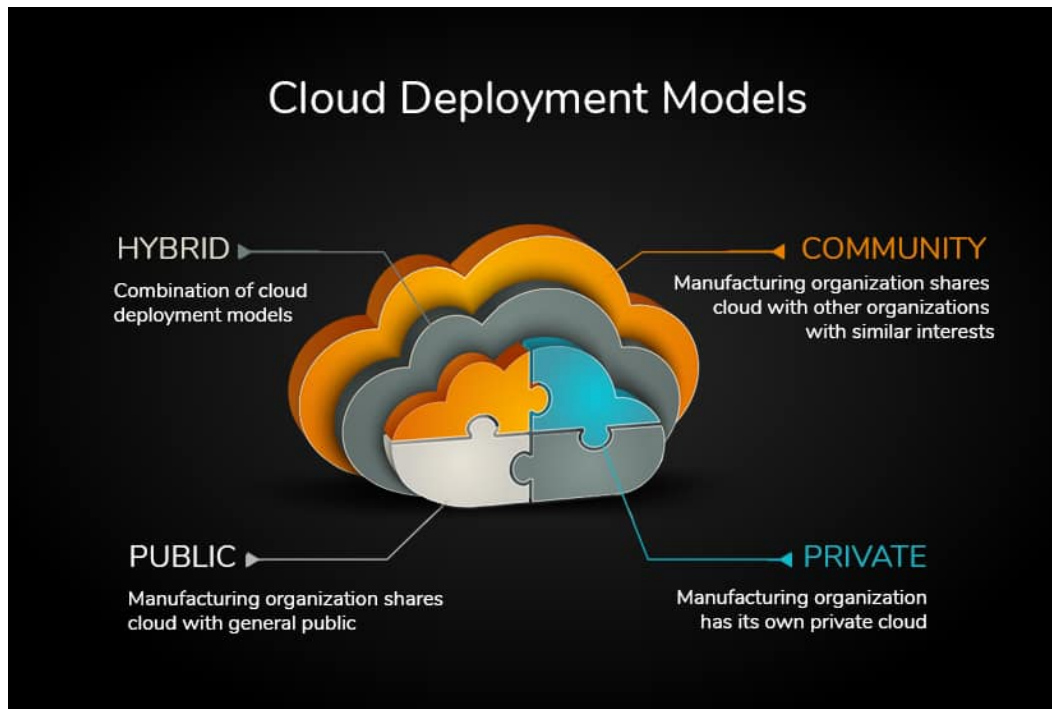


FIG. 6 – Modèles de déploiement du cloud computing

### 1.3.4 Avantages du Cloud Computing

Le cloud computing offre les avantages clés suivants[9]:

- Flexibilité: Des outils et fonctionnalités prédéfinis avec une adaptation à la demande et la possibilité de déterminer le niveau de contrôle tout en garantissant la sécurité des données.
- Efficacité: Permet de mettre les applications rapidement sur le marché sans se soucier des coûts d'infrastructure ou de maintenance garantissant une accessibilité depuis n'importe quel appareil connecté à Internet tout en conservant les données grâce aux sauvegardes maintenues sur le réseau.
- Valeur stratégique: Le cloud computing met à disposition des technologies innovantes tout en assurant la gestion de l'infrastructure conférant un avantage concurrentiel.

### 1.3.5 Défis du cloud computing

Les auteurs de [9] évoquent, également, un certain nombre de difficultés quand à l'adoption du cloud computing:

- La sécurité des données liée à la perte ou l'indisponibilité des données. Effectivement, l'utilisation du cloud expose les données à de nombreux risques, notamment la perte et le vol lors des transferts ou à cause de failles chez les fournisseurs.
- Les problèmes de gouvernance et de souveraineté. Les données peuvent être sujet de contrôle juridique pouvant limiter l'usage du cloud.
- Les problèmes de conformité pouvant émerger d'où à l'obligation de respecter des réglementations et normes strictes comme le RGPD (Règlement Général sur la Protection des données et HIPAA (Health Insurance Portability and Accountability Act).
- Le manque de standardisation dans la manière dont les technologies en constante évolution s'intègrent et interagissent, rendant l'intégration difficile pour les systèmes existants.

- Le manque de contrôle direct sur les couches physiques de l'infrastructure ce qui limite la visibilité sur les performances, la sécurité et la localisation des données.
- La dépendance au fournisseur de services cloud complique le passage à d'autres fournisseurs en raison de l'incompatibilité des services ou des données. Ce qui peut ralentir l'innovation ou augmenter les coûts.
- La dépendance sur une connexion Internet stable car une panne réseau. En cas de panne Internet, les services et ressources deviennent inaccessibles.

## 1.4 Conclusion

Dans ce chapitre, nous avons présenté les principes fondamentaux du cloud computing et du développement agile, en mettant en évidence les avantages qu'ils apportent au processus de développement logiciel. Cependant, ce dernier continue de faire face à diverses problématiques. Ces difficultés peuvent être adressées en étendant les pratiques agiles grâce aux nombreux services offerts par le cloud computing. Cette thématique fera l'objet de l'analyse du prochain chapitre.

## Chapitre 2: État de l'existant et Proposition d'un Framework

## 2 Chapitre 2: État de l'existant et Proposition d'un Framework

### 2.1 Introduction

Le développement logiciel agile repose sur un nombre de principes qui ont réussi à répondre aux besoins rapides et évolutives de l'industrie au fil des années. Une industrie qui continue d'évoluer à travers l'adoption de nouvelles technologies, notamment le cloud computing qui offre une multitude de services permettant de réduire les coûts et d'améliorer les performances du développement logiciel.

Dans ce chapitre, nous allons étudier des articles scientifiques qui ont exploré la notion du développement agile basé sur le cloud computing. Une étude comparative de ces documents nous permettrait d'établir une méthodologie agile-cloud qui sera appliquée dans le développement d'une application web et mobile.

### 2.2 Étude critique sur les travaux existants

Dans cette section, nous allons analyser un certain nombre de travaux dont les auteurs ont évalué la conjonction des pratiques agiles avec le cloud computing.

#### 2.2.1 Développement logiciel agile utilisant le cloud computing: Une étude de cas

À travers cette étude réalisée par (Younas et al, 2019), les auteurs réalisent une étude de cas afin d'évaluer un cadre méthodologique associant le Développement Agile et le cloud computing (ADCC – Agile Development and Cloud Computing) proposé dans une étude antérieure, comme reflété la figure 7.

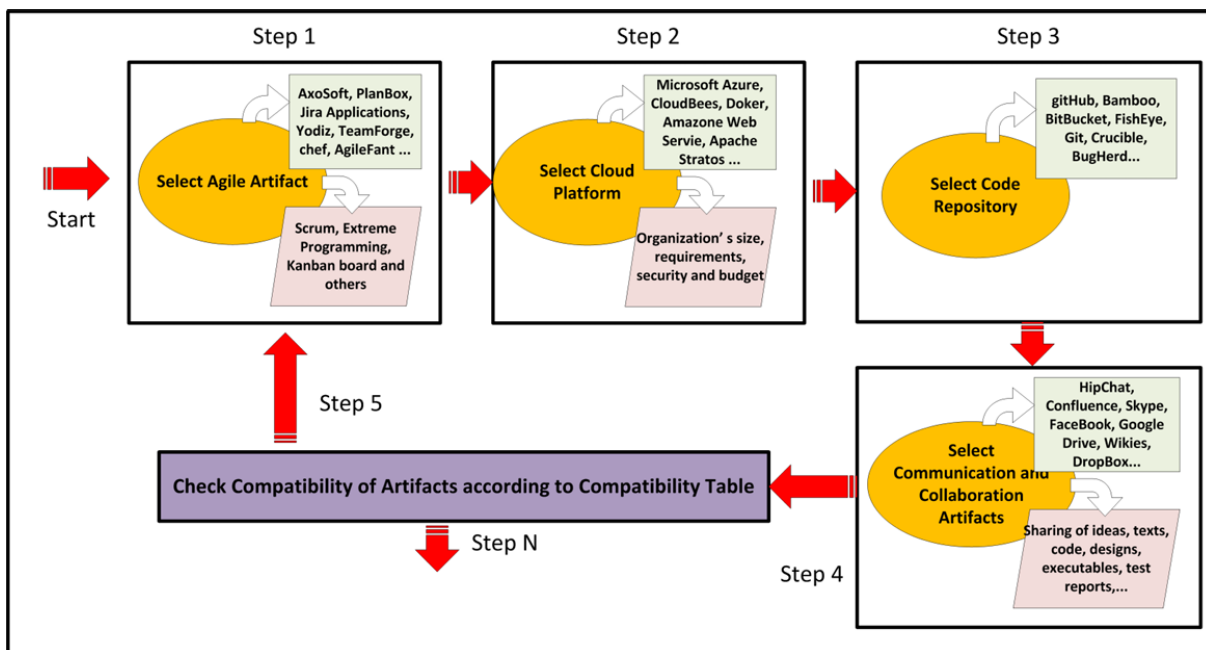


FIG. 7 – Procédure d'implémentation pour la conception du Framework

L'étude évalue le développement logiciel d'un système de gestion hospitalière conçu pour remplacer un système manuel et basé sur le papier, à la fois dans un environnement local et distribué. Le projet est réalisé selon quatre scénarios par quatre équipes consistantes de 4 étudiants provenant de deux universités (GCUF, Pakistan & UTM, Malaysia) qui ont été formés sur le Framework ADCC, comme illustré dans le tableau 1.

Utilisant	Développement logiciel agile simple	framework ADCC
Environnement local	Scénario 1	Scénario 2
Environnement distribué	Scénario 3	Scénario 4

TAB. 1 – Scénarios de réalisation du projet

Dans chaque environnement, le processus de développement est évalué à travers une comparaison entre l'utilisation des méthodes agiles classiques et l'utilisation du ADCC framework. Dans chaque scénario, une application web est développée, en 6 versions (releases), et évaluée selon le nombre de jours nécessaires pour achever les trois phases de développement :

- Recueil des besoins
- Planification, conception et programmation
- Tests et déploiement

À travers cette étude de cas, les auteurs continuellement soulignent la façon dans lequel les services cloud permettent l'accès aux ressources nécessaires pour améliorer le développement logiciel agile en offrant:

- Des compilations (builds) plus rapides
- Un traitement en parallèle
- Une communication en temps réel
- Un retour d'information plus rapide et plus efficace
- Un système de gestion des versions permettant ainsi de tenir les membres de l'équipe informés des avancées et des changements

Cependant, cette étude se concentre uniquement sur Scrum et n'inclut pas d'autres méthodologies agiles qui pourraient interagir différemment avec les technologies cloud, rendant ainsi difficile la généralisation des résultats. De plus, bien que l'informatique en nuage améliore divers aspects du développement agile, les inconvénients potentiels ne sont pas abordés comme les risques de sécurité.

### 2.2.2 Méthodes agiles pour le cloud computing

Cet article concrétisé par (Kalem et al, 2013) a été réalisé dans le but de mettre en évidence les avantages de l'utilisation des méthodes agiles en conjonction avec l'informatique en nuage, à travers une étude de cas portant sur une application de gestion d'entrepôt. Le projet est mené au sein de l'entreprise Sprint – une organisation spécialisée dans le développement logiciel et ayant effectué une transition vers le cloud computing – selon deux scénarios : le premier utilisant uniquement les méthodes agiles, et le second combinant les méthodes agiles avec l'informatique en nuage via Google App Engine. Dans les deux cas, la méthode DSDM (Dynamic Systems Development Method) est adoptée. L'application est développée par une équipe de 4 étudiants au cours de 3 itérations, et les processus de développement des deux scénarios sont comparés.

À travers cette étude de cas, les auteurs démontrent de manière efficace les avantages et les améliorations du

développement logiciel en reliant les méthodes agiles au cloud computing, tels que :

- La mise à disposition de l'infrastructure nécessaire (logicielle et matérielle)
- L'amélioration de la communication et de la collaboration grâce à des outils comme Skype, Google Docs et Google Forms.
- La réduction des longs cycles de retour d'information grâce à l'accès à l'application depuis n'importe quel endroit disposant d'une connexion Internet.
- La diminution des retards dans les phases de test et de déploiement grâce au traitement en parallèle.

Cet article met en lumière les bénéfices du lien entre développement agile et cloud computing, bien qu'il manque de validation sur des projets plus vastes et multiples. De plus, les auteurs n'ont pas abordé les défis potentiels liés à l'adoption des technologies cloud.

### 2.2.3 Développement de produit logiciel basé sur l'agile utilisant les services du cloud computing: Résultats d'une étude de cas

Dans cet article, (Zarinah et Kasirun, 2013) cherchent à examiner comment le cloud computing améliore l'efficacité du développement logiciel basé sur les méthodes agiles, ainsi que la collaboration entre les équipes agiles. Dans cette étude de cas, un projet de sécurité mobile sur Internet est mené au sein de l'organisation Alpha – une organisation internationale de développement de produits – dans un environnement distribué, en utilisant Scrum comme méthodologie agile, et CloudForge (PaaS) comme service cloud. L'application est développée par une équipe de 9 membres : 5 ingénieurs logiciels, 1 ingénieur qualité, 1 chef de produit, 1 propriétaire de produit et 1 designer UX (4 membres dans le bureau en Asie et 5 dans le bureau en Europe). La figure 8 est une illustration de cela.

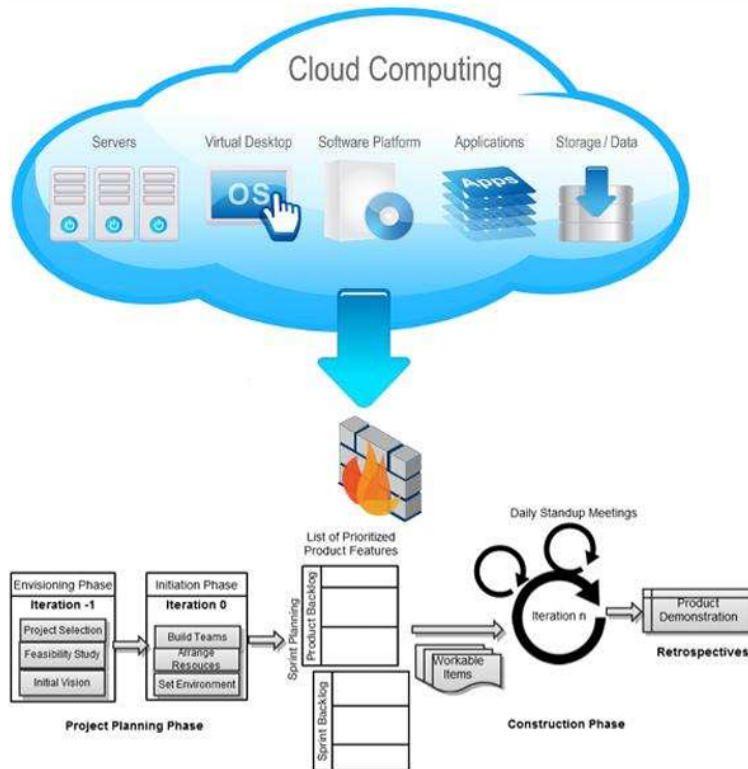


FIG. 8 – Développement agile basé sur les services cloud

Cette étude renforce l'argument selon lequel le cloud computing améliore le développement logiciel agile à travers ses différentes phases, comme le montrent les résultats de cette étude de cas :

- Traitement en parallèle grâce à la disponibilité des ressources
- Infrastructure nécessaire permettant aux membres de l'équipe de travailler efficacement, ce qui se traduit par des compilations plus rapides
- Intégration continue, gestion des versions et tests continus du code
- Retours continus du client et échanges entre les équipes
- Communication constante entre les membres de l'équipe, assurant une transparence totale

Cependant, ce travail ne prend en compte que la méthodologie Scrum et se base sur une seule étude de cas, ce qui représente un champ d'analyse trop limité pour généraliser ces résultats à l'ensemble du secteur.

#### **2.2.4 Implémentation de Scrum dans le développement logiciel agile basé sur le cloud**

Cette étude effectuée par (Luca, 2025) explore la mise en œuvre de Scrum dans le développement logiciel basé sur le cloud. Il aborde les avantages, les défis, et met également en lumière les meilleures pratiques pour intégrer Scrum au cloud computing afin d'améliorer l'efficacité du développement logiciel de plusieurs façons, notamment en termes de :

- Scalabilité
- Rentabilité
- Collaboration en temps réel
- Fonctions de sécurité et conformité
- Intégration et déploiement continus

Parmi les façons dont le cloud computing améliore le développement logiciel, on retrouve :

- Un développement et un déploiement plus rapides grâce aux pipelines CI/CD basés sur le cloud
- Des ressources à la demande pour répondre aux exigences des sprints
- Une meilleure collaboration facilitant la communication et le partage des connaissances entre les équipes agiles
- Un soutien aux équipes distribuées, assurant un accès global

En tirant parti du cloud computing dans des environnements agiles, cela permet :

- Une flexibilité et une scalabilité pour une utilisation efficace des ressources
- Une collaboration renforcée
- L'intégration et le déploiement continus (CI/CD)
- Une rentabilité accrue
- Une meilleure visibilité et transparence

Tout au long de cet article, l'auteur met en avant de nombreux outils permettant de simplifier Scrum et d'améliorer le développement agile. L'article conclut en réaffirmant l'impact de Scrum et du cloud computing dans le développement agile, tout en mentionnant ses limites telles que les préoccupations en matière de sécurité, la gestion des coûts et la coordination des équipes à distance, qui nécessitent une planification stratégique et les outils adéquats pour être surmontés.

### 2.2.5 Accentuation sur les tests d'assimilation en utilisant le Cloud

Les auteurs de cet article (Raj et al, 2015) traitent de l'intégration de Scrum et du cloud computing pour améliorer le processus de développement agile, en mettant particulièrement l'accent sur les tests logiciels via le TaaS (Testing as a Service), comme représenté par la figure 9.

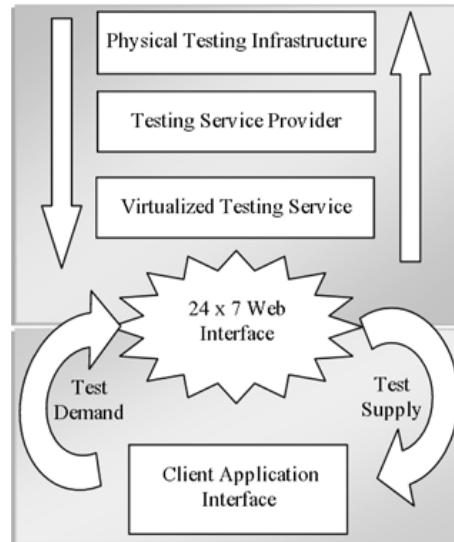


FIG. 9 – *Le Modele TAAS*

La combinaison du cloud computing et de Scrum présente de nombreux avantages en termes de collaboration d'équipe, de flexibilité, de scalabilité et d'efficacité des coûts. L'intégration du TaaS dans le développement logiciel accélère la phase de test grâce à des laboratoires de test (qui incluent des outils de test, exécutent l'application sous test (Application under test "AUT") et surveillent les données), ainsi qu'à l'exécution parallèle des tests dans plusieurs environnements, ce qui améliore la qualité du logiciel et élimine les retards liés à l'acquisition et à la configuration des serveurs.

Le TaaS fournit plusieurs services qui apportent une solution optimale à la multitude de problèmes rencontrés par les tests traditionnels, tels que :

- la gestion du TaaS,
- la gestion des environnements de test,
- Services de tests à la demande

Cependant, les tests basés sur le cloud rencontrent quelques problèmes, comme les risques liés à la sécurité des données (accès non autorisé, violations de données, etc.) et les problèmes de conformité, ce qui représente un enjeu critique pour les organisations adoptant des services cloud.

Bien que cette étude présente une intégration intéressante de Scrum et du cloud computing, elle néglige les défis rencontrés dans le processus de développement et n'explore pas comment adapter cette intégration à des projets complexes basés sur le cloud.

## 2.3 Étude Comparative

L'adaptabilité des équipes, le développement itératif, les cycles de développement courts et les retours fréquents sont parmi les facteurs clés qui expliquent le succès du développement agile dans l'industrie. Les besoins de ce secteur

évoluent constamment, d'où la recherche constante de nouvelles technologies pour répondre aux exigences croissantes du développement logiciel. Les cinq articles analysés précédemment explorent tous l'une de ces technologies : le cloud computing et son association avec le développement agile de logiciels. Le tableau 2 représente une étude comparative de ces travaux.

Article	[3]	[4]	[5]	[6]	[7]
Portée de l'étude	Développement d'un projet dans un environnement local et distribué	Développement d'un projet en utilisant le cloud et sans l'utiliser	Développement d'un projet en utilisant le cloud	Bénéfices d'implémenter scrum avec le cloud computing	Développement d'un projet en utilisant le cloud en mettant l'accent sur la phase de Test
Méthodologie d'étude	Etude de cas	Article méthodologique	Etude de cas	Article de synthèse	Article méthodologique
Méthode agile	Scrum	DSDM (Dynamic System Development method)	Scrum	Scrum	Scrum
Service cloud	MyRen cloud	Google app engine	CloudForge	(non spécifié)	Test-as-a-Service

TAB. 2 – Étude comparative des travaux

## 2.4 Proposition et description d'un cadre méthodologique (Framework) agile-cloud

Après une étude approfondie de plusieurs articles examinant la conjonction de l'agile et du cloud computing dans le processus de développement logiciel, nous allons proposer notre propre cadre méthodologique qui sera mis en application au cours du développement d'une application.

### 2.4.1 Description globale du framework proposé

Le Framework proposé est une combinaison de la méthode agile ScrumBan et du cloud computing (SBCC). La méthode agile choisie pour ce cadre méthodologique est une méthode hybride qui combine les meilleures fonctionnalités de Scrum et de Kanban. En effet, Scrum est une méthode de gestion de projet agile avec pour objectif d'améliorer la productivité des équipes de développement, apportant souplesse et performance à la réalisation du projet. Une approche itérative et incrémentale qui repose sur la notion de sprints (un délai défini pour l'accomplissement d'un certain nombre de tâches) donnant lieu à une meilleure flexibilité, des stand-ups quotidien pour assurer le bon avancement du projet, des Feedbacks réguliers permettant une forte implication des parties prenantes et une récolte des retours des clients et des réunions rétrospective évaluant les performances de l'équipe pour un meilleur déroulement des prochains sprints. Chaque équipe se constitue d'un Scrum Master (responsable de faire respecter le cadre Scrum et gestion de l'équipe), Product Owner (gestionnaire du backlog et responsable du produit) et équipe de développement (développeurs, architectes, designers, testeurs, etc).

Par ailleurs, Kanban est une méthode de gestion visuelle du travail utilisée pour organiser, visualiser et améliorer le flux des tâches pendant le processus de développement. Elle est illustrée sous forme d'un tableau composé de colonnes représentant l'état d'accomplissement des fonctionnalités tout au long du processus (To do, In progress et Done) et de cartes représentant les tâches à accomplir. Kanban offre la possibilité de visualiser l'état d'avancement du projet, de remédier à la rigidité de Scrum en permettant de débiter le développement de fonctionnalités prévues pour les prochains sprints, même si celles du sprint en cours ne sont pas entièrement terminées et de limiter le

travail en cours, connu par WIP (Work In Progress), pour éviter la surcharge et optimiser le flux du travail donnant lieu à une meilleure productivité des équipes et une meilleure qualité du produit final.

Ainsi, la combinaison de Scrum et Kanban (Scrumban) apporte une flexibilité accrue, des livraisons continues et une réduction de la surcharge de travail tout en assurant la satisfaction des clients.

Afin de renforcer ce processus de développement logiciel, nous allons y incorporer le cloud computing. Un modèle qui propose une multitude d'outils et de services accompagnant l'équipe agile dans toutes les phases du développement en apportant une scalabilité dynamique, un accès collaboratif et centralisé aux documents, outils et capacités de déploiement, d'intégration et de livraison continus améliorant la productivité tout au long du cycle de développement.

Notre cadre méthodologique proposé se décompose en une pré-phase, une phase de développement et une post-phase, comme étant illustré dans la figure 10.

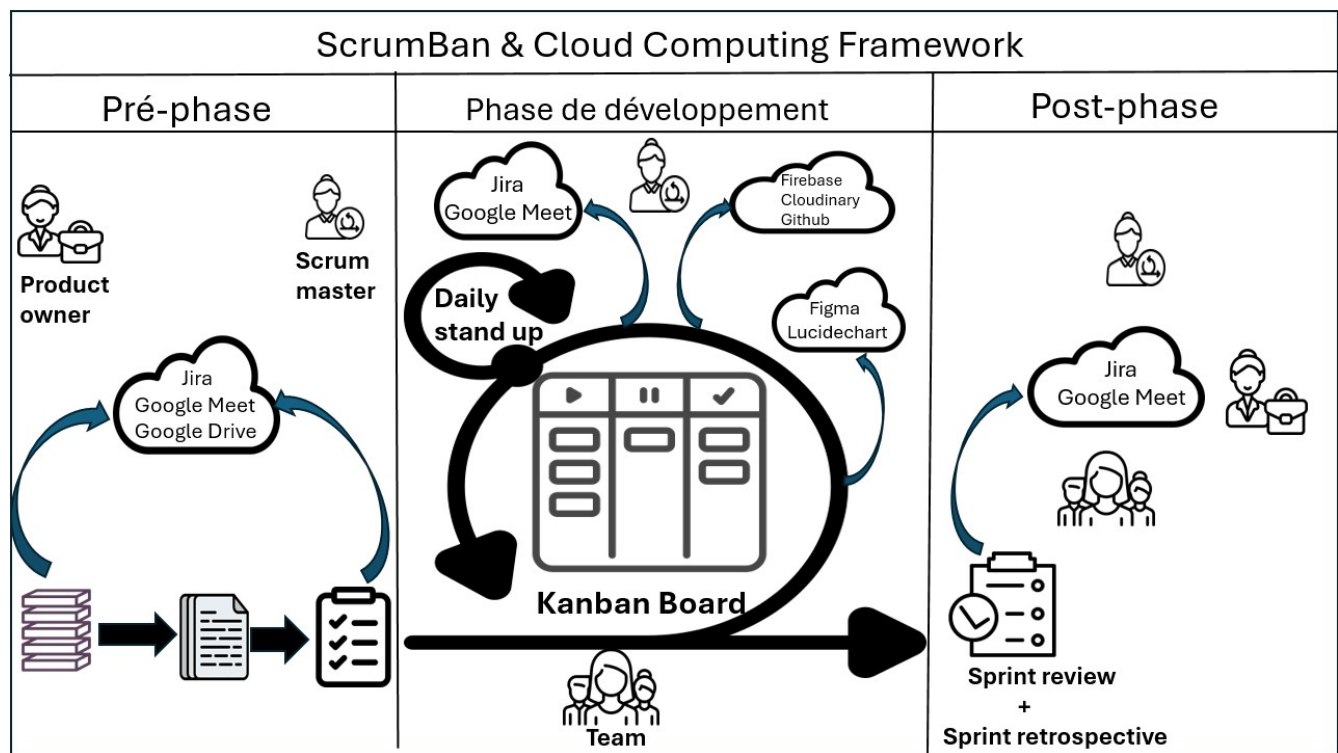


FIG. 10 – Framework SBCC

#### 2.4.2 Présentation des acteurs du framework

Les principaux acteurs intervenant dans le processus de développement sont:

- Product owner responsable du product Backlog. Il intervient durant la pré-phase pour définir les besoins et prioriser les fonctionnalités en continuant à ajuster le backlog au fil des sprints selon les retours et l'évolution du projet.
- Scrum master responsable de s'assurer de la bonne application du cadre méthodologique. Supervise le long du processus animant les stand-ups quotidiens et accompagnant l'équipe de développement dans l'application des bonnes pratiques.

- Équipe de développement qui compte les développeurs, testeurs, architectes et designers. Ils se chargent de la phase de développement en réalisant la conception, l'implémentation, les tests et le déploiement. L'équipe participe également à la post-phase qui leur permet d'évaluer leur travail pendant le sprint et d'apporter des améliorations pour une meilleure productivité.

### 2.4.3 Description détaillée des différentes phases du framework

Dans cette section, nous allons décrire les étapes à suivre dans chaque phase du processus de réalisation d'un projet.

- Description de la phase 1 : Pré-phase

Cette phase consiste à préparer le terrain pour la phase de développement. Le product owner commence par définir le product backlog (carnet de produit) dans lequel il formule des user stories listant les fonctionnalités souhaitées tout en les priorisant. Le product owner peut s'entretenir avec le scrum master via Google meet pour une meilleure compréhension et communication des besoins. En plus de définir le product backlog, il se charge de sa gestion à travers l'outil Jira (un outil de gestion de projet) en utilisant Google Drive pour toutes autres informations nécessaires.

Ayant à disposition le carnet de produit, le Scrum master et l'équipe de développement se réunit sur Google meet afin de planifier les fonctionnalités à accomplir pendant le sprint tout en définissant le sprint backlog détaillant les tâches à accomplir via Jira.

- Description de la phase 2: Phase de développement

Durant cette phase, l'équipe de développement se charge de développer les fonctionnalités délimitées pour le sprint. L'équipe de développement enchaîne cette phase par la conception de l'architecture système via Lucidchart.

Ensuite, l'étape d'implémentation à travers l'éditeur de texte Visual studio code avec une intégration Github pour l'hébergement et la gestion du code. Cette étape est appuyée par la plateforme cloud Firebase qui dispose de plusieurs services supplémentaires pour le déploiement d'applications, l'hébergement de données et les tests, avec la plateforme Cloudinary pour la gestion des médias, ce qui permet à l'équipe de développer d'une manière rapide et efficace. L'équipe de développement se doit d'actualiser l'avancement des tâches qui leurs ont été assignées via Jira à travers le tableau Kanban (Kanban Board), ce qui permet au scrum master de suivre le flux du travail avec visualisation des tâches à faire, celles en cours et celles terminées avec l'auteur de chacune permettant une complète transparence vis-à-vis de l'avancement du projet.

Cette phase se termine par la présentation des interfaces utilisateur développées pendant le sprint via Figma.

Le scrum master est, aussi, chargé de superviser la phase de développement à travers des stand-ups quotidiens avec l'équipe de développement s'assurant que le tout est cohérent et garantissant la bonne implémentation des pratiques ScrumBan.

- Description de la phase 3: Post-phase

Cette phase, après le développement d'une version fonctionnelle et opérationnelle du produit, permet d'évaluer le flux du travail, de communiquer les difficultés rencontrées et de proposer des solutions et améliorations à travers des Retrospective de sprint (Sprint Retrospective) entre le scrum master et l'équipe de développement via Google meet.

De plus, toutes les parties prenantes du projet (product owner, scrum master et l'équipe de développement) se réunit pendant des revus de sprints (Sprint Review), via Google meet, qui consiste à démontrer les user stories qui ont été accomplies pendant le sprint. L'équipe de développement scrum récolte les retours du client, permettant une itération du sprint en cas de besoins, et enregistre tout raffinement du product backlog, via Jira, permettant une meilleure réactivité aux changements.

## 2.5 Conclusion

A travers ce chapitre, un cadre méthodologique agile-cloud a été proposé en prenant en compte des articles scientifiques évaluant l'intégration du cloud dans le processus de développement agile. Il n'y a pas de doutes que

le cloud computing est un atout révolutionnaire pour le processus de développement logiciel agile et afin d'appuyer cet argument, le framework proposé sera mis en œuvre dans le prochain chapitre à travers le développement d'une application web et mobile de gestion de crise.

# Chapitre 3: Validation de la proposition par une étude de cas

## 3 Chapitre 3: Validation de la proposition par une étude de cas

### 3.1 Introduction

Dans ce chapitre, nous allons mettre en œuvre le Framework ultérieurement proposé dans le développement d'une application web et mobile qui sera mise à la disposition des autorités, des secours et des citoyens.

Cette application se présente comme une solution logiciel pour la gestion de crises naturelles en Algérie, conçue pour permettre aux autorités de communiquer, non seulement avec les autorités, mais aussi avec les citoyens. Ainsi, donnant lieu à une meilleure coordination des secours en intervention et permettant aux citoyens de rester informés sur l'évolution de ces crises.

L'objectif de cette application est de faciliter et d'améliorer le processus de gestion de crise et qui non seulement prévient les désastres, dans certains cas, mais aide, également, à l'intervention en minimisant les pertes et favorisant la communication avec la population.

### 3.2 Définition de gestion de crise

Les crises sont des événements qui peuvent survenir à tout moment sous différentes formes tels que les catastrophes naturels, les attaques terroristes, les pandémies, etc. Des situations qui instaurent un déséquilibre et une incertitude élevée susceptible de nuire aux personnes, aux biens ou aux processus opérationnels. La gestion de crises désigne une approche fondée sur un ensemble de mesures, procédures et de stratégies permettant d'identifier une crise, de la suivre et de la résoudre ou de la réduire d'une manière rapide mais surtout efficace.

Le cycle de gestion de crise comprend quatre phases fondamentales pour anticiper, répondre efficacement et reconstruire après une situation critique. Ces étapes, énumérés par [13] se décrivent comme suit:

- Prédiction et Prévention: Il s'agit d'une période qui précède la survenue d'un incident, dont l'objectif est d'identifier et limiter les risques, ainsi, réduire la probabilité qu'une crise ait lieu. Cependant, il est nécessaire de souligner que certains incidents ne peuvent être évités ni prédits tels que les désastres naturels et que d'autres, bien qu'en théorie évitables, peuvent être détectés trop tard, rendant leur prévention impossible. La prévention, étant la première ligne de défense, vise à:
  - Analyser les menaces et vulnérabilités d'une organisation, d'un système ou d'une structure.
  - Mettre en place des politiques de sécurité et de conformité dans le respect des lois, normes et réglementation.
  - Maintenance préventive des équipements, systèmes ou processus, afin d'éviter les pannes ou incidents.
  - Audits réguliers s'assurant que tous fonctionnent correctement dans le respect des normes.Ces activités, et bien d'autres, permettent de se donner les moyens d'agir avant qu'il ne soit tard.
- Préparation: Cette étape consiste à organiser la réponse vis-à-vis d'une crise avec des moyens et des plans afin d'être prêt à agir et réagir rapidement et efficacement. Cela implique:
  - La création de cellules de crises opérationnelles prêtes à être activées à tout moment.
  - L'élaboration de plans d'urgences détaillés mettant à disposition une structure et un langage commun sur quoi faire, qui fait quoi et comment agir en cas de crise. Cela permet la gestion immédiate de la situation tout en assurant la continuités des activités essentielles malgré les crises (panne informatique, cyberattaques, etc).
  - La sensibilisation et formation du personnel à travers des simulations, des exercices d'alertes et des formations permettant de tester la réaction des équipes face à une crise fictive dans un environnement contrôlé.
  - Le stockage préventive de ressources critiques pour pouvoir faire face à une crise. Ces stocks incluent du matériel médical, de la nourriture, de l'énergie, etc

L'investissement dans ces activités permet de se structurer avant que la crise ne frappe afin d'éviter le chaos.

- **Intervention:** Cette phase concerne la période où l'incident survient et juste après sa survenue en fonction du type de crise. Elle consiste à réagir rapidement en gérant activement la crise avec pour objectif de contenir et limiter les impacts négatifs et stabiliser la situation. Les actions typiques entretenues pendant cette période se caractérisent par:
  - Mise en œuvre immédiate du plan de crise prévu avec les procédures, les rôles et les actions à entreprendre dès la détection d'une crise pour y faire face.
  - Communication interne (vers les employés, les équipes et les responsables) et externe (vers les médias, les autorités et le public) afin d'informer, de rassurer et éviter la panique et la désinformation.
  - Coordination des équipes de secours avec une organisation efficace des parties prenantes et la mise en place des moyens et ressources nécessaires pour réagir rapidement et limiter les dégâts engendrés par la crise.
- **Rétablissement:** Cette dernière étape qui concerne la période qui suit la survenue d'une crise, cette dernière étant conclue et achevée. Elle vise à restaurer le fonctionnement normal des activités, des organisations et de la société tout en profitant de tirer des leçons dans le but d'améliorer les dispositifs mis en place pour la gestion de crise et de renforcer la résilience face à de futurs événements. Pour cela, les tâches suivantes sont prises en main:
  - Élaboration d'un bilan de crise qui s'agit d'analyser ce qui a fonctionné ou ce qui a manqué pendant la crise afin de comprendre les réussites, les erreurs et les points à améliorer.
  - Réparation et reconstruction de ce qui a été endommagé pour relancer les activités interrompues le plus rapidement possible.
  - Offrir du soutien psychologique et financier si nécessaire aux personnes touchées par la crise et aux équipes d'intervention.
  - Amélioration et renforcement des plans et dispositifs de gestion en vue d'éventuelles futures crises.

Ces quatre phases de gestion sont essentielles et ne doivent pas être négligées. Elle forme un cycle continu qui transforme une crise en une opportunité d'améliorer et de renforcer la capacité d'anticipation de la suivante. Elle sont illustrées à travers la figure 11.

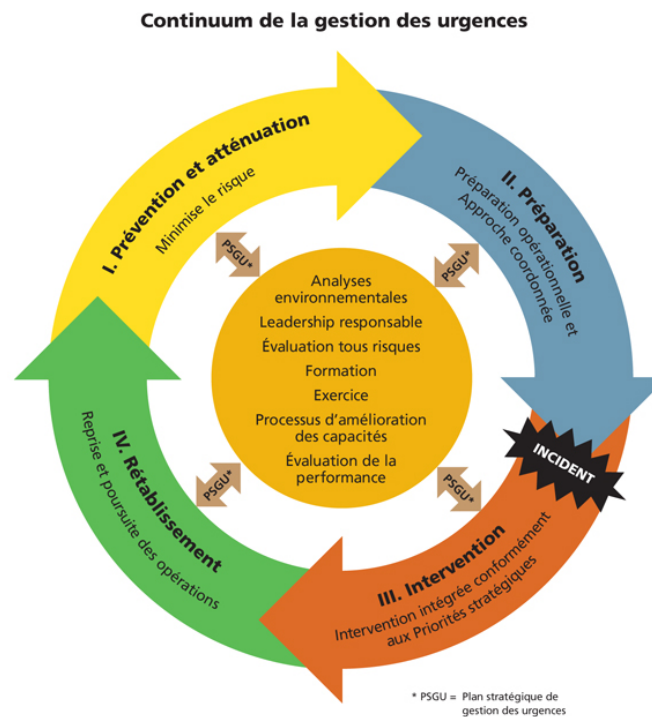


FIG. 11 – Phases de crises

### 3.3 Types de crise

Il existe plusieurs types de crises, notamment:

- Crises naturelles: Des événements souvent imprévisibles et incontrôlables causés par la nature. Par exemple, séismes, inondations, incendies de forêt, ouragan et tempêtes.
- Crises économiques ou financières: Des situations qui perturbent l'économie et la stabilité des organisations. Ces perturbations peuvent se manifester en tant que récessions, inflation, faillite d'entreprise, crises bancaires, etc.
- Crises sanitaires: Des situation qui concerne la santé publique à grands échelles tels les pandémies et les épidémies.
- Crises politiques et sociales: se manifeste en tension interne ou externe menaçant la stabilité sociale et l'ordre public, notamment, des coups d'états, des guerres, des émeutes, des grèves, etc.
- Crises écologiques: Causés par la dégradation de l'environnement à grandes échelles souvent liés à l'activité humaine. On note, par exemple, la pollution, le réchauffement politique, la perte de biodiversité et la déforestation massive.
- Crises sécuritaires: Des situations menacent la sécurité des personnes et d'infrastructures tels que les attentats terroristes et les intrusions sur des sites sensibles.

Il existe bien plus de types de crises que celles mentionnées précédemment, mais cette étude de cas se focalise sur la gestion des crises naturelles.

### 3.4 Gestion de crise en Algérie

Les moyens et dispositifs législatifs, institutionnels et opérationnels mis en place pour faire face aux aléas naturels varient de pays en pays. L'Algérie étant victime de nombreuses catastrophes naturels dévastatrices comme le tremblement de Terre de Boumerdès en 2003, le système algérien de gestion des crises a connu plusieurs amélioration en fur des années avec une vision plus large qui comprend, non seulement la gestion, mais aussi l'anticipation et la réduction des risques. Ces différents changements se sont portés sur:

- Le plan législatif et juridique: Des lois et des décrets créant un environnement favorable à la réduction des risques de catastrophe.
  - Loi n. 04-20 du 25 décembre 2004 [14] qui définit les bases du système de gestion des catastrophes en Algérie, en définissant les responsabilités des autorités publiques et des collectivités locales. Elle couvre les différents aspects du cycle de gestion des catastrophes, à savoir la prévention, la préparation et la réponse.
  - Loi n.24-04 du 26 février 2024 [15] qui définit les règles de prévention, d'intervention et de réduction des risques de catastrophes dans le cadre du développement durable tout en précisant les types de risques couverts, tels que les risques sismiques, d'inondations, d'incendies de forêt, etc.
- Le plan institutionnel: Mise en place d'instituts et d'organismes avec une coordination interministérielle accrue pour objectif d'élaboration et de mise en œuvre de la stratégie nationale de gestion des risques. Parmi ces organismes:
  - La direction générale de la protection civile (DGPC): L'institution nationale chargée de la prévention, de la gestion des risques et des interventions d'urgence en Algérie placée sous la tutelle du Ministère de l'Intérieur.
  - Ministère de l'Intérieur, des Collectivités Locales et de l'Aménagement du Territoire (MICLAT): L'organisme qui supervise la DGPC et qui est chargé de coordonner les efforts des différentes institutions impliquées dans la gestion des crises tout en élaborant des politiques nationales en matière de prévention des risques.
  - Autres ministères sectoriels comme le ministère de l'agriculture qui se charge des risques liés aux feux de forêt, le ministère des ressources en eau qui gère les risques d'inondation et de sécheresse, etc.

- Collectivités locales (wilayas et communes): Les autorités locales sont responsables de l'élaboration des plans communaux de sauvegarde et de la coordination des secours au niveau local.

Ces dernières représentent les institutions principales dans la gestion des risques et des crises, cependant, il existe d'autres organisations pouvant apporter du soutien au système de gestion de crises tels que les sociétés civiles et les Organisations Non Gouvernementales (croissant rouge, etc), les Institutions académiques et de recherche et Partenaires internationaux tel que le Programme des Nations Unies pour le Développement (PNUD), etc.

- Le plan opérationnel: Des procédures, des outils et moyens matériels et des ressources humaines mises en main de la protection civile tel que:
  - Le Plan ORSEC (Organisation des Secours): Un dispositif national de gestion des catastrophes visant à organiser, coordonner et mobiliser les moyens humains et matériels nécessaires en cas de sinistre majeur. Il a pour but le sauvetage et secours des personnes, la coordination des secours, la mobilisation des ressources nécessaires pour faire face à la catastrophe et la gestion de la post-catastrophe mettant en place des mesures de réhabilitation et de soutien aux sinistrés.

### 3.4.1 Scénario Typique d'une gestion de crise en Algérie

Afin d'illustrer les divers procédés employés lors de la survenue d'une crise en Algérie, nous proposons un scénario typique d'une crise naturelle, notamment un séisme d'une magnitude importante dans une wilaya du nord du pays, sachant que c'est une zone sismique. L'objectif est de mettre en valeur les différents dispositifs mis en place pour chaque phase du cycle de gestion de crise en prenant en compte la revue [8]:

- Phase 1 (Prédiction et Prévention):
  - Le Centre de Recherche en Astronomie, Astrophysique et Géophysique (CRAAG) effectue une surveillance sismique permanente.
- Phase 2 (Préparation):
  - La protection civile effectue régulièrement des exercices de simulation de séismes et des formations de premiers secours dans les écoles et différentes organisations.
  - Elaboration de plans d'urgence dans les zones de risques
  - Organisation de simulations et formations pour toutes les parties prenantes dans le système de gestion de crise.
- Phase 3 (Intervention): Supposant qu'un séisme de magnitude 6 sur l'échelle de Richter à 7h du matin dans la wilaya de Béjaïa.
  - Des signalements et rapports sont détectés venant de citoyens ou autorités faisant état d'un tremblement de terre.
  - Le Centre de Recherche en Astronomie, Astrophysique et Géophysique (CRAAG) détecte immédiatement le séisme, confirme sa survenue et relai d'informations aux autorités locales et au ministère de l'intérieur.
  - Le ministère de l'intérieur soulève l'alerte et active le plan ORSEC.
  - Déploiement des secours dans les zones sinistrées (la protection civile, le croissant-rouge, l'armée, etc) afin d'évacuer les survivants, de mettre en place des refuges temporaires pour les accueillir et évaluer les zones à hauts risques tout en sollicitant de l'aide des wilayas adjacentes et du soutien international si nécessaire.
  - Communication continue avec les médias et la population pour éviter la désinformation et les informer du progrès de la situation.
- Phase 4 (Retour à la normal):
  - Dresser le bilan humain et matériel
  - Évaluation post-crise organisée par la DGPC.
  - Révision du plan ORSEC local pour renforcer le système de gestion de crise.

Malgré la présence d'un système de gestion de crise traitant la plupart des points essentiels d'une crise, le système algérien est régi de défis persistants tels:

- Manque de dispositifs de prévention dans certaines régions et sur certains aléas naturels.
- Manque de coordination entre les secteurs.
- Inégalités des capacités locales entre wilayas en termes de ressources humaines et matériels et de formations des autorités locales.
- Manque de communication optimale avec la population.

Par cette étude cas, nous souhaitons intégrer la technologie dans le but d'apporter des améliorations aux différentes difficultés rencontrées dans le système de gestion de crise en Algérie.

### 3.5 Outils de développement

Les outils de développement utilisés dans la réalisation de notre application sont:



**Visual studio code:** Un éditeur de code source léger mais puissant [16], développé en open source par Microsoft. Il est gratuit, multiplateforme et met à la disposition des développeurs une interface conviviale disposant d'un riche écosystème d'extensions pour plusieurs langages et environnements d'exécution.



**GitHub:** Une plateforme d'hébergement [17] de code basée sur le cloud qui offre des outils et fonctionnalités pour la gestion de projets, la collaboration et le contrôle de versions.



**Figma:** Une application de conception d'interfaces utilisateur (UI) et d'expérience utilisateur (UX) [18] basée sur le cloud qui offre une gamme complète d'outils pour la conception, la création, le partage et le test de divers prototypes et animations de sites Web, d'applications mobiles et d'autres produits numériques.



**Google meet:** Un service de visioconférence [19] développé par Google, offrant la possibilité d'organiser et de participer à des réunions vidéo sécurisées en ligne, avec partage d'écran, chat en direct, enregistrement, etc. Ce qui permet aux utilisateurs de se connecter à distance que ça soit pour des fins professionnelles ou bien personnelles.



**Google drive:** Un service de stockage basé sur le cloud [20] permettant de sauvegarder, de partager et d'accéder à des documents en ligne. Une plateforme qui permet aux utilisateurs de collaborer en temps réel tout en garantissant le contrôle des versions des documents.



**Lucidchart:** Un outil multiplateforme [21] qui permet aux utilisateurs de créer des diagrammes, des organigrammes, des maquettes, en ligne. Une application souvent utilisée dans la conception de l'architecture logicielle d'applications à travers des schémas UML, etc.



**Firestore:** Plateforme cloud [22] de développement d'applications web et mobiles proposée par Google, fournissant plusieurs services, tel qu'une base de données en temps réel, l'authentification et l'hébergement, afin de développer des applications web ou mobiles. Cette plateforme agit comme un Backend (Backend as a Service) pour aider les développeurs à développer et offrir des expériences d'applications plus riches.



**Jira:** Outil de gestion de projet agile [23], développé par Atlassian. Une plateforme souvent utilisés dans le suivi du flux de travail des équipes de développement logiciel, le suivi des bugs et la gestion des test, facilitant le travail et la collaboration des équipes agiles.



**Cloudinary:** Plateforme cloud [24] développée en 2012 pour permettre aux développeurs de mieux télécharger, gérer et livrer une multitude de photos et vidéos. Cette plateforme propose de livrer une expérience visuelles rapides et irréprochables à grande échelle.



**Expo Go:** Un environnement sandbox [26] qui permet aux développeurs d'expérimenter, d'une manière rapide et efficace, le développement d'applications mobiles Android et IOS native en exécutant du code Javascript.

### 3.6 Langages et frameworks de programmation

Les lagages et frameworks utilisés pour développer notre application sont:



**HTML (Hyper Text Markup Langage):** Un langage de balisage [27] qui est utilisé pour définir la structure du contenu d'une page web. En d'autres termes, HTML est employé pour représenter le squelette d'un site web.



**CSS (cascading Style Sheet):** Langage de feuille de style en cascade [28] qui permet de définir l'apparence et la présentation d'une page web écrite en HTML. CSS détermine la façon dont laquel les éléments sont rendus à l'écran sur différents appareils.

**JSX (JavaScript XML):** Une extension de syntaxe de JavaScript [29] qui permet aux développeurs React d'écrire un balisage de type HTML dans React avec du code Javascript. JSX simplifie l'écriture et la compréhension du code qui défine l'apparence et la structure des interfaces utilisateurs.



**Bootstrap:** Framework front-end open source [30] qui facilite la conception de sites web et d'applications web modernes, réactifs et adaptatifs garantissant une cohérence visuelle sur tous types d'écrans. Bootstrap a été construit sur HTML, CSS et Js permettant de créer des sites web plus rapidement sans se soucier des commandes et fonctions de base.



**Javascript:** Langage de programmation ou de script [31] qui permet de créer des pages web interactives et dynamiques en implémentant des fonctionnalités complexes afin d'améliorer l'expérience utilisateur.



**React:** Bibliothèque open source JavaScript basé sur les composants [32], facilitant la création d'interfaces utilisateur interactives et dynamiques. Destinée au développement d'applications web, elle utilise le concept du DOM virtuel pour des mises à jour plus rapides, améliorant les performances et minimisant les mises à jour du DOM réel du navigateur.



**React Native:** framework d'applications mobiles open source [34] qui permet de concevoir les interfaces utilisateurs d'applications mobiles natives pour IOS et Android sans compromettre l'expérience utilisateur. React Native offre un ensemble de composants natifs correspondant aux blocs de construction de l'interface utilisateur native de la plateforme.



**Vite:** Un outil de développement moderne [35] qui s'intègre facilement avec des frameworks, notamment React. Il permet d'améliorer et d'alléger l'expérience de développement en offrant des capacités avancées, le tout avec une configuration minimale.



**Expo:** Un framework qui étend React Native [26] qui est utilisé pour développer des applications Android et iOS natives en utilisant une seule base de code en Javascript. Expo est doté de multiple SDK (Software Development Kit) qui permettent d'accéder à des fonctionnalités natives d'un appareil, comme la caméra.

### 3.7 Mise en application du framework proposé

Dans cette section, nous allons mettre en application le framework SBCC (ScrumBan et Cloud computing), que nous avons précédemment proposé, dans le but de développer une application web et mobile afin de faciliter la gestion de crise en temps réel.

### 3.7.1 Pré-phase

Dans cette phase nous allons présenter les parties prenantes dans la réalisation de ce projet, en spécifiant le rôle de chacun, comme le représente le tableau 3

Product owner	Représentant du ministère de l'Intérieur
Scrum master	Docteur Karima Ait Abdelouhab
Equipe de développement	Abderrahmane Cylia

TAB. 3 – Rôles des parties prenantes du projets

De plus, nous allons définir le product backlog de l'application à travers une analyse et spécification des besoins.

- Spécification des besoins

Dans cette étape, nous allons définir les exigences fonctionnels et non fonctionnels de notre application.

- Besoins fonctionnels

- ★ Inscription: chaque citoyen doit créer un compte afin de pouvoir utiliser les fonctionnalités de l'application et accéder à ces données.
- ★ Authentification: chaque utilisateur doit s'authentifier afin de pouvoir utiliser les fonctionnalités de l'application et accéder à ces données.
- ★ Signalement d'incident: permet aux utilisateurs de signaler un incident avec une localisation, photo/vidéo et toute autre information nécessaire.
- ★ Valider le rapport d'incident: permet aux autorités d'approuver le rapport d'incident afin d'affirmer la présence d'une catastrophe pour déclencher le système de gestion de crise.
- ★ Notifications et messages d'alerte: permet d'alerter les utilisateurs et de communiquer avec la population les informants de l'avancement de la crise.
- ★ Gestion des ressources: permet de gerer les ressources humaines et matérielles de chaque organisme intervenant dans une crise.
- ★ Système de communication: permet de communiquer entre les différentes parties prenantes dans la gestion de crises.
- ★ Gestion des utilisateurs: permet de définir les rôles des utilisateurs afin d'adapter l'interface selon le profil.
- ★ Système de géolocalisation: permet de localiser les zones touchés par la crise, les équipe d'intervention sur le terrain, les routes bloqués, les centres d'évacuation et les établissement médicaux.
- ★ Archivage historique des crises: permet d'archiver l'historique de chaque incident afin d'évaluer la performance dans parties intervenantes.
- ★ Annonces et communiqués: permettre de publier des annonces ou communiqués à la population à tout moment.

- Besoins non fonctionnels

- ★ Performance: l'application doit être rapide, réactive et disponible à tout moment offrant une expérience utilisateur agréable.
- ★ Évolutivité: l'application doit être capable de s'adapter à un grand nombre d'utilisateurs simultanés en cas de crise sans compromettre ses performances.
- ★ Fiabilité: implique la capacité de l'application à fonctionner correctement et en continu, tout en garantissant la sécurité des données de l'utilisateur pour éviter la perte de données et les temps d'arrêt. Elle peut inclure la sauvegarde automatique des données, la détection et la correction des erreurs, etc.
- ★ Disponibilité: s'assurer que l'application soit disponible 24 heures sur 24 et 7 jours sur 7 surtout en cas de crise avec un temps d'arrêt minimal.

- ★ Convivialité : veiller à ce que l'interface utilisateur soit facile à utiliser pour les utilisateurs en dépit de leurs niveaux de compétences technologiques.
  - ★ Sécurité : s'assurer l'authentification sécurisée des utilisateurs, chiffrement des données sensibles et la protection contre toutes attaques.
  - ★ Compatibilité : faire en sorte que l'application soit capable de fonctionner sans difficultés sur différents navigateurs web et plateformes.
  - ★ Ergonomie : s'assurer que l'application soit facile à utiliser, intuitive et offrant une navigation aisée à travers ses fonctionnalités.
  - ★ Interopérabilité : veiller à ce que les différents composants de l'application soit capable de travailler.
- Diagramme de cas d'utilisation globale
 

La figure 12 représente le diagramme de cas d'utilisation globale de l'application, prenant compte de la spécification des besoins.

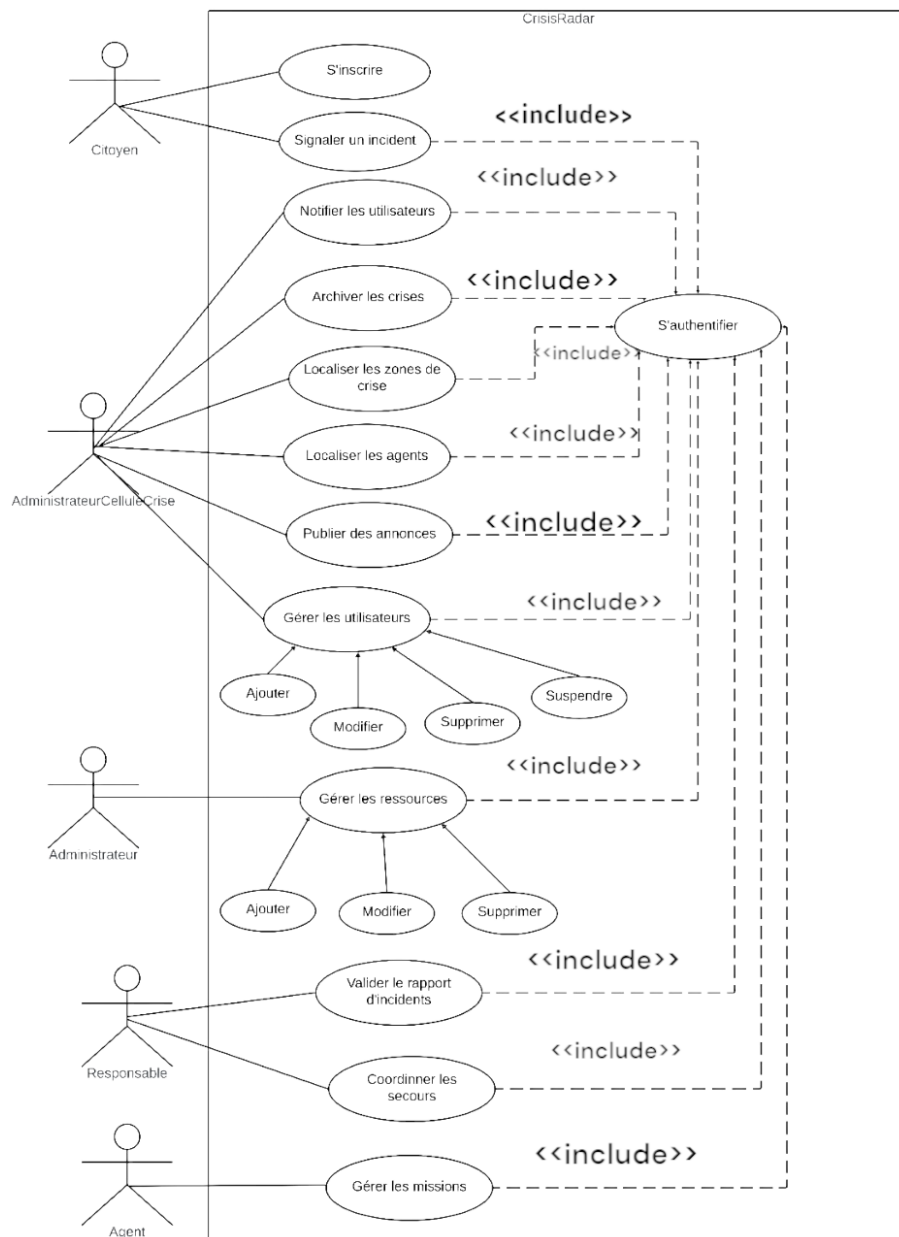


FIG. 12 – Diagramme de cas d'utilisation globale

- Product Backlog

Dans cette section, nous allons définir le carnet de produit, à travers la spécification des besoins, tout en priorisant les fonctionnalités, comme illustré dans le tableau 3.

Fonctionnalité	Utilisateurs	Priorité
Inscription	Citoyens	1
Authentification	Citoyens et autorités	1
Signalement d'incident	Citoyens	1
Gestion des utilisateurs	Autorités	1
Valider le rapport d'incident	Autorités	2
Notifications et messages d'alerte	Autorités	2
Gestion des ressources	Autorités	2
Système de géolocalisation	Autorités	3
Coordination des secours	Autorités	3
Archivage historique des crises	Autorités	4
Annonces et communiqués	Autorités	4

TAB. 4 – *Product Backlog*

- Présentation des acteurs du système:

Les utilisateurs qui interagissent avec l'application sont :

- Les citoyens: représentant les personnes civiles.
- Les autorités: représentant les différents acteurs impliqués dans la gestion de crise, qui sont:
  - ★ Responsables tel que le directeur de la protection civile, le wali, le ministre de l'intérieur, le chef de garde, le chef de point transit, etc
  - ★ Administrateur de la cellule de crise qui se charge du suivi de la crise et tout elements adjacents.
  - ★ Administrateurs des différentes organisations tels que l'administrateur de la protection civile chargé de gérer ses effectifs, etc.
  - ★ Agents de terrain qui représente toute personne intervenantes sur les lieux d'une crise.

- Planification des sprints

Dans le tableau 5, nous délimitons les fonctionnalités à réaliser dans chaque sprint.

N° du sprint	Fonctionnalité	Durée
Sprint n°01	<ul style="list-style-type: none"> <li>- Inscription</li> <li>- Authentification</li> <li>- Signalement d'incident</li> <li>- Gestion des utilisateurs</li> </ul>	3 jours
Sprint n°02	<ul style="list-style-type: none"> <li>- Valider le rapport d'incident</li> <li>- Notifications et messages d'alerte</li> <li>- Gestion des ressources</li> </ul>	5 jours
Sprint n°03	<ul style="list-style-type: none"> <li>- Système de géolocalisation</li> <li>- Coordination des secours</li> </ul>	1 semaine
Sprint n°04	<ul style="list-style-type: none"> <li>- Archivage historique des crises</li> <li>- Annonces et communiqués</li> </ul>	3 jours

TAB. 5 – Planification des sprints

- Sprint backlog

Au cours du premier sprint, nous allons développer les user stories représenté dans la figure 13:

ID	Description	Priority	Count
CR-7	En tant que citoyen je peux me connecter à mon compte, pour accéder...	À FAIRE	1
CR-16	En tant qu'administrateur, je peux gérer les utilisateurs.	À FAIRE	1
CR-23	En tant qu'administrateur de la cellule de crise, je peux me connecter...	À FAIRE	1
CR-8	En tant que citoyen, je peux signaler un incident.	À FAIRE	1
CR-1	En tant que citoyen, je peux créer un compte.	À FAIRE	1

FIG. 13 – Sprint 1 Backlog

Au cours du deuxième sprint, nous allons développer les user stories représenté dans la figure 14:

ID	Description	Priority	Count
CR-9	En tant que responsable, je peux valider le rapport d'incident.	À FAIRE	2
CR-15	En tant qu'administrateur, je peux gérer les ressources.	À FAIRE	2
CR-17	En tant qu'administrateur, je peux envoyer des alertes et notifications.	À FAIRE	2

FIG. 14 – Sprint 2 Backlog

Au cours du troisième sprint, nous allons développer les user stories représenté dans la figure 15:

ID	Description	Priority	Count
CR-11	En tant que responsable, je peux localiser les zones touchées.	À FAIRE	3
CR-13	En tant qu'utilisateur (responsable et agent de terrain), je peux comm...	À FAIRE	3
CR-14	En tant qu'agent de terrain, je peux consulter mes missions.	À FAIRE	3
CR-19	En tant qu'administrateur, je peux localiser les équipes d'intervention.	À FAIRE	3
CR-20	En tant que responsable, je peux coordonner les secours.	À FAIRE	3

FIG. 15 – Sprint 3 Backlog

Au cours du quatrième sprint, nous allons développer les user stories représenté dans la figure 16:

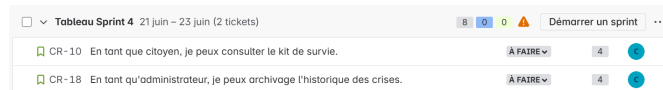


FIG. 16 – *Sprint 4 Backlog*

### 3.7.2 Phase de développement

**Réalisation du sprint n°01** Dans cette partie du chapitre, nous allons décrire les étapes de développement des fonctionnalités du premier sprint.

★ Analyse

– Diagramme de cas d'utilisation

La figure 17 illustre le diagramme de cas d'utilisation du premier sprint.

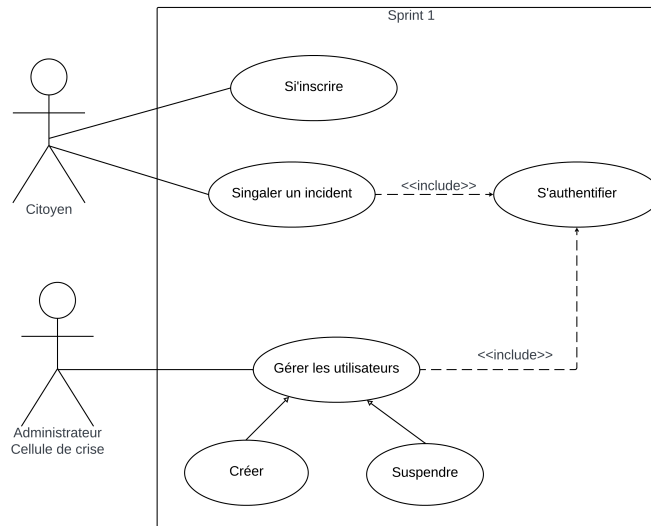


FIG. 17 – *Diagramme de cas d'utilisation du sprint 1*

– Description textuelle

A travers cette section, nous allons définir la description textuelle de chaque cas d'utilisation du sprint 01:

– Cas d'utilisation “S’inscrire” illustré par le tableau 6

Description textuelle
<p><b>Nom du cas:</b> Inscription  <b>But:</b> Permettre aux citoyens de créer un compte utilisateur.  <b>Acteur:</b> citoyen</p>
<p><b>Séquencement:</b> le citoyen est devant l'écran de connexion.  <b>Pré-conditions:</b> L'application est installée sur l'appareil du citoyen.  <b>Enchaînement nominale:</b>  1. Le citoyen clique sur le lien "s'inscrire"  2. Le système affiche la page d'inscription.  3. Le citoyen remplit le formulaire et clique sur le bouton "s'inscrire"  4. Le système vérifie la syntaxe et valide.  5. Le système crée un compte utilisateur.  <b>Enchaînement alternatif:</b>  A1. La syntaxe est invalide. A partir du point 4.  5. Le système affiche un message d'erreur.  6. Le citoyen ressaisie ses informations et clique sur le bouton "s'inscrire".  Le séquencement continue à partir du point 4.  <b>Post-conditions:</b>  Le citoyen a un compte utilisateur.</p>

TAB. 6 – Description textuelle du cas d'utilisation "S'inscrire"

– Cas d'utilisation "Se connecter" illustré par le tableau 7.

Description textuelle
<p><b>Nom du cas:</b> Connexion  <b>But:</b> Permettre aux utilisateurs de se connecter à l'application.  <b>Acteur:</b> utilisateur</p>
<p><b>Séquencement:</b> le citoyen est devant l'écran de connexion.  <b>Pré-conditions:</b>  L'application est installée sur l'appareil du citoyen.  L'utilisateur a déjà un compte.  Le compte utilisateur n'est pas suspendu. <b>Enchaînement nominale:</b>  1. Le citoyen remplit le formulaire et clique sur le bouton "se connecter"  2. Le système vérifie les identifiants de connexion et valide.  3. Le système affiche la page d'accueil de l'application.  4.  <b>Enchaînement alternatif:</b>  A1. Les identifiants de connexion sont invalides. A partir du point 2.  3. Le système affiche un message d'erreur.  4. L'utilisateur ressaisie ses informations et clique sur le bouton "se connecter".  Le séquencement continue à partir du point 2.  A2. l'utilisateur a oublié son mot de passe. à partir du point 1.  1.L'utilisateur clique sur le lien "Mot de passe oublié". 2.Le système affiche la page de réinitialisation du mot de passe.  3. L'utilisateur saisie son email. 4. Le système envoie un email de réinitialisation.  Le séquencement reprend à partir du point 1. <b>Post-conditions:</b>  Le compte utilisateur n'est pas suspendu.</p>

TAB. 7 – Description textuelle du cas d'utilisation "Connexion"

- Cas d'utilisation "Signaler un incident" illustré par le tableau 8.

Description textuelle
<p><b>Nom du cas:</b> Signalement d'incident  <b>But:</b> Permettre aux citoyens de signaler un incident aux autorités.  <b>Acteur:</b> citoyen</p>
<p><b>Séquencement:</b> le citoyen est devant la page de signalement d'incident.  <b>Pré-conditions:</b> L'application est installée sur l'appareil du citoyen. Le citoyen est connecté à son compte. <b>Enchaînement nominale:</b>  1. Le citoyen remplit le formulaire et clique sur le bouton "signaler"  2. Le système vérifie que les champs obligatoires sont remplis et valides.  <b>Enchaînement alternatif:</b>  A1. Un champ obligatoire est vide. A partir du point 2.  3. Le système affiche un message d'alerte.  4. Le citoyen saisit les champs vides et clique sur "signaler". Le séquencement continue à partir du point 2.  <b>Post-conditions:</b>  Le système enregistre le signalement.</p>

TAB. 8 – Description textuelle du cas d'utilisation "Signaler un incident"

- Cas d'utilisation "Gérer les utilisateurs" illustré par le tableau 9.

Description textuelle
<p><b>Nom du cas:</b> Gestion des utilisateurs  <b>But:</b> Permettre à l'administrateur de la cellule de crise de gérer les utilisateurs.  <b>Acteur:</b> Administrateur de la cellule de crise</p>
<p><b>Séquencement:</b> le citoyen est devant la page utilisateur.  <b>Pré-conditions:</b> Administrateur de la cellule de crise est connecté à son compte utilisateur.  <b>Enchaînement nominale:</b>  1. L'administrateur clique sur un bouton (ajouter, modifier, supprimer ou suspendre).  2. Le système affiche un formulaire pour l'ajout ou la modification et une alerte pour la suppression et la suspension.  3. L'administrateur clique sur le bouton "confirmer"  4. Le système vérifie les données et valide.  5. Le système exécute l'action.  <b>Enchaînement alternatif:</b>  A1. L'administrateur annule son action. A partir du point 2.  Le séquencement s'arrête ici.  <b>Post-conditions:</b>  Les informations sont mises à jour dans Firestore.</p>

TAB. 9 – Sprint 4 Backlog

- Diagramme de séquence

Dans cette partie, nous allons définir les diagrammes de séquences système pour chaque cas d'utilisation.

- Cas d'utilisation "S'inscrire" représenté par la figure 18

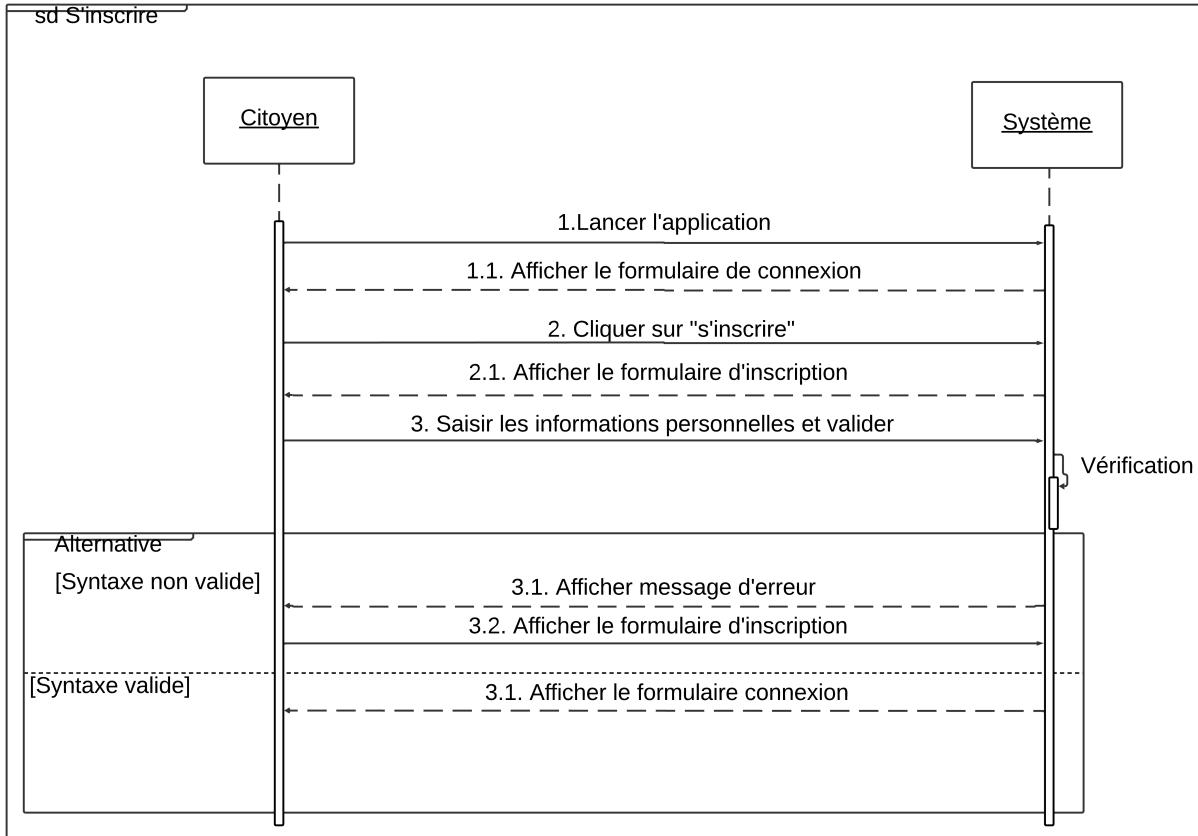


FIG. 18 – Diagramme de cas d'utilisation "Inscription"

– Cas d'utilisation "Se connecter" représenté par la figure 19.

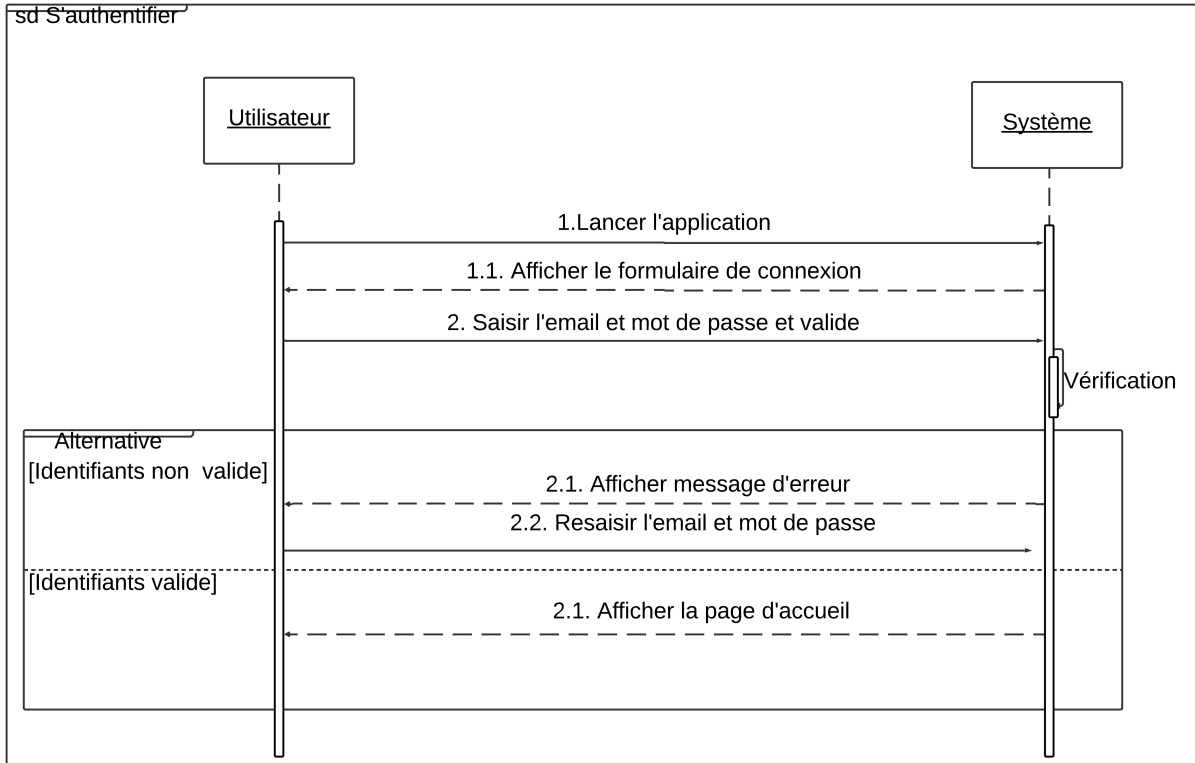


FIG. 19 – Diagramme de cas d'utilisation "Connexion"

– Cas d'utilisation "Signaler incident" représenté par la figure 20.

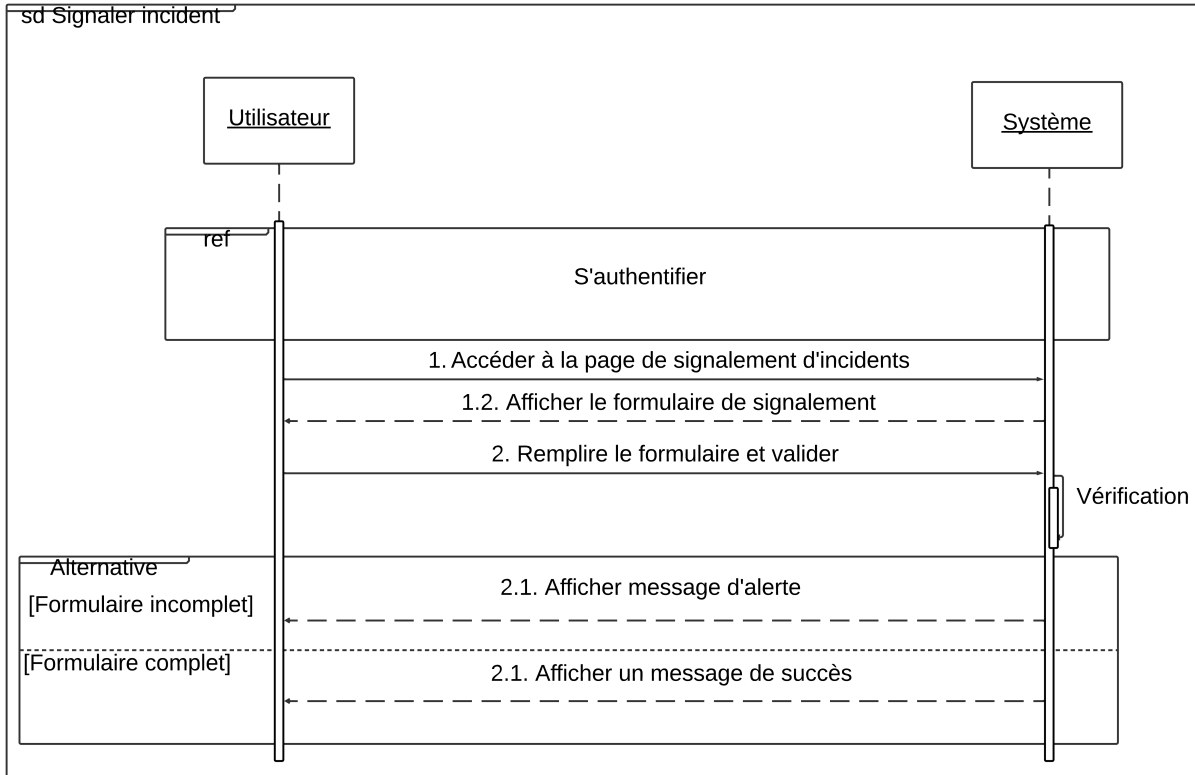


FIG. 20 – Diagramme de séquence “Signalement d’incident”

– Cas d’utilisation “Gestion des utilisateurs” représenté par la figure 21.

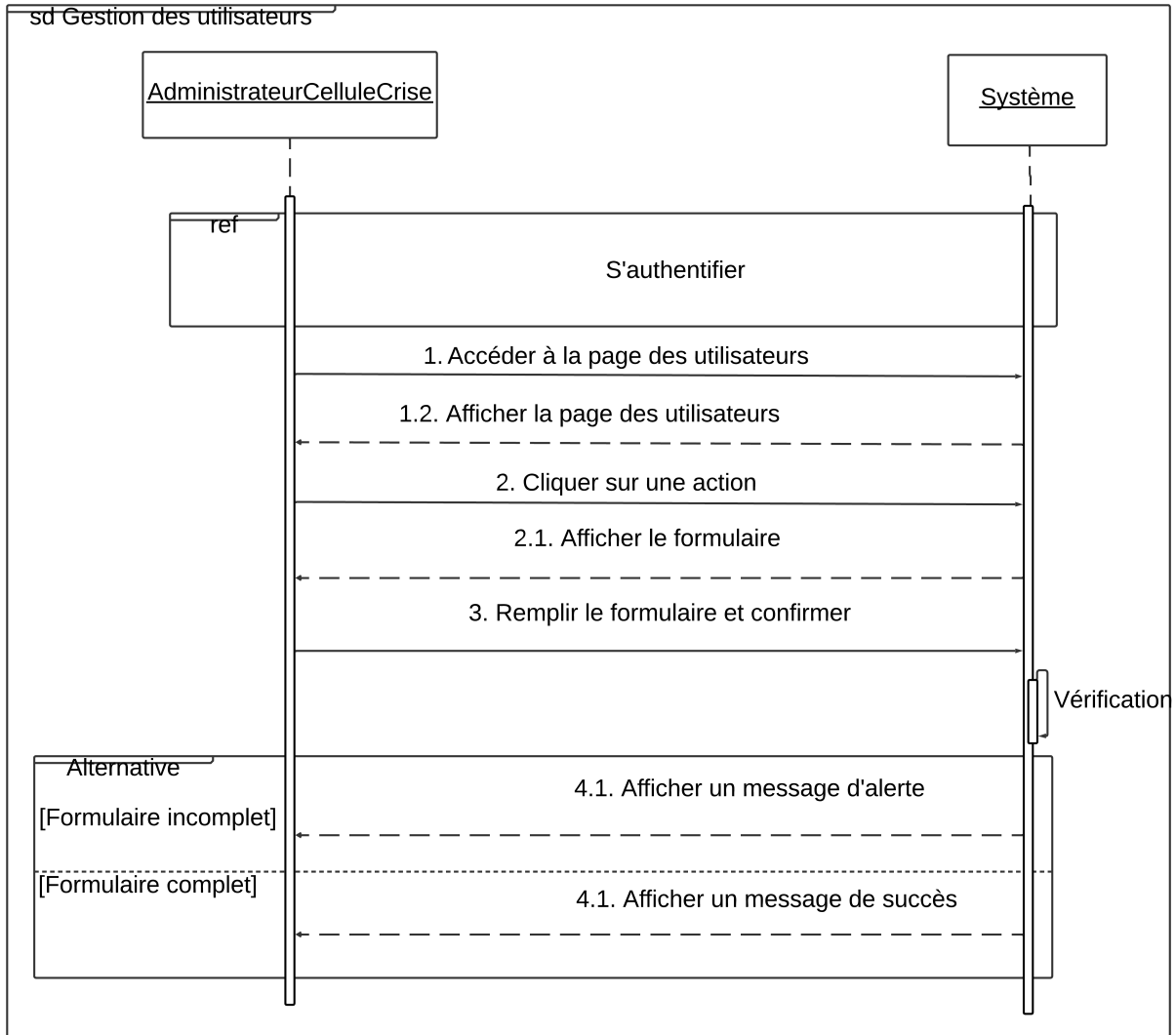


FIG. 21 – *Diagramme de séquence “Gestion des utilisateurs”*

★ Conception

- Diagrammes de séquence détaillés
- Cas d’utilisation “Inscription” illustré par la figure 22.

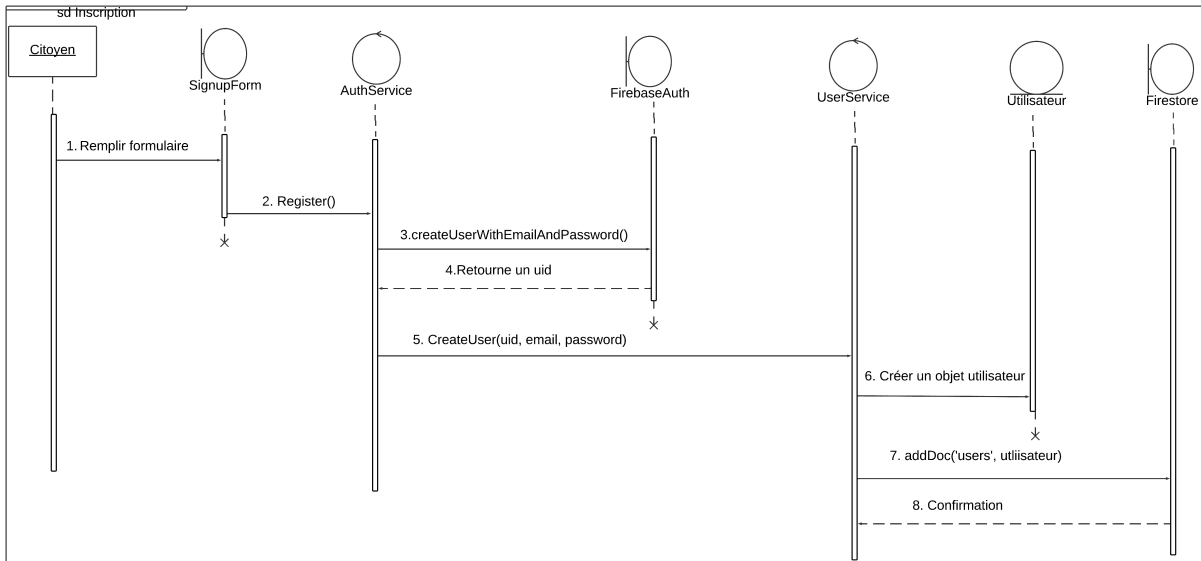


FIG. 22 – Diagramme de séquence détaillé de “Inscription”

– Cas d’utilisation “Connexion” illustré par la figure 23.

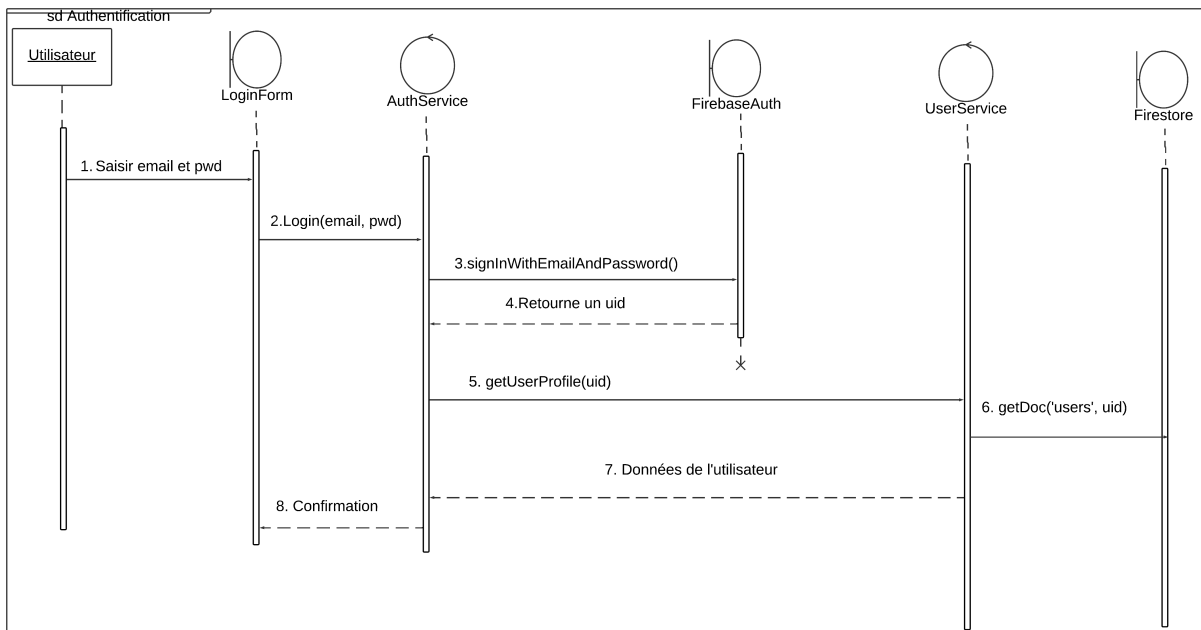


FIG. 23 – Diagramme de séquence détaillé de “Connexion”

– Cas d’utilisation “Signalement d’incidents” illustré par la figure 24.

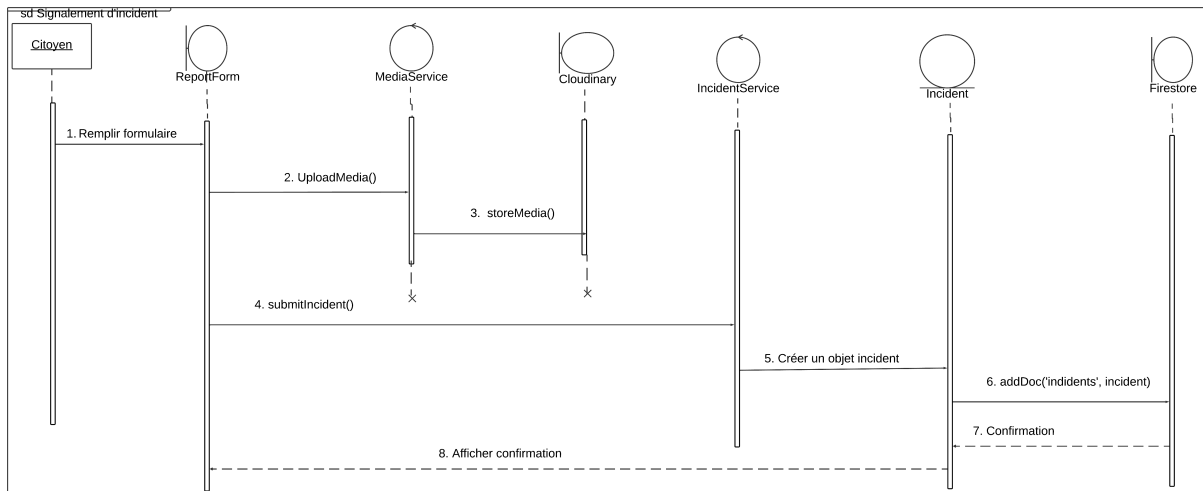


FIG. 24 – Diagramme de séquence détaillé de “Signalement d’incidents”

– Cas d’utilisation “Gestion des utilisateurs”

Ce cas d’utilisation se décompose en les scénarios suivants:

– Scénario « Ajouter un utilisateur » illustré par la figure 25.

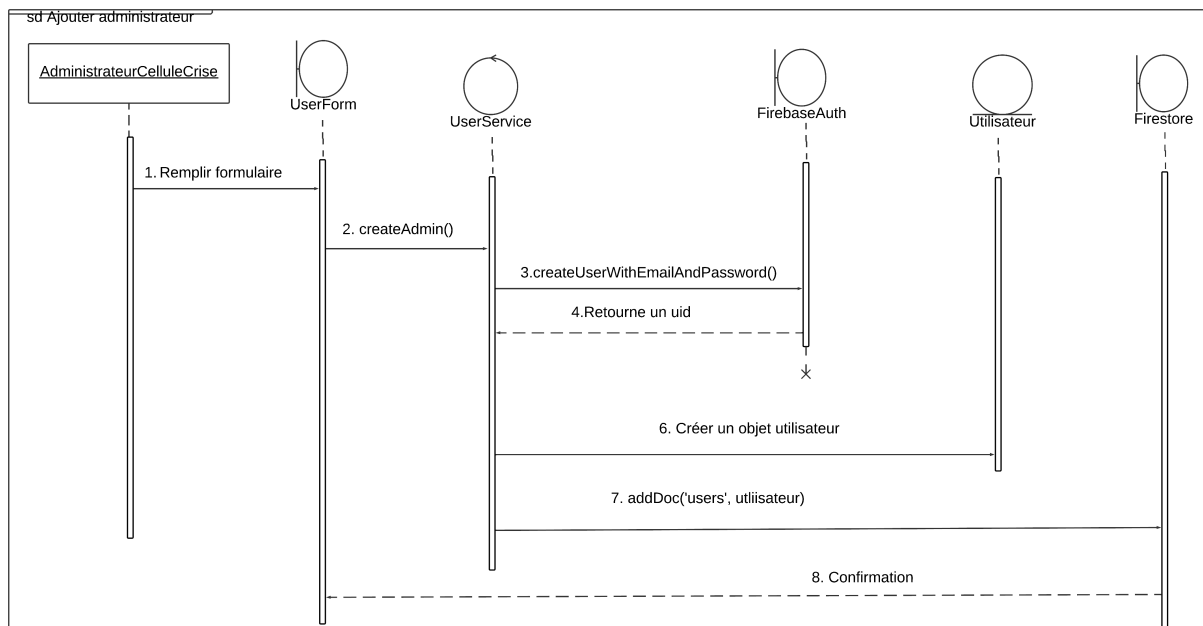


FIG. 25 – Diagramme de séquence détaillé de “Ajouter un utilisateur”

– Scénario « Modifier un utilisateur » illustré par la figure 26.

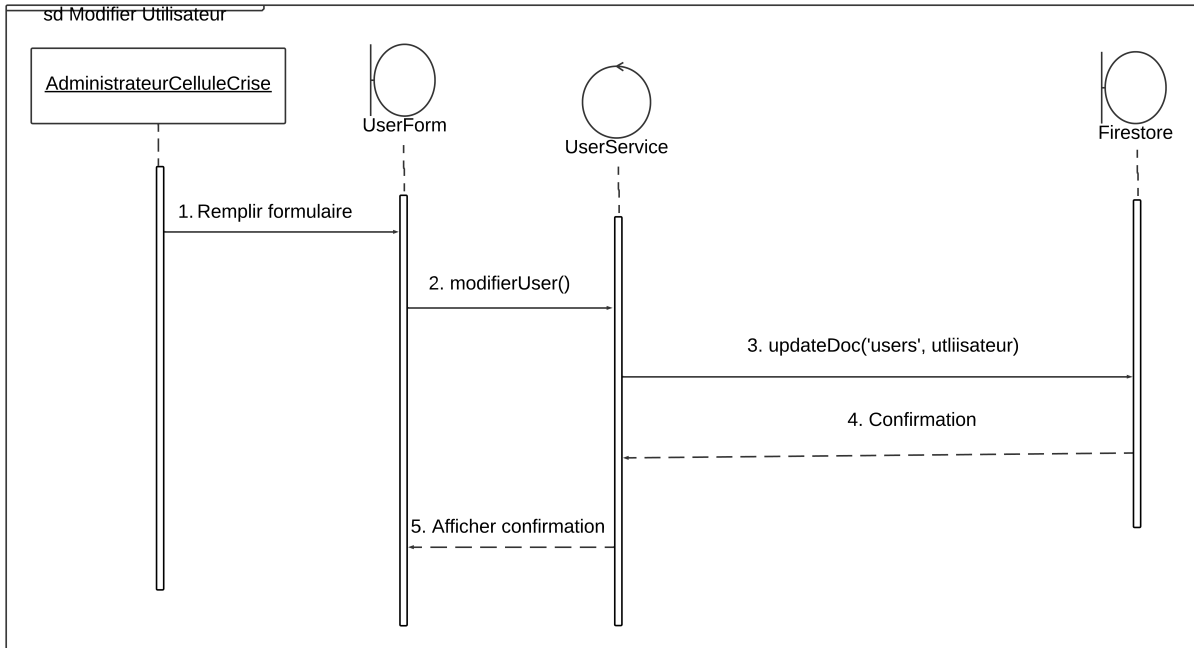


FIG. 26 – Diagramme de séquence détaillé de “Modifier un utilisateur”

– Scénario « Supprimer un utilisateur » illustré par la figure 27.

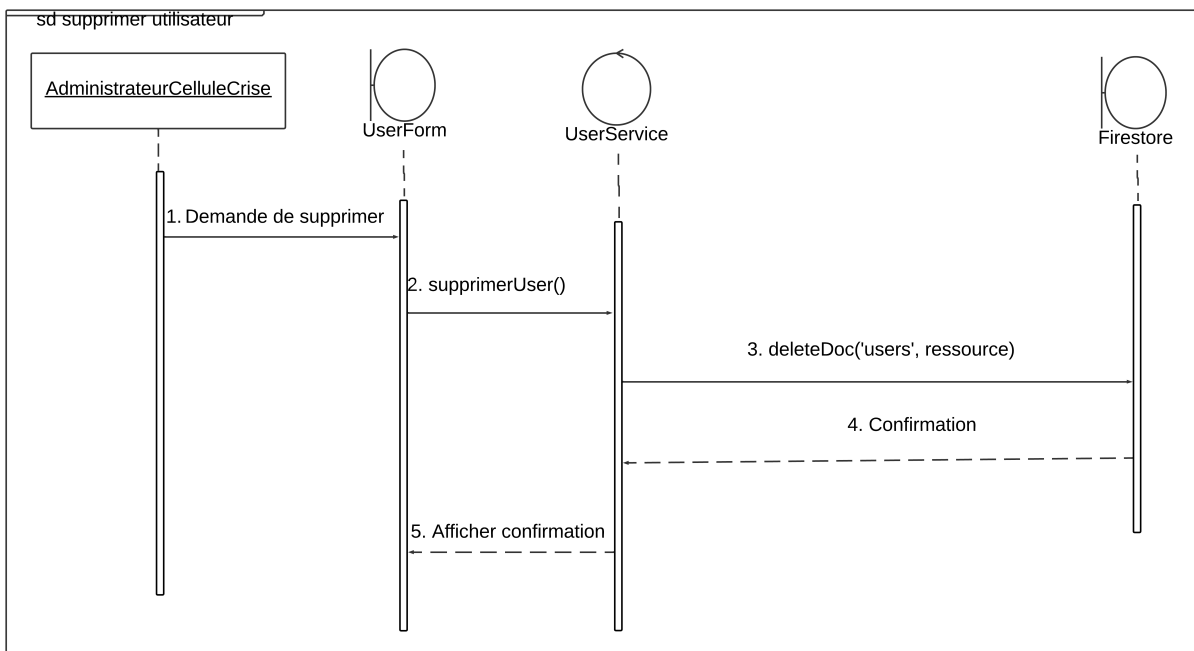


FIG. 27 – Diagramme de séquence détaillé de “Supprimer un utilisateur”

– Scénario « Suspendre un utilisateur » illustré par la figure 28.

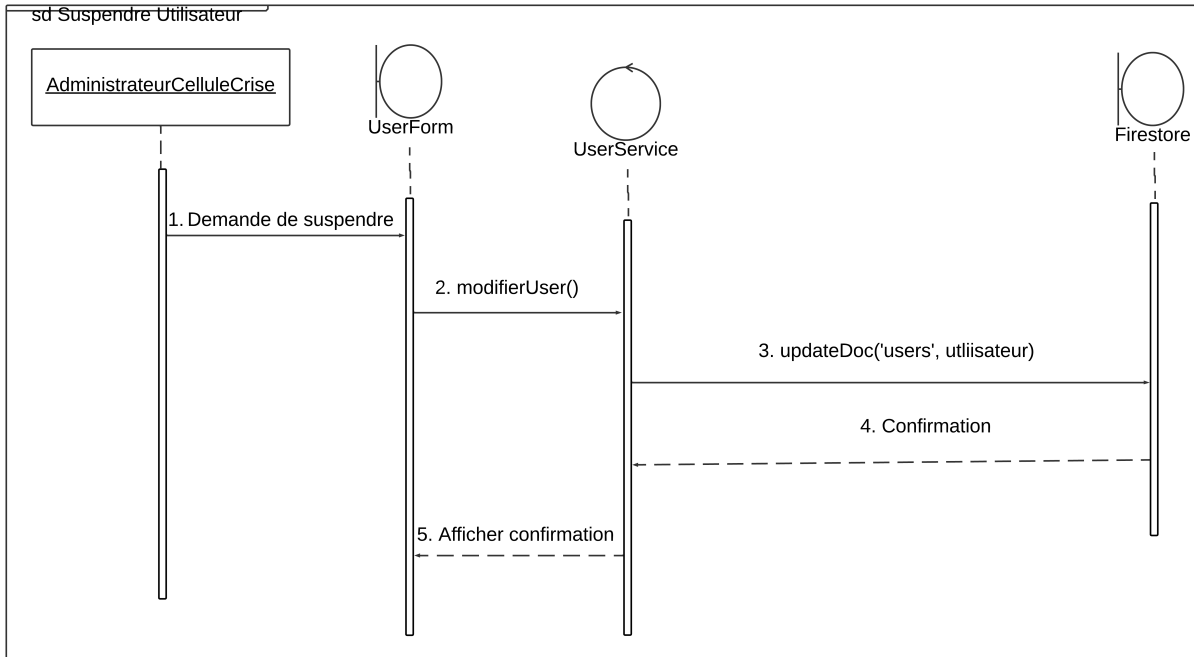


FIG. 28 – Diagramme de séquence détaillé de “Suspendre un utilisateur”

– Diagramme de classe

La figure 29 représente le diagramme de classe du sprint 1.

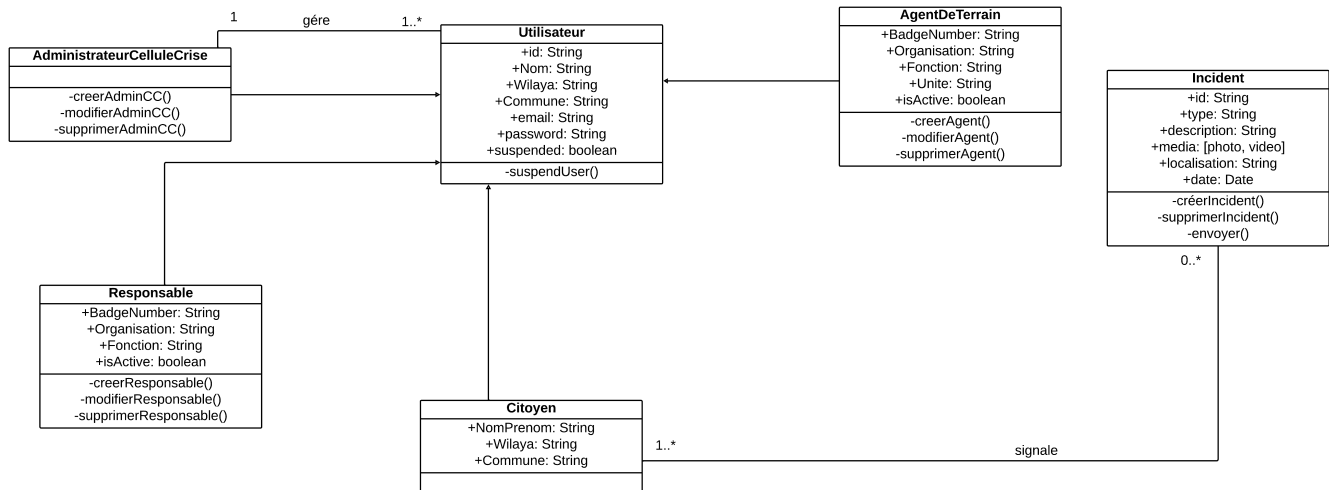


FIG. 29 – Diagramme de classe du sprint 1

#### ★ Implémentation

Nous commençant le développement de chaque user story du sprint 1 backlog avec l’éditeur de texte Vs code, en utilisant Firebase comme Backend av Firebase authentification avec email et mot de passe, Firestore comme base de données et

#### ★ Test de fonctionnalités

Pour s’assurer du bon fonctionnement du livrable à la fin du sprint, nous avons effectuer:

- Des tests manuelles sur le navigateur et l’émulateur Expo Go.
- Des tests automatisés effectués via le service Test Lab de Firebase.

★ Présentation des interfaces

Dans cette partie, nous allons présenter les interfaces utilisateurs développés dans ce sprint.

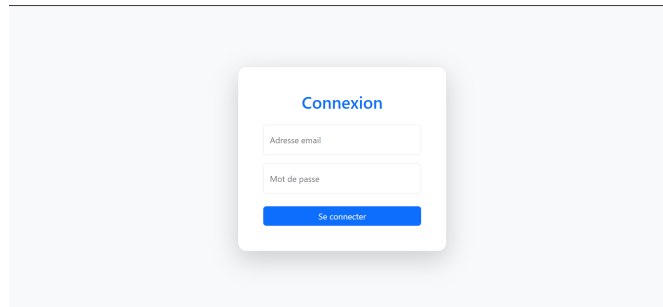


FIG. 30 – Interfaces de Connexion de l'administrateur

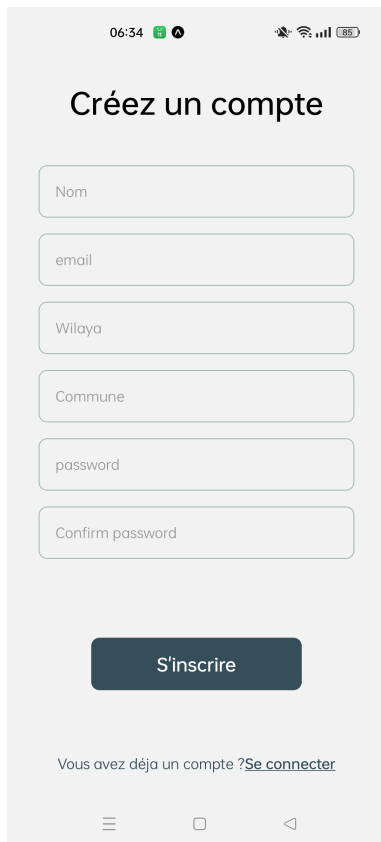


FIG. 31 – Interfaces de Inscription des citoyens

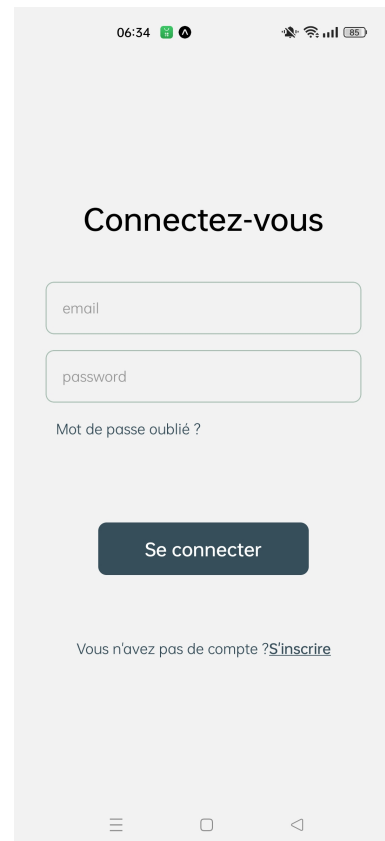


FIG. 32 – Interfaces de Connexion des utilisateurs de l'application mobile

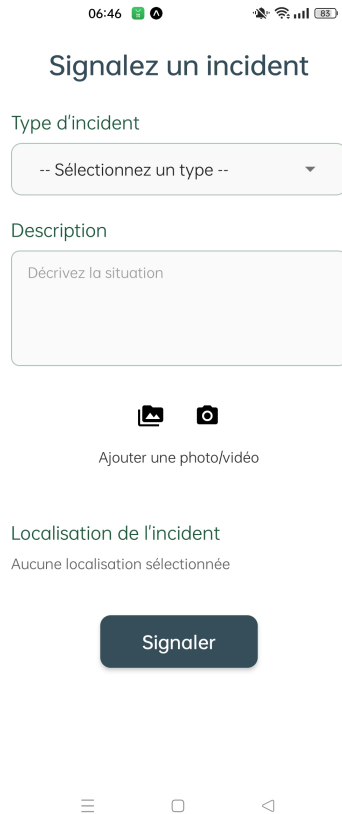


FIG. 33 – Interfaces de Signalement d'incident



FIG. 34 – Interfaces de Gestion des utilisateurs citoyens

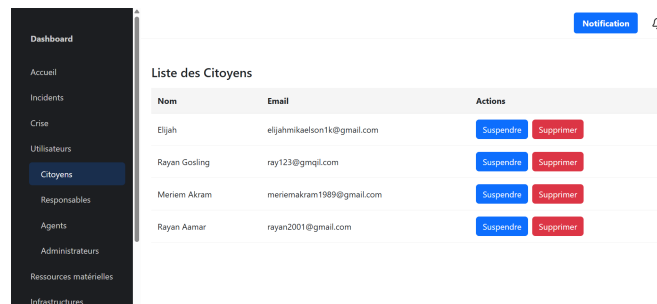


FIG. 35 – Interfaces de Gestion des utilisateurs agents de terrain



FIG. 36 – Interfaces de Gestion des utilisateurs responsables

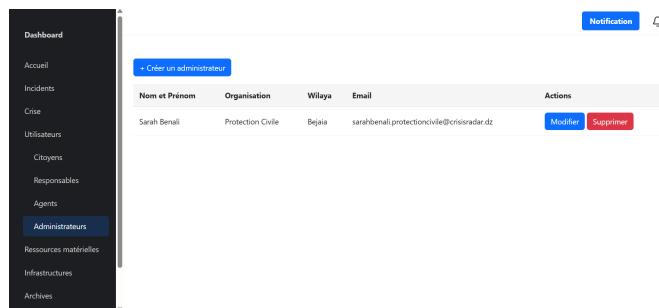


FIG. 37 – Interfaces de Gestion des administrateurs

### 3.7.3 Post-phase

Pendant cette phase, nous allons, non seulement, évaluer la version fonctionnelle et opérationnelle du produit résultant de ce sprint avec le product owner, mais aussi, évaluer le flux de travail pendant le sprint.

#### – Sprint review

Pendant une réunion avec le scrum master et product owner, ce dernier a émet de la satisfaction vis-à-vis du produit développé pendant le premier sprint.

Le product owner souligne une condition en ce qui concerne le développement d'une fonctionnalité du deuxième sprint, qui est que, lors de la gestion de ressources humaines, l'administrateur de l'organisation fait la distinction entre agents et responsable lui même à travers le tableau de bord.

#### – Sprint retrospective

Lors d'une réunion avec le scrum master, on a fait un point sur la qualité du flux de travail en prenant en compte les stand-ups quotidiens pendant le sprint. On a observé un non respect du délai délimité pour le développement du sprint, et après une discussion approfondie, on a déduis la cause de ces retards qui est une mauvaise organisation du travail pendant la journée.

Afin de remédier à ce problème, pour un meilleur déroulement du processus de développement, nous avons décider d'établir des emplois du temps plus rigoureux permettant de développer une consistance dans le quotidien des développeurs.

**Réalisation du sprint n°02** Dans cette partie, nous allons décrire le déroulement du processus de développement du deuxième sprint.

#### ★ Analyse

– Diagramme de cas d'utilisation

La figure 38 représente le diagramme de cas d'utilisation du sprint 2.

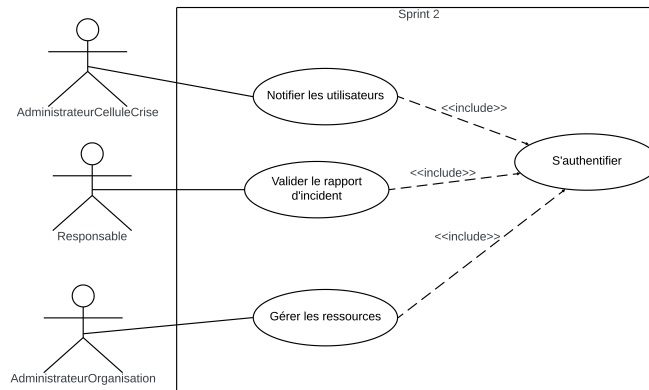


FIG. 38 – Diagramme de cas d'utilisation du sprint 2

– Description textuelle A travers cette section, nous allons définir la description textuelle de chaque cas d'utilisation du sprint 01:

- Cas d'utilisation “Gérer les ressources” illustré par le tableau 10.

Description textuelle
<p><b>Nom du cas:</b> Gestion des ressources</p> <p><b>But:</b> Permettre à l’administrateur d’une organisation de gérer les ressources de cette dernière.</p> <p><b>Acteur:</b> Administrateur d’une organisation</p>
<p><b>Séquencement:</b> l’administrateur est devant la page de ressources.</p> <p><b>Pré-conditions:</b> Administrateur de l’organisation est connecté à son compte utilisateur.</p> <p>Le compte utilisateur de l’administrateur n’est pas suspendu. <b>Enchaînement nominale:</b></p> <ol style="list-style-type: none"> <li>1. L’administrateur clique sur un bouton (ajouter, modifier ou supprimer).</li> <li>2. Le système affiche un formulaire.</li> <li>3. L’administrateur clique sur le bouton “confirmer”</li> <li>4. Le système vérifie les données et valide.</li> <li>5. Le système exécute l’action.</li> </ol> <p><b>Enchaînement alternatif:</b></p> <p>A1. L’administrateur annule son action. A partir du point 2.</p> <p>Le séquencement s’arrête ici.</p> <p><b>Post-conditions:</b></p> <p>Les informations sont mises à jour dans Firestore.</p>

TAB. 10 – Description textuelle du cas d'utilisation “Gestion des ressources”

- Cas d'utilisation “Notifier les utilisateurs” illustré par le tableau 11.

Description textuelle
<p><b>Nom du cas:</b> Notification des utilisateurs  <b>But:</b> Permettre à un utilisateur d'envoyer des notification.  <b>Acteur:</b> L'administrateur de la cellule de crise.</p>
<p><b>Séquencement:</b> L'administrateur de la cellule de crise est devant la page d'accueil.  <b>Pré-conditions:</b>  L'administrateur de la cellule de crise est déjà connecté.  <b>Enchaînement nominale:</b>  1.L'administrateur de la cellule de crise clique sur le bouton "envoyer une notification".  2.L'administrateur de la cellule de crise remplit le formulaire et clique sur le bouton "envoyer"  3. Le système vérifie les informations et valide.  <b>Enchaînement alternatif:</b>  A1. Les champs du formulaire ne sont pas tous remplis. A partir du point 3.  3. Le système affiche un message d'alerte.  4. L'utilisateur ressaisie ses informations et clique sur le bouton "envoyer".  Le séquencement continue à partir du point 2.  <b>Post-conditions:</b>  Les utilisateurs reçoivent les notifications.</p>

TAB. 11 – Description textuelle du cas d'utilisation "Notification des utilisateurs"

– Cas d'utilisation "Valider le rapport d'incident" illustré par le tableau 12.

Description textuelle
<p><b>Nom du cas:</b> Validation du rapport de crise.  <b>But:</b> Permettre à un responsable de valider ou de réfuter un rapport de crise.  <b>Acteur:</b> responsable</p>
<p><b>Séquencement:</b> le responsable est devant la page des rapports.  <b>Pré-conditions:</b>  L'application est installée sur l'appareil du responsable.  Le compte utilisateur n'est pas suspendu. <b>Enchaînement nominale:</b>  1. Le responsable clique sur le bouton "valider" ou le bouton "réfuter".  2. Le système affiche le formulaire.  3. Le responsable remplit le formulaire et clique sur le bouton "confirmer".  4. Le système vérifie les données et valide.  5. Le système affiche un message de succès.  <b>Enchaînement alternatif:</b>  Les champs du formulaire ne sont pas tous remplis. A partir du point 4.  3. Le système affiche un message d'alerte.  4. L'utilisateur ressaisie ses informations et clique sur le bouton "envoyer".  Le séquencement continue à partir du point 4.  <b>Post-conditions:</b>  En cas de réfutation, une notification est envoyé à l'administrateur de la cellule de crise.  En cas de validation, un rapport détaillé est envoyé à l'administrateur de la cellule de crise.</p>

TAB. 12 – Description textuelle du cas d'utilisation "Validation des rapports d'incident"

– Diagramme de séquence

Dans cette partie, nous allons présenter les diagrammes de séquence système qui consiste à décrire les interactions entre les acteurs et le système.

– Cas d'utilisation "Gestion des ressources" illustré par la figure 39.

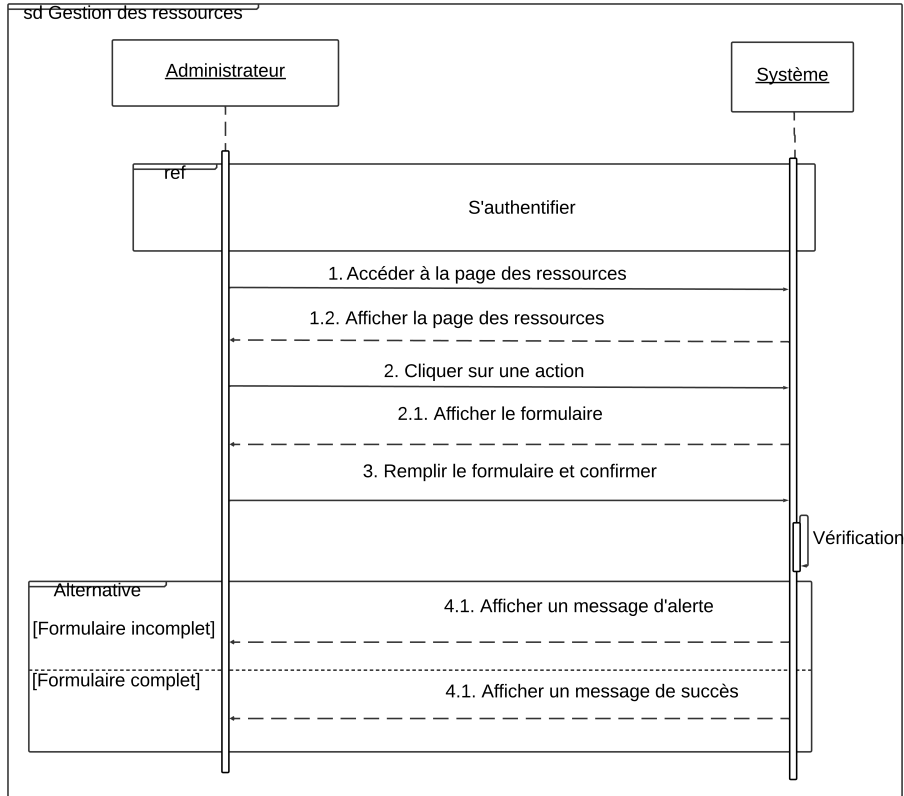


FIG. 39 – Diagramme de séquence “Gestion des ressources”

– Cas d’utilisation “Notifier les utilisateurs” illustré par la figure 40.

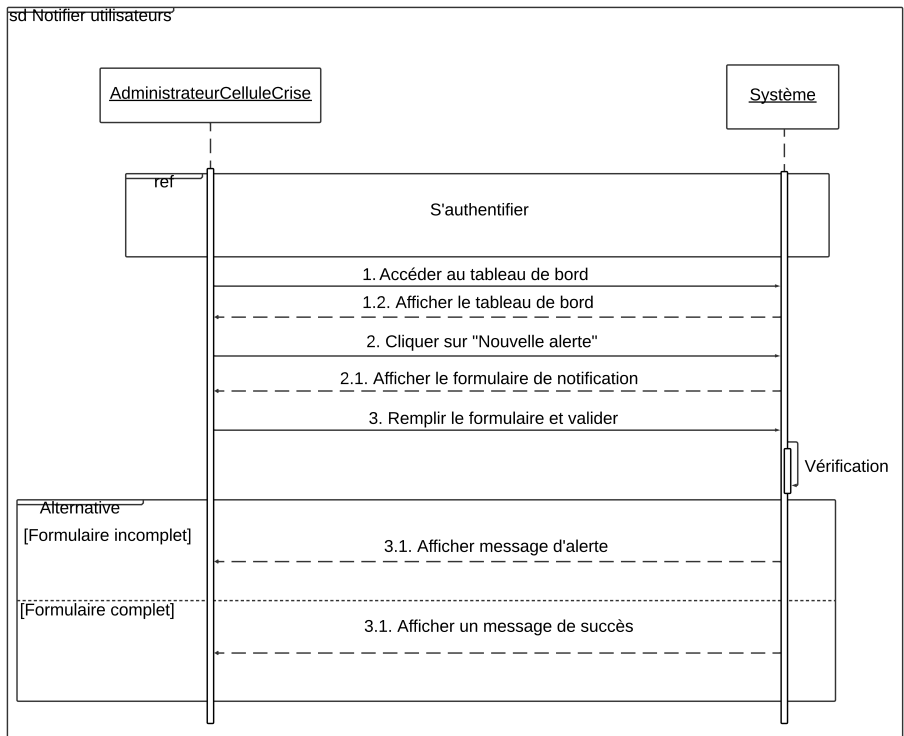


FIG. 40 – Diagramme de séquence “Notifier les utilisateurs”

– Cas d'utilisation "Valider le rapport d'incident" illustré par la figure 41.

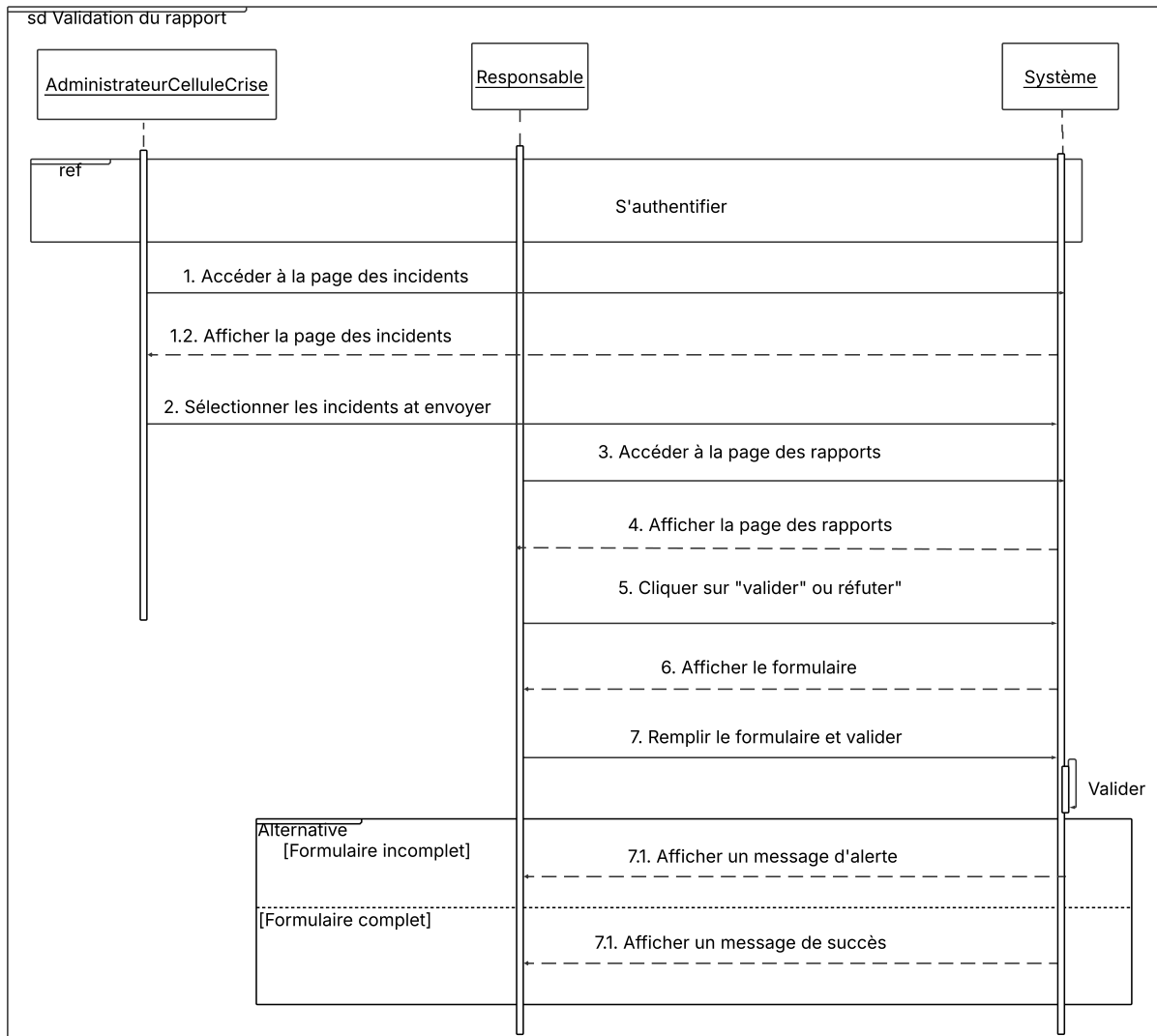


FIG. 41 – Diagramme de séquence "Valider le rapport d'incident"

★ Conception

– Diagrammes de séquence détaillés

Dans cette partie, nous allons présenter les diagrammes de séquence détaillés qui consiste à décrire les interactions entre les objets du système.

– Cas d'utilisation "Gestion des utilisateurs"

Ce cas d'utilisation se décompose en plusieurs scénarios, qui sont:

– Scénario « Ajouter une ressources » illustré par la figure 42.

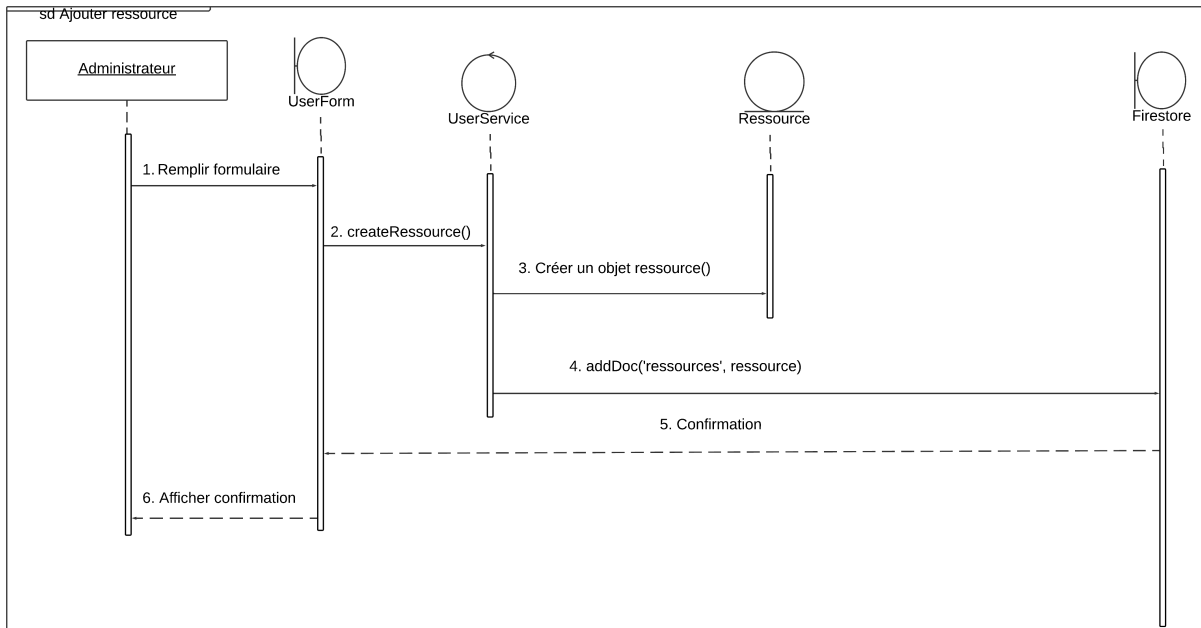


FIG. 42 – Diagramme de séquence détaillé de “Ajouter une ressource”

– Scénario « Modifier une ressource » illustré par la figure 43.

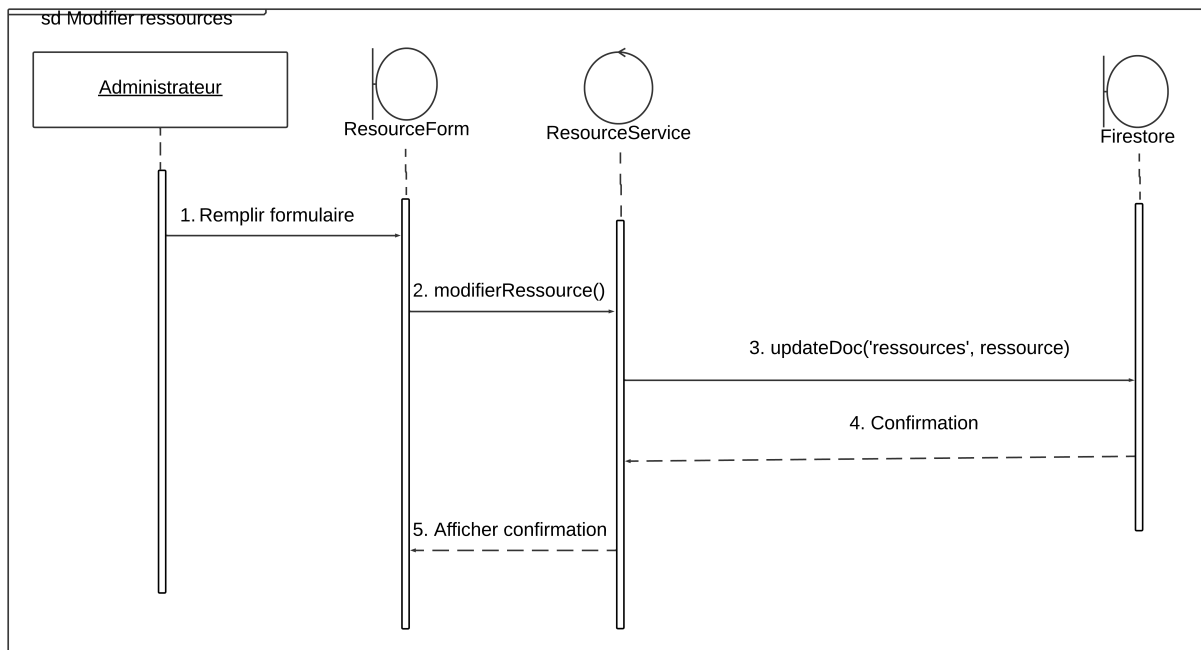


FIG. 43 – Diagramme de séquence “Modifier une ressource”

– Scénario « Supprimer une ressource » illustré par la figure 44.

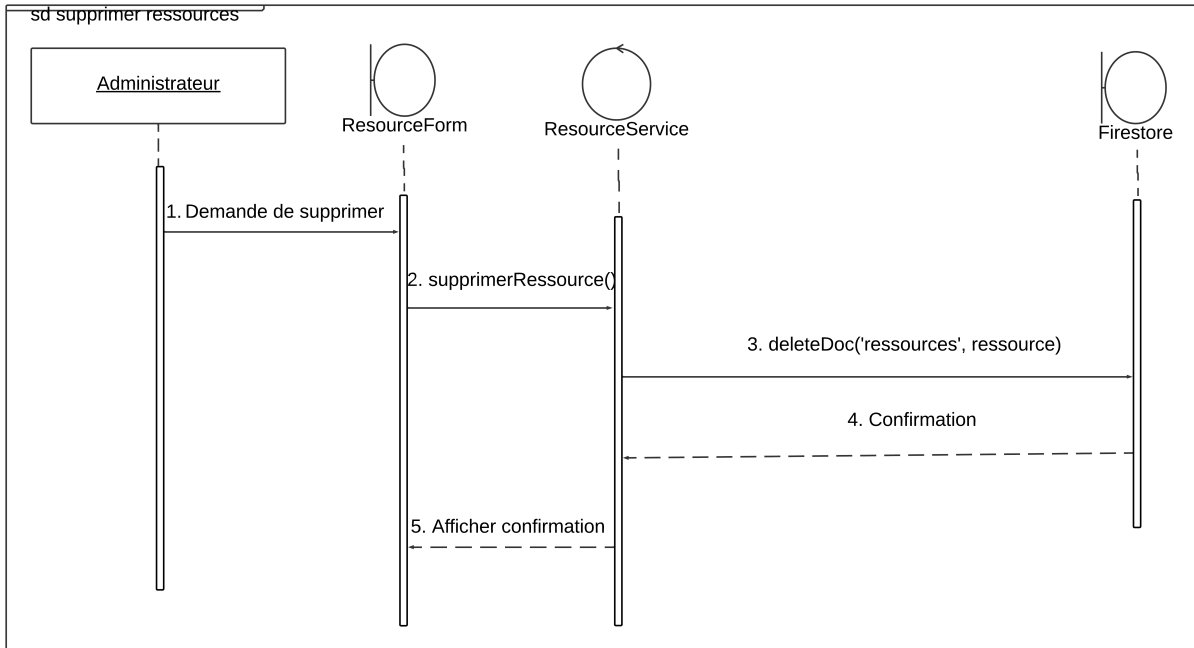


FIG. 44 – Diagramme de séquence “Supprimer une ressource”

– Cas d’utilisation “Notifier les utilisateurs” illustré par la figure 45.

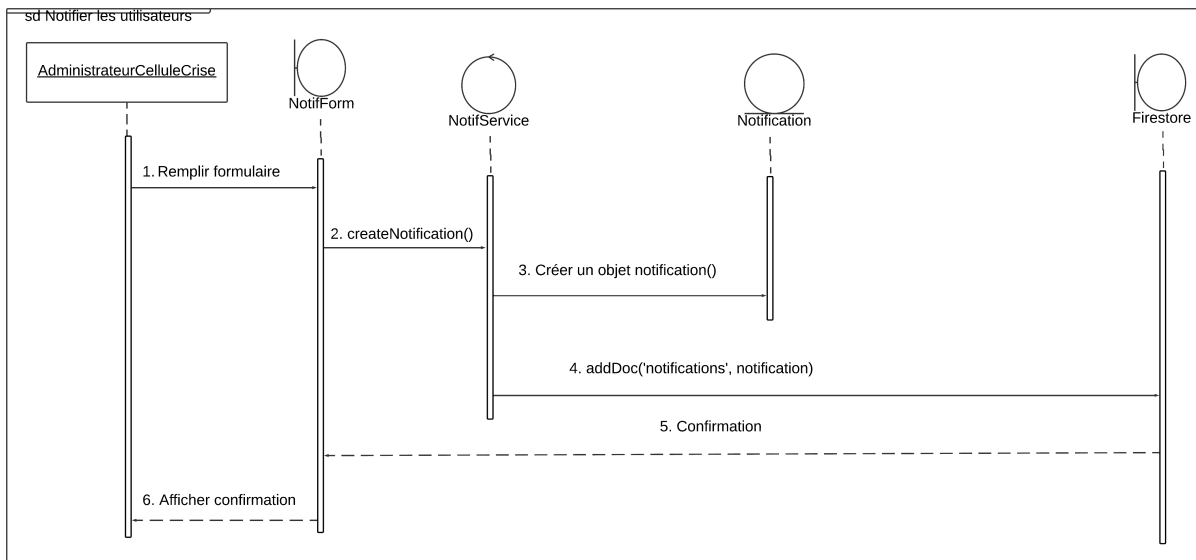


FIG. 45 – Diagramme de séquence détaillé “Notifier les utilisateurs”

– Cas d’utilisation “Valider le rapport d’incidents” illustré par la figure 46.

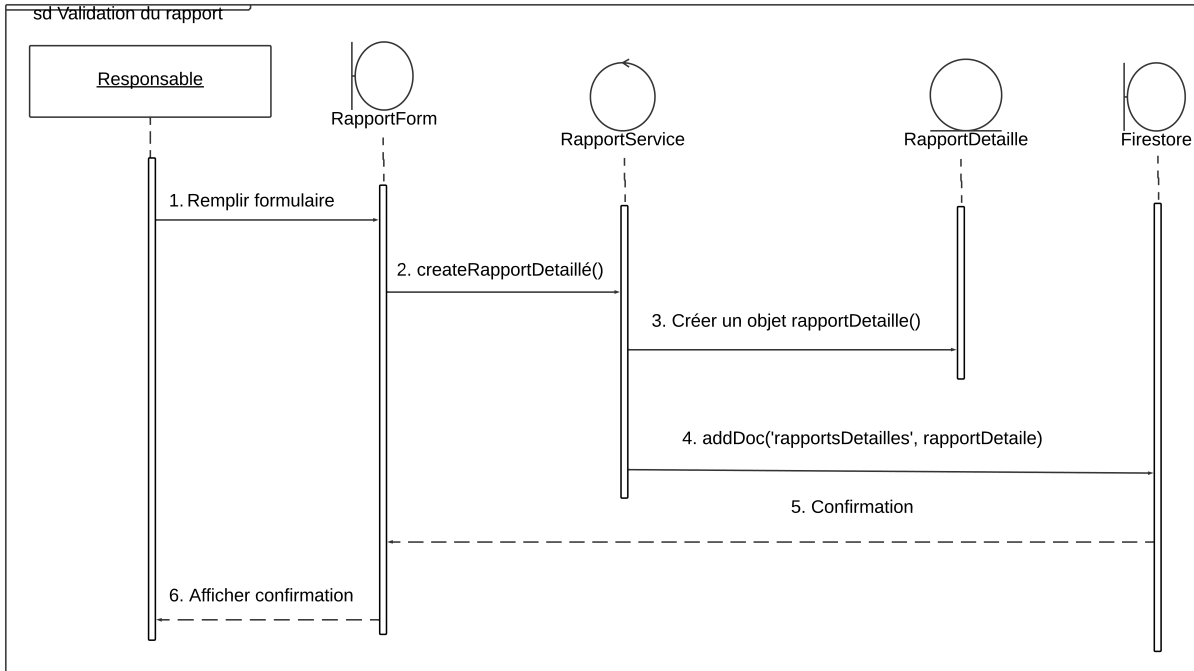


FIG. 46 – Diagramme de séquence détaillé de “Valider le rapport d’incident”

– Diagramme de classe

La figure 47 représente le diagramme de classe du sprint 2.

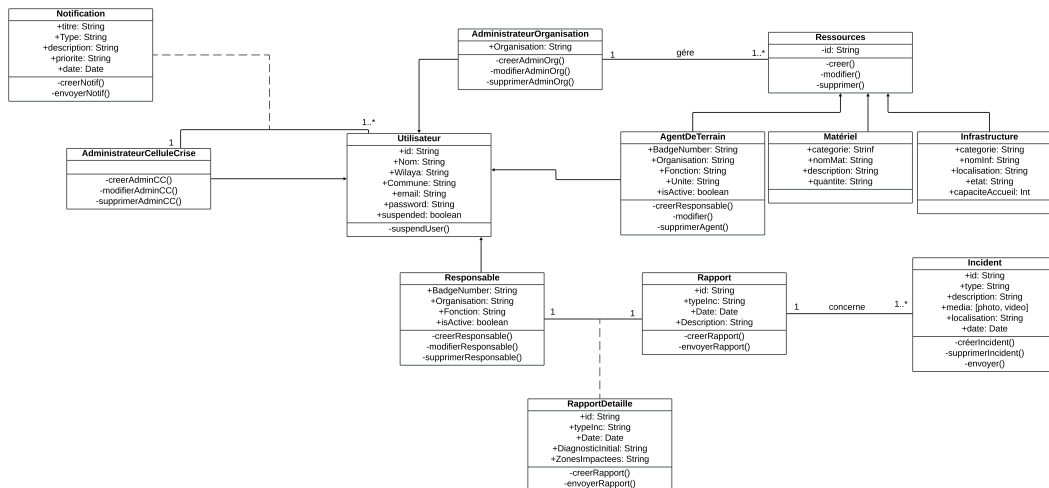


FIG. 47 – Diagramme de classe du sprint 2

★ Implémentation

Nous utilisant Firebase authentication avec email et mot de passe et Firestore comme base de données.

★ Test de fonctionnalités

Pour s’assurer du bon fonctionnement du livrable à la fin du sprint, nous avons effectuer:

- Des tests manuelles sur le navigateur et l’émulateur Expo Go.
- Des tests automatisés effectués via le service Test Lab de Firebase.

★ Présentation des interfaces

Dans cette partie, nous allons présenter les interfaces utilisateurs développées dans ce sprint.

Dashboard

Accueil

Responsables

**Agents**

Ressources matérielles

Infrastructures

Notification

### Liste des agents de terrain

+ Créer un agent

Numéro de badge	Nom et Prénom	Fonction	Organisation	Unité	Wilaya	Statut	Actions
PC439000	Zineb Aidat	Paramédicale	Protection Civile	100	Béjaia	Actif	Modifier Supprimer
PC320043	Amine Belkacem	Coordinateur	Protection Civile	123		Actif	Modifier Supprimer
PC125460	Maya Bishop	Pompier	Protection Civile	108		Non actif	Modifier Supprimer
PC430043	Rayan Amran	Secouriste	Protection civile	130		Actif	Modifier Supprimer

FIG. 48 – Interfaces de Gestion des ressources agents

Dashboard

Accueil

Responsables

Agents

Ressources matérielles

**Infrastructures**

Notification

### Liste des Infrastructures disponibles

+ Créer une infrastructure

Nom	Catégorie	Description	Adresse	Etat	Capacité d'accueil	Organisation	Actions
Centre de Secours	Centre d'hébergement	Centre temporaire d'accueil pour les personnes évacuées	Boulevard Amirouche, Béjaia	En rénovation	200	Protection Civile	Modifier Supprimer

FIG. 49 – Interfaces de Gestion des infrastructures

Dashboard

Accueil

Responsables

Agents

**Ressources matérielles**

Infrastructures

Notification

### Liste des Ressources disponibles

+ Créer une ressource

Nom	Catégorie	Description	Quantité	Organisation	Actions
Camion-citerne incendie	Véhicule	Véhicule équipé pour la lutte contre les incendies, notamment dans les zones rurales ou difficiles d'accès.	3	Protection civile	Modifier Supprimer

FIG. 50 – Interfaces de Gestion des ressources responsables

Dashboard

Accueil

**Responsables**

Agents

Ressources matérielles

Infrastructures

Notification

### Liste des Responsables

+ Créer un responsable

Badge	Nom	Fonction	Unité	Organisation	Actions
PC430055	Anais Azout	Chef de garde	32	Protection Civile	Modifier Supprimer
PC000032	Abdenour Boukhefifa	Capitaine		Protection civile	Modifier Supprimer
PC650011	Abdenour Belaid	Directeur		Protection Civile	Modifier Supprimer
PC453211	Amira Boukhefifa	paramédicale		Protection civile	Modifier Supprimer
PC708790	Karim Mebarli	Pompier		Protection civile	Modifier Supprimer

FIG. 51 – Interfaces de Gestion des ressources matérielles

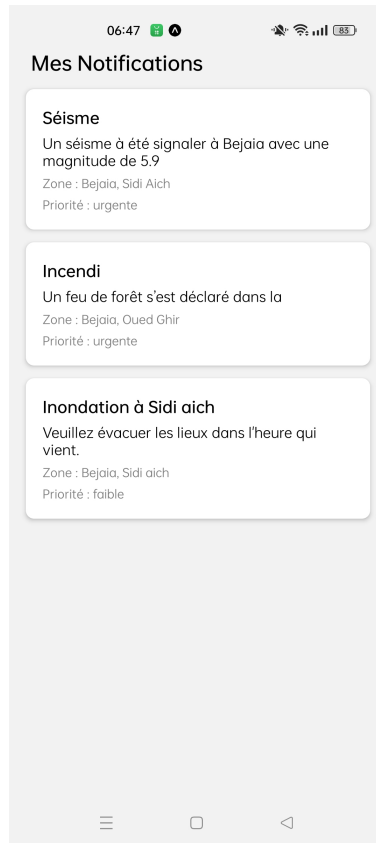


FIG. 52 – Interfaces de Notifications

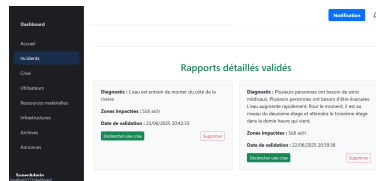


FIG. 53 – Interfaces de validation des rapports

### 3.7.4 Post-phase

Pendant cette phase, nous allons, non seulement, évaluer la version fonctionnelle et opérationnelle du produit résultant de ce sprint avec le product owner, mais aussi, évaluer le flux de travail pendant le sprint.

- Sprint review  
Pendant une réunion avec le scrum master et product owner, ce dernier a émet de la satisfaction vis-à-vis du produit développé pendant le deuxième sprint.
- Sprint retrospective  
Lors d'une réunion avec le scrum master, après l'implémentation d'un emploi du temps plus rigoureux, nous avons observer des améliorations au niveau du flux de travail avec les délais de développement respectés.

**Réalisation du sprint n°03** Dans cette partie, nous allons décrire les composants du processus de développement du troisième sprint.

★ Analyse

– Diagramme de cas d'utilisation

La figure 54 est une illustration du diagramme de cas d'utilisation du sprint 3.

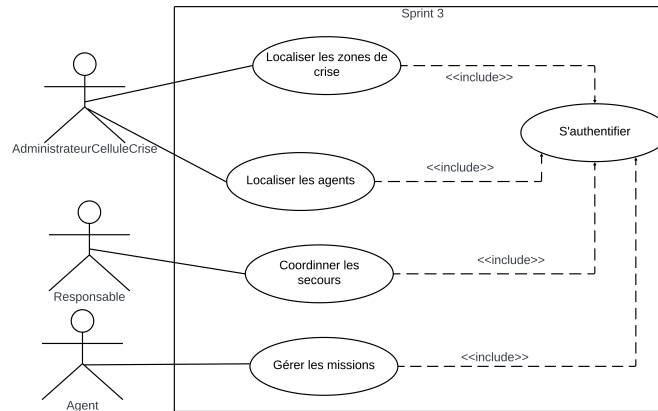


FIG. 54 – Diagramme de cas d'utilisation du sprint 3

– Description textuelle

A travers cette section, nous allons définir la description textuelle de chaque cas d'utilisation du sprint 01:

- Cas d'utilisation "Localiser les agents" illustré par le tableau 13.

Description textuelle
<p><b>Nom du cas:</b> Géolocalisation des agents.  <b>But:</b> Permettre à l'administrateur de localiser les zones impactées par la crise.  <b>Acteur:</b> Administrateur de la cellule de crise.</p>
<p><b>Séquencement:</b> l'administrateur est devant la page d'accueil du tableau de bord.  <b>Pré-conditions:</b>  l'administrateur est déjà connecté.  L'agent a accepté une mission.  <b>Enchaînement nominale:</b>  1. Le citoyen remplit le formulaire et clique sur le bouton "se connecter"  2. Le système vérifie les identifiants de connexion et valide.  3. Le système affiche la page d'accueil de l'application.  4.  <b>Enchaînement alternatif:</b>  A1. Les identifiants de connexion sont invalides. A partir du point 2.  3. Le système affiche un message d'erreur.  4. L'utilisateur ressaisie ses informations et clique sur le bouton "se connecter".  Le séquencement continue à partir du point 2.  A2. l'utilisateur a oublié son mot de passe. à partir du point 1.  1. L'utilisateur clique sur le lien "Mot de passe oublié". 2. Le système affiche la page de réinitialisation du mot de passe.  3. L'utilisateur saisie son email. 4. Le système envoie un email de réinitialisation.  Le séquencement reprend à partir du point 1. <b>Post-conditions:</b>  Le compte utilisateur n'est pas suspendu.</p>

TAB. 13 – Description textuelle du cas d'utilisation "Géolocalisation des agents."

– Cas d'utilisation "Localiser les zones de crise" illustré par le tableau 14.

Description textuelle
<p><b>Nom du cas:</b> Géolocalisation des zones de crise.  <b>But:</b> Permettre à l'administrateur de localiser les zones impactées par la crise.  <b>Acteur:</b> Administrateur de la cellule de crise.</p>
<p><b>Séquencement:</b> l'administrateur est devant la page d'accueil du tableau de bord.  <b>Pré-conditions:</b>  L'administrateur est déjà connecté.  L'administrateur a déjà déclenché une crise. <b>Enchaînement nominale:</b>  1. Le système récupère les coordonnées et informations de la crise  2. Le système affiche la zone de crise sur une carte avec un cercle rouge.  <b>Enchaînement alternatif:</b>  A1. Il n'y a pas de crise en cours.  3. Le système affiche une carte vierge.  <b>Post-conditions:</b></p>

TAB. 14 – Description textuelle du cas d'utilisation "Géolocalisation des crises"

– Cas d'utilisation "Coordonner les secours" illustré par le tableau 15.

Description textuelle
<p><b>Nom du cas:</b> coordination des secours  <b>But:</b> Permettre à un responsable de coordonner les agents de terrain.  <b>Acteur:</b> responsable</p>
<p><b>Séquencement:</b> le responsable est devant l'écran de coordination.  <b>Pré-conditions:</b>  L'application est installée sur l'appareil du responsable.  Le responsable est déjà connecté.  Le compte utilisateur n'est pas suspendu. <b>Enchaînement nominale:</b>  1. Le responsable clique sur le bouton "créer une mission".  2. Le système affiche le formulaire.vérifie les identifiants de connexion et valide.  3. Le responsable remplit le formulaire et clique sur le bouton "créer".  4.Le système vérifie les champs du formulaire et valide.  5. Le système affiche un message de succès. <b>Enchaînement alternatif:</b>  A1. Un champs obligatoire est vide. A partir du point 2.</p> <p>3. Le système affiche un message d'alerte.  4. Le responsable ressaisie ses informations et clique sur le bouton "créer".  Le séquencement continue à partir du point 2.  <b>Post-conditions:</b>  Les missions sont affichés aux destinataires.</p>

TAB. 15 – Description textuelle du cas d'utilisation "Coordination des secours"

– Cas d'utilisation "Gérer les missions" illustré par le tableau 16.

Description textuelle
<p><b>Nom du cas:</b> Gestion des mission  <b>But:</b> Permettre à un agent de gérer ses missions.  <b>Acteur:</b> agent</p>
<p><b>Séquencement:</b> l'agent est devant la page des missions.  <b>Pré-conditions:</b>  L'application est installée sur l'appareil de l'agent.  L'agent est déjà connecté.  Le compte utilisateur n'est pas suspendu. <b>Enchaînement nominale:</b>  1. L'agent clique sur le bouton "accepter" de la mission.  2. Le système met à jour l'état de la mission.  3. L'agent clique sur le bouton "terminer" de la mission.  4.Le système met à jour l'état de la mission. <b>Enchaînement alternatif:</b>  Aucun <b>Post-conditions:</b>  Les missions sont mises à jour chez le responsable.</p>

TAB. 16 – Description textuelle du cas d'utilisation "Gestion des mission"

– Diagramme de séquence

Dans cette partie, nous allons présenter les diagrammes de séquence système qui consiste à décrire les interactions entre les acteurs et le système.

– Cas d'utilisation "Localisation des agents" illustré par la figure 55.

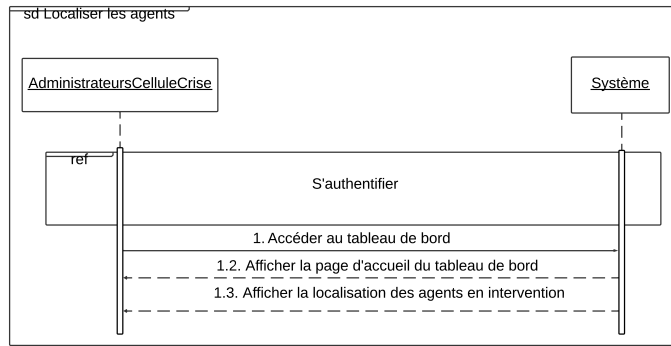


FIG. 55 – Diagramme de séquence de “Localisation des agents”

– Cas d’utilisation “Localisation des zones de crises” illustré par la figure 56.

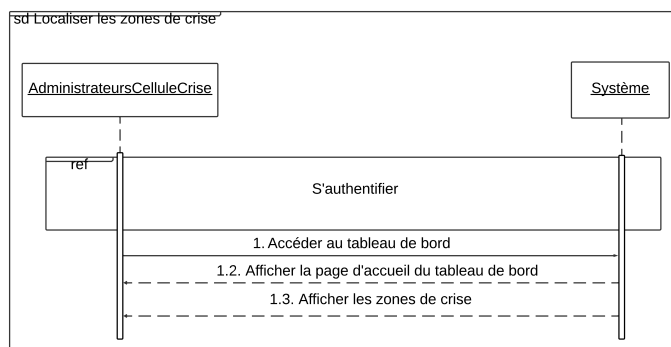


FIG. 56 – Diagramme de séquence de “Localisation des zones de crise”

– Cas d’utilisation “Coordination des secours” illustré par la figure 57.

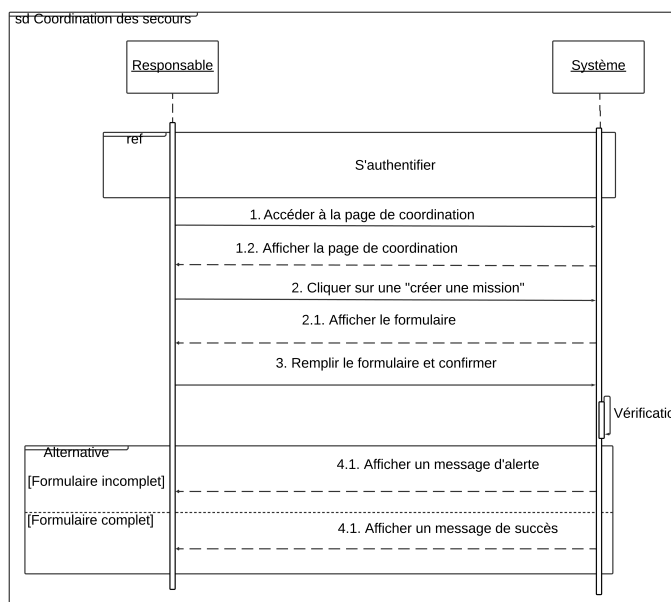


FIG. 57 – Diagramme de séquence de “Coordination des secours”

– Cas d’utilisation “Gestion des missions” illustré par la figure 58.

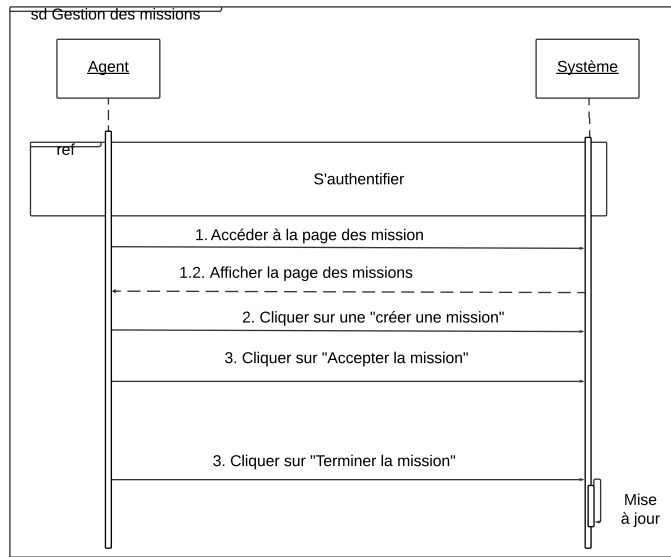


FIG. 58 – Diagramme de séquence de “Gestion des missions”

★ Conception

Cette étape vise à définir les interactions entre les acteurs et les objets du système à travers des diagrammes.

- Diagrammes de séquence détaillés
  - Dans cette partie, nous allons présenter les diagrammes de séquence détaillés qui consiste à décrire les interactions entre les objets du système.
- Cas d'utilisation “Localisation des agents” illustré par la figure 59.

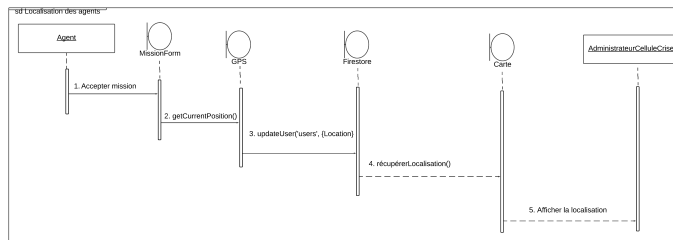


FIG. 59 – Diagramme de séquence détaillé de “Localisation des agents”

- Cas d'utilisation “Localisation des zones de crises” illustré par la figure 55.

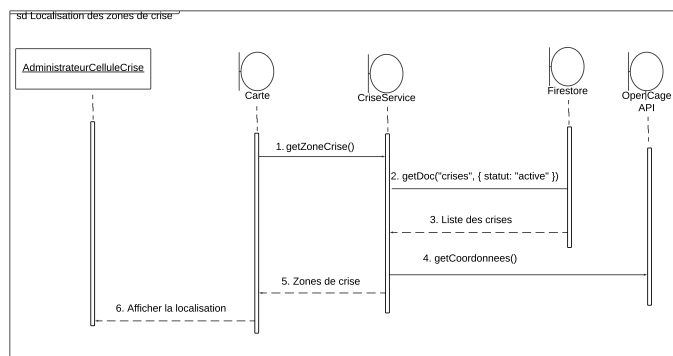


FIG. 60 – Diagramme de séquence détaillé de “Localisation des zones de crises ”

- Cas d'utilisation “Coordination des secours” illustré par la figure 60.

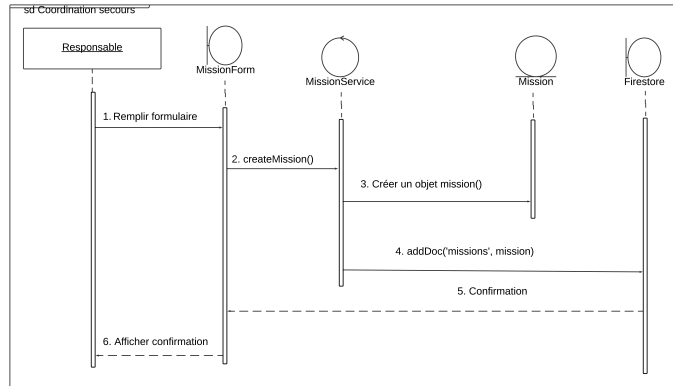


FIG. 61 – Diagramme de séquence détaillé de “Coordination des secours”

- Cas d’utilisation “Gestion des missions” illustré par la figure 61.

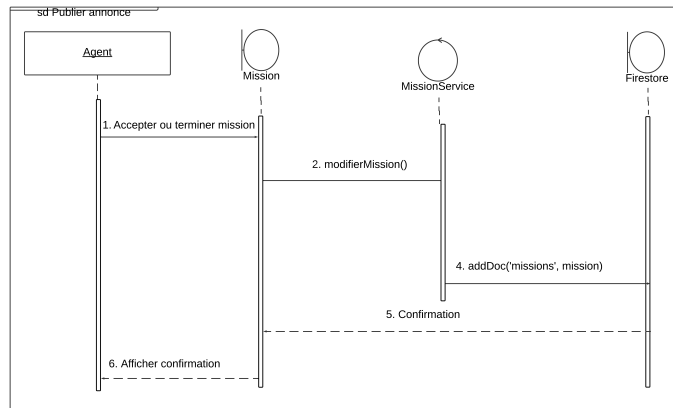


FIG. 62 – Diagramme de séquence détaillé de “Gestion des missions”

- Diagramme de classe  
La figure 62 représente le diagramme de classe du sprint 3.

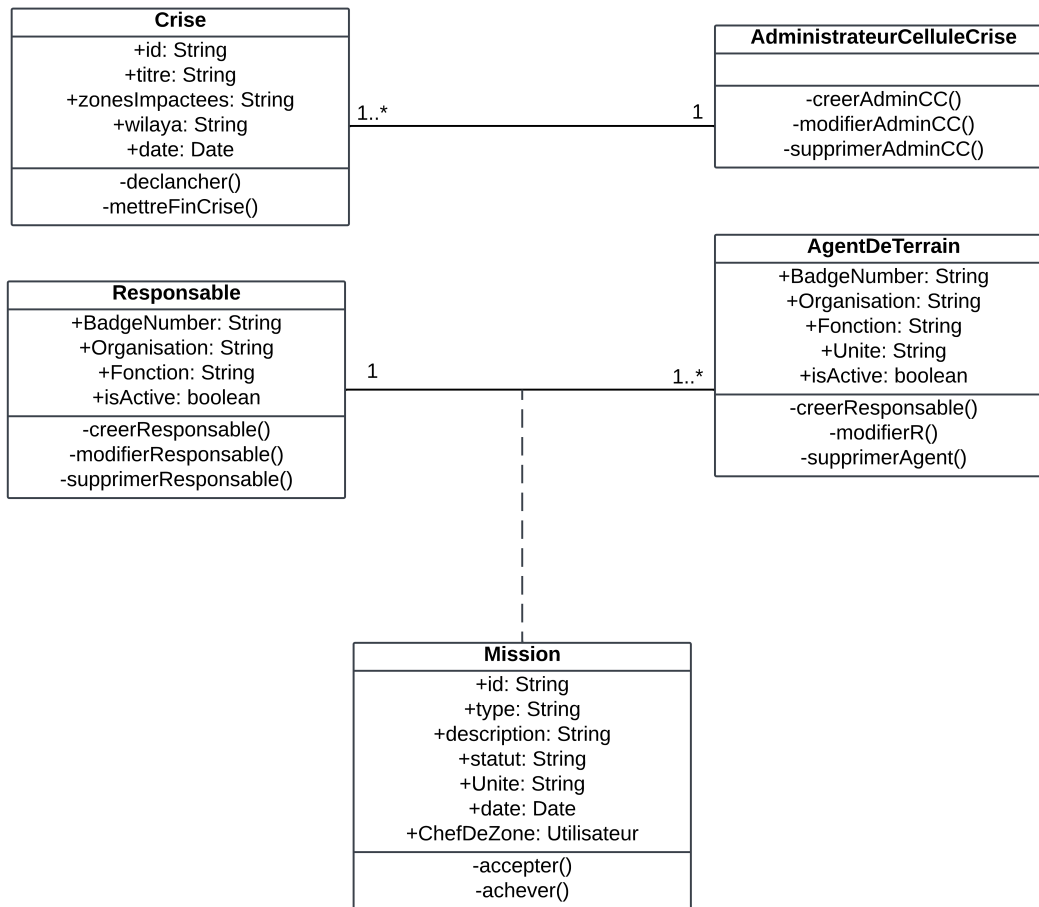


FIG. 63 – Diagramme de classe du sprint 3

★ Implémentation

Nous avons utiliser Firebase authentication, Firestore comme base de données et l’API OpenCage pour obtenir des coordonnées GPS.

★ Test de fonctionnalités

Pour s’assurer du bon fonctionnement du livrable à la fin du sprint, nous avons effectuer:

- Des tests manuelles sur le navigateur et l’émulateur Expo Go.
- Des tests automatisés effectués via le service Test Lab de Firebase.

★ Présentation des interfaces

Dans cette partie, nous allons présenter les interfaces utilisateurs développées dans ce sprint.

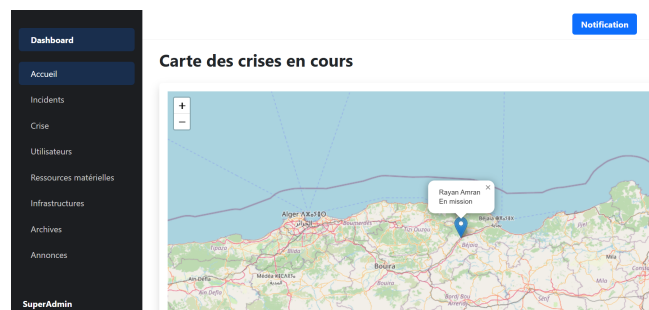


FIG. 64 – Système de géolocalisation

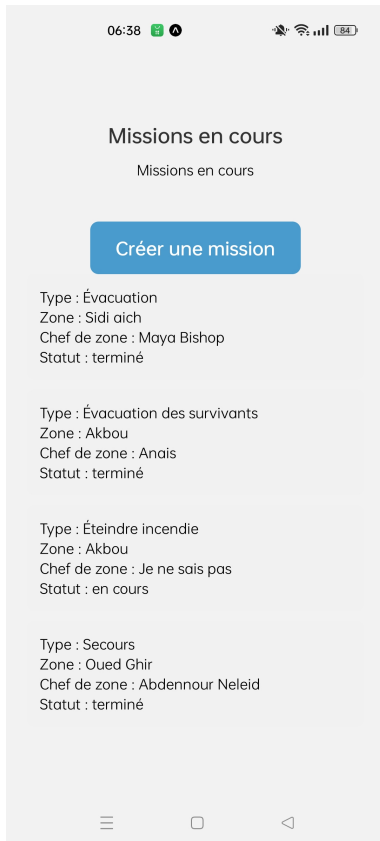


FIG. 65 – *Gestion des missions*

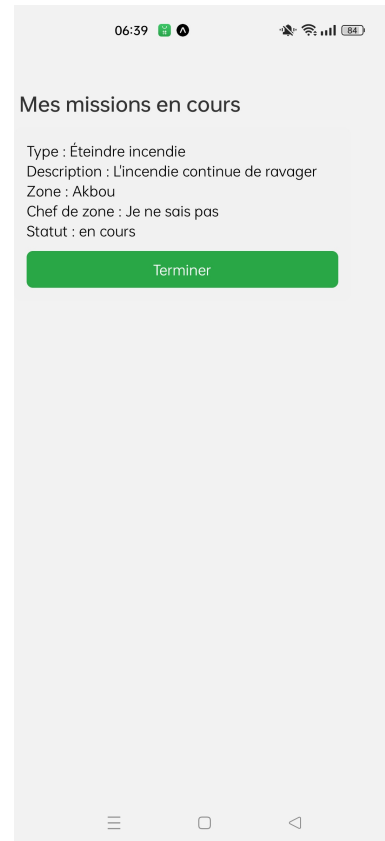


FIG. 66 – *Coordination des secours*

### 3.7.5 Post-phase

Pendant cette phase, nous allons, non seulement, évaluer la version fonctionnelle et opérationnelle du produit résultant de ce sprint avec le product owner, mais aussi, évaluer le flux de travail pendant le sprint.

- Sprint review  
Pendant une réunion avec le scrum master et product owner, ce dernier a approuvé les fonctionnalités développées pendant se sprint.
- Sprint retrospective  
Lors d'une réunion avec le scrum master, nous avons observer un flux de travail optimal et une bonne collaboration entre les membres de l'équipe.

**Réalisation du sprint n°04** Cette section décrit le processus de développement du quatrième, et dernier, sprint.

#### ★ Analyse

Cette étape consiste a pour objectif de comprendre, en profondeur, les besoins des utilisateurs à travers des diagrammes et des descriptions textuelles.

- Diagramme de cas d'utilisation  
La figure 67 illustre le diagramme de cas d'utilisation du sprint 4.

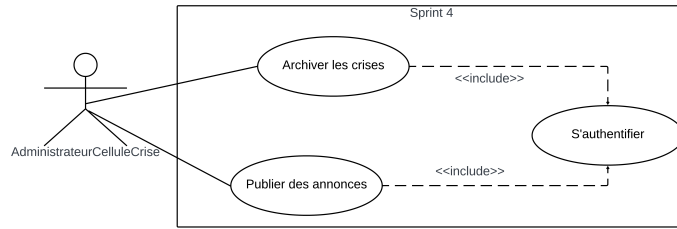


FIG. 67 – Diagramme de cas d'utilisation du sprint 4

– Description textuelle

A travers cette section, nous allons définir les cas d'utilisation à travers textuelle de chaque cas d'utilisation du sprint 04:

- Cas d'utilisation “Archiver la crise” illustré par le tableau 17.

Description textuelle
<p><b>Nom du cas:</b> Archivage de la crise  <b>But:</b> Permettre à l'administrateur de la cellule de crise d'archiver la crise.  <b>Acteur:</b> utilisateur</p>
<p><b>Séquencement:</b> l'administrateur de la cellule de crise est devant la page de crises.  <b>Pré-conditions:</b>  L'administrateur de la cellule de crise est connecté et il existe une crise en cours.  <b>Enchaînement nominale:</b>  1. L'administrateur de la cellule de crise clique sur le bouton “Terminer la crise”  2. Le met à jour l'état de la crise et enregistre ses informations dans la base de données.</p>
<p><b>Enchaînement alternatif:</b>  Aucun <b>Post-conditions:</b>  Les données archivées sont affichées dans la page d'archives du tableau de bord.</p>

TAB. 17 – Description textuelle du cas d'utilisation “Archiver une crise”

- Cas d'utilisation “Publier annonce” illustré par le tableau 18.

Description textuelle
<p><b>Nom du cas:</b> Publication d'annonces et de communiqués  <b>But:</b> Permettre à l'administrateur de la cellule de crise de publier des annonces.  <b>Acteur:</b> Administrateur de la cellule de crise</p>
<p><b>Séquencement:</b> la page d'annonces.  <b>Pré-conditions:</b>  L'application est installée sur l'appareil de l'administrateur de la cellule de crise est déjà connecté. <b>Enchaînement nominale:</b>  1. L'administrateur de la cellule de crise remplit le formulaire et clique sur le bouton “publier”  2. Le système vérifie que les champs sont tous remplis et valide.  3. Le système enregistre l'annonce.</p>
<p><b>Enchaînement alternatif:</b>  A1. Les champs ne sont pas tous remplis. A partir du point 2.  3. Le système affiche un message d'alerte.  4. L'utilisateur ressaisie ses informations et clique sur le bouton “publier”.  Le séquencement continue à partir du point 2.</p>
<p><b>Post-conditions:</b>  Le système affiche les annonces dans la page d'accueil de l'application.</p>

TAB. 18 – Description textuelle du cas d'utilisation “Publier annonces”

★ Diagramme de séquence

Dans cette partie, nous allons présenter les diagrammes de séquence détaillés qui consiste à décrire les interactions entre les acteurs et le système.

- Cas d'utilisation "Archivage des crises" illustré par la figure 68.

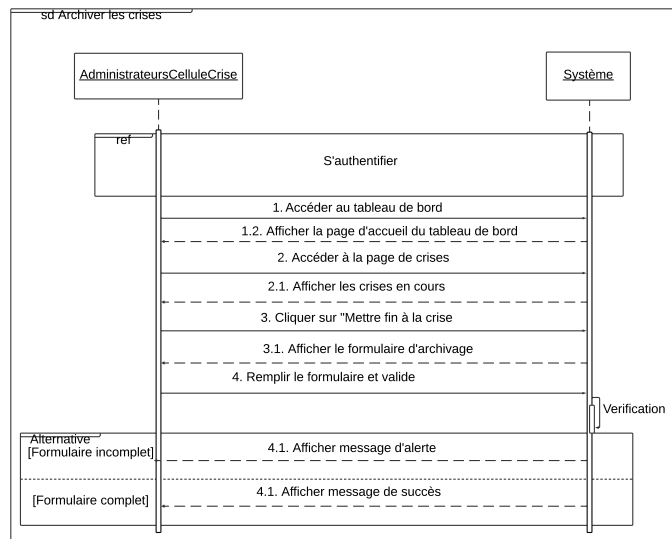


FIG. 68 – Diagramme de séquence de "Archivage des crises"

- Cas d'utilisation "Publication d'annonces et de communiqués" illustré par la figure 69.

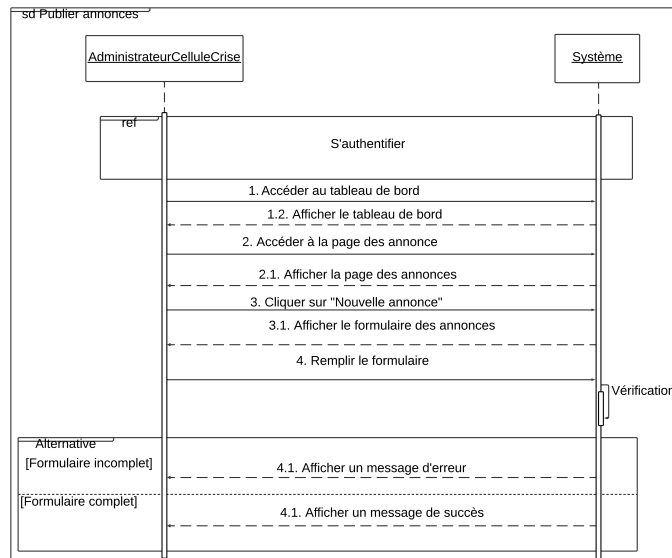


FIG. 69 – Diagramme de séquence de "Publication d'annonces et de communiqués"

Conception

Cette étape vise à définir les interactions entre les acteurs et les objets du système à travers des diagrammes.

- Diagrammes de séquence détaillés

Dans cette partie, nous allons présenter les diagrammes de séquence détaillés qui consiste à décrire les interactions entre les objets du système.

- Cas d'utilisation "Archivage des crises" illustré par la figure 70.

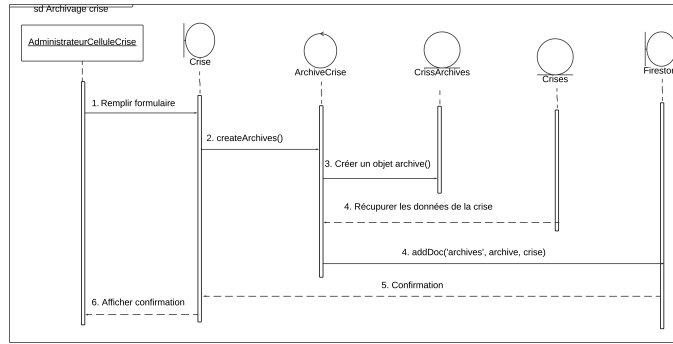


FIG. 70 – Diagramme de séquence détaillé de “Archivage des crises”

– Cas d’utilisation “Publication d’annonces et de communiqués” illustré par la figure 71.

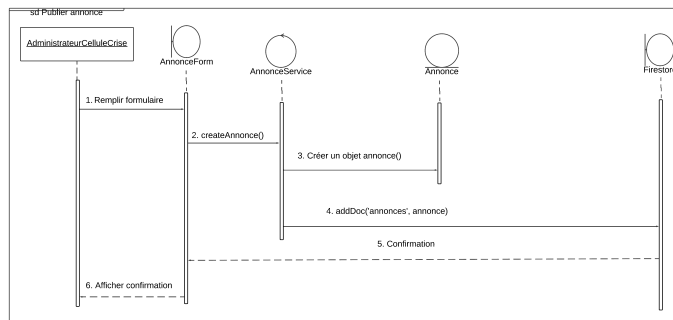


FIG. 71 – Diagramme de séquence détaillé de “Publication d’annonces et de communiqués”

– Diagramme de classe

La figure 72 représente le diagramme de classe du sprint 4.

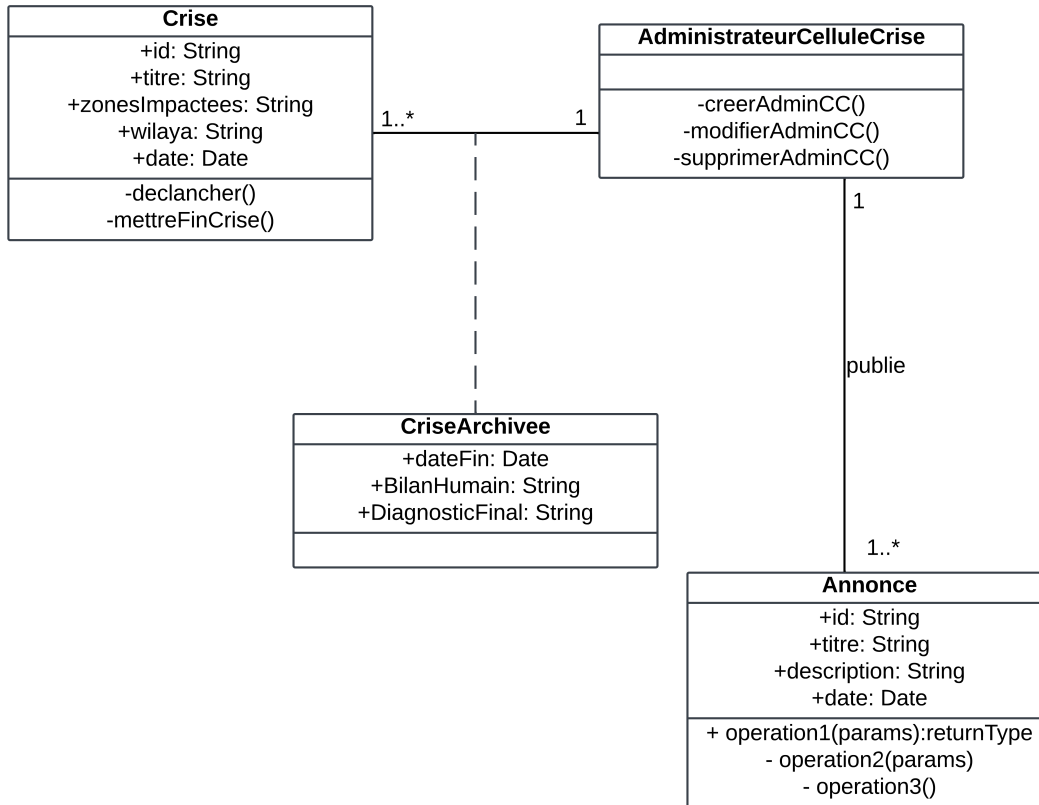


FIG. 72 – Diagramme de classe du sprint 4

### Implémentation

Nous commençant le développement de chaque user story du sprint 1 backlog avec l'éditeur de texte Vs code, en utilisant Firebase comme Backend av Firebase authentification avec email et mot de passe, Firestore comme base de données et

### Test de fonctionnalités

Pour s'assurer du bon fonctionnement du livrable à la fin du sprint, nous avons effectuer:

- Des tests manuelles sur le navigateur et l'émulateur Expo Go.
- Des tests automatisés effectués via le service Test Lab de Firebase.

### Présentation des interfaces

Dans cette partie, nous allons présenter les interfaces utilisateurs réalisées dans ce sprint.

### 3.7.6 Post-phase

Pendant cette phase, nous allons, non seulement, évaluer la version fonctionnelle et opérationnelle du produit résultant de ce sprint avec le product owner, mais aussi, évaluer le flux de travail pendant le sprint.

- Sprint review  
Pendant une réunion avec le product owner a salué la qualité des fonctionnalités du produit développées

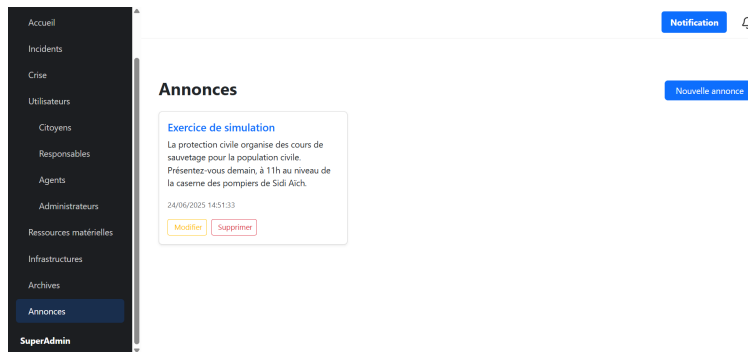


FIG. 73 – Publication d'annonces



FIG. 74 – Archivage des crises

pendant se sprint.

– Sprint retrospective

Pendant une réunion, le scrum master et l'équipe de développement ont su développer les fonctionnalités du sprint dans un environnement optimal, ainsi, un flux de travail a adopté lors de futurs projets.

### 3.8 Conclusion

A travers ce chapitre, nous avons développés une application web et mobile dédiée à la gestion de crise, pour objectif de valider le cadre méthodologique SBCC (ScrumBan et Cloud Computing) proposé lors du deuxième chapitre.

En mettant en œuvre le framework SBCC, nous avons observé une amélioration du processus de développement, notamment, des livrables plus fréquents, une collaboration entre les membres de l'équipe agile a été plus fluide et efficace, une documentation optimale et un contrôle continu des versions, etc.

Enfin, nous avons pu atteindre les objectifs initiaux de l'application de gestion de crise. Néanmoins, certains aspects de l'application peuvent encore être améliorés.

## Conclusion générale et perspectives

# Conclusion générale et perspectives

Dans le cadre de ce mémoire, nous avons développé une application web et mobile dédiée à la gestion des crises engendrés par les aléas naturels. Cette application a été conçue en adoptant un cadre méthodologique agile-cloud inspiré par l'approche agile Scrumban en conjonction avec la plateforme cloud Firebase.

Au cours de ce projet, nous avons étudié une multitude de travaux traitant la thématique de la conjonction du développement agile et du cloud computing, donc, nous avons décidé de réaliser une application ayant pour objectif de démontrer les avantages d'unir l'approche agile et le cloud computing. Cette application propose plusieurs fonctionnalités avec l'intention d'améliorer le processus de gestion de crise et de faciliter la communication entre les autorités et la population.

Ce processus de développement a été une expérience fructueuse, à la fois sur le plan pédagogique et sur le plan technique, qui nous a permis d'enrichir et d'approfondir nos connaissances avec diverses technologies d'actualité telles que la bibliothèque React, la plateforme Firebase, etc.

Le développement agile basé sur le cloud computing reste un sujet d'actualité en évolution continue. Pour les travaux futurs, il existe plusieurs aspects du domaine de la technologie qui peuvent être exploités afin d'étendre cette thématique, notamment, l'intégration de l'intelligence artificielle pour enrichir les fonctionnalités, l'automatisation des intégration et des déploiements continus pour accélérer les cycles de livraison de l'agile, la scalabilité intelligente pour une adaptation dynamique plus optimale aux flux des utilisateurs, etc. En continuant l'exploration de nouvelles pratiques dans le développement agile-cloud, nous offrant la possibilité d'améliorer la qualité des logiciels développés dans le but absolu de satisfaire les exigences évolutives des clients et organisations.

# Annexe



# Résumé

Dans ce mémoire, nous avons développé une application qui vise à gérer les désastres naturels en mettant en application un cadre méthodologique (ScrumBan et Cloud Computing). Ce framework fait l'union entre l'approche agile et le cloud computing dans le but d'améliorer le processus du développement.

Nous avons adopté la méthode agile hybride qui uni les meilleures pratiques de la méthode Scrum et de la méthode Kanban, afin de garantir une bonne collaboration entre les parties prenantes dans le projet et un flux de travail plus efficace. Nous avons utilisé UML (Unified Modeling Language) pour la conception et la modélisation du système. Nous avons intégré le cloud computing à travers Firebase (comme Backend as a Service) en utilisant Firebase Authentication pour l'authentification des utilisateurs, Firestore comme base de données et la plateforme Cloudinary pour l'hébergement des médias (photos/vidéos).

Mots clés: Agile, Cloud Computing, UML, Firebase, ScrumBan, Cloudinary, Web, Mobile

# Abstract

Through this project, we developed an application that aims to manage natural disasters by implementing a framework (ScrumBan). This framework makes the union between the agile approach and cloud computing with the sole purpose of enhancing the process of development.

We adopted the hybrid agile methodology that merges the best practices of both Scrum and Kanban, in order to guarantee a good collaboration between the different actors taking part in this project and a more efficient workflow. We utilised UML ( Unified Modeling Language) for the conception and modelis of the system.

We integrated cloud computing through Firebase (as a Backend as a Service) by using Firebase Authentication for user authentication, Firestore as a database and the Cloudinary platform for media (photos/videos) storage.

Key words: Agile, Cloud Computing, UML, Firebase, ScrumBan, Cloudinary, Web, Mobile

# Bibliographie

## Références

- [1] MELL, Peter, GRANCE, Tim, et al. The NIST definition of cloud computing. 2011.
- [2] BECK, Kent, BEEDLE, Mike, VAN BENNEKUM, Arie, et al. Manifeste pour le développement Agile de logiciels. En ligne: <http://agilemanifesto.org/iso/fr>, 2001.
- [3] YOUNAS, Muhammad, JAWAWI, Dayang Norhayati Abang, MAHMOOD, Ahmad Kamil, et al. Agile software development using cloud computing: a case study. *IEEE Access*, 2019, vol. 8, p. 4475-4484.
- [4] KALEM, S., DONKO, D., et BOSKOVIC, D. Agile methods for cloud computing. In: 2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2013. p. 1079-1083.
- [5] ZARINAH, IISSS et KASIRUN, M. Agile-based software product development using cloud computing services: Findings from a case study. *Sci. Int.(Lahore)*, 2013, p. 1045-1052.
- [6] LUCA, Charlie. Implementing Scrum in Cloud-Based Agile Software Development.
- [7] RAJ, Gaurav, YADAV, Komal, et JAISWAL, Arunima. Emphasis on testing assimilation using cloud computing for improvised agile SCRUM framework. In: 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE). IEEE, 2015. p. 219-225.
- [8] Direction générale de la protection civile et de l'aide humanitaire de la Commission européenne. \*Revue par les pairs de la protection civile – Algérie: Rapport final\*. Bruxelles: Union européenne, mars 2020. Disponible à l'adresse: [https://civil-protection-humanitarian-aid.ec.europa.eu/system/files/2020-03/peer\\_review\\_-\\_report\\_algeria\\_fr\\_final.pdf](https://civil-protection-humanitarian-aid.ec.europa.eu/system/files/2020-03/peer_review_-_report_algeria_fr_final.pdf).
- [9] IBM. Introduction to cloud computing. Consulté le avril 2025. Disponible à l'adresse: <https://www.coursera.org/learn/introduction-to-cloud>.
- [10] Claire Drumond. Atlassian. Scrum. Consulté le mai 2025. Disponible à l'adresse: <https://www.atlassian.com/fr/agile/scrum>.
- [11] Dan Radigan. Atlassian. Kanban. Consulté le mai 2025. Disponible à l'adresse: <https://www.atlassian.com/fr/agile/kanban>.
- [12] Agile Alliance. Extreme Programming. Consulté le mai 2025. Disponible à l'adresse: <https://www.agilealliance.org/glossary/xp/>
- [13] Craddock, RJ. Thales Research and Technology (UK) Limited. Crisis management models and timelines. 2006.
- [14] JOURNAL OFFICIEL DE LA REPUBLIQUE ALGERIENNE N°84. Loi relative à la prévention des risques majeurs et à la gestion des catastrophes dans le cadre du développement durable. 2004.
- [15] JOURNAL OFFICIEL DE LA REPUBLIQUE ALGERIENNE N° 16. Loi portant les règles de prévention, d'intervention et de réduction des risques de catastrophes dans le cadre du développement durable. 2024.
- [16] Microsoft. Visual Studio: IDE et Editeur de code pour les développeurs de logiciels et les équipes. Consulté le avril 2025. Disponible à l'adresse: <https://visualstudio.microsoft.com/fr/>

- [17] Github. A propos de Github.Consulté le avril 2025. Disponible à l'adresse: <https://docs.github.com/fr/get-started/start-your-journey/about-github-and-git>
- [18] Figma. What is Figma?.Consulté le avril 2025. Disponible à l'adresse: <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>
- [19] Max Gayler. What is Google meet?. Consulté le avril 2025. Disponible à l'adresse: <https://www.claap.io/blog/what-is-a-google-meet>
- [20] Katie Terrell Hanna. What is Google drive?. Consulté le avril 2025. Disponible à l'adresse: <https://www.techtarget.com/searchmobilecomputing/definition/Google-Drive>
- [21] San José State University. Lucidchart. 2008. Consulté le avril 2025. Disponible à l'adresse: <https://www.sjsu.edu/it/services/applications/lucidchart.php>
- [22] Nicolas Belhamri. Firebase. 2020. Consulté le avril 2025. Disponible à l'adresse: <https://www.boryl.fr/glossaire/firebase/>
- [23] Kate Brush. What is Jira?. 2024. Consulté le avril 2025. Disponible à l'adresse: <https://www.techtarget.com/searchsoftwarequality/definition/Jira>
- [24] Cloudinary. Consulté le avril 2025. Disponible à l'adresse: <https://cloudinary.com/about>
- [25] SHARP, Helen et ROBINSON, Hugh. An ethnographic study of XP practice. Empirical Software Engineering, 2004, vol. 9, no 4, p. 353-375.
- [26] Sergio Panagia et Giacomo Alonzi. What is Expo and why it matters for app development. Consulté le avril 2025. Disponible à l'adresse: <https://www.mozestudio.com/journal/what-is-expo-and-why-it-matters-for-app-development/>
- [27] HTML: HyperText Markup Language. Consulté le avril 2025. Disponible à l'adresse: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [28] CSS: Cascading Style Sheets. Consulté le avril 2025. Disponible à l'adresse: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [29] What is JSX?. Consulté le avril 2025. Disponible à l'adresse: <https://www.geeksforgeeks.org/reactjs/jsx-full-form/>
- [30] Andrew Zola. Bootstrap. 2022. Consulté le avril 2025. Disponible à l'adresse: <https://www.techtarget.com/whatis/definition/bootstrap>
- [31] What is JavaScript?. Consulté le avril 2025. Disponible à l'adresse: [https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development/Core/Scripting/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript)
- [32] React Introduction. 2025. Consulté le avril 2025. Disponible à l'adresse: <https://www.geeksforgeeks.org/reactjs/reactjs-introduction/>
- [33] React Native. Consulté le avril 2025. Disponible à l'adresse: <https://reactnative.dev/>

[34] Stanley Uili. Vite.js: A Beginner's Guide. 2025. Consulté le avril 2025. Disponible à l'adresse: <https://betterstack.com/community/guides/scaling-nodejs/vitejs-explained/>

[35] Sergio Panagia et Giacomo Alonzi. What is Expo and why it matters for app development. Consulté le avril 2025. Disponible à l'adresse: <https://www.mozestudio.com/journal/>

# Résumé

Dans ce mémoire, nous avons développé une application qui vise à gérer les désastres naturels en mettant en application un cadre méthodologique (ScrumBan et Cloud Computing). Ce framework fait l'union entre l'approche agile et le cloud computing dans le but d'améliorer le processus du développement.

Nous avons adopté la méthode agile hybride qui uni les meilleures pratiques de la méthode Scrum et de la méthode Kanban, afin de garantir une bonne collaboration entre les parties prenantes dans le projet et un flux de travail plus efficace. Nous avons utilisé UML (Unified Modeling Language) pour la conception et la modélisation du système. Nous avons intégré le cloud computing à travers Firebase (comme Backend as a Service) en utilisant Firebase Authentication pour l'authentification des utilisateurs, Firestore comme base de données et la plateforme Cloudinary pour l'hébergement des médias (photos/vidéos).

Mots clés: Agile, Cloud Computing, UML, Firebase, ScrumBan, Cloudinary, Web, Mobile

# Abstract

Through this project, we developed an application that aims to manage natural disasters by implementing a framework (ScrumBan). This framework makes the union between the agile approach and cloud computing with the sole purpose of enhancing the process of development.

We adopted the hybrid agile methodology that merges the best practices of both Scrum and Kanban, in order to guarantee a good collaboration between the different actors taking part in this project and a more efficient workflow. We utilised UML ( Unified Modeling Language) for the conception and modelis of the system.

We integrated cloud computing through Firebase (as a Backend as a Service) by using Firebase Authentication for user authentication, Firestore as a database and the Cloudinary platform for media (photos/videos) storage.

Key words: Agile, Cloud Computing, UML, Firebase, ScrumBan, Cloudinary, Web, Mobile