

Remerciements

*Je voudrais tout d'abord remercier du fond du cœur, **la direction générale de l'A.R.R.U**, pour avoir cru en mon profil en me donnant l'opportunité d'effectuer mon stage de PFE.*

*Je souhaite ainsi faire part de mes sentiments de gratitude à mon encadrant et maître de stage **Mr Aladine LAABIDI**. Il a su me faire confiance lors de cette aventure dans le monde professionnel. Sa disponibilité et la qualité de son encadrement m'ont permis d'enrichir mes connaissances et d'acquérir les compétences nécessaire pour réussir mon projet de fin d'études.*

Je souhaite ensuite adresser mes remerciements à tous mes professeurs ainsi que l'administration de la FSEGT, pour la qualité de l'enseignement qui est de haut niveau et le soutien de l'équipe administrative.

*Quant à **Mme Aziza Dagdagui**, mon encadrante universitaire et ma très chère amie, je voudrais qu'elle sache que j'apprécie, tous ses efforts pour permettre d'assurer un contenu et une rédaction sans fautes de mon rapport de stage, ses conseils pour qu'à aucun moment je ne décide de baisses les bras, ainsi que la qualité de son enseignement tout au long des années de licence. Merci beaucoup Madame.*

Je voudrais enfin remercier tous les membres du jury pour leur présence et leur lecteur attentive de mon rapport de projet de fin d'études.

Dédicaces

J'ai l'honneur de dédier ce modeste travail

À

Mes chers parents, pour leur sacrifice, leur précieux conseils, leur soutien infini tout au long de mes études. Merci de m'avoir, toujours guidé et éclairé. Je vous promets de tout faire pour vous rendre fiers de ma réussite dont vous avez toujours rêvé.

À

Mes deux grands frères, pour le soutien moral et l'encouragement qu'ils m'ont toujours apporté et surtout pendant cette période.

À

Toute ma famille et mes amis qui m'ont toujours encouragés, et à qui je souhaite plus de succès.

Table des matières

Table des figures	vii
Liste des tableaux	xi
Introduction générale	1
1 Cadre général du projet	3
Introduction	4
1.1 Cadre du Projet	4
1.2 Présentation de l'organisme d'accueil	4
1.2.1 Présentation générale de l'ARRU	4
1.2.2 Intervention	6
1.2.3 Modes d'intervention de l'Agence	6
1.3 Analyse de l'existant	6
1.3.1 Description de l'existant	6
1.3.2 Critique de l'existant	8
1.3.3 Solution proposée	8
1.4 Méthodologie adoptée	8
1.4.1 Classification de méthodologie de conduite de projets	9
1.4.2 Méthodes agiles vs. Classiques	9
1.4.3 Choix de la méthodologie de conception	11
1.4.3.1 Comparaison entre SCRUM, XP et DSDM	11
1.4.3.2 Choix de la méthodologie : SCRUM	11
1.4.4 Scrum	12
1.4.4.1 Les bases de SCRUM	12
1.4.4.2 Les acteurs dans SCRUM	12
1.4.4.3 Terminologie Scrum	13
1.4.4.4 Processus SCRUM	14
1.5 Langage de modélisation	15
1.5.1 Processus de modélisation	15
1.5.2 Langage de Modélisation Unifié : UML	15

TABLE DES MATIÈRES

1.6	Environnement de développement	15
1.6.1	Environnement matériel	15
1.6.2	Environnement logiciel	16
1.6.2.1	Outil de conception	16
1.6.2.2	Outil de rédaction du rapport	16
1.6.2.3	Outils de tests et intégrations	17
1.6.2.4	Technologies utilisées et Langages de développement	17
1.7	Architecture logicielle du projet	21
1.7.1	Architecture 3-tiers :	21
1.7.2	Modèle de conception (Design pattern)	22
1.7.2.1	Modèle de conception Backend (MVC)	22
1.7.2.2	Modèle de conception Frontend (MVVM)	23
	Conclusion	23
2	Sprint 0 : Analyse et Spécification des Besoins	24
	Introduction	25
2.1	Capture de besoins	25
2.1.1	Identification des acteurs	25
2.1.2	Identification des besoins fonctionnels	25
2.1.3	Identification des besoins non fonctionnels	26
2.2	Diagramme de classes	28
2.3	Pilotage du projet avec SCRUM	29
2.4	Diagramme de cas d'utilisation global	29
2.5	Product Backlog	30
2.6	Planification des sprints	31
	Conclusion	31
3	Sprint 1	32
	Introduction	33
3.1	Spécification des besoins	33
3.1.1	Diagramme de cas d'utilisation du sprint 1	33
3.1.2	Backlog du sprint 1	34
3.2	Analyse	35
3.2.1	Raffinement des cas d'utilisation	35
3.2.1.1	Cas d'utilisation « S'authentifier »	36
3.2.1.2	Cas d'utilisation « Gérer directions »	37
3.2.1.3	Cas d'utilisation « Gérer types tickets »	38
3.2.2	Diagrammes de séquences système	40
3.2.2.1	Cas d'utilisation « S'authentifier »	40
3.2.2.2	Cas d'utilisation « Ajouter direction »	41

TABLE DES MATIÈRES

3.2.2.3	Cas d'utilisation « Consulter direction »	42
3.2.2.4	Cas d'utilisation « Supprimer Direction »	43
3.3	Conception	44
3.3.1	Diagrammes de séquences détaillés	44
3.3.1.1	Cas d'utilisation « S'authentifier »	44
3.3.1.2	Cas d'utilisation « Ajouter direction »	45
3.3.1.3	Cas d'utilisation « Consulter direction »	46
3.3.1.4	Cas d'utilisation « Supprimer Direction »	47
3.3.2	Diagrammes d'activité	48
3.3.2.1	Cas d'utilisation « S'authentifier »	48
3.3.2.2	Cas d'utilisation « Ajouter direction »	49
3.3.2.3	Cas d'utilisation « Supprimer direction »	50
3.4	Réalisation	51
3.5	Tests et validation	54
	Conclusion	54
4	Sprint 2	55
	Introduction	56
4.1	Spécification des besoins	56
4.1.1	Diagramme de cas d'utilisation du sprint 2	56
4.1.2	Backlog du sprint 2	56
4.2	Analyse	58
4.2.1	Raffinement des cas d'utilisation	58
4.2.1.1	Cas d'utilisation « Gérer personnel »	58
4.2.1.2	Gérer tickets	62
4.2.2	Diagrammes de séquences système	64
4.2.2.1	Cas d'utilisation « Ajouter une personne »	65
4.2.2.2	Cas d'utilisation « Consulter personnel »	66
4.2.2.3	Cas d'utilisation « Supprimer une personne »	67
4.2.2.4	Cas d'utilisation « Consulter une personne »	68
4.2.2.5	Cas d'utilisation « Modifier une personne »	69
4.2.2.6	Cas d'utilisation « Rechercher personnel »	70
4.2.2.7	Cas d'utilisation « Consulter tickets »	71
4.2.2.8	Cas d'utilisation « Consulter tickets approuvés »	72
4.3	Conception	72
4.3.1	Diagrammes de séquences détaillés	73
4.3.1.1	Cas d'utilisation « Ajouter une personne »	73
4.3.1.2	Cas d'utilisation « Consulter personnel »	74
4.3.1.3	Cas d'utilisation « Supprimer une personne »	74

TABLE DES MATIÈRES

4.3.1.4	Cas d'utilisation « Consulter une personne »	75
4.3.1.5	Cas d'utilisation « Consulter tickets »	75
4.3.1.6	Cas d'utilisation « Modifier une personne »	76
4.3.1.7	Cas d'utilisation « Consulter tickets approuver »	77
4.3.2	Diagrammes d'activités	78
4.3.2.1	Cas d'utilisation « Modifier une personne »	78
4.3.2.2	Cas d'utilisation « Rechercher personnel »	79
4.4	Réalisation	80
4.5	Tests et validation	83
	Conclusion	83
5	Sprint 3	84
	Introduction	85
5.1	Spécification des besoins	85
5.1.1	Backlog du sprint 3	85
5.1.2	Diagramme de cas d'utilisation du sprint 3	86
5.2	Analyse	86
5.2.1	Raffinement des cas d'utilisation	86
5.2.1.1	Ca d'utilisation « Approuver tickets en attente »	87
5.2.1.2	Cas d'utilisation « Accepter tickets approuvés »	89
5.2.2	Diagrammes de séquences système	91
5.2.2.1	Cas d'utilisation « Consulter tickets en attente »	91
5.2.2.2	Cas d'utilisation « Approuver tickets en attente »	92
5.2.2.3	Cas d'utilisation « Refuser ticket en attente »	93
5.3	Conception	94
5.3.1	Diagrammes de séquences détaillés	94
5.3.1.1	Cas d'utilisation « Consulter tickets en attente »	94
5.3.1.2	Cas d'utilisation « Approuver ticket en attente »	95
5.3.1.3	Cas d'utilisation « Refuser ticket en attente »	96
5.3.2	Diagrammes d'activités	97
5.3.2.1	Cas d'utilisation « Approuver tickets en attente »	97
5.3.2.2	Cas d'utilisation « Refuser ticket en attente »	98
5.4	Réalisation	99
5.5	Tests et validation	101
	Conclusion	101
Conclusion et perspectives		102
Bibliographie		104

Table des figures

1.1	Carte d'identité de l'ARRU	5
1.2	Organigramme de l'ARRU	5
1.3	Demande fournitures	7
1.4	Demande de gestion d'archive	7
1.5	Comparaison entre les méthodes Agiles vs les méthodes Classiques.	10
1.6	Processus Scrum	14
1.7	Langage UML	15
1.8	Logo de StarUml.	16
1.9	Logo de Overleaf.	16
1.10	Logo de Postman.	17
1.11	Logo de MongoDB	17
1.12	Logo de Angular.	18
1.13	Logo de NodeJs	18
1.14	Logo de Express Js.	18
1.15	Logo de HTML.	19
1.16	Logo de CSS.	19
1.17	Logo de Bootstrap.	19
1.18	Logo de NPM.	20
1.19	Logo de TypeScript.	20
1.20	Logo de JSON.	20
1.21	Architecture de l'application	21
1.22	Design Pattern MVC	22
1.23	Design Pattern MVVM	23
2.1	Hachage mot de passe.	27
2.2	Diagramme de classes global.	28
2.3	Diagramme de cas d'utilisation global.	29
2.4	Planification des sprints	31
3.1	Diagramme de cas d'utilisation du sprint 1	33
3.2	Diagramme de cas d'utilisation « S'authentifier ».	36

TABLE DES FIGURES

3.3	Diagramme de cas d'utilisation "Gérer directions"	37
3.4	Diagramme de cas d'utilisation « Gérer types de tickets ».	38
3.5	Diagramme de séquences système du cas d'utilisation « S'authentifier »	40
3.6	Diagramme de séquences système du sous cas d'utilisation « Ajouter Direction »	41
3.7	Diagramme de séquences système du sous cas d'utilisation « Consulter direction ».	42
3.8	Diagramme de séquences système du sous cas d'utilisation « Supprimer Direction ».	43
3.9	Diagramme de séquences détaillé du cas d'utilisation « S'authentifier »	44
3.10	Diagramme de séquences détaillé du cas d'utilisation « Ajouter direction ».	45
3.11	Diagramme de séquences détaillé du cas d'utilisation « Consulter Directions ».	46
3.12	Diagramme de séquences détaillé du cas d'utilisation « Supprimer direction ».	47
3.13	Diagramme d'activité du cas d'utilisation « S'authentifier ».	48
3.14	Diagramme d'activité du cas d'utilisation « Ajouter direction ».	49
3.15	Diagramme d'activité du cas d'utilisation « Supprimer direction ».	50
3.16	Interface : Authentification	51
3.17	Interface : Consulter directions	51
3.18	Interface : Ajouter direction	52
3.19	Interface : Supprimer direction	52
3.20	Interface : Consulter types tickets	53
3.21	Interface : Ajouter type ticket	53
4.1	Diagramme de cas d'utilisation du sprint 2	56
4.2	Diagramme de cas d'utilisation « Gérer personnel ».	58
4.3	Diagramme de cas d'utilisation "Gérer tickets".	62
4.4	Diagramme de séquences système du sous-cas d'utilisation « Ajouter une personne »	65
4.5	Diagramme de séquences système du sous-cas d'utilisation « Consulter personnel ».	66
4.6	Diagramme de séquences système du sous-cas d'utilisation « Supprimer une personne ».	67
4.7	Diagramme de séquences système du sous-cas d'utilisation « Consulter une personne ».	68
4.8	Diagramme de séquences système du sous-cas d'utilisation « Modifier une personne ».	69
4.9	Diagramme de séquences système du sous-cas d'utilisation « Rechercher personnel ».	70

TABLE DES FIGURES

4.10 Diagramme de séquences système du sous-cas d'utilisation « Consulter tickets »	71
4.11 Diagramme de séquences système du sous-cas d'utilisation « Consulter tickets approuvés »	72
4.12 Diagramme de séquences détaillé du sous-cas d'utilisation « Ajouter une personne »	73
4.13 Diagramme de séquences détaillé du sous-cas d'utilisation « Consulter personnel »	74
4.14 Diagramme de séquences détaillé du sous-cas d'utilisation « Supprimer une personne »	74
4.15 Diagramme de séquences détaillé du sous-cas d'utilisation « Consulter une personne »	75
4.16 Diagramme de séquences détaillé du sous-cas d'utilisation « Consulter tickets »	75
4.17 Diagramme de séquences détaillé du sous-cas d'utilisation « Modifier une personne »	76
4.18 Diagramme de séquences détaillés du sous-cas d'utilisation « Consulter tickets approuvés »	77
4.19 Diagramme d'activité du sous-cas d'utilisation « Modifier une personne »	78
4.20 Diagramme d'activité du sous-cas d'utilisation « Rechercher personnel »	79
4.21 Interface : Consulter personnel	80
4.22 Interface : Ajouter une personne	80
4.23 Interface : Consulter une personne	81
4.24 Interface : Modifier une personne	81
4.25 Interface : Consulter tickets	82
4.26 Interface : Consulter tickets	82
4.27 Interface : Consulter refusés	83
5.1 Diagramme de cas d'utilisation du sprint 3	86
5.2 Diagramme de cas d'utilisation « Approuver tickets en attente »	87
5.3 Diagramme de cas d'utilisation « Accepter tickets approuvés »	89
5.4 Diagramme de séquences système du sous-cas d'utilisation « Consulter tickets en attente »	91
5.5 Diagramme de séquences système du sous-cas d'utilisation « Approuver ticket en attente »	92
5.6 Diagramme de séquences système du sous-cas d'utilisation « Refuser ticket en attente »	93
5.7 Diagramme de séquences détaillé du cas d'utilisation « Consulter tickets en attente »	94

TABLE DES FIGURES

5.8	Diagramme de séquences détaillé du sous-cas d'utilisation « Approuver ticket en attente »	95
5.9	Diagramme de séquences détaillé du sous-cas d'utilisation « Approuver ticket en attente ».	96
5.10	Diagramme d'activité du sous-cas d'utilisation « Approuver ticket en attente ».	97
5.11	Diagramme d'activité du sous-cas d'utilisation « Refuser ticket en attente ».	98
5.12	Interface : Consulter tickets en attente	99
5.13	Interface : Approuver ticket en attente	99
5.14	Interface : Refuser ticket en attente	100
5.15	Interface : Consulter tickets en approuvés	100

Liste des tableaux

1.1	Table comparative des méthodes agiles Scrum, XP et DSDM [HI12]	11
1.2	Environnement matériel.	15
2.1	Table des acteurs.	25
2.2	Le backlog du produit	30
3.1	Le Backlog du sprint 1	34
3.2	Description textuelle du cas d'utilisation « S'authentifier »	36
3.3	Description textuelle du sous cas d'utilisation « Ajouter direction »	37
3.4	Description textuelle du sous cas d'utilisation « Consulter directions »	37
3.5	Description textuelle du sous cas d'utilisation « Supprimer direction »	38
3.6	Description textuelle du sous cas d'utilisation « Ajouter type ticket »	39
3.7	Description textuelle du sous cas d'utilisation « Consulter types tickets »	39
3.8	Description textuelle du sous cas d'utilisation « Supprimer type de ticket »	39
3.9	Table des acteurs.	54
3.10	Tests de validation	54
4.1	Le Backlog du sprint 2	57
4.2	Description textuelle du sous cas d'utilisation « Ajouter une peronne »	59
4.3	Description textuelle du sous cas d'utilisation « Consulter personnel »	59
4.4	Description textuelle du sous-cas d'utilisation « Supprimer une personne »	60
4.5	Description textuelle du sous-cas d'utilisation « Consulter une personne »	60
4.6	Description textuelle du sous-cas d'utilisation « Modifier une personne »	61
4.7	Description textuelle du sous-cas d'utilisation « Rechercher personnel »	61
4.8	Description textuelle du sous cas d'utilisation « Ajouter ticket »	62
4.9	Description textuelle du sous cas d'utilisation « Consulter tickets »	62
4.10	Description textuelle du sous cas d'utilisation « Consulter tickets en attentes »	63
4.11	Description textuelle du sous cas d'utilisation « Consulter tickets approuvés »	63
4.12	Description textuelle du sous cas d'utilisation « Consulter tickets acceptés »	63
4.13	Description textuelle du sous cas d'utilisation « Consulter tickets refusés »	64
4.14	Tests et validation	83

LISTE DES TABLEAUX

5.1	Le Backlog du sprint 3	85
5.2	Description textuelle du sous-cas d'utilisation « Consulter tickets en attente »	87
5.3	Description textuelle du sous-cas d'utilisation « Approuver ticket en attente »	88
5.4	Description textuelle du sous-cas d'utilisation « Refuser ticket en attente »	88
5.5	Description textuelle du sous-cas d'utilisation « Consulter tickets approuvés »	89
5.6	Description textuelle du sous-cas d'utilisation « Accepter ticket approuvé »	90
5.7	Description textuelle du sous-cas d'utilisation « Refuser ticket approuvé » .	90
5.8	Liste des tests	101

Introduction générale

Les défis actuels et futurs et l'évolution rapide du marché dans le contexte de mondialisation et de globalisation obligent les entreprises à se doter de moyens technologiques de pointe pour gérer les flux d'informations afin de rester à jour, puisque l'importance de l'informatique augmente dans presque tous les domaines de la vie et son concept devient extrêmement important, pour cela, il est prêt à déployer un grand nombre de ressources pour garantir sa compétitivité et son évolutivité.

Le stage du PFE m'a donné l'opportunité d'avoir un contact direct avec le monde professionnel afin de consolider notre pratique.

Mon projet consiste à concevoir et développer une application web appelée «HelpDesk», qui permettra aux différents employés de l'A.R.R.U de créer les tickets (Approvisionnement/Entretien) qui seront traités respectivement par les responsables et le chargé des demandes.

Le présent rapport s'organise comme suit :

- **Le premier chapitre** est réservé au cadre général du projet, l'organisme d'accueil, la critique de l'existant, la solution proposée et le choix méthodologique et conceptuel et l'environnement pour la réalisation du projet.
- **Le deuxième chapitre** concerne la planification et l'architecture : Dans ce chapitre, nous allons définir la méthodologie de gestion de projet, le contexte du système, déterminer les besoins fonctionnels, les besoins non fonctionnels, le Backlog du produit, planifier les sprints et enfin donner le diagramme de cas d'utilisation global ainsi que le diagramme de classes.
- **Le troisième chapitre** présente la réalisation du Sprint 1 après l'avoir analysé et détaillé.
- **Le quatrième chapitre** présente la réalisation du Sprint 2 après l'avoir analysé et détaillé.
- **Le cinquième chapitre** présente la réalisation du Sprint 3 après l'avoir analysé et détaillé.

A l'issue de ce travail, les perspectives d'amélioration de la vision et de développement de cette application seront discutées en guise de conclusion.

Chapitre 1

Cadre général du projet

Sommaire

Introduction	4
1.1 Cadre du Projet	4
1.2 Présentation de l'organisme d'accueil	4
1.2.1 Présentation générale de l'ARRU	4
1.2.2 Intervention	6
1.2.3 Modes d'intervention de l'Agence	6
1.3 Analyse de l'existant	6
1.3.1 Description de l'existant	6
1.3.2 Critique de l'existant	8
1.3.3 Solution proposée	8
1.4 Méthodologie adoptée	8
1.4.1 Classification de méthodologie de conduite de projets	9
1.4.2 Méthodes agiles vs. Classiques	9
1.4.3 Choix de la méthodologie de conception	11
1.4.4 Scrum	12
1.5 Langage de modélisation	15
1.5.1 Processus de modélisation	15
1.5.2 Langage de Modélisation Unifié : UML	15
1.6 Environnement de développement	15
1.6.1 Environnement matériel	15
1.6.2 Environnement logiciel	16
1.7 Architecture logicielle du projet	21
1.7.1 Architecture 3-tiers :	21
1.7.2 Modèle de conception (Design pattern)	22
Conclusion	23

Introduction

L'objectif de ce premier chapitre est de mettre ce travail dans son contexte général. En premier lieu, je vais présenter le cadre du projet. En second lieu, je passerai à l'étude de l'existant et la proposition de solution. En dernier lieu, je présenterai la méthodologie adoptée et l'environnement de la réalisation du projet.

1.1 Cadre du Projet

Ce stage s'inscrit dans le cadre de la validation de mon projet de fin d'études en vue de l'obtention du diplôme de licence en Business Intelligence (BI) à la Faculté des Sciences Economiques et de Gestion de Tunis (FSEG). Pendant ce stage, je suis appelé à réaliser un projet mettant l'accent sur mes connaissances, mes capacités techniques et pratiques acquises pendant ma formation académique. Je suis alors accueilli au sein de L'Agence de Réhabilitation et de Rénovation Urbaine (ARRU) pour effectuer un stage d'une durée de 4 mois

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation générale de l'ARRU

L'Agence de Réhabilitation et de Rénovation Urbaine (ARRU) est une entreprise publique Tunisienne à caractère industriel et commercial, créée en Août 1981 par la loi n° 81-69 du 1er Août 1981.

Elle est chargée de l'exécution de la politique de l'Etat dans les domaines de la réhabilitation et de la rénovation Urbaine, sous la tutelle du Ministère de l'Equipement de l'Habitat et de l'Infrastructure, pour le compte de l'Etat et des collectivités locales, principalement les communes.

Son intervention s'effectue dans le cadre des programmes nationaux ou dans le cadre contractuel avec les collectivités publiques locales, maitres d'ouvrages des projets, qui se chargent de mobiliser les financements des différents projets.

L'ARRU réalise pour son propre compte des projets de promotion immobilière et ce, dans le but d'améliorer ses propres ressources d'équilibrer ses comptes et de participer à la réalisation des objectifs nationaux en matière d'habitat.

CHAPITRE 1. CADRE GÉNÉRAL DU PROJET

La figure ci-après présente la carte d'identité de l'ARRU



FIGURE 1.1 – Carte d'identité de l'ARRU

La figure suivante illustre l'organigramme de l'ARRU

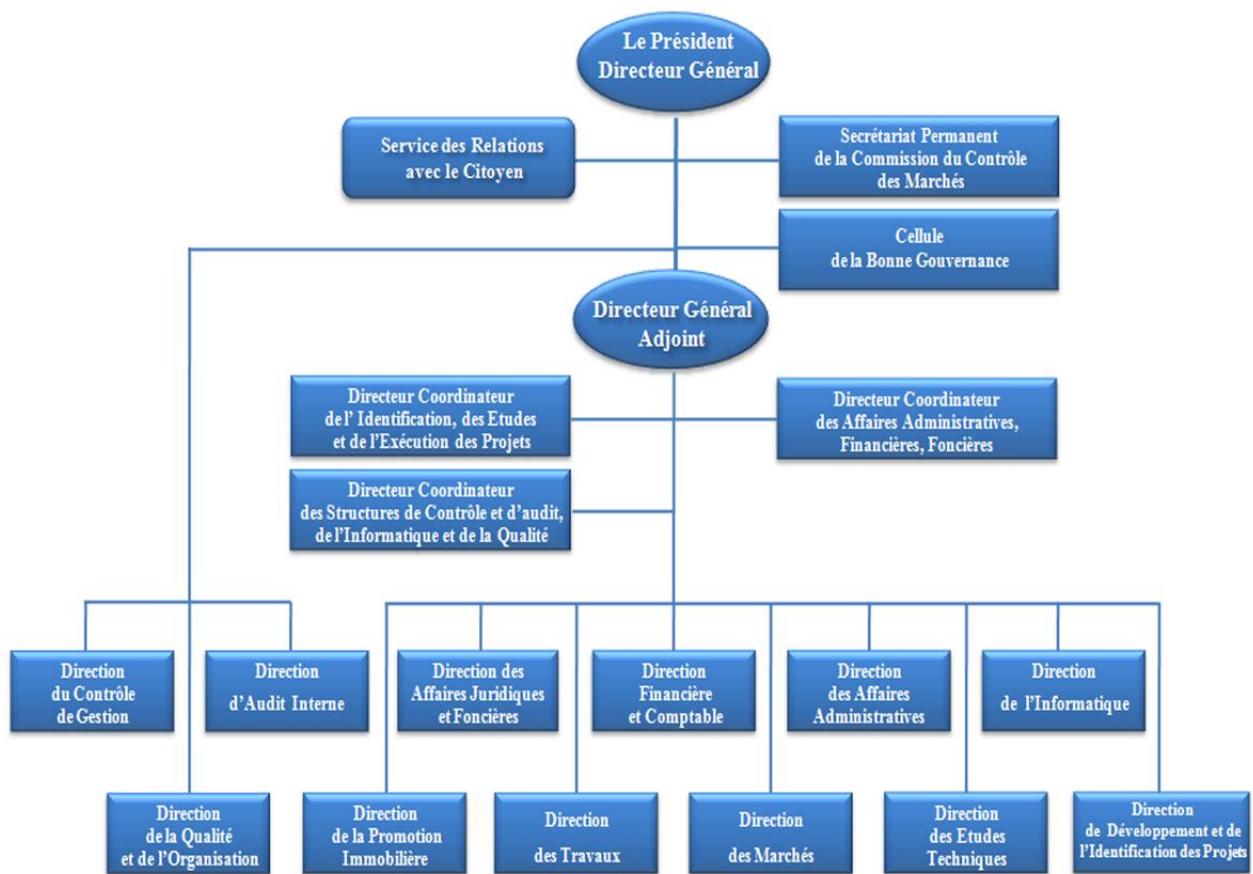


FIGURE 1.2 – Organigramme de l'ARRU

1.2.2 Intervention

L'intervention de l'A.R.R.U, depuis plus de 30 ans, dans plus de 1000 quartiers populaires et centres anciens répartis sur tout le territoire, a contribué :

- À l'amélioration du cadre de vie des habitants à travers la mise en oeuvre et l'extension des infrastructures de base (eaux potables, assainissements eaux usées , voiries, drainage, éclairage public..).
- La construction d'équipements publics (écoles, dispensaires, terrains/salles de sports..) et des équipements marchands de même que l'aménagement de zones d'activités et d'aires de jeux pour les enfants.

1.2.3 Modes d'intervention de l'Agence

- **Maitrise d'ouvrage déléguée** : réalisation des programmes nationaux de réhabilitation et de rénovation urbaine pour le compte de l'Etat et des collectivités publiques locales.
- **Maitrise d'ouvrage directe** : réalisation des opérations d'aménagement des terrains nus et de promotion immobilière et ce dans le but d'améliorer ses propres ressources et d'équilibrer ses comptes.

1.3 Analyse de l'existant

L'analyse de l'existant est une étape importante pour bien comprendre le système actuel. Elle a pour objectif de dégager les lacunes du système existant et de proposer les solutions adéquates et définir les objectifs à atteindre au titre de perfectionnement.

1.3.1 Description de l'existant

Les employés de l'entreprise peuvent être confrontés à une variété de supports, tels que des dizaines de pages éparpillées et des processus très longs. Actuellement, L'ARRU utilise la méthode classique de gestion des demandes qui transitent dans les directions de la société.

En effet, un employé doit remplir manuellement sa demande et la transporter d'un responsable à un autre jusqu'à atteindre la validation ou le refus.

Ce processus aussi long et lent rend l'évolution d'une demande très compliquée, peu pratique pour le partage et la mise à jour des informations et non évolutive.

Les figures suivantes présentent quelques types de demandes.

CHAPITRE 1. CADRE GÉNÉRAL DU PROJET

	العدد --20..	التاريخ/..../20..	مطلوب في التزويد	وكالة التهذيب والتجديد المعراني ادارة الشؤون الادارية مكتب المعدات
الاسم و اللقب: الادارة او المصلحة :				
ملاحظات		الكمية المطلوبة	ال حاجيات	
امضاء الطالب: تأشيره المدير المبادر: تأشيره ادارة الشؤون الادارية				

FIGURE 1.3 – Demande fournitures

يملأ من قبل مصلحة التصرف في الوثائق والأرشيف		وزارة النقل والتجهيز وكالة التهذيب والتجديد المعراني			
تحويل عدد لسنة		جدول تحويل وشأنق شبه نشطة الى مصلحة التصريف في الوثائق والأرشيف			
.....		يملأ من قبل المصلحة المحولة			
التعريف بالوثائق المحولة - عدد الحافظات - وعاء الوثائق		التعريف بالمصلحة المنشئة للوثائق - الوحدة المنشئة - اسم المسئول			
قائمة في الوثائق المحولة					
يملأ من قبل مصلحة التصرف في الوثائق والأرشيف		يملأ من قبل المصلحة المحولة			
الملاحظات	تاريخ الإلاضف او الترحيل	عدد قاعدة الحفظ بجدول مدد الاستبقاء	التاريخ الاخير والاخصى للوثائق	عنوان الحافظات	رقم الحافظة
		

FIGURE 1.4 – Demande de gestion d'archive

1.3.2 Critique de l'existant

L'étude de l'existant nous a permis de dégager les points faibles détaillés ci dessous :

- L'intervention manuelle cause une lenteur du déroulement des tâches ainsi que le risque de la non fiabilité ;
- Les tâches à effectuer nécessitent beaucoup d'attention et de temps pour une vérification manuelle et régulière ;
- La redondance des tâches ;
- L'indisponibilité : en cas d'absence d'un responsable, l'employé doit attendre son retour pour que le processus de sa demande puisse avancer.
- Le nombre de demandeurs est trop élevé et les demandes sont variées ce qui entraîne une mauvaise organisation dans la gestion des demandes, ainsi qu'il serait très difficile de prendre en charge toutes les demandes ;
- Le risque d'oubli de traitement ou de perte de la demande ;
- Le manque de suivi : inexistence de système permettant au demandeur de suivre l'état de sa demande.

1.3.3 Solution proposée

Pour remédier aux difficultés présentées dans la section précédente, j'ai proposé au responsable de mon stage au sein de l'ARRU de concevoir et mettre en place une application web permettant d'automatiser la gestion des demandes. Cette application sera utilisée par tout le personnel participant dans le traitement d'une demande dès son remplissage jusqu'à la dernière signature.

L'objectif principal de l'application est la mise en place d'un système fiable capable d'effectuer la gestion des demandes du personnel. Le futur système doit :

- Améliorer considérablement la qualité des services offerts ;
- Garantir une transparence opérationnelle ;
- Permettre de gagner le temps, limiter les efforts, diminuer la perte d'informations ;
- Faciliter le suivi des demandes.

1.4 Méthodologie adoptée

Pour pouvoir choisir la bonne méthodologie, il faut avoir une idée sur les avantages et les inconvénients de quelques unes afin de pouvoir choisir celle qui convient mieux à mon projet. Dans ce qui suit, je présenterai un aperçu sur les différentes méthodologies utilisées.

1.4.1 Classification de méthodologie de conduite de projets

Les méthodologies de conduite de projets aident à accomplir chaque étape d'un projet, de la planification à la mise en oeuvre, dans un souci d'efficacité et rentabilité. Ces méthodologies sont réparties en deux grandes familles :

- **Les méthodes traditionnelles :**

Dites aussi méthodes classiques. Une succession de phases qui se déverse les unes dans les autres. Une phase de l'approche classique ne peut commencer tant que la précédente n'est pas terminée. Alors, le plan du projet se fait étape par étape.

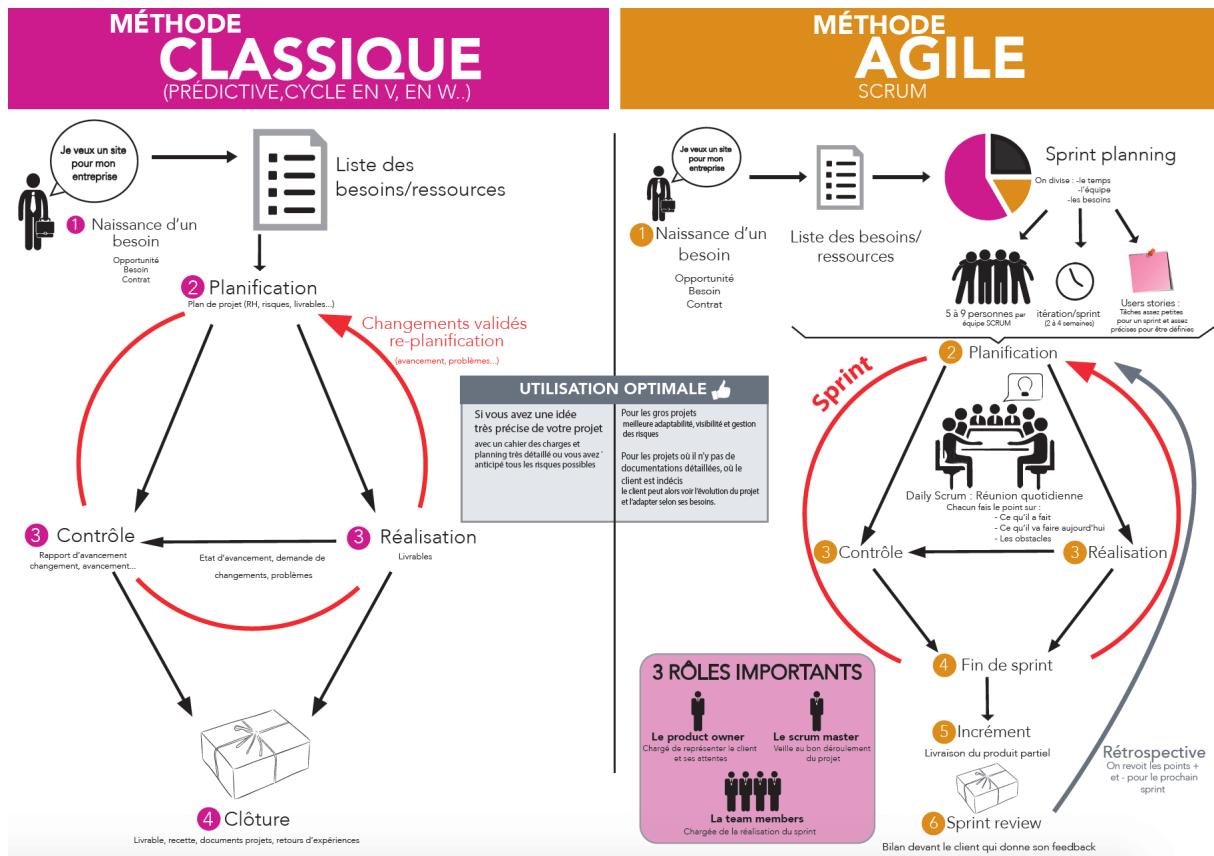
- **Les méthodes Agiles :**

Plus efficaces et moins rigides que les méthodes classiques. Ces dernières placent les besoins du client au centre des priorités du projet. Une méthode Agile est une approche itérative et collaborative, capable de prendre en compte les besoins initiaux du client et ceux liés aux évolutions.

1.4.2 Méthodes agiles vs. Classiques

Lorsqu'on démarre un projet, il est normal de se questionner sur l'organisation que l'on va adopter pour sa mise en place et sa réalisation. Combien de réunions ? Quel rôle va jouer chaque personne de l'équipe ? Comment faire pour éviter de recommencer le travail depuis le début si le client change d'avis ? Depuis quelque temps, on entend parler de la méthode Agile, ou encore Scrum (la méthodologie Agile la plus utilisée). Mais cette méthode de gestion de projet n'est pas la plus facile à comprendre, et la plupart des sites internet qui proposent une explication se perdent souvent dans des explications textuelles longues dont on décroche facilement.

Pour simplifier cela, la figure ci-après [Guy18]est un schéma explicatif du processus de la méthodologie Agile, comparé au processus de la méthodologie Classique de gestion de projet .



On privilégiera plutôt la méthode classique lorsqu'on a une idée précise du projet, avec un planning bien détaillé et où on a anticipé tous les risques possibles, quant à la méthode Agile, on la choisirra plutôt pour les gros projets, celle-ci permettant une meilleure adaptabilité, visibilité et gestion des risques. On privilégiera également la méthode Agile pour les projets où il n'y a pas de documents détaillés, ou quand vous sentez que votre client est indécis. Le client pourra alors voir l'évolution du projet et l'adapter à ses besoins sans pour autant vous obliger à recommencer tout le travail que vous avez fourni depuis le début [Guy18].

Les méthodes agiles répondent aux méthodes classiques, trop prédictives et trop rigides, en exposant de nouveaux principes plus souples dont l'anticipation, l'auto-régulation, le feedback et la collaboration. Elles renforcent aussi la capacité d'une organisation « apprenante » au changement et à la transformation [CS16].

Le mouvement Agile est en effet né en réponse à l'échec des paradigmes dominants de gestion de projets et de développement logiciels (y compris la méthodologie en cascade).

1.4.3 Choix de la méthodologie de conception

1.4.3.1 Comparaison entre SCRUM, XP et DSDM

Après la sélection de la méthodologie Agile comme étant la méthodologie de conduite du projet, je doit choisir l'une de ces populaires méthodes pour mon projet. Pour cela, je vais procéder à la comparaison de trois méthodes agiles : SCRUM, XP et DSDM [HI12].

La table ci-après décrit le principe de chacune de ces méthodes ainsi que ses points forts et ses points faibles.

TABLE 1.1 – Table comparative des méthodes agiles Scrum, XP et DSDM [HI12]

Méthode	Points Forts	Points Faibles
SCRUM	<ul style="list-style-type: none"> • Personnels engagés • Meilleure vue de l'ensemble du projet • Courtes itérations • Amélioration de la communication • Augmentation de la productivité • Réduction des bugs • Mise à jour des priorités • Qualité du produit mise en avant 	<ul style="list-style-type: none"> • Peu de place pour les étapes postérieures ainsi que les étapes antérieures. • Faible documentation et donc facile à détourner.
XP	<ul style="list-style-type: none"> • Une forte réactivité au changement des besoins du client. • Un travail d'équipe. • Un code de qualité. • Une efficacité importante. • Un coût de modification linéaire. 	<ul style="list-style-type: none"> • Méthodologie déroulante. • Petits et moyens projets seulement. • Nécessité une forte implication du client
DSDM	<ul style="list-style-type: none"> • Une gestion de projet efficace et un contrôle solide sur le cycle de vie du projet • Exigences en matière d'approche prioritaire utiles à assurer en premier. 	<ul style="list-style-type: none"> • Client relégué au 2nd plan. • La documentation est complexe. • Consomme beaucoup de temps.

L'analyse de la table précédente, me conduit à éliminer la méthode DSDM car le client n'est pas impliqué au premier plan, sa documentation est complexe et sa maintenance est difficile. En plus, j'élimine la méthode XP qui, néglige l'étude fonctionnelle et la capture des besoins fonctionnels et techniques. Par voie de conséquence, je vais opter pour la méthode agile Scrum pour plusieurs raisons. Sa principale caractéristique consiste à valoriser les individus et leurs interactions, le fonctionnement du logiciel, la collaboration avec les clients d'où elle constitue une bonne solution tenant compte du processus, de la documentation complète, de la négociation des contrats et des plans préétablis.

1.4.3.2 Choix de la méthodologie : SCRUM

Mon projet suivra le processus de développement agile SCRUM (processus de mêlée) vu qu'elle présente la méthodologie la plus appropriée à ma démarche projet. Elle fournit

un ensemble de pratiques de base qui laissent volontairement le choix libre des techniques de développement. Dans ce qui suit, je cite quelques avantages offerts par la méthodologie SCRUM [CS16] :

- Une flexibilité : Je priorise moi-même les exigences selon les besoins de mon organisation et les circonstances changeantes.
- Une mise à jour des tâches et des priorités : le client bénéficie d'une flexibilité au niveau de la définition, de l'évolution des fonctionnalités et des séquences d'activités.
- Une idée claire de l'état du projet : je peux consulter l'avancement du projet, évaluer éventuellement les obstacles rencontrés et suivre les tâches de différentes équipes.
- Une planification correcte : une planification détaillée à court terme et une planification globale à long terme.

1.4.4 Scrum

1.4.4.1 Les bases de SCRUM

La méthodologie SCRUM est basée sur 3 principes qui sont au même temps les principes de la culture agile qui sont [Aub11] :

- **Transparence** : Scrum met l'accent sur le fait d'avoir un langage commun.
- **Inspection** : À intervalle régulier, Scrum propose de faire le point sur les différents artefacts produits, afin de détecter toute variation indésirable.
- **Adaptation** : Si une dérive est constatée pendant l'inspection, le processus doit alors être adapté.

1.4.4.2 Les acteurs dans SCRUM

Scrum définit 3 rôles [Aub11] :

- **Le Product owner** : C'est le maître d'ouvrage, également le propriétaire du produit il est responsable de maximiser le travail de l'équipe. Il est le seul de gèrent du carnet du produit il définit les exigences que doit satisfaire le produit, ajuste et règle les fonctionnalités du produit au cours de chaque itération. C'est lui également qui donne l'ordre d'approbation ou les refus de travail présenté.
- **L'équipe de développement** : Celle-ci est responsable de transformer les besoins exprimés dans le Product Backlog par le Product owner en fonctionnalités concrètes et utilisables. Elle doit livrer un incrément, une version du logiciel, à la fin de chaque sprint. L'équipe de développement est auto – organisée. Nulle personne ne peut lui indiquer comment transformer les histoires du Product backlog. On dit qu'elle est autonome. Elle est pluridisciplinaire car elle peut contenir d'autres rôles tels que développeur, concepteur, designer.

- **Le Scrum master :** Le scrum master doit maîtriser SCRUM dans son intégralité. Il doit s'assurer que la méthode scrum doit être mise en œuvre correctement, que la démarche de l'équipe de développement s'adapte selon les pratiques de la méthode, généralement il est confondu avec le chef du projet. Il a comme mission de :
 - Diriger et aider les membres de l'équipe ;
 - Gérer efficacement le carnet du produit, afin qu'il soit concis et clair pour l'équipe ;
 - Aider l'équipe à éliminer les obstacles à son progrès ;
 - S'assurer que l'équipe est productive.

1.4.4.3 Terminologie Scrum

Dans Scrum, nous utilisons la terminologie suivante [CS16] :

- **Daily Scrum (ou mêlée quotidienne)** : moment de partage, courte « cérémonie » ou réunion (environ 15 minutes) menée chaque jour avec les membres de l'équipe et le product owner pour informer sur l'activité de tous, déterminer les tâches de la journée et identifier les obstacles empêchant l'équipe d'atteindre l'objectif du sprint.
- **Product Backlog (ou produit backlog)** : liste priorisée des fonctionnalités du produit, soit une liste partagée des choses à faire. Rétrospective : « cérémonie » à l'issue de chaque sprint dont l'objectif est de discuter, en présence du Scrum master et du product owner, des axes d'amélioration de l'équipe. Moment capital qui incarne l'un des principes fondamentaux énoncés par le Manifeste agile : à intervalles réguliers, l'équipe réfléchit sur la manière de préparer le futur et d'accroître son efficacité, puis adapte et ajuste son comportement en conséquence.
- **Sprint Backlog** : liste des fonctionnalités à réaliser lors d'un sprint. La liste des tâches correspondantes est établie lors d'un sprint planning. La charge des tâches est déterminée par les développeurs.
- **Sprint Planning (ou planification)** : il réunit l'équipe et le product owner pour déterminer l'objectif et le contenu du sprint à venir. Les user stories sont découpées en tâches faisant chacune l'objet d'une estimation des charges, exprimée en heures ou en jours, par l'équipe. Le principal résultat est le sprint backlog.
- **Sprint Review (ou revue de sprint)** : elle consiste pour l'essentiel, au terme de chaque cycle, à faire une démonstration publique du résultat du sprint, à collecter le feedback des parties prenantes et à prendre des décisions. User Story : élément du backlog. Une exigence du système à développer, une brève description d'une fonctionnalité telle que vue par l'utilisateur (client). Vision produit (ou vision) : décrivant les principaux objectifs, jalons, utilisateurs visés et rédigée par le Product Owner, elle contribue à guider et à fédérer les acteurs du projet.

1.4.4.4 Processus SCRUM

La Figure ci-après présente le fonctionnement général de la méthode Scrum.

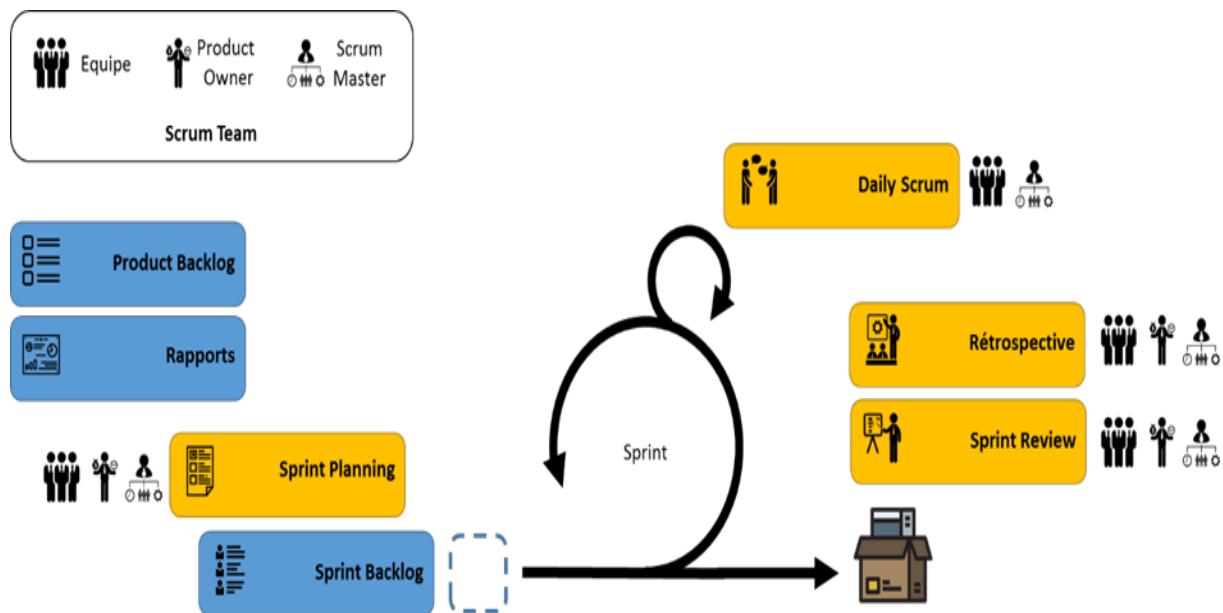


FIGURE 1.6 – Processus Scrum

L'examen de la Figure 1.4 me permet de conclure que :

- Scrum est basé sur des itérations (périodes courtes de développement dont les objectifs sont définis à l'avance) appelées sprints.
- Les objectifs font partie d'un référentiel d'exigences appelé le product backlog qui est et tenu à jour par le product owner ("Correspondant métier").
- Ce référentiel est composé de fonctionnalités constamment priorisées. Avant chaque sprint, les fonctionnalités les plus prioritaires passent dans le sprint backlog et deviennent donc les objectifs à réaliser durant le sprint (itération).
- Un sprint démarre toujours par sa planification en partant de discussions entre le product owner et l'équipe concernant le product backlog. A l'issue de cette rencontre, des tâches sont définies et le sprint peut débuter.
- L'équipe de développement est pilotée par le Scrum Master qui a pour but de résoudre les obstacles, participer au développement en cas de besoin et tout mettre en oeuvre pour que les objectifs soient réalisés durant le sprint.
- Chaque sprint améliore la valeur ajoutée du produit en ajoutant de nouvelles fonctionnalités qui peuvent être livrées au client.

1.5 Langage de modélisation

1.5.1 Processus de modélisation

Le processus de modélisation vise à obtenir une solution acceptable du système informatique. La solution finalement retenue n'est pas obtenue en une seule itération. Plusieurs étapes sont nécessaires ; ces étapes successives permettent de raffiner le niveau de détails du système à réaliser. Les premières étapes donnent une vision à très gros grains et permettent d'avancer dans la compréhension du problème [MG00].

1.5.2 Langage de Modélisation Unifié : UML

UML est un langage de modélisation unifié basé sur des notations graphiques. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement [MG00].



FIGURE 1.7 – Langage UML

1.6 Environnement de développement

1.6.1 Environnement matériel

Pour la conception et la réalisation de ce projet, j'ai utilisé un ordinateur portable dont la configuration est présentée dans la TABLE 1.2.

TABLE 1.2 – Environnement matériel.

Processeur	RAM	Système d'exploitation
Intel(R) Core(TM) i5-10500H	8 Giga octets	Microsoft Windows 10 de 64 bits

1.6.2 Environnement logiciel

Dans cette partie, je vais présenter l'environnement logiciel de mon projet de point de vue Technologies, Langages, Frameworks, Logiciels et Services.

1.6.2.1 Outil de conception

StarUml

C'est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux comme Rational Rose¹ ou Together².



FIGURE 1.8 – Logo de StarUml.

1.6.2.2 Outil de rédaction du rapport

Overleaf

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

Cette plateforme est très compatible avec différents supports tels que tablettes et smartphones [Bas16].



FIGURE 1.9 – Logo de Overleaf.

1.6.2.3 Outils de tests et intégrations

Postman

C'est un API client populaire qui permet aux développeurs de créer, partager, tester et documenter facilement les API. Pour ce faire, il permet aux utilisateurs de créer et d'enregistrer des requêtes HTTP/S simples et complexes, ainsi que de lire leurs réponses [Pos].



FIGURE 1.10 – Logo de Postman.

1.6.2.4 Technologies utilisées et Langages de développement

MongoDB

C'est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données



FIGURE 1.11 – Logo de MongoDB

Angular 13

C'est un cadre de conception d'applications et une plateforme de développement permettant de créer des applications monopages efficaces et sophistiquées [Ang].



FIGURE 1.12 – Logo de Angular.

NodeJs

C'est un environnement d'exécution JavaScript open source et multiplateforme. C'est un outil populaire pour presque tous les types de projets! Node.js exécute le moteur JavaScript V8, le cœur de Google Chrome, en dehors du navigateur. Cela permet à Node.js d'être très performant

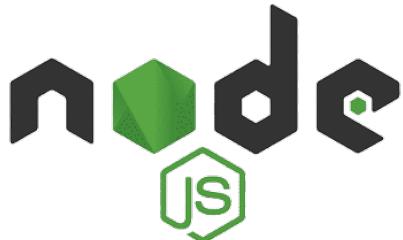


FIGURE 1.13 – Logo de NodeJs

Express Js

C'est une infrastructure d'applications Web Node.js minimalist et flexible qui fournit un ensemble de fonctionnalités robuste pour les applications Web et mobiles.



FIGURE 1.14 – Logo de Express Js.

HTML

Le langage de balisage hypertexte, HTML, est un type de langage informatique descriptif. Plus précisément, c'est un format de données utilisé dans le monde de l'Internet pour la mise en forme des pages Web. [RLJ+99].



FIGURE 1.15 – Logo de HTML.

CSS

Les feuilles de style en cascade (CSS) sont un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage tel que le HTML. [Mey06].



FIGURE 1.16 – Logo de CSS.

Bootstrap

C'est un ensemble d'outils utiles à la création du design d'applications et de sites et web. C'est une collection qui contient des codes CSS et HTML, des outils de navigation, boutons, formulaires, et autres éléments interactifs, ainsi que des extensions JavaScript en option. [Bha15].



FIGURE 1.17 – Logo de Bootstrap.

NPM : Node Package Manager

C'est le gestionnaire de paquets pour Node.js. Il a été créé en 2009 en tant que projet open source pour aider les développeurs JavaScript à partager facilement des modules de code empaquetés. Le registre npm est une collection publique de paquets de code open-source pour Node.js, les applications web frontales, les applications mobiles, les robots, les routeurs et d'innombrables autres besoins de la communauté JavaScript. NPM est le client de ligne de commande qui permet aux développeurs d'installer et de publier ces paquets [npm].



FIGURE 1.18 – Logo de NPM.

TypeScript

C'est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript [Typ].



FIGURE 1.19 – Logo de TypeScript.

JSON : JavaScript Object Notation

C'est un format léger d'échange de données. Il est facile à lire et à écrire pour les humains. Il est facile à analyser et à générer pour les machines. Il est basé sur un sous-ensemble de la norme du langage de programmation JavaScript. JSON est un format texte totalement indépendant du langage, mais qui utilise des conventions familières aux programmeurs de la famille des langages C, notamment C, C++, C#, Java, JavaScript, Perl, Python et bien d'autres [JSO].

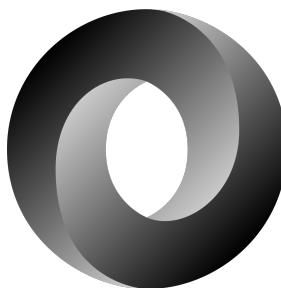


FIGURE 1.20 – Logo de JSON.

1.7 Architecture logicielle du projet

En informatique, architecture désigne la structure générale inhérente à un système informatique, l'organisation des différents éléments du système (logiciels et/ou matériels et/ou humains et/ou informations) et des relations entre ces éléments. Plus simplement l'architecture d'une application sert à décrire d'une manière symbolique et schématique les différents éléments de cette dernière, leurs interrelations et leurs interactions. Comme architecture de notre application, nous allons choisir l'architecture 3-tiers.

1.7.1 Architecture 3-tiers :

L'architecture 3-tiers (le terme "tier" vient de l'anglais et signifie "niveau") est un modèle d'architecture d'application. Son principe de base consiste à séparer trois couches logicielles contenues dans une application :

- **Un client** (le téléphone demandeur de ressources) équipé d'une interface utilisateur.
- **Un serveur d'application** (appelé middleware) qui fournit la ressource mais en faisant appel à un autre serveur.
- **Un serveur de données** qui fournit au serveur d'application les données (et/ou les traitements) requises pour répondre au client.

La figure ci-dessus montre l'architecture de notre application côté front et back :

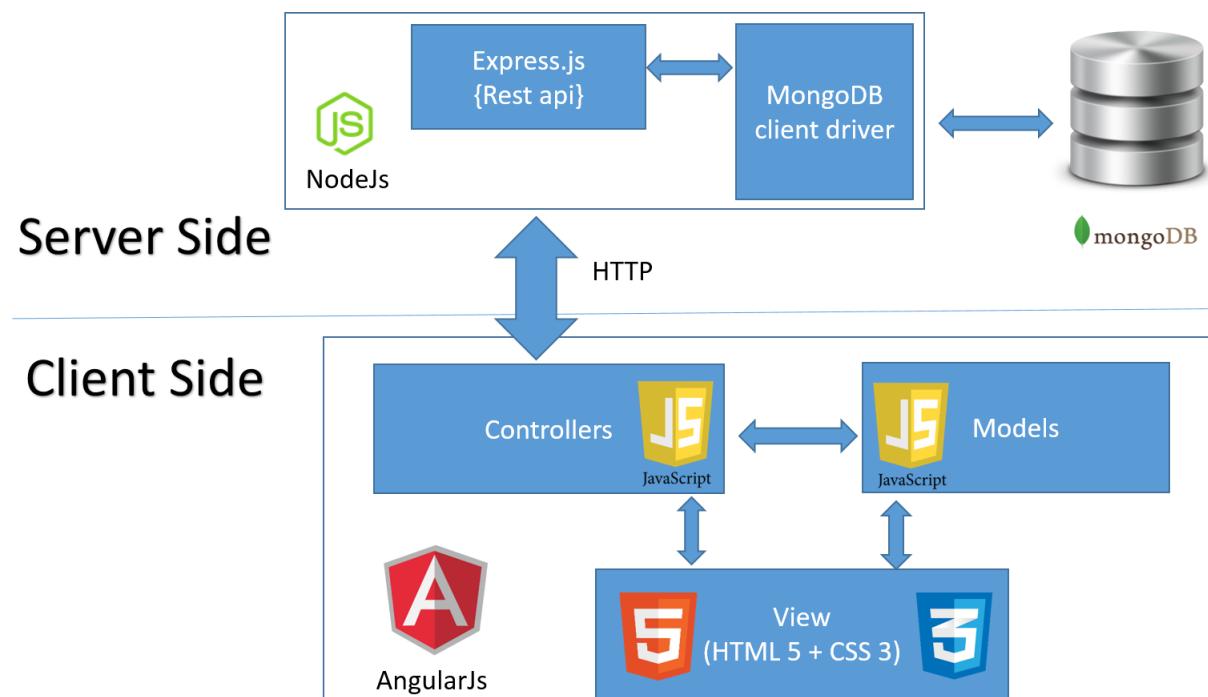


FIGURE 1.21 – Architecture de l'application

1.7.2 Modèle de conception (Design pattern)

Un modèle de conception (ou patron de conception) est un ensemble de “bonnes pratiques”, qui répond à un problème de conception d’une application. Il décrit une manière d’architecturer une application informatique en la décomposant en sous-parties. Pour notre application nous allons utiliser le modèle MVC pour le projet de Backend et MVVM pour le projet Frontend.

1.7.2.1 Modèle de conception Backend (MVC)

L’architecture Model-Vue-Controller consiste à découper son code pour qu’il appartienne à l’une des trois composantes du MVC :

- **Modèle** : Le modèle décrit la logique métier et se caractérise par un ensemble de classes. Il travaille sur la conception de règles métier pour les données et sur la façon dont les données sont modifiées ou traitées.
- **La Vue** : Elle représente les composants de l’interface utilisateur comme HTML, CSS, jQuery, etc. La vue affiche les données que le contrôleur renvoie sous forme de résultat. Le modèle peut également être converti en interface utilisateur à l’aide de la vue.
- **Le Contrôleur** : Cette partie gère la logique du code qui prend des décisions. C’est en quelque sorte l’intermédiaire entre le modèle et la vue : le contrôleur va demander les données au modèle, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue.

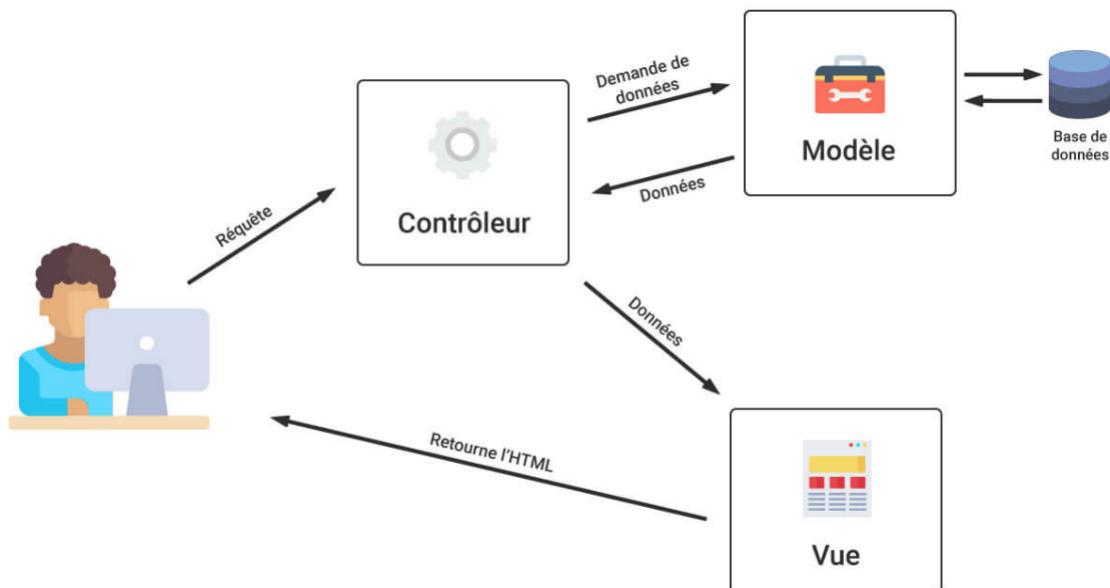


FIGURE 1.22 – Design Pattern MVC

1.7.2.2 Modèle de conception Frontend (MVVM)

Model-View-ViewModel (MVVM) est un modèle de conception structurelle qui sépare les objets en trois groupes distincts :

- Comme on le voit dans le cas de MVC, le modèle ici aussi est utilisé pour décrire la logique métier et un ensemble de classes la caractérisant. Il travaille sur la conception de règles métier pour les données sur la façon dont les données sont modifiées ou traitées. Vue Encore une fois, comme dans le cas de MVC et MVP, la vue ici représente les composants de l'interface utilisateur comme HTML, CSS, jQuery, etc.
- **La Vue** : Affiche les données que le contrôleur renvoie sous forme de résultat. Le modèle peut également être converti en interface utilisateur à l'aide de la vue.
- **View-Model** : Il est de la responsabilité du View-Model de présenter les méthodes et les fonctions. Il doit présenter des commandes pour faire fonctionner le modèle, maintenir l'état de la vue et activer les événements dans la vue.



FIGURE 1.23 – Design Pattern MVVM

Conclusion

Dans ce chapitre, j'ai commencé par cerner le cadre général du projet. Ensuite, j'ai présenté la méthodologie de conception que je vais suivre dans mon application. Enfin, j'ai listé les différents outils et technologies nécessaires pour la mise en place de ma solution. A ce stade, nous pouvons désormais passer au chapitre suivant, qui contiendra le plan du projet et l'identification des fonctionnalités de notre application.

Chapitre 2

Sprint 0 : Analyse et Spécification des Besoins

Sommaire

Introduction	25
2.1 Capture de besoins	25
2.1.1 Identification des acteurs	25
2.1.2 Identification des besoins fonctionnels	25
2.1.3 Identification des besoins non fonctionnels	26
2.2 Diagramme de classes	28
2.3 Pilotage du projet avec SCRUM	29
2.4 Diagramme de cas d'utilisation global	29
2.5 Product Backlog	30
2.6 Planification des sprints	31
Conclusion	31

Introduction

Une étape essentielle de tout cycle de développement logiciel ou conceptuel consiste à effectuer une étude préalable. Le but de cette phase est de comprendre le contexte du système. Il s'agit d'éclaircir au mieux les besoins fonctionnels et non fonctionnels, faire apparaître les acteurs et identifier les cas d'utilisation. Dans ce chapitre, je vais essayer de cerner les besoins sous forme de diagrammes de cas d'utilisation.

2.1 Capture de besoins

Dans cette partie, je vais identifier les acteurs et leurs rôles. Par la suite, je répondrai aux questions suivantes « que doit faire le système ? » et « Quelles sont les contraintes ? » afin d'expliquer les différents besoins fonctionnels et non fonctionnels que mon projet.

2.1.1 Identification des acteurs

On s'intéresse tout d'abord à la présentation des acteurs qui sont en contact direct avec l'application et qui profitent des différents accès aux données. (Notons ici que tout genre d'accès est obligatoirement authentifié) La table ci-après présente les différents acteurs interagissant avec le système.

TABLE 2.1 – Table des acteurs.

Acteur	Description
Administrateur	La personne accédant à la plateforme en ayant le rôle de gérer les directions, gérer le personnel et gérer les types de tickets.
Utilisateur	La personne accédant à la plateforme en ayant la possibilité de gérer les tickets.
Responsable	L'utilisateur accédant à la plateforme en ayant la possibilité d'approuver ou refuser les tickets demandés par les utilisateurs.
Chargé de tickets	L'utilisateur accédant à la plateforme en ayant la possibilité d'accepter ou refuser les tickets déjà approuvés par le responsable.

2.1.2 Identification des besoins fonctionnels

Les besoins fonctionnels correspondent aux fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée/sortie du système.

Je présenterai ainsi tous les besoins fonctionnels de mon projet :

- **S'authentifier**

Chaque membre dispose d'une adresse mail et d'un mot de passe spécifique, qui assurent la vérification de son identité et lui autorisent l'accès pour bénéficier des services fournis par l'application de manière totalement sécurisée.

- **Gérer directions**

Le système permet à l'administrateur de consulter, ajouter et supprimer les directions.

- **Gérer types de tickets**

Le système permet à l'administrateur de consulter, modifier, ajouter et supprimer les types des tickets.

- **Gérer personnel**

L'administrateur peut consulter, modifier, ajouter et supprimer les personnels.

- **Gérer tickets**

Le système permet à l'utilisateur de créer , consulter et suivre ses tickets

- **Approuver tickets**

Le système permet au responsable d'approuver ou refuser les tickets demandés.

- **Accepter tickets**

Le système permet au chargé des tickets d'accepter ou refuser les tickets approuvés par le responsable.

2.1.3 Identification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Concernant mon application, j'ai dégagé les besoins suivants :

- **Temps de réponse** : Le temps de réponse aux actions des utilisateurs doit être raisonnable.
- **Robustesse** : Le système ne doit pas se planter ou perdre des données lorsqu'il est soumis à des sollicitations inhabituelles. En effet, il s'agit d'une mesure de disponibilité des systèmes et de l'intégrité des informations en situation de stress.
- **Disponibilité** : La disponibilité du système est un enjeu majeur pour les utilisateurs. En effet, il doit être toujours accessible 24h/24, 7j/7.
- **Portabilité** : La compatibilité avec diverses plateformes, facilité de remplacement d'autres systèmes en place, facilité d'installation et de désinstallation de l'application
- **Extensibilité** : L'application devra être extensible, c'est-à dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.
- **Convivialité et simplicité** : L'application doit être facile à utiliser. En effet, les interfaces utilisateur doivent être simples, ergonomiques et adaptées à l'utilisateur.

- **Sécurité** : La notion de sécurité est nécessaire dans mon système, les différents composants doivent communiquer entre eux de manière sécurisée.

The screenshot shows a MongoDB interface with the database name `helpDesk.users`. At the top right, it displays "2 DOCUMENTS" and "3 INDEXES". Below the header, there are tabs for `Documents`, `Aggregations`, `Schema`, `Explain Plan`, `Indexes`, and `Validation`. The `Documents` tab is selected.

On the left, there is a search bar with the query `{ field: 'value' }` and buttons for `FILTER`, `OPTIONS`, `FIND`, `RESET`, and three dots. Below the search bar are buttons for `ADD DATA`, `VIEW`, and three icons. To the right, it says "Displaying documents 1 - 2 of 2" with navigation arrows and a `REFRESH` button.

The first document listed has the following fields:

```
_id: ObjectId('62535dd5c658b59e344fb4f7')
firstname: "alaa"
lastname: "khouja"
email: "alaa@gmail.com"
matricule: "azerty"
statu: false
hierarchy: "null"
directionname: "FIN"
role: "chFourniture"
password: "$2b$10$TlxWa]XsDisGyyg8T4j2ReNIQH]6j2qvMfSG9WKaX/wMhA0qhNuow"
_v: 0
```

The second document listed has the following fields:

```
_id: ObjectId('6255ddbd8fa2fb1fd66d6230')
firstname: "admin"
lastname: "admin"
email: "admin@gmail.com"
matricule: "admin"
statu: false
hierarchy: "null"
directionname: "ADMIN"
role: "admin"
password: "$2b$10$xJ1tBCueicGQ2hkNGrcvle/xFogg6FZtg96bhhlfcnxV0Y84w3Iem"
_v: 0
```

FIGURE 2.1 – Hachage mot de passe.

2.2 Diagramme de classes

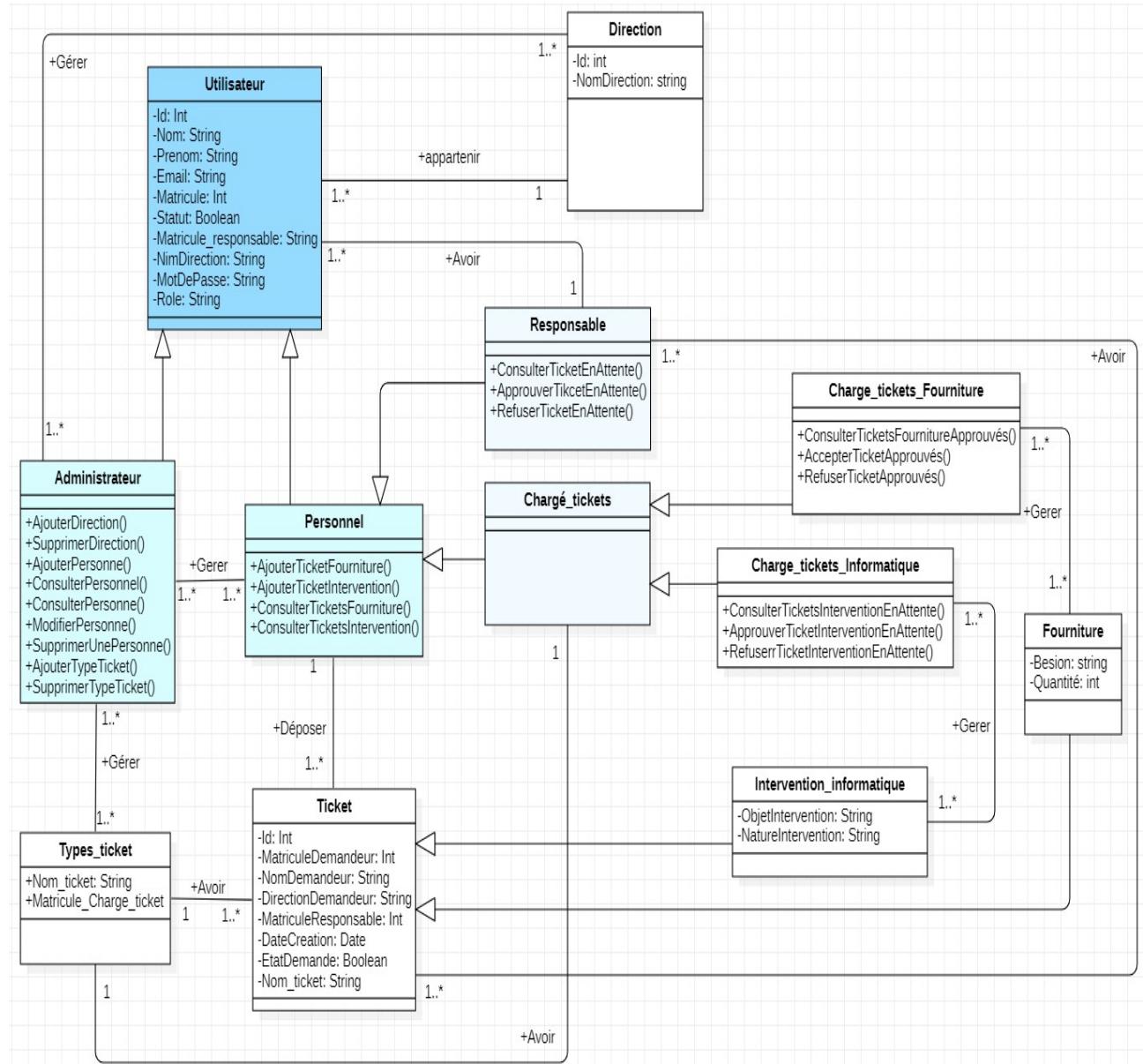


FIGURE 2.2 – Diagramme de classes global.

2.3 Pilotage du projet avec SCRUM

Comme indiqué dans le chapitre précédent, j'ai choisi la méthodologie agile SCRUM lors de la mise en place de l'application. En effet, la méthodologie SCRUM nécessite la collaboration de plusieurs intervenants, pour mon projet :

- Product Owner : Mr LAABIDI Aladine
- Scrum Master : Mme DEGDEGUI Aziza.
- Scrum Team : KHOUJA Zied.

2.4 Diagramme de cas d'utilisation global

Pour mieux comprendre le fonctionnement de l'application, je présente le diagramme de cas d'utilisation système. C'est un diagramme général qui englobe les fonctionnalités principales que doit fournir l'application qui est présenté dans la Figure ci-après.

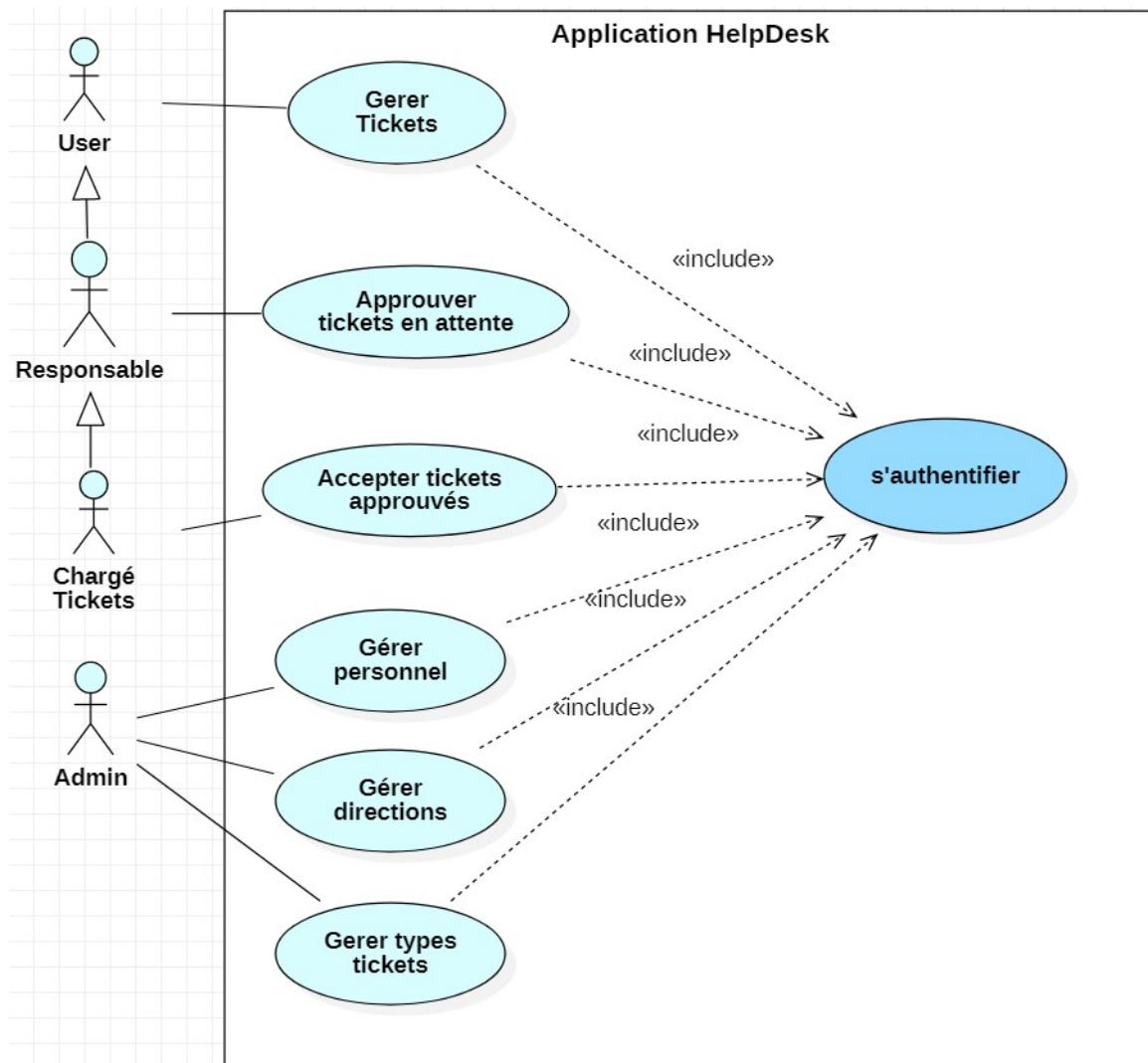


FIGURE 2.3 – Diagramme de cas d'utilisation global.

2.5 Product Backlog

Un Backlog est une liste de fonctionnalités ou de tâches, jugées nécessaires et suffisantes pour la réalisation satisfaisante du projet. Les éléments du Backlog sont appelés des histoires utilisateur (user story). Chacune de ces histoires doit avoir un but métier . Pour le traitement des histoires, j'ai choisi de commencer par les tâches les plus prioritaires. En effet, chaque histoire utilisateur possède une estimation qui est l'évaluation initiale de l'équipe sur la quantité de travail nécessaire en jours pour implémenter cette exigence. Ce backlog servira à piloter l'équipe de développement et pourra évoluer tout au long de mon projet. Pour estimer la complexité des User Story, je vais affecter un nombre entier allant de 1 à 3.

La TABLE suivante résume le backlog produit de l'application.

TABLE 2.2 – Le backlog du produit

ID	Fonctionnalité	User stories	Priorité	Complexité
1	S'authentifier	En tant qu'administrateur /utilisateur/responsable/charge de tickets, je dois m'authentifier afin d'accéder à mon interface de travail	Élevée	1
2	Gérer directions	En tant qu'administrateur, je peux gérer les directions	Élevée	1
3	Gérer types tickets	En tant qu'administrateur, je peux gérer les types tickets	Élevée	1
4	Gérer personnel	En tant qu'administrateur, je peux gérer le personnel	Élevée	3
5	Gérer tickets	En tant qu' utilisateur/responsable/charge de tickets, je peux gérer les tickets	Élevée	3
6	Approuver tickets en attente	En tant que responsable, je peux approuver ou refuser les tickets en attente	Élevée	1
7	Accepter tickets approuvés	En tant que chargé de tickets, je peux accepter ou refuser les tickets approuvés	Élevée	1

2.6 Planification des sprints

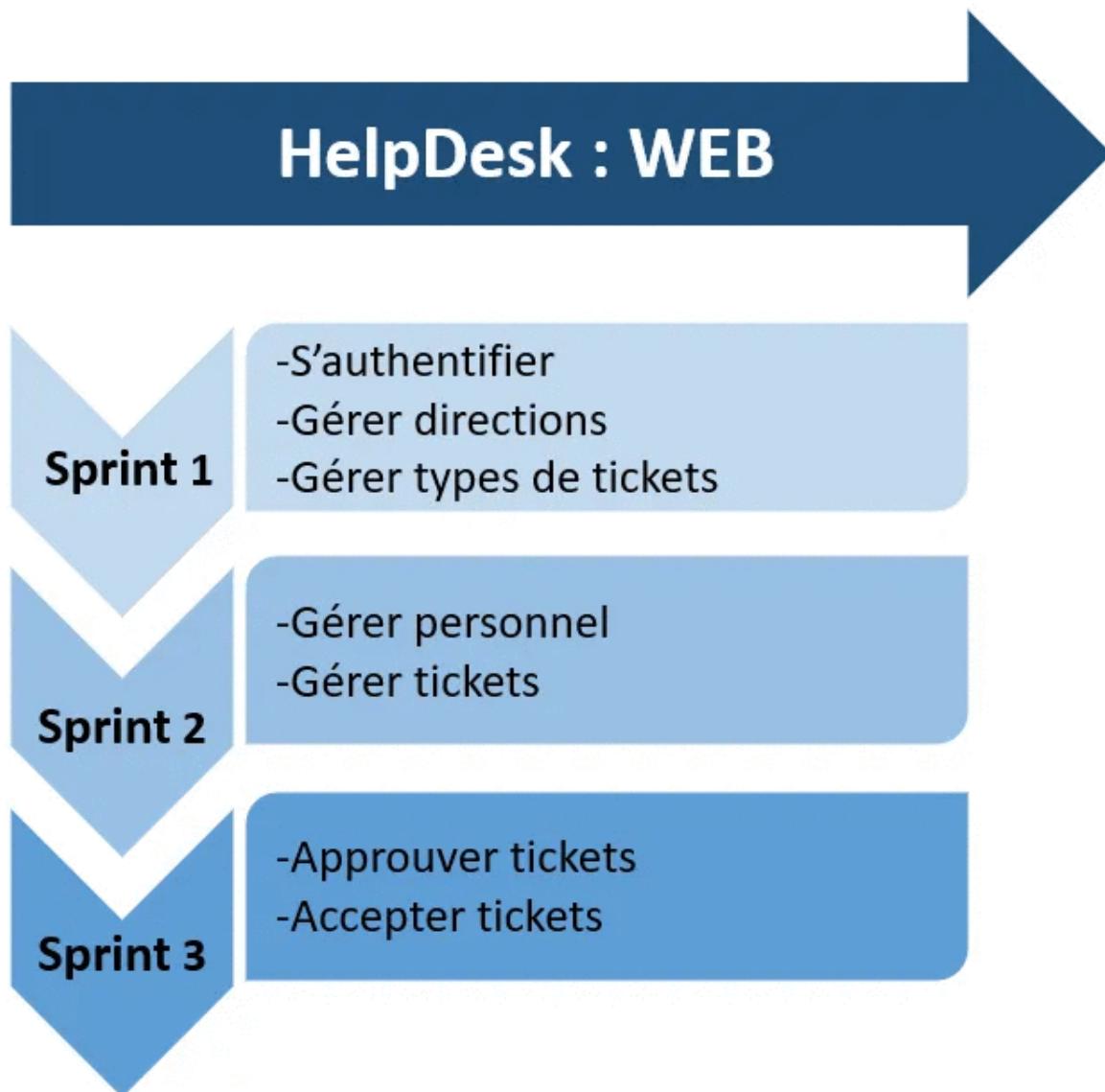


FIGURE 2.4 – Planification des sprints

Conclusion

Tout au long de ce chapitre, j'ai détaillé les besoins attendus de ma solution. J'ai défini par la suite le diagramme de cas d'utilisation général, le diagramme de classes et le backlog produit ainsi que la planification des différents sprints.

Le chapitre suivant sera consacré à la description des différentes étapes de mise en place du premier sprint.

Chapitre 3

Sprint 1

Sommaire

Introduction	33
3.1 Spécification des besoins	33
3.1.1 Diagramme de cas d'utilisation du sprint 1	33
3.1.2 Backlog du sprint 1	34
3.2 Analyse	35
3.2.1 Raffinement des cas d'utilisation	35
3.2.2 Diagrammes de séquences système	40
3.3 Conception	44
3.3.1 Diagrammes de séquences détaillés	44
3.3.2 Diagrammes d'activité	48
3.4 Réalisation	51
3.5 Tests et validation	54
Conclusion	54

Introduction

Ce chapitre est consacré aux travaux effectués durant le premier sprint et ceci en décomposant les User Stories du sprint en des tâches présentées dans le Backlog de sprint. Je passe par une phase de spécification fonctionnelle et une phase de conception, pour clôturer enfin par quelques interfaces réalisées lors de la phase de réalisation.

Rappelons que ce premier sprint contient les fonctionnalités suivantes :

- S'authentifier
- Gérer les directions
- Gérer les types de tickets

3.1 Spécification des besoins

Dans cette partie, je présente le diagramme de cas d'utilisation du sprint ainsi que le backlog du sprint.

3.1.1 Diagramme de cas d'utilisation du sprint 1

La Figure ci-après représente le diagramme de cas d'utilisation du sprint 1.

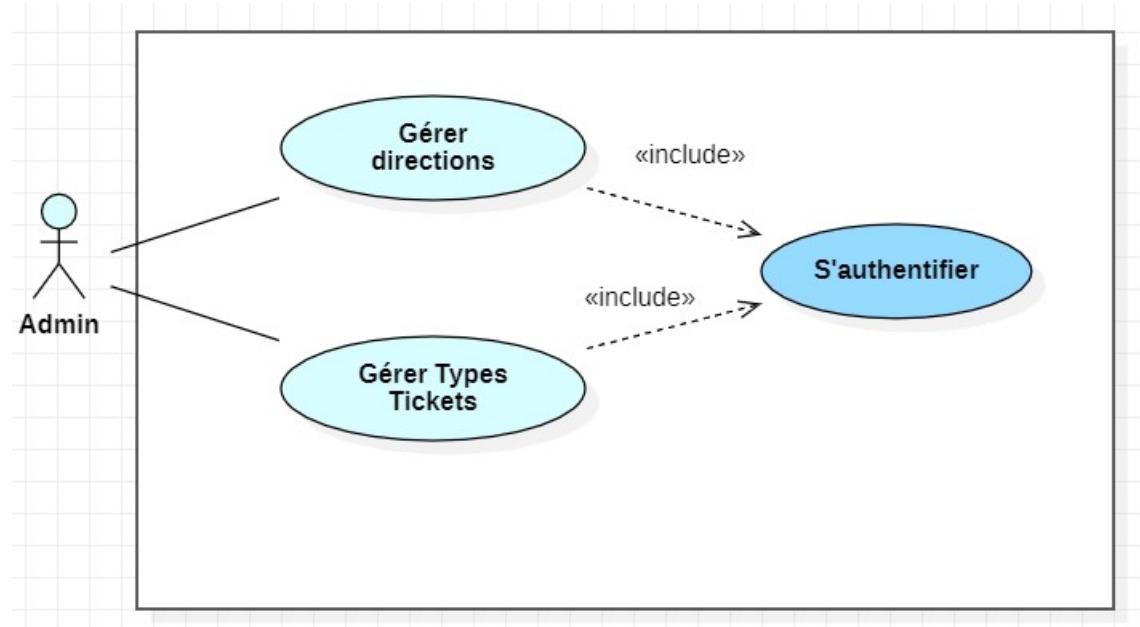


FIGURE 3.1 – Diagramme de cas d'utilisation du sprint 1

3.1.2 Backlog du sprint 1

Avant de commencer le travail et après le choix des User Stories à réaliser durant le sprint 0, j'arrive à l'étape de décomposition des User Stories en des tâches élémentaire. Je présente à travers le tableau 3.1 le Backlog du sprint 1 détaillé avec complexité de chaque tâche qui peut avoir 1 au minimum et 3 au maximum.

TABLE 3.1 – Le Backlog du sprint 1

ID	User Stories	ID	Type	Tache	Complexité
1	En tant que responsable de développement, je veux adopter les meilleures technologies qui répondent au mieux aux besoins, de même que monter en compétences.	1.1	-	Rechercher et déterminer les meilleures technologies qui seront utilisées dans mon projet.	-
2	En tant qu'administrateur je dois m'authentifier.	2.1	Front-End	Mettre en place un bouton devant chaque catégorie pour y accéder.	1
		2.2	Front-End	Développer une barre latérale commune mais personnaliser pour chaque catégorie de notification.	3
		-	Test	Test et validation.	-
		-	Conception	Conception des diagrammes.	-
3	En tant qu'administrateur je peux consulter, ajouter et supprimer une direction.	3.1	Back-end	Créer l'entité 'direction' dans la base de données.	1
		3.2	Back-End	Créer un web service 'directionAPI' pour gérer les direction.	1
		3.3	Front-End	Créer un service web front.	2

		3.4	Front-End	Créer l'interface gérer "direction".	2
		-	Test	Test et validation.	-
		-	Conception	Conception des diagrammes.	-
4	En tant qu'administrateur je peux consulter, ajouter et supprimer une type de ticket.	4.1	Back-end	Créer l'entité 'type ticket' dans la base de données.	1
		4.2	Back-End	Créer un web service 'typeTicketAPI' pour gérer les types tickets.	1
		4.3	Front-End	Créer un service web front.	2
		4.4	Front-End	Créer l'interface gérer "type tickets".	2
		-	Test	Test et validation.	-
		-	Conception	Conception des diagrammes.	-

3.2 Analyse

Afin de mieux assimiler les cas d'utilisation, j'ai établi leurs raffinements pour livrer une description détaillée sur les différents scénarios possibles.

3.2.1 Raffinement des cas d'utilisation

Dans cette sous-section, je vais présenter les raffinements des diagrammes de cas d'utilisation liés au premier sprint, ainsi qu'une description textuelle des principaux cas d'utilisation.

3.2.1.1 Cas d'utilisation « S'authentifier »

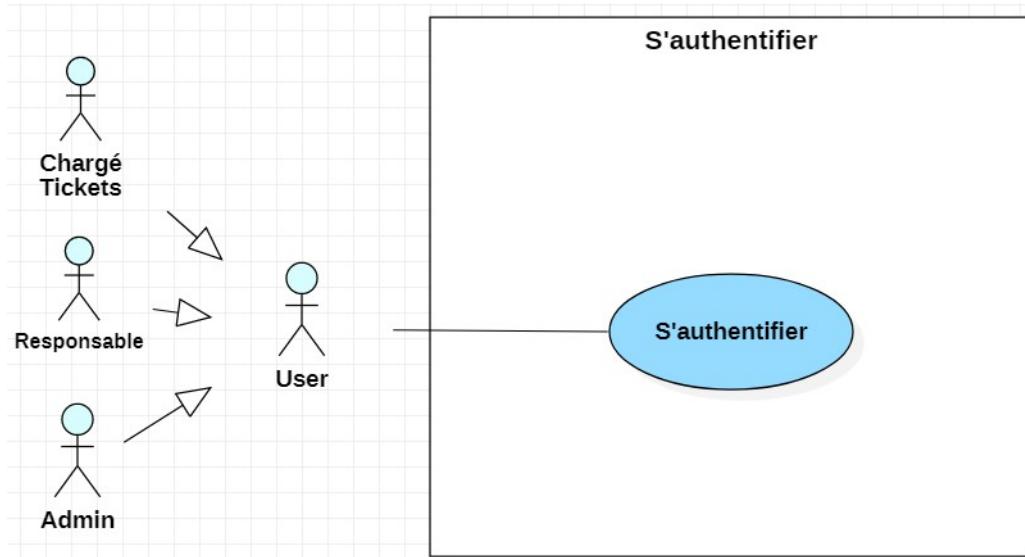


FIGURE 3.2 – Diagramme de cas d'utilisation « S'authentifier ».

TABLE 3.2 – Description textuelle du cas d'utilisation « S'authentifier »

Cas d'utilisation	S'authentifier
Acteurs	User / Responsable / Chargé tickets / Administrateur
Pré-condition	L'utilisateur doit avoir un compte et le login et le mot de passe saisis doivent être corrects.
Scénario Nominal	<ol style="list-style-type: none"> Afficher l'interface de l'authentification Saisir le login et le mot de passe et valider Le système vérifie les coordonnées Le système affiche l'interface d'accueil propre à l'utilisateur.
Post-condition	L'acteur accède à la configuration des notifications.
Exception	Données saisies invalides : Affichage d'un message d'erreur

3.2.1.2 Cas d'utilisation « Gérer directions »

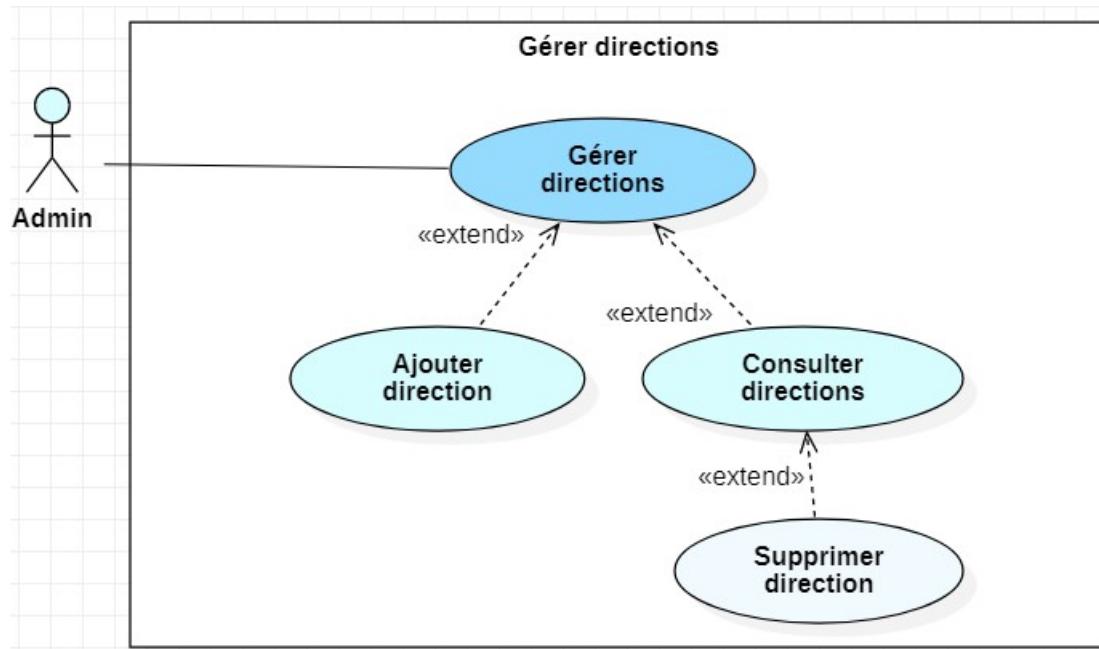


FIGURE 3.3 – Diagramme de cas d'utilisation ”Gérer directions ”.

TABLE 3.3 – Description textuelle du sous cas d'utilisation « Ajouter direction »

Cas d'utilisation	Ajouter direction
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	1. L'administrateur clique sur bouton «Ajouter direction». 2. Le système affiche le formulaire d'ajout d'une direction. 3. L'administrateur remplit le formulaire d'ajout d'une direction et clique sur le bouton «Enregistrer». 4. Le système ajoute la direction à la liste des directions.
Post-condition	Direction ajoutée.
Exception	Données invalides : Affichage d'un message d'erreur "Direction déjà existante dans la base de données"

TABLE 3.4 – Description textuelle du sous cas d'utilisation « Consulter directions »

Cas d'utilisation	Consulter directions
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	1. L'administrateur ouvre l'interface Gérer directions. 2. La table des directions s'affiche.
Post-condition	La liste des directions est affichée.
Exception	Problème de connexion.

TABLE 3.5 – Description textuelle du sous cas d'utilisation « Supprimer direction »

Cas d'utilisation	Supprimer direction
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. Afficher la liste des directions. 2. L'administrateur choisit la direction à supprimer et clique sur le bouton supprimer. 3. Le système affiche un message de confirmation 4. L'administrateur clique sur le bouton «Confirmer» . 5. Le système affiche un message de succès de suppression de direction. 6. Le système affiche la nouvelle liste des direction.
Post-condition	Direction Supprimée.
Exception	<ul style="list-style-type: none"> *Affichage d'un message d'erreur " impossible de supprimer cette Direction ". *Problème de connexion à la base de données

3.2.1.3 Cas d'utilisation « Gérer types tickets »

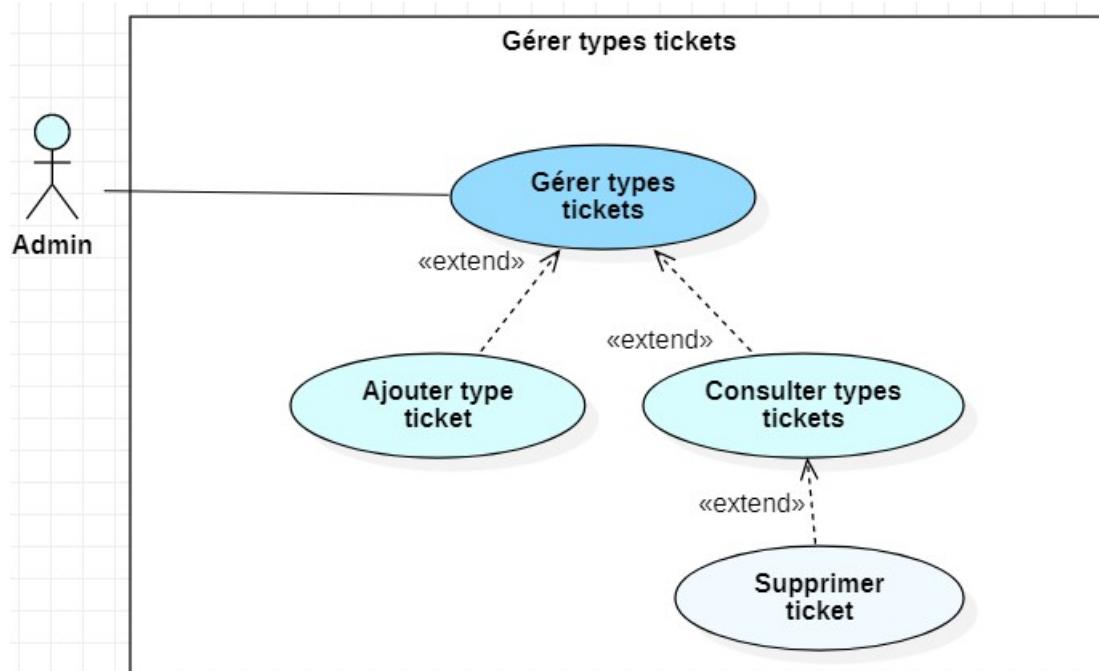


FIGURE 3.4 – Diagramme de cas d'utilisation « Gérer types de tickets ».

TABLE 3.6 – Description textuelle du sous cas d'utilisation « Ajouter type ticket »

Cas d'utilisation	Ajouter type de ticket
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur clique sur bouton «Ajouter type ticket». 2. Le système affiche le formulaire d'ajout d'un type de ticket. 3. L'administrateur remplit le formulaire d'ajout d'un type de ticket et clique sur le bouton «Enregistrer». 4. Le système ajoute le type de ticket à la liste des types tickets.
Post-condition	Type de ticket ajouté.
Exception	Données invalides : Affichage d'un message d'erreur "Type de ticket déjà existant dans la base de données". Retour à l'étape 2 du scénario nominal.

TABLE 3.7 – Description textuelle du sous cas d'utilisation « Consulter types tickets »

Cas d'utilisation	Consulter type tickets
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur ouvre l'interface Gérer type de tickets. 2. La table Type tickets s'affiche.
Post-condition	La liste Type de tickets est affichée.
Exception	Problème de connexion à la base de données.

TABLE 3.8 – Description textuelle du sous cas d'utilisation « Supprimer type de ticket »

Cas d'utilisation	Supprimer type de ticket
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. Afficher la liste des types tickets. 2. L'administrateur choisit le type de ticket à supprimer et clique sur le bouton supprimer. 3. L'administrateur clique sur le bouton «Confirmer» . 4. Le système affiche un message de succès de suppression de type de ticket. 5. Le système affiche la nouvelle liste types tickets.
Post-condition	Type de ticket Supprimé.
Exception	Problème de connexion à la base de données.

3.2.2 Diagrammes de séquences système

Le diagramme de séquences modélise l'aspect dynamique du système, il s'agit d'une séquence d'interactions d'un point de vue temporel entre le système et les acteurs. Son rôle était de décrire graphiquement un scénario d'un cas d'utilisation [DV16].

3.2.2.1 Cas d'utilisation « S'authentifier »

La Figure suivante illustre le diagramme de séquences système du cas d'utilisation « S'authentifier ».

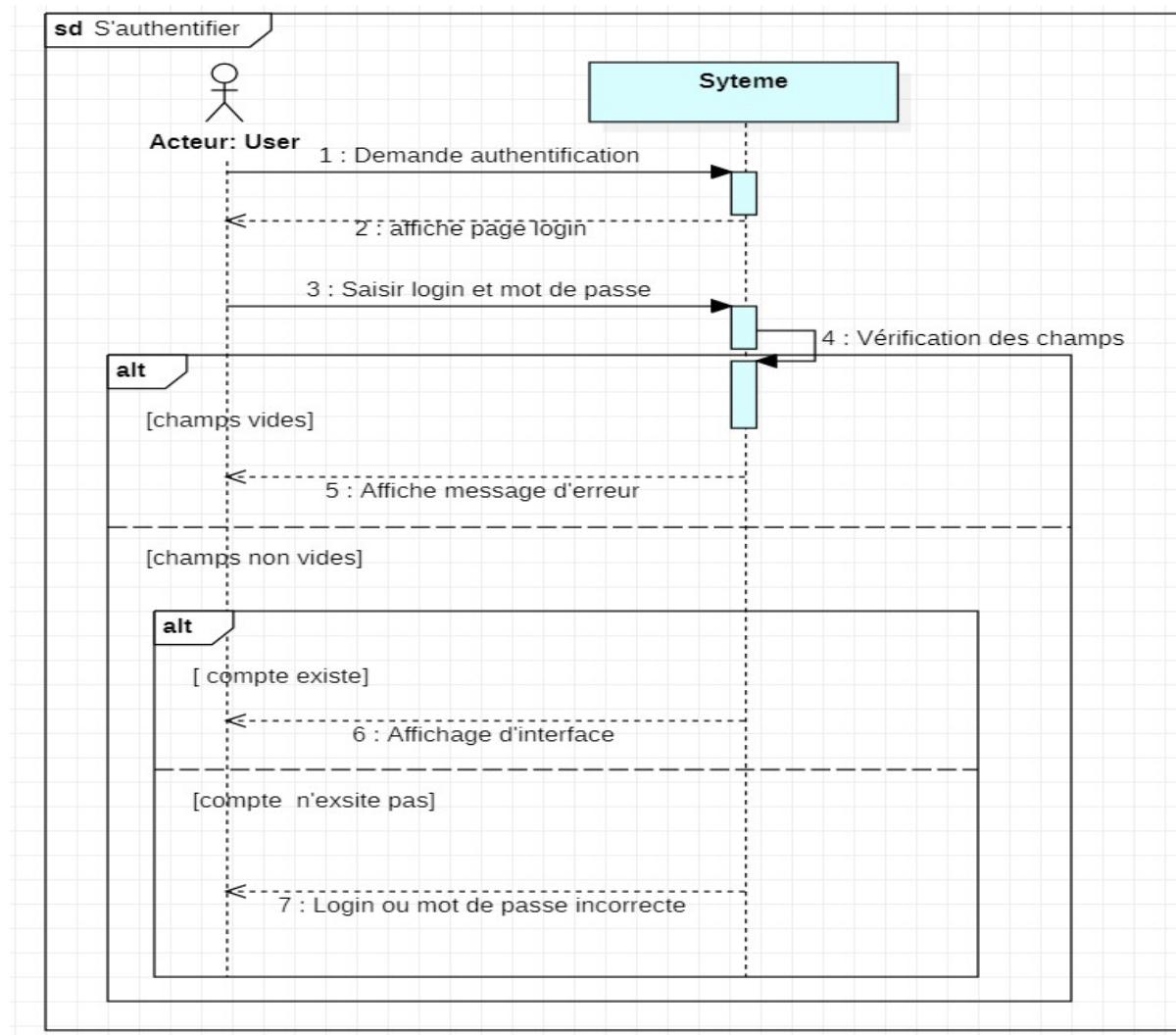


FIGURE 3.5 – Diagramme de séquences système du cas d'utilisation « S'authentifier ».

3.2.2.2 Cas d'utilisation « Ajouter direction »

Dans ce qui suit le diagramme de séquences système du sous cas d'utilisation « Ajouter Direction ».

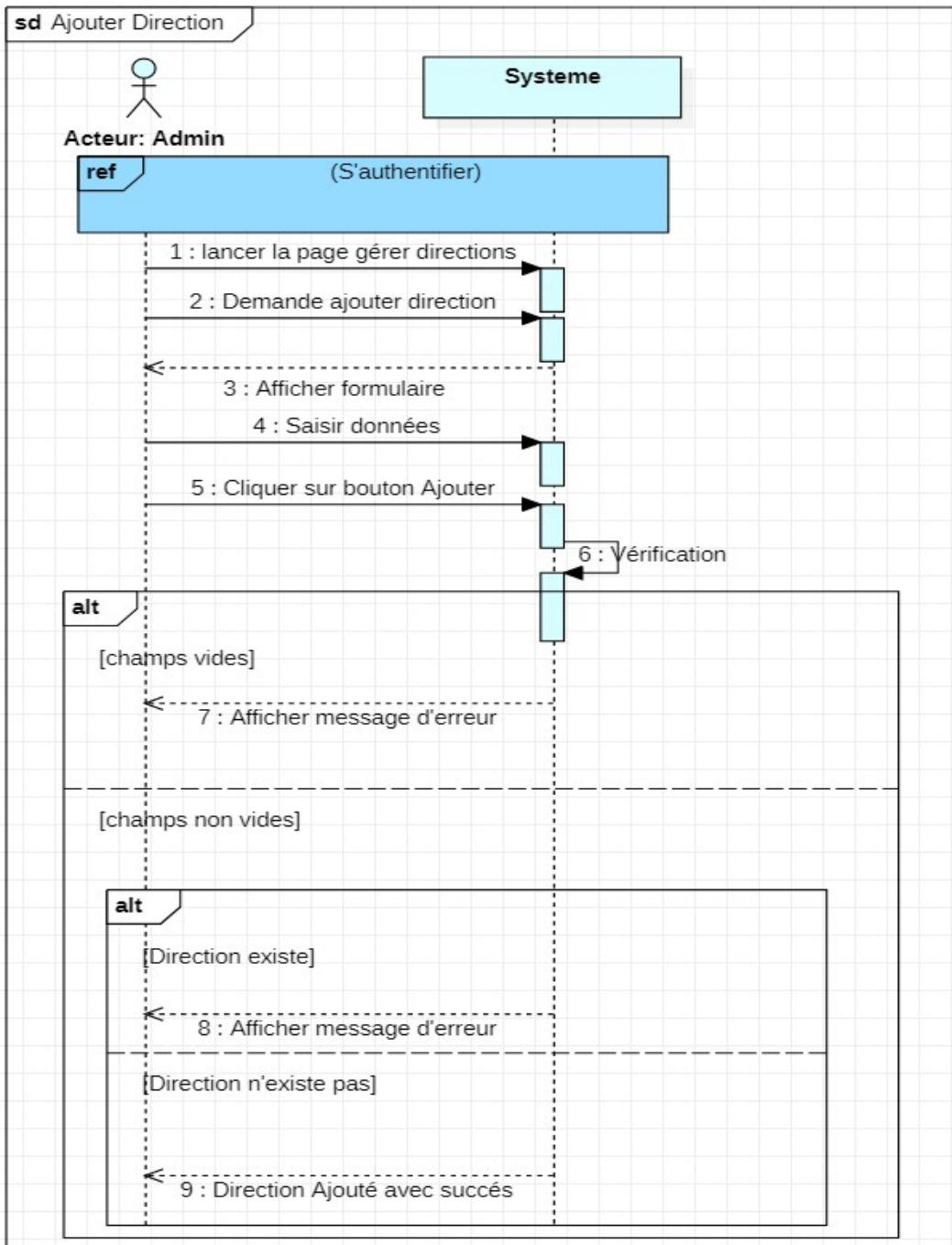


FIGURE 3.6 – Diagramme de séquences système du sous cas d'utilisation « Ajouter Direction »

3.2.2.3 Cas d'utilisation « Consulter direction »

La Figure ci-dessous présente le diagramme de séquences système du sous cas d'utilisation « Consulter Directions ».

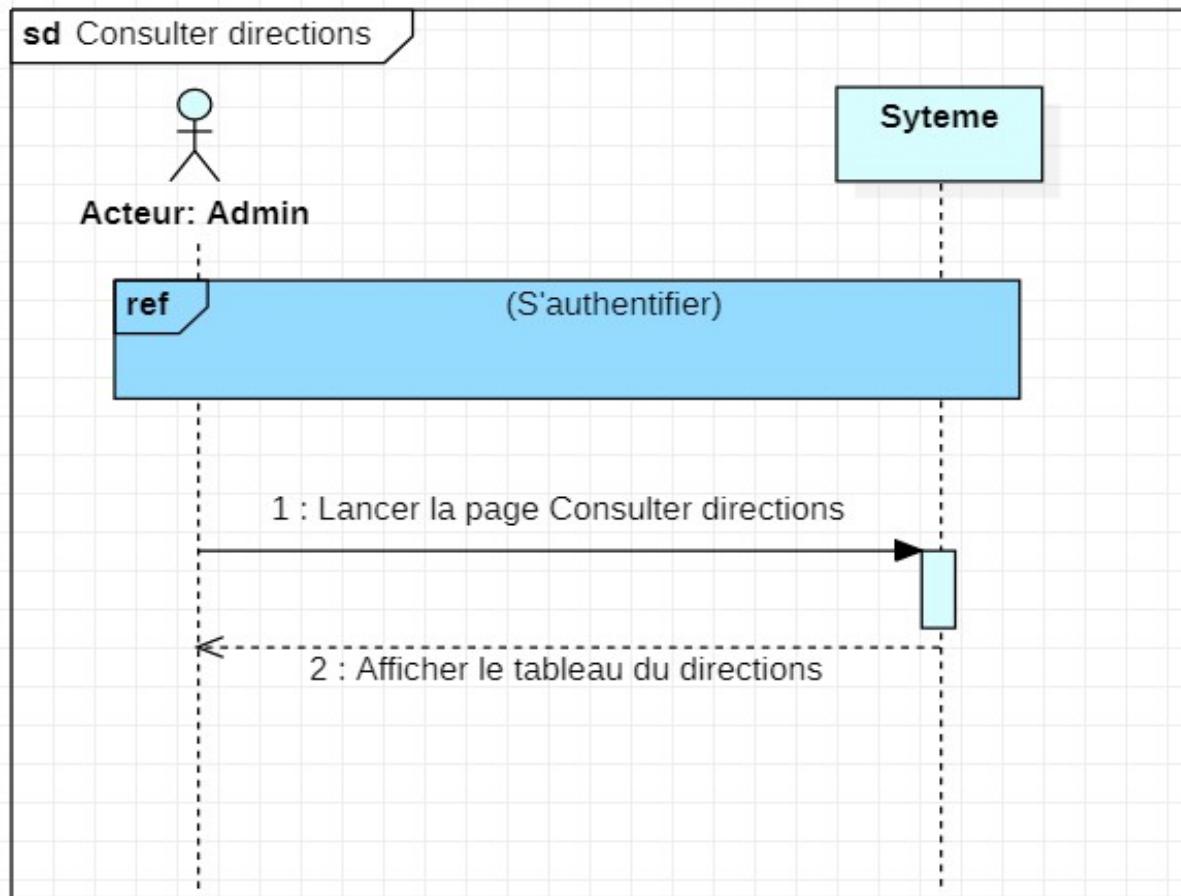


FIGURE 3.7 – Diagramme de séquences système du sous cas d'utilisation « Consulter direction ».

3.2.2.4 Cas d'utilisation « Supprimer Direction »

La Figure qui suit présente le diagramme de séquences système du sous cas d'utilisation « Supprimer Direction ».

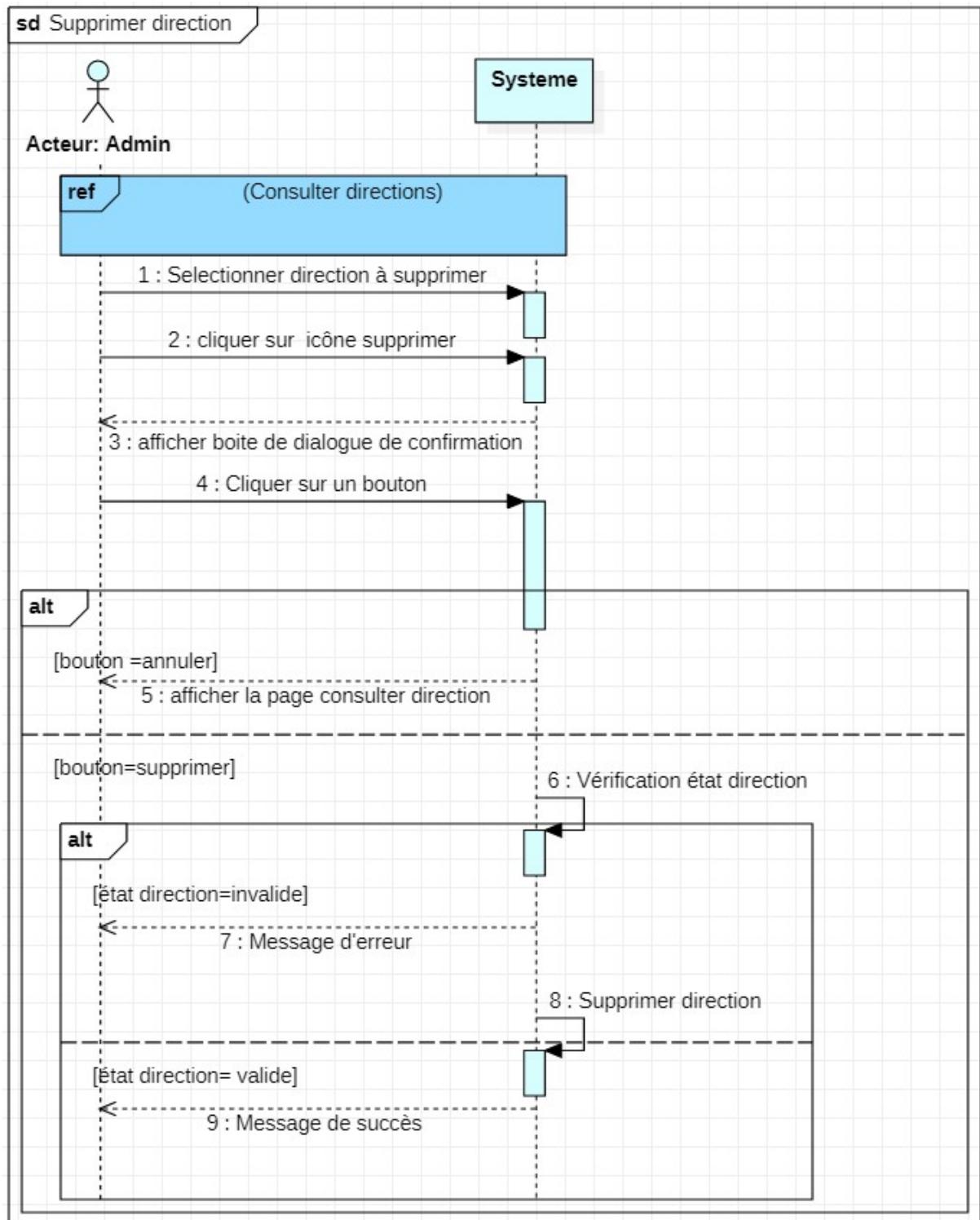


FIGURE 3.8 – Diagramme de séquences système du sous cas d'utilisation « Supprimer Direction ».

3.3 Conception

La conception présente une entrée majeure pour les activités d'implémentation et de test. Elle se traduit, dans mon cas, par les diagrammes d'activité et les diagrammes de séquences détaillés.

3.3.1 Diagrammes de séquences détaillés

Un diagramme de séquences détaillé permet de représenter en détail les interactions entre les objets métiers de notre système selon un ordre chronologique.

3.3.1.1 Cas d'utilisation « S'authentifier »

La Figure suivante présente le diagramme de séquences détaillé du cas d'utilisation « S'authentifier ».

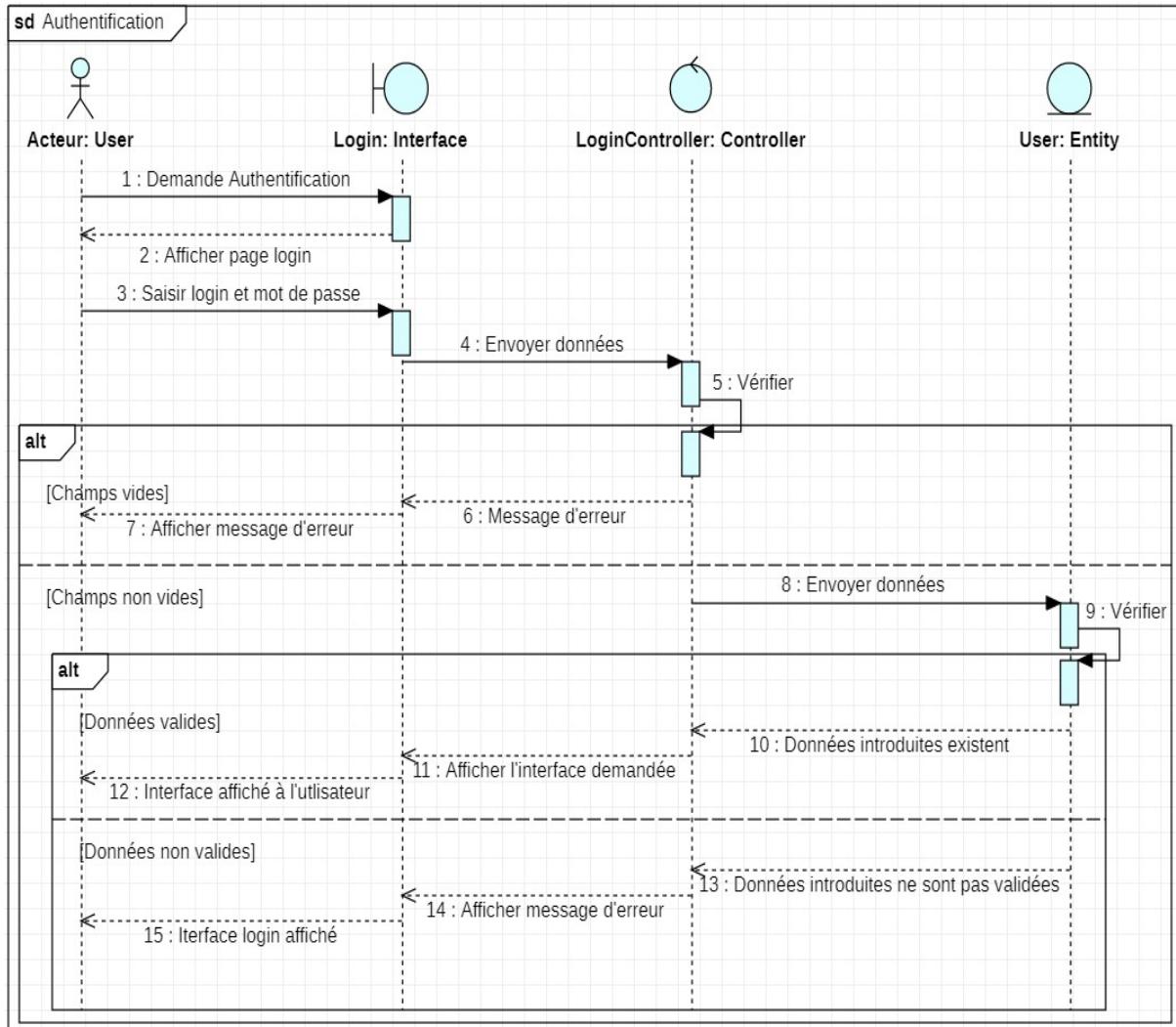


FIGURE 3.9 – Diagramme de séquences détaillé du cas d'utilisation « S'authentifier ».

3.3.1.2 Cas d'utilisation « Ajouter direction »

La Figure suivante présente le diagramme de séquences détaillé du sous-cas d'utilisation « Ajouter direction ».

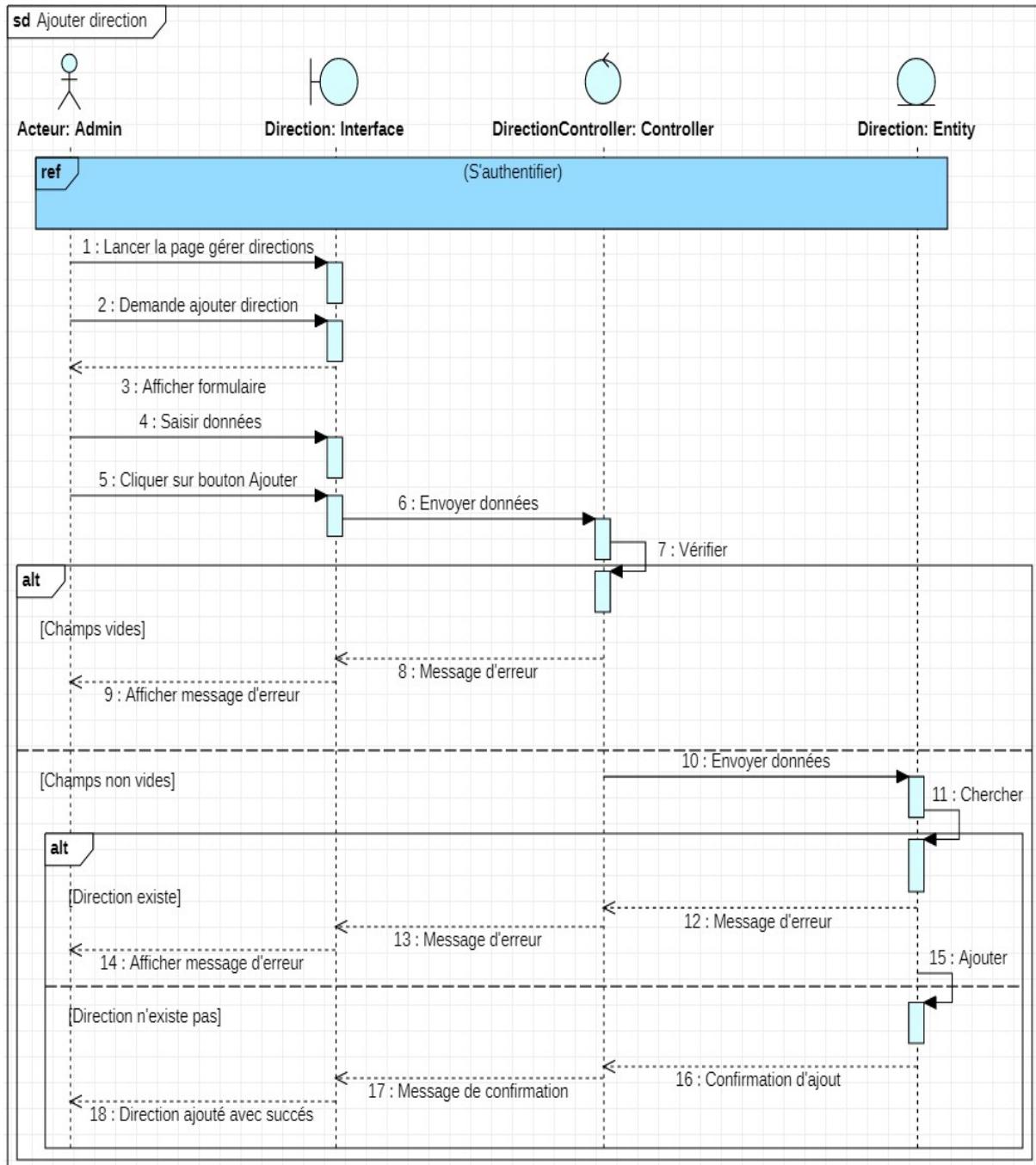


FIGURE 3.10 – Diagramme de séquences détaillé du cas d'utilisation « Ajouter direction ».

3.3.1.3 Cas d'utilisation « Consulter direction »

La Figure ci-après montre le diagramme de séquences détaillé du sous-cas d'utilisation « Consulter direction ».

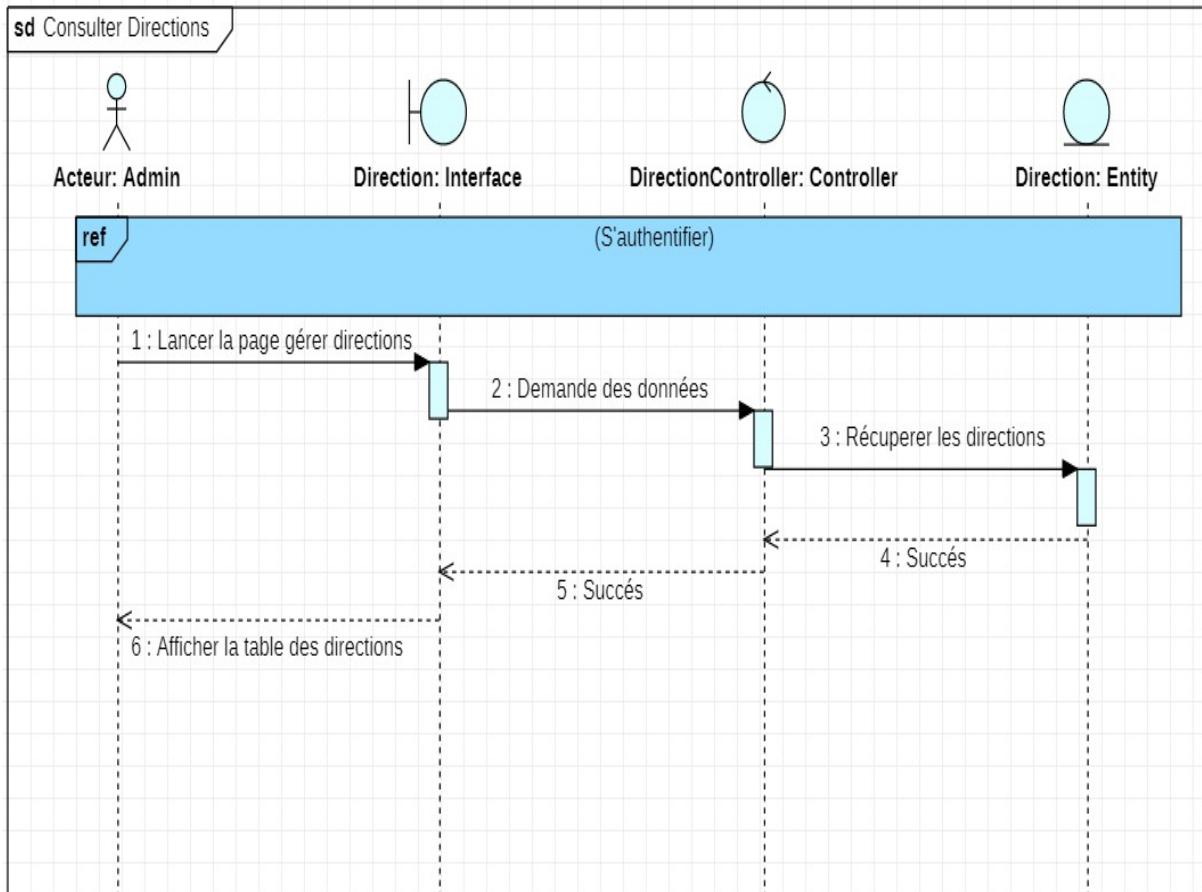


FIGURE 3.11 – Diagramme de séquences détaillé du cas d'utilisation « Consulter Directions ».

3.3.1.4 Cas d'utilisation « Supprimer Direction »

La Figure suivante présente le diagramme de séquences détaillés du sous-cas d'utilisation « Supprimer Direction ».

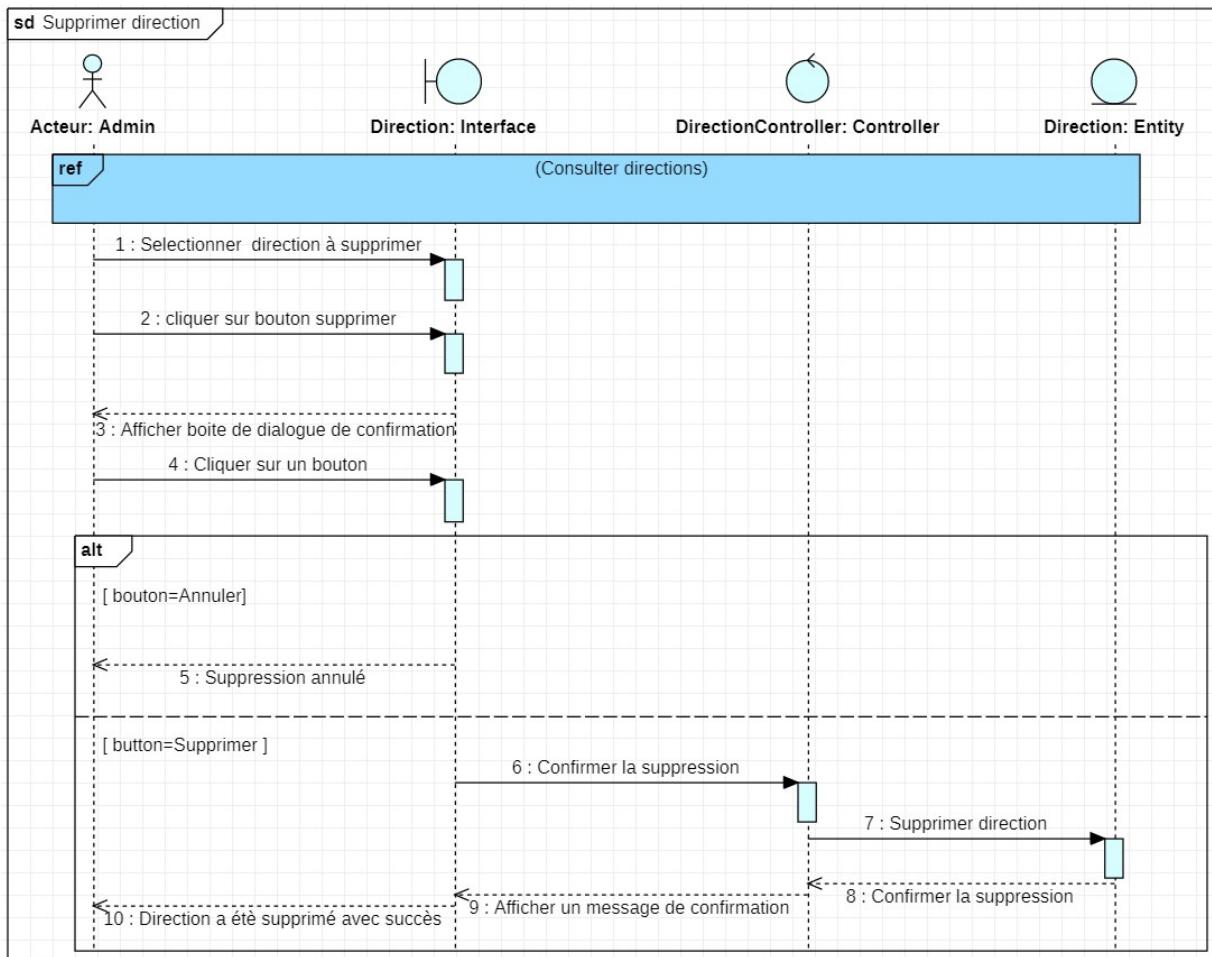


FIGURE 3.12 – Diagramme de séquences détaillé du cas d'utilisation « Supprimer direction ».

3.3.2 Diagrammes d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation [Fou12].

3.3.2.1 Cas d'utilisation « S'authentifier »

La Figure suivante présente le diagramme d'activité du cas d'utilisation « S'authentifier ».

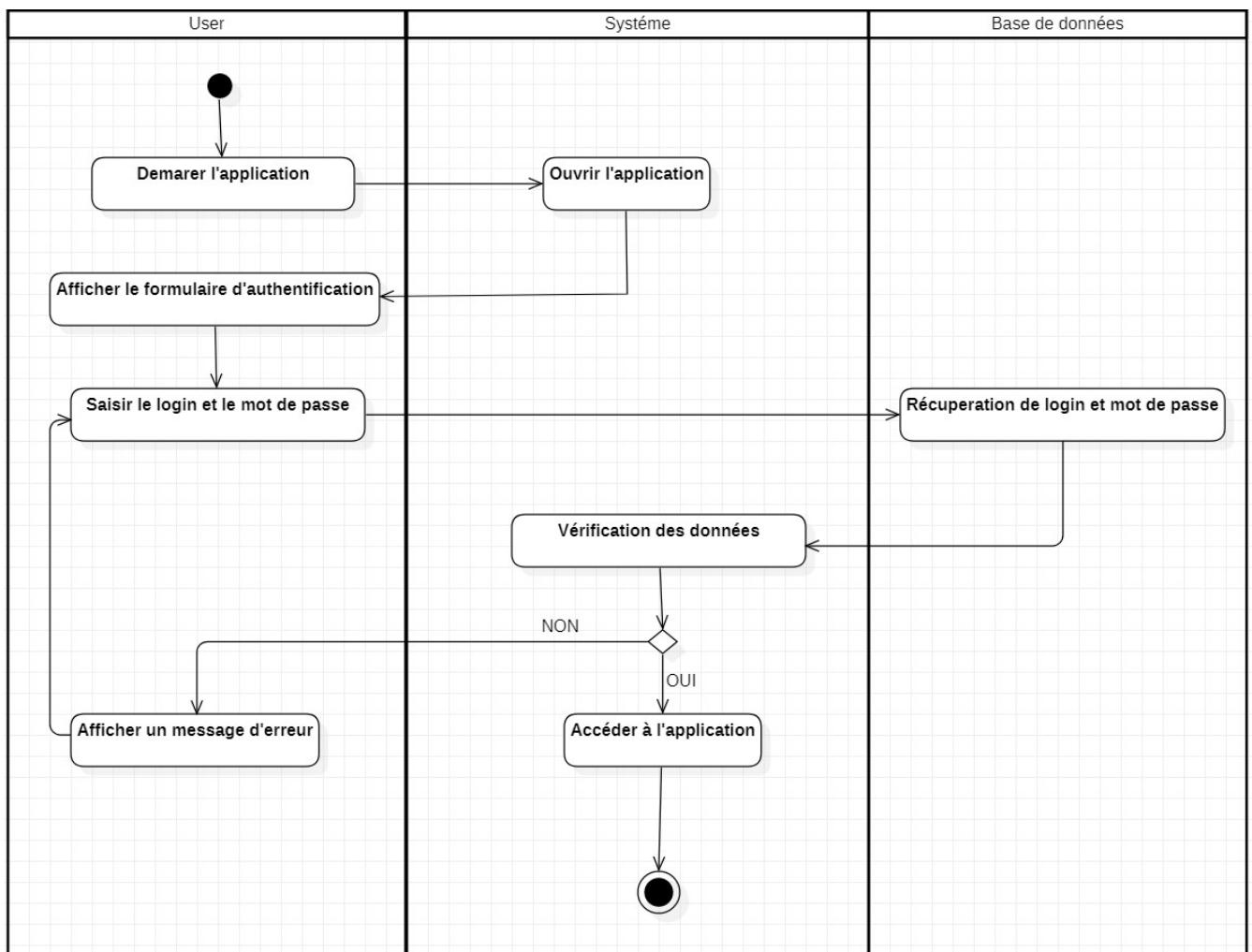


FIGURE 3.13 – Diagramme d'activité du cas d'utilisation « S'authentifier ».

3.3.2.2 Cas d'utilisation « Ajouter direction »

La Figure suivante présente le diagramme d'activité du cas d'utilisation « Ajouter direction ».

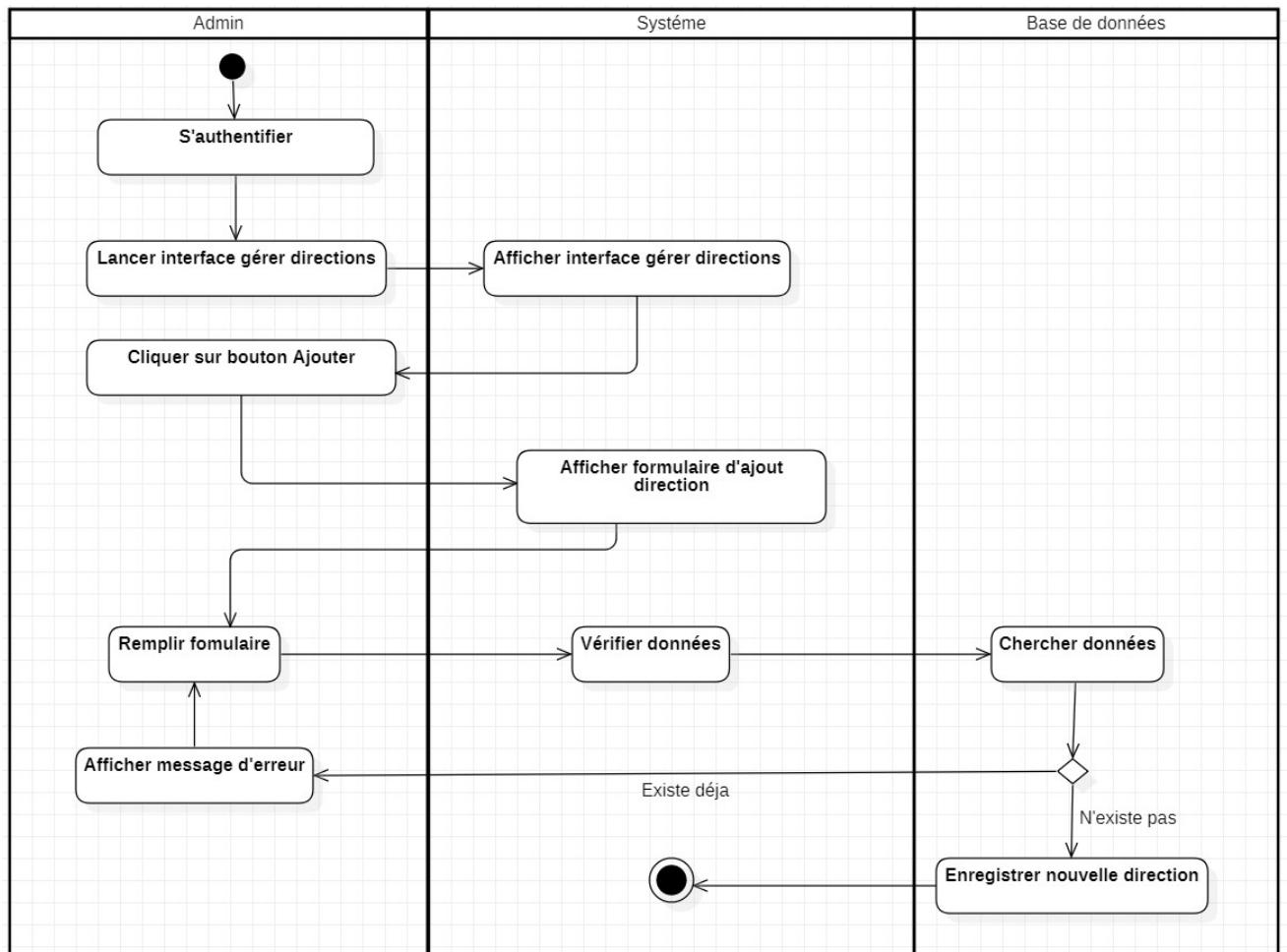


FIGURE 3.14 – Diagramme d'activité du cas d'utilisation « Ajouter direction ».

3.3.2.3 Cas d'utilisation « Supprimer direction »

La Figure suivante présente le diagramme d'activité du cas d'utilisation « Supprimer direction ».

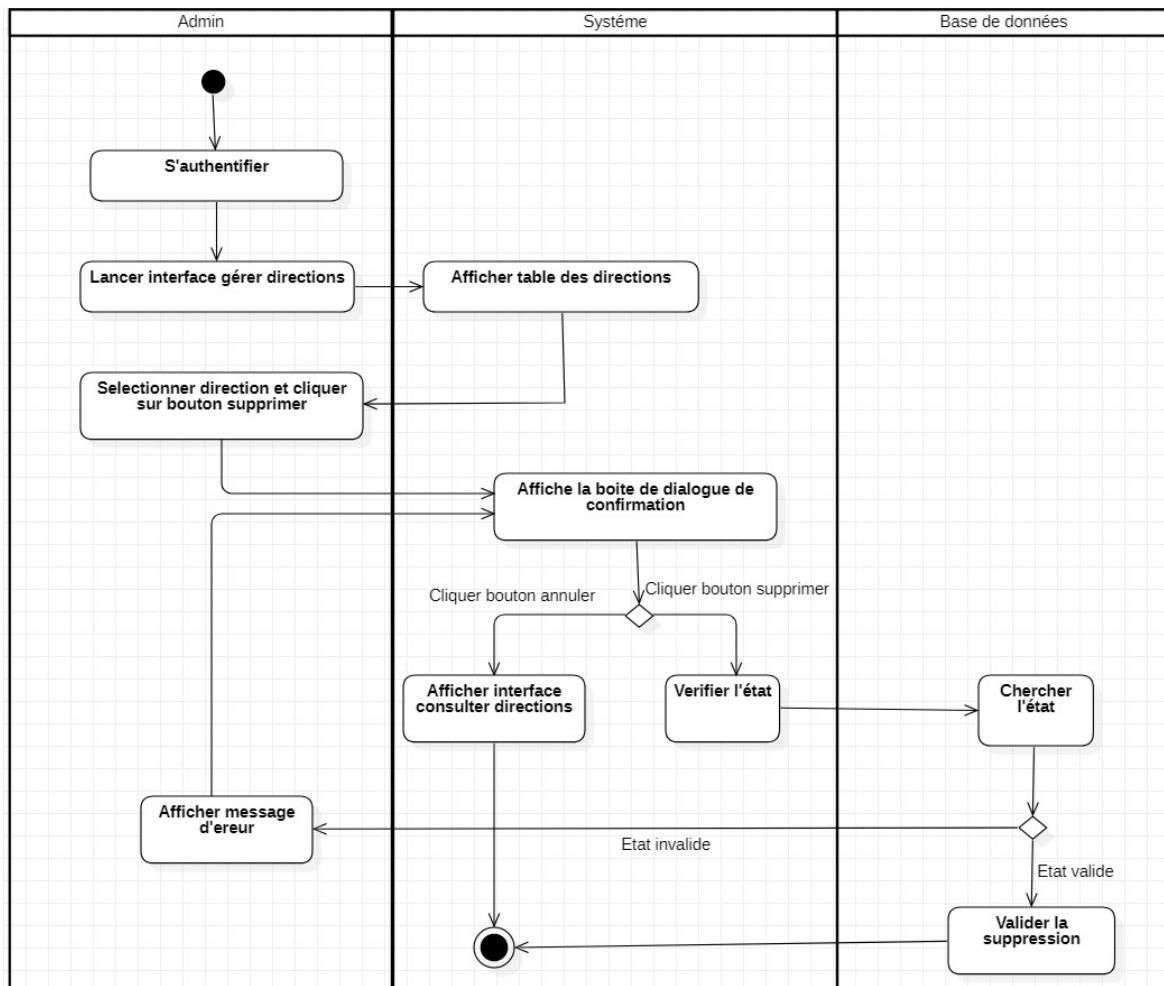


FIGURE 3.15 – Diagramme d'activité du cas d'utilisation « Supprimer direction ».

3.4 Réalisation

- Interface : Authentification

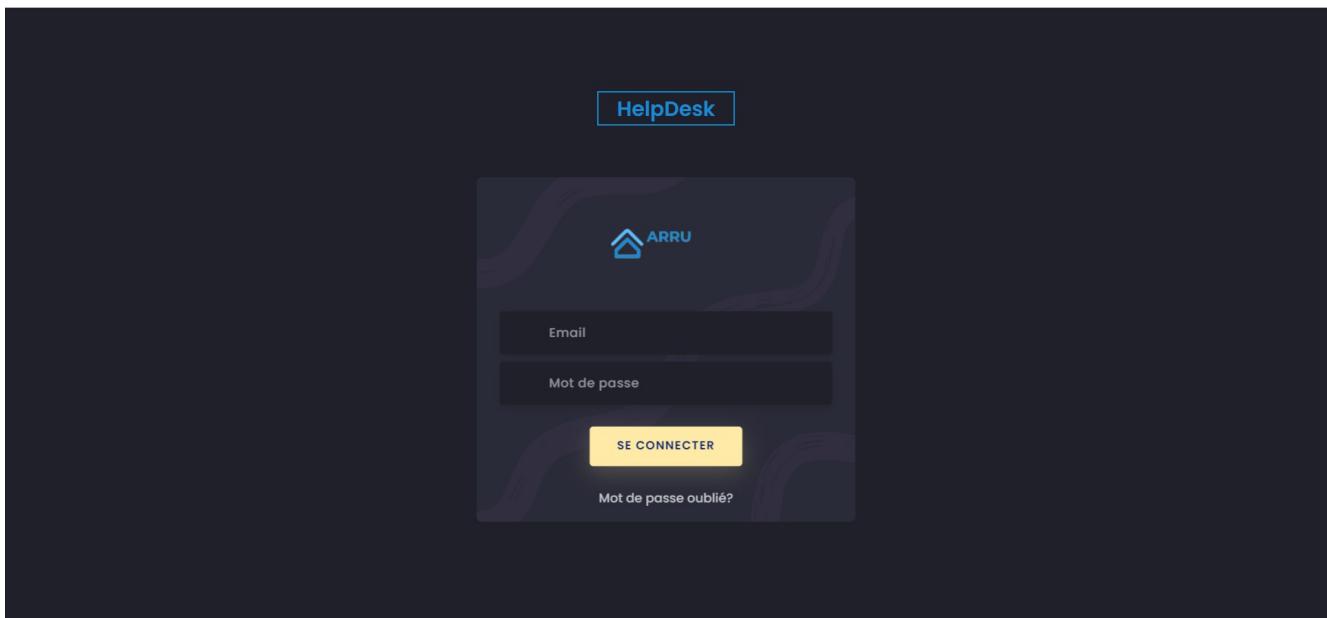


FIGURE 3.16 – Interface : Authentification

- Interface : Consulter directions

A screenshot of the "Espace Admin" section of the application. On the left is a sidebar with icons for Accueil, Personnel, Directions, Types tickets, and Logout. The main area is titled "ESPACE ADMIN" and contains a table titled "Directions". The table has four columns: "NOM DIRECTION" (Direction Name), "FIN" (Status FIN), "ADMIN" (Status ADMIN), and "APV" (Status APV). Each row has a red delete icon at the end. A blue button labeled "+ Ajouter" (Add) is located in the top right corner of the table area.

FIGURE 3.17 – Interface : Consulter directions

- Interface : Ajouter direction

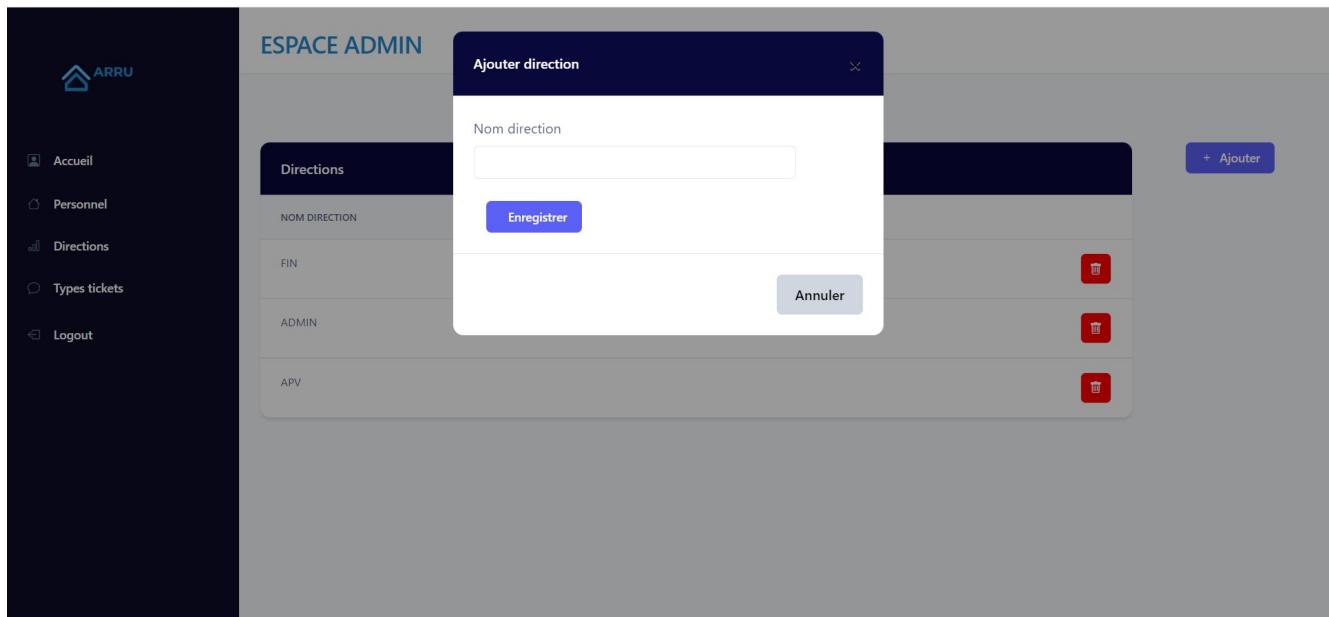


FIGURE 3.18 – Interface : Ajouter direction

- Interface : Supprimer direction

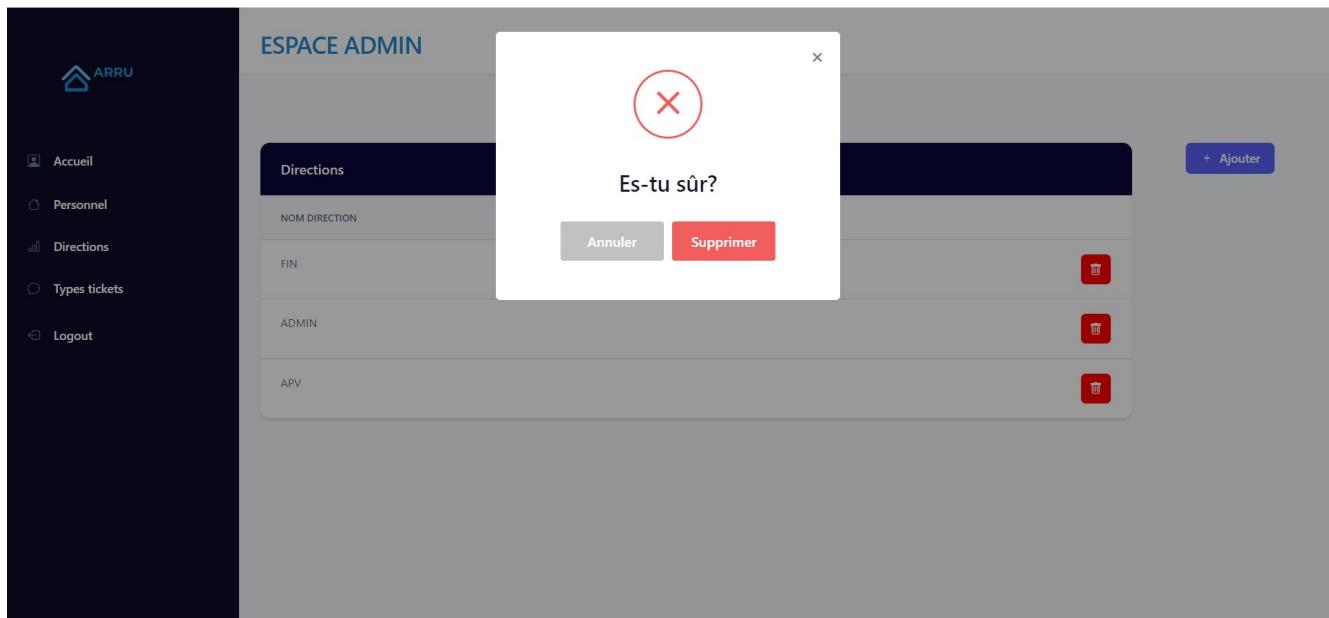


FIGURE 3.19 – Interface : Supprimer direction

- Interface : Consulter types tickets

The screenshot shows the ARRU application's admin interface. On the left is a dark sidebar with icons for Accueil, Personnel, Directions, Types tickets, and Logout. The main area is titled "ESPACE ADMIN". A sub-section titled "Types tickets" displays two rows of ticket types. Each row contains the ticket type name, the assigned staff member's matricule, and edit/delete icons. The first row shows "Fourniture" assigned to "azerty". The second row shows "Intervention informatique" assigned to "6536T". A blue button labeled "+ Ajouter" is located in the top right corner of the "Types tickets" section.

FIGURE 3.20 – Interface : Consulter types tickets

- Interface : Ajouter type ticket

The screenshot shows the ARRU application's admin interface. On the left is a dark sidebar with icons for Accueil, Personnel, Directions, Types tickets, and Logout. The main area is titled "ESPACE ADMIN". A sub-section titled "Ajouter type ticket" contains three input fields: "Type ticket", "Nom directions de chargé ticket", and "Matricule chargé tickets". A blue "Enregistrer" button is located in the bottom right corner of the form.

FIGURE 3.21 – Interface : Ajouter type ticket

3.5 Tests et validation

En informatique, un test désigne une procédure de vérification partielle d'un système. Le tableau suivant présente les tests effectués lors sprint1 en collaboration avec le product owner comme le montre le tableau suivant :

TABLE 3.9 – Table des acteurs.

Fonctionnalité	Résultat
S'authentifier .	Conforme
Gérer Directions.	Conforme
Gérer Types Tickets.	Conforme

TABLE 3.10 – Tests de validation

Conclusion

Tout au long de ce chapitre, j'ai présenté le backlog du sprint 1 ainsi que la conception, l'analyse, et les interfaces des cas d'utilisation **"S'authentifier"**, **"Gérer directions"** et **"Gérer types de tickets"**. Dans le prochain chapitre, je vais présenter le deuxième sprint.

Chapitre 4

Sprint 2

Sommaire

Introduction	56
4.1 Spécification des besoins	56
4.1.1 Diagramme de cas d'utilisation du sprint 2	56
4.1.2 Backlog du sprint 2	56
4.2 Analyse	58
4.2.1 Raffinement des cas d'utilisation	58
4.2.2 Diagrammes de séquences système	64
4.3 Conception	72
4.3.1 Diagrammes de séquences détaillés	73
4.3.2 Diagrammes d'activités	78
4.4 Réalisation	80
4.5 Tests et validation	83
Conclusion	83

Introduction

Durant ce chapitre, j'entame le deuxième sprint. Pour cela, je vais commencer par le diagramme de cas d'utilisation raffiné du sprint, suivi par le backlog du sprint et je terminerai par l'analyse et la conception.

Rappelons que ce premier sprint contient es fonctionnalités suivantes :

- Gérer personnel
- Gérer ticktes

4.1 Spécification des besoins

Dans cette partie, je présente le diagramme de cas d'utilisation du sprint suivi par le backlog du sprint.

4.1.1 Diagramme de cas d'utilisation du sprint 2

La Figure 4.1 présente le diagramme de cas d'utilisation du sprint 2.

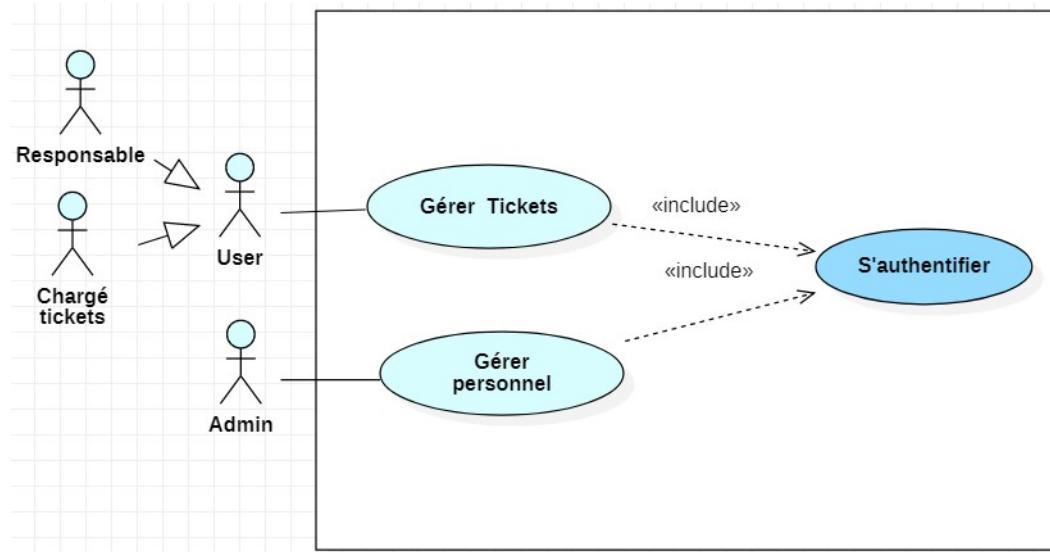


FIGURE 4.1 – Diagramme de cas d'utilisation du sprint 2.

4.1.2 Backlog du sprint 2

En partant du même principe que le sprint précédent, je commence par définir le but du deuxième sprint. La table 4.1 détaille le backlog du sprint 2.

TABLE 4.1 – Le Backlog du sprint 2

ID	User Stories	ID	Type	Tache	Complexité
1	En tant qu'administrateur je peux consulter, ajouter modifier et supprimer une personne.	1.1	Back-end	Créer l'entité 'utilisateur' dans la base de données.	1
		1.2	Back-End	Créer un web service 'utilisateurAPI' pour gérer les utilisateurs.	1
		1.3	Front-End	Créer un service web front.	2
		1.4	Front-End	Créer l'interface CRUD "utilisateur".	2
		-	Test	Test et validation.	-
		-	Conception	Conception des diagrammes.	-
2	En tant qu'utilisateur je peux consulter, ajouter un ticket.	2.1	Back-end	Créer l'entité 'ticket' dans la base de données.	1
		2.2	Back-End	Créer un web service 'ticketAPI' pour gérer les tickets.	1
		2.3	Front-End	Créer un service web front.	2
		2.4	Front-End	Créer l'interface gérer "ticket".	2
		-	Test	Test et validation.	-
		-	Conception	Conception des diagrammes.	-

4.2 Analyse

Afin de mieux assimiler les cas d'utilisation, j'établis leurs raffinements pour livrer une description sur les différents scénarios possibles.

4.2.1 Raffinement des cas d'utilisation

Dans cette sous-section, je présente les raffinements des diagrammes de cas d'utilisation liés au deuxième sprint, ainsi qu'une description textuelle des principaux cas d'utilisation.

4.2.1.1 Cas d'utilisation « Gérer personnel »

La Figure 4.2 représente le diagramme de cas d'utilisation de « Gérer personnel ».

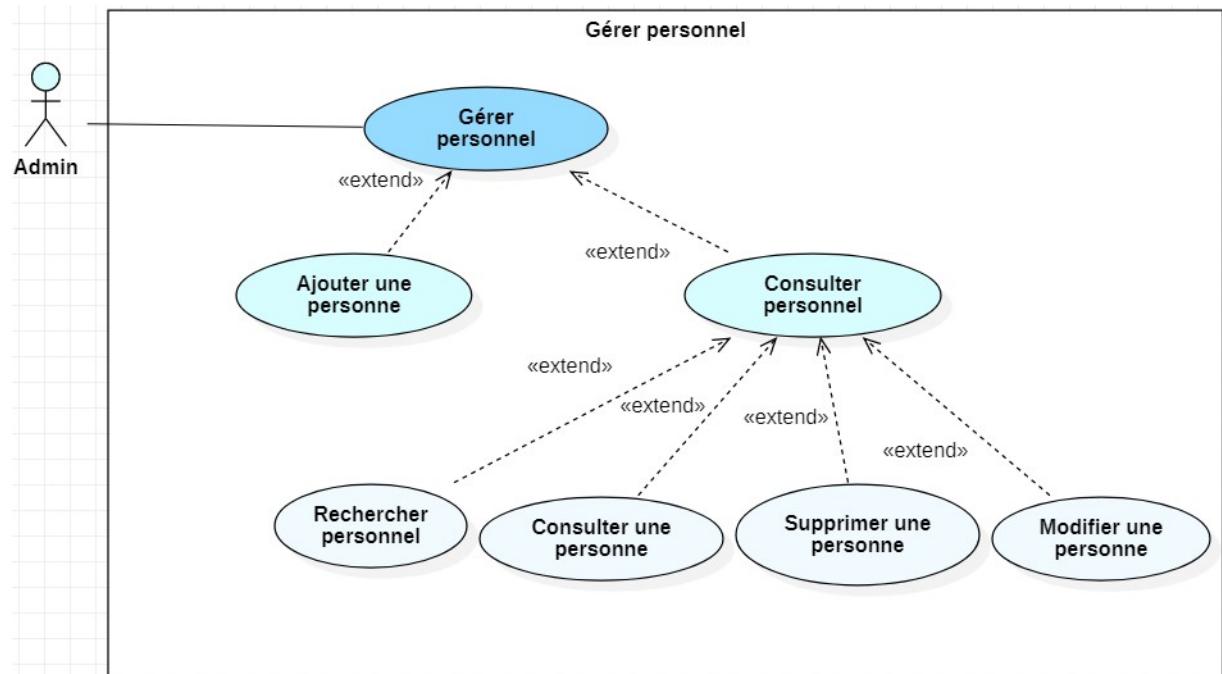


FIGURE 4.2 – Diagramme de cas d'utilisation « Gérer personnel ».

TABLE 4.2 – Description textuelle du sous cas d'utilisation « Ajouter une peronne »

Cas d'utilisation	Ajouter une personne
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur ouvre l'interface gerer personnel. 2. L'administrateur clique sur bouton «Ajouter». 3. Le système affiche le formulaire d'ajout d'une personne. 4. L'administrateur remplit le formulaire d'ajout d'une personne et clique sur le bouton «Enregittrer». 5. Le système ajoute une personne à la liste personnel.
Post-condition	Une personne ajouté.
Exception	<ul style="list-style-type: none"> *Les informations manquantes *Affichage d'un message d'erreur "adresse e-mail est déjà utilisée " . *Affichage d'un message d'erreur "matricul est déjà utilisée " .

TABLE 4.3 – Description textuelle du sous cas d'utilisation « Consulter personnel »

Cas d'utilisation	Consulter personnel
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur ouvre l'interface gerer personnel. 2. La table personnel s'affiche.
Post-condition	La liste personnel est affichée.
Exception	Problème de connexion à la base de données.

TABLE 4.4 – Description textuelle du sous-cas d'utilisation « Supprimer une personne »

Cas d'utilisation	Supprimer une personne
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. Afficher la liste du personnel. 2. L'administrateur choisit la personne à supprimer et clique sur le bouton supprimer. 3. Le système affiche un message de confirmation 4. L'administrateur clique sur le bouton «Confirmer». 5. Le système affiche un message de succès de suppression d'une personne. 6. Le système affiche la nouvelle liste du personnel.
Post-condition	Une personne supprimé.
Exception	<ul style="list-style-type: none"> *Affichage d'un message d'erreur "impossible de supprimer un administrateur". *Affichage d'un message d'erreur "impossible de supprimer un responsable". *Affichage d'un message d'erreur "impossible de supprimer un chargé tickets". *Affichage d'un message d'erreur "impossible de supprimer une personne qui possède un ticket en cours". *Problème de connexion à la base de données.

TABLE 4.5 – Description textuelle du sous-cas d'utilisation « Consulter une personne »

Cas d'utilisation	Consulter une personne
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. Afficher la liste personnel. 2. L'administrateur choisit une personne à consulter et clique sur le bouton consulter. 3. Le système affiche toutes les informations relatives à la personne.
Post-condition	Detail personne affiché.
Exception	Problème de connexion à la base de données.

TABLE 4.6 – Description textuelle du sous-cas d'utilisation « Modifier une personne »

Cas d'utilisation	Modifier une personne
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur clique sur bouton «Modifier». 2. Le système affiche le formulaire de modification d'une personne. 3. L'administrateur remplit le formulaire de modification d'une personne et clique sur le bouton «Enregitsrer». 4. Le système met à jour les informations de la personne modifiée à la liste du personnel.
Post-condition	Personnel modifié.
Exception	<p>*Affichage d'un message d'erreur "impossible de modifier cette personne".</p> <p>Problème de connexion à la base de données.</p>

TABLE 4.7 – Description textuelle du sous-cas d'utilisation « Rechercher personnel »

Cas d'utilisation	Rechercher personnel
Acteurs	Administrateur
Pré-condition	Administrateur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur saisit l'élément de recherche. 2. Le système affiche la liste des utilisateurs vérifiant le critère de recherche.
Post-condition	Liste personnel affichée.
Exception	<p>L'administrateur saisit une valeur inexistante.</p> <p>Le système affiche un message d'erreur</p>

4.2.1.2 Gérer tickets

La figure 4.3 représente le diagramme de cas d'utilisation de "Gérer tickets".

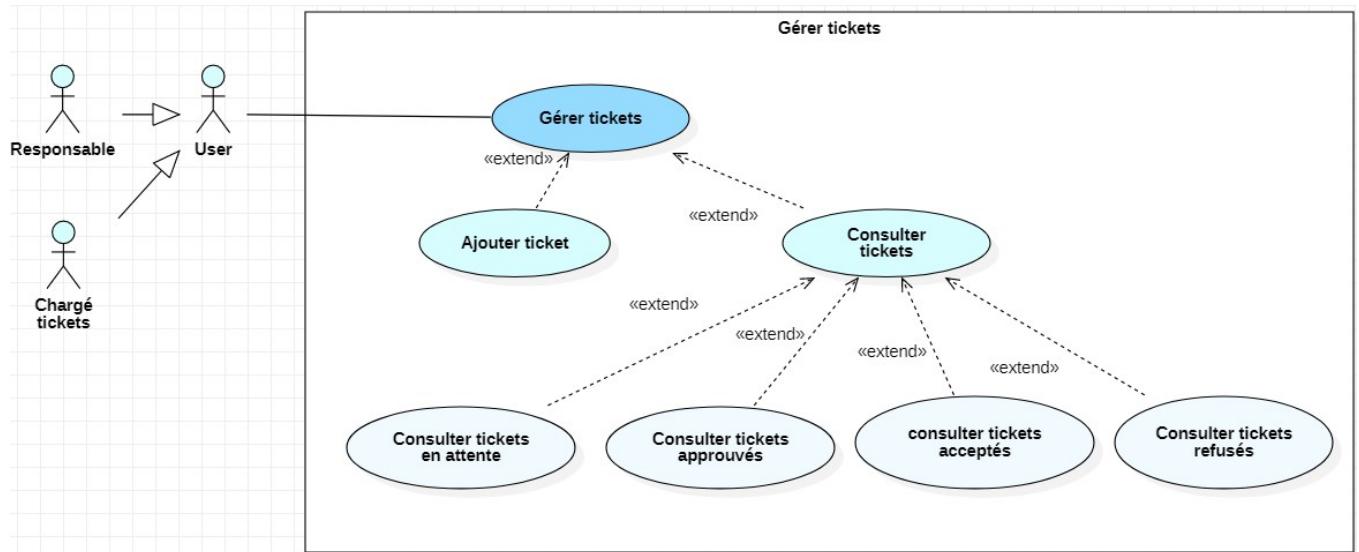


FIGURE 4.3 – Diagramme de cas d'utilisation "Gérer tickets".

TABLE 4.8 – Description textuelle du sous cas d'utilisation « Ajouter ticket »

Cas d'utilisation	Ajouter ticket
Acteurs	User/Responsable/Chargé ticket
Pré-condition	Acteur authentifié
Scénario Nominal	1. L'acteur clique sur bouton «Ajouter». 2. Le système affiche le formulaire d'ajout d'un ticket. 3. L'acteur remplit le formulaire d'ajout d'un ticket et clique sur le bouton «Enregistrer». 4. Le système ajoute un ticket à la liste des tickets.
Post-condition	Un ticket ajouté.
Exception	Les informations manquantes

TABLE 4.9 – Description textuelle du sous cas d'utilisation « Consulter tickets »

Cas d'utilisation	Consulter tickets
Acteurs	User/Responsable/Chargé ticket
Pré-condition	Acteur authentifié
Scénario Nominal	1. L'acteur ouvre l'interface gérer tickets. 2. L'interface gérer tickets.
Post-condition	L'interface gérer tickets est affichée.
Exception	Problème de connexion à la base de données.

TABLE 4.10 – Description textuelle du sous cas d'utilisation « Consulter tickets en attentes »

Cas d'utilisation	Consulter tickets en attente
Acteurs	User/Responsable/Chargé ticket
Pré-condition	Acteur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'acteur ouvre l'interface gérer tickets. 2. L'interface gérer tickets s'affiche. 3. L'acteur clique sur le button tickets en attente 4. Les tickets en attente s'affichent.
Post-condition	La liste des tickets en attente est affichée.
Exception	Problème de connexion à la base de données.

TABLE 4.11 – Description textuelle du sous cas d'utilisation « Consulter tickets approuvés »

Cas d'utilisation	Consulter tickets approuvés
Acteurs	User/Responsable/Chargé tickets
Pré-condition	Acteur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'acteur ouvre l'interface gérer tickets. 2. L'interface gérer tickets s'affiche. 3. L'acteur clique sur le button tickets approuvés 4. La liste des tickets approuvés s'affiche.
Post-condition	La liste des tickets approuvés est affichée.
Exception	Problème de connexion à la base de données.

TABLE 4.12 – Description textuelle du sous cas d'utilisation « Consulter tickets acceptés »

Cas d'utilisation	Consulter tickets acceptés
Acteurs	User/Responsable/Chargé ticket
Pré-condition	Acteur ticket authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'acteur ouvre l'interface gérer tickets. 2. L'interface gérer tickets s'affiche. 3. L'acteur clique sur le button tickets acceptés 4. La liste des tickets acceptés s'affiche.
Post-condition	La liste des tickets acceptés est affichée.
Exception	Problème de connexion à la base de données.

TABLE 4.13 – Description textuelle du sous cas d'utilisation « Consulter tickets refusés »

Cas d'utilisation	Consulter tickets refusés
Acteurs	User/Responsable/Chargé ticket
Pré-condition	Acteur authentifié
Scénario Nominal	<ol style="list-style-type: none">1. L'acteur ouvre l'interface gérer tickets.2. L'interface gérer tickets s'affiche.3. L'acteur clique sur le button tickets refusés4. La liste des tickets refusés s'affiche.
Post-condition	La liste des tickets refusés est affichée.
Exception	Problème de connexion à la base de données.

4.2.2 Diagrammes de séquences système

Le diagramme de séquences modélise l'aspect dynamique du système, il s'agit d'une séquence d'interactions d'un point de vue temporel entre le système et les acteurs. Son rôle était de décrire graphiquement un scénario d'un cas d'utilisation [DV16].

4.2.2.1 Cas d'utilisation « Ajouter une personne »

Dans ce qui suit, le diagramme de séquences système du sous-cas d'utilisation « Ajouter une personne ».

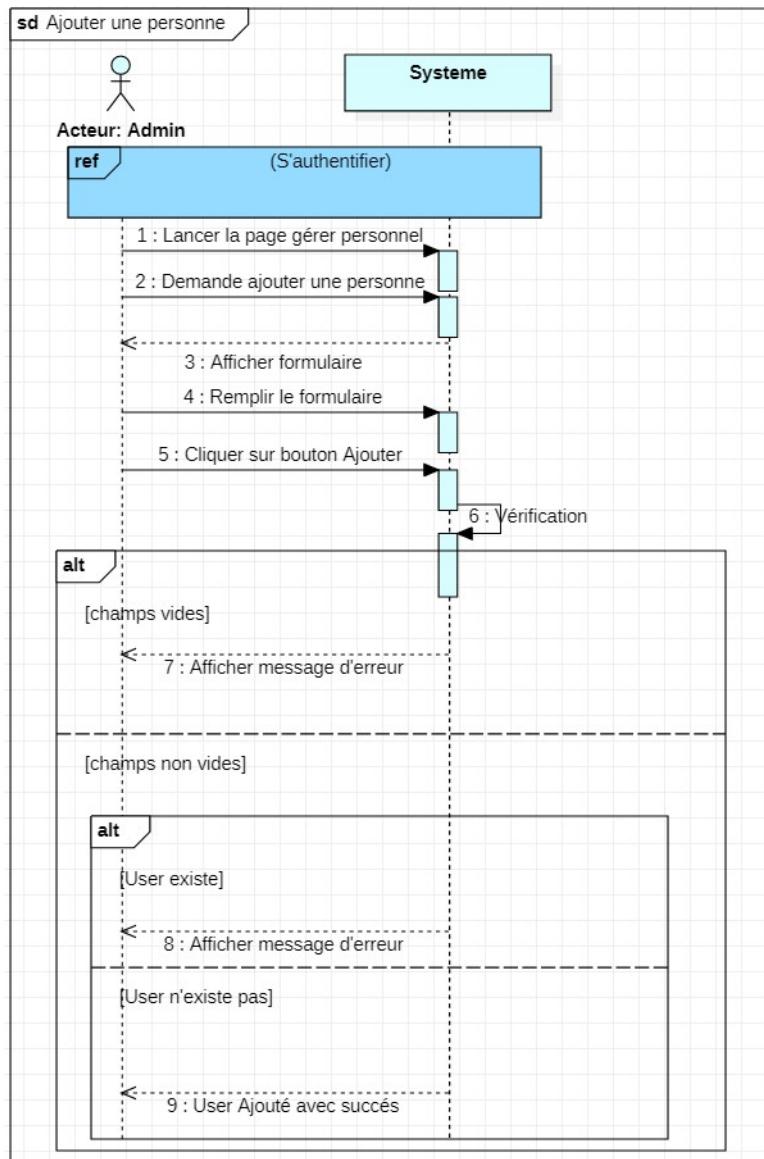


FIGURE 4.4 – Diagramme de séquences système du sous-cas d'utilisation « Ajouter une personne »

4.2.2.2 Cas d'utilisation « Consulter personnel »

Dans ce qui suit le diagramme de séquences système du sous-cas d'utilisation « Consulter personnel ».

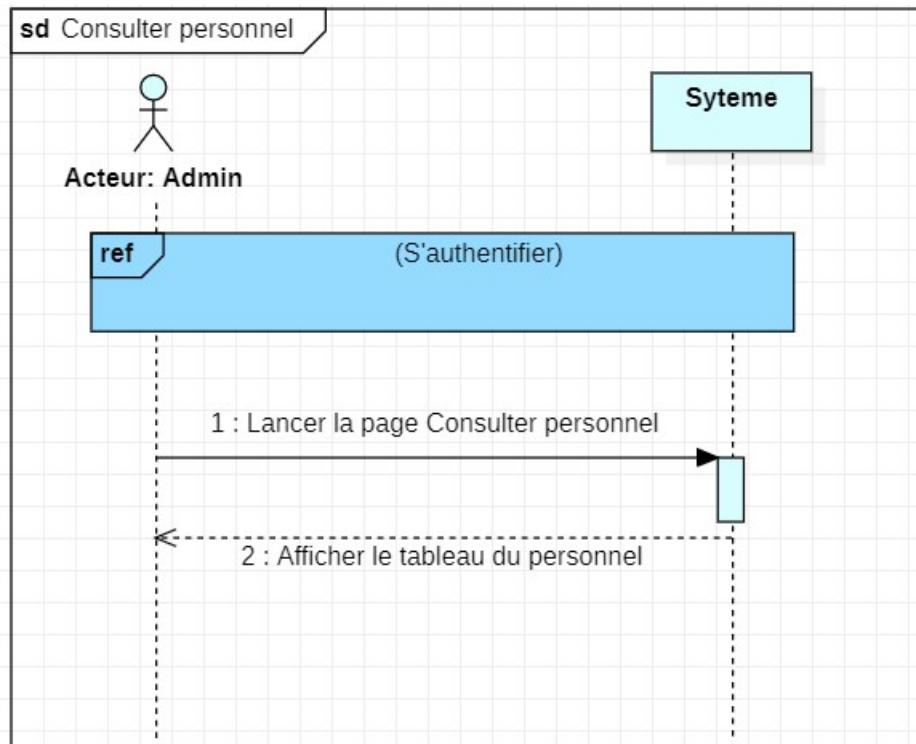


FIGURE 4.5 – Diagramme de séquences système du sous-cas d'utilisation « Consulter personnel »

4.2.2.3 Cas d'utilisation « Supprimer une personne »

Dans ce qui suit le diagramme de séquences système du sous-cas d'utilisation « Supprimer une personne ».

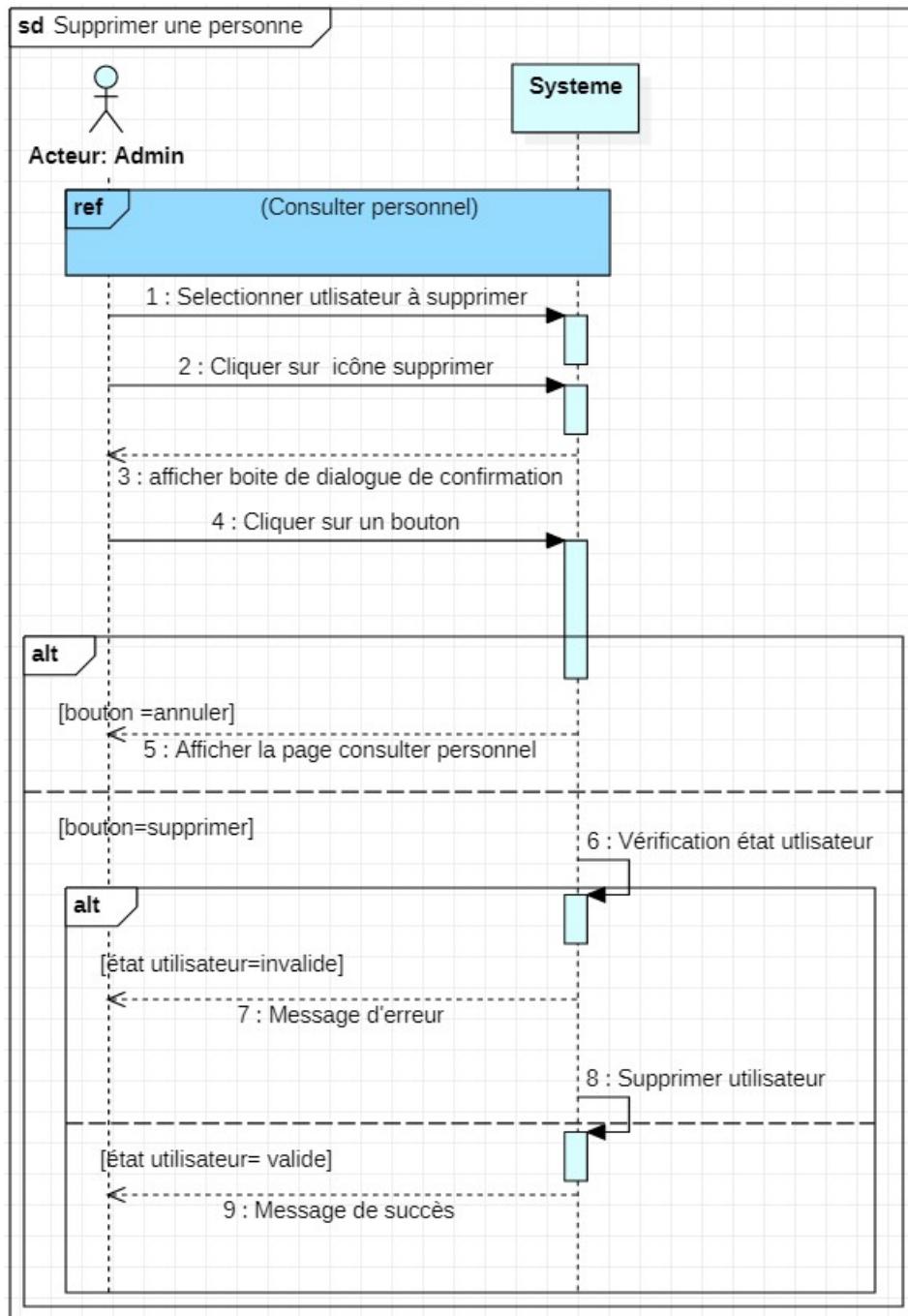


FIGURE 4.6 – Diagramme de séquences système du sous-cas d'utilisation « Supprimer une personne »

4.2.2.4 Cas d'utilisation « Consulter une personne »

Dans ce qui suit le diagramme de séquences système du sous-cas d'utilisation « Consulter une personne ».

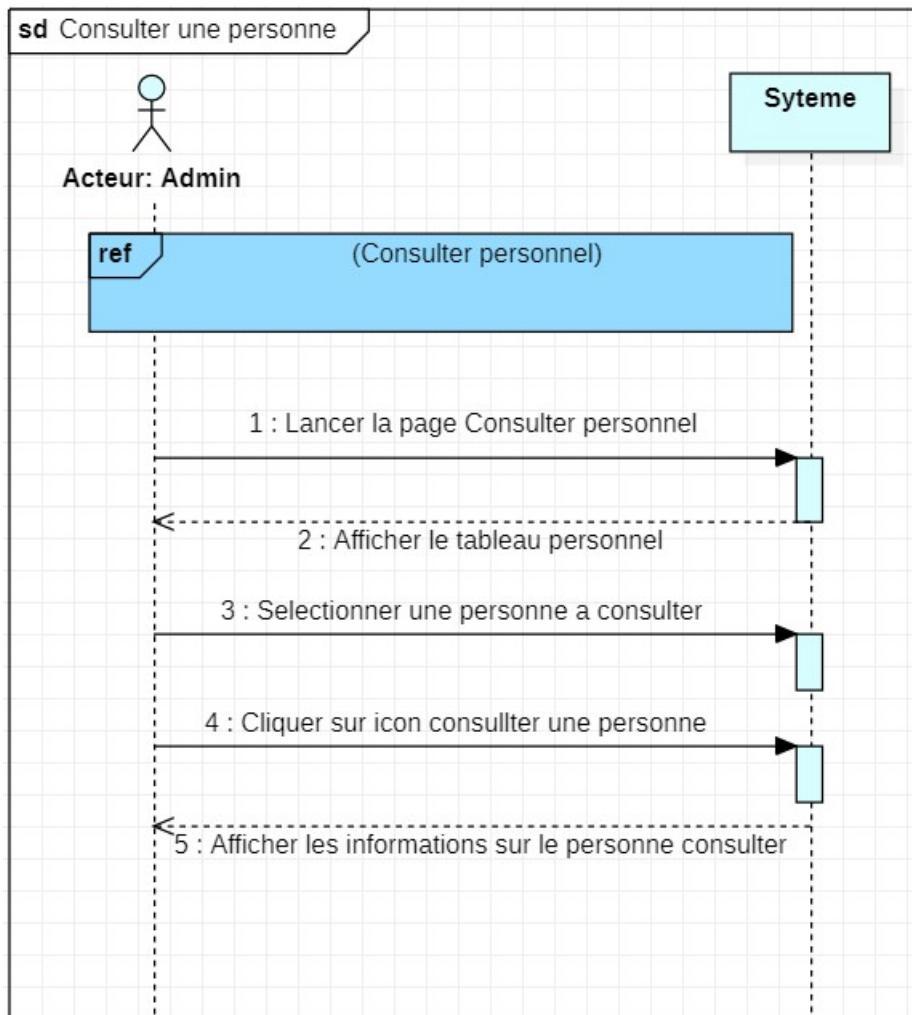


FIGURE 4.7 – Diagramme de séquences système du sous-cas d'utilisation « Consulter une personne »

4.2.2.5 Cas d'utilisation « Modifier une personne »

Dans ce qui suit le diagramme de séquences système du sous-cas d'utilisation « Modifier une personne ».

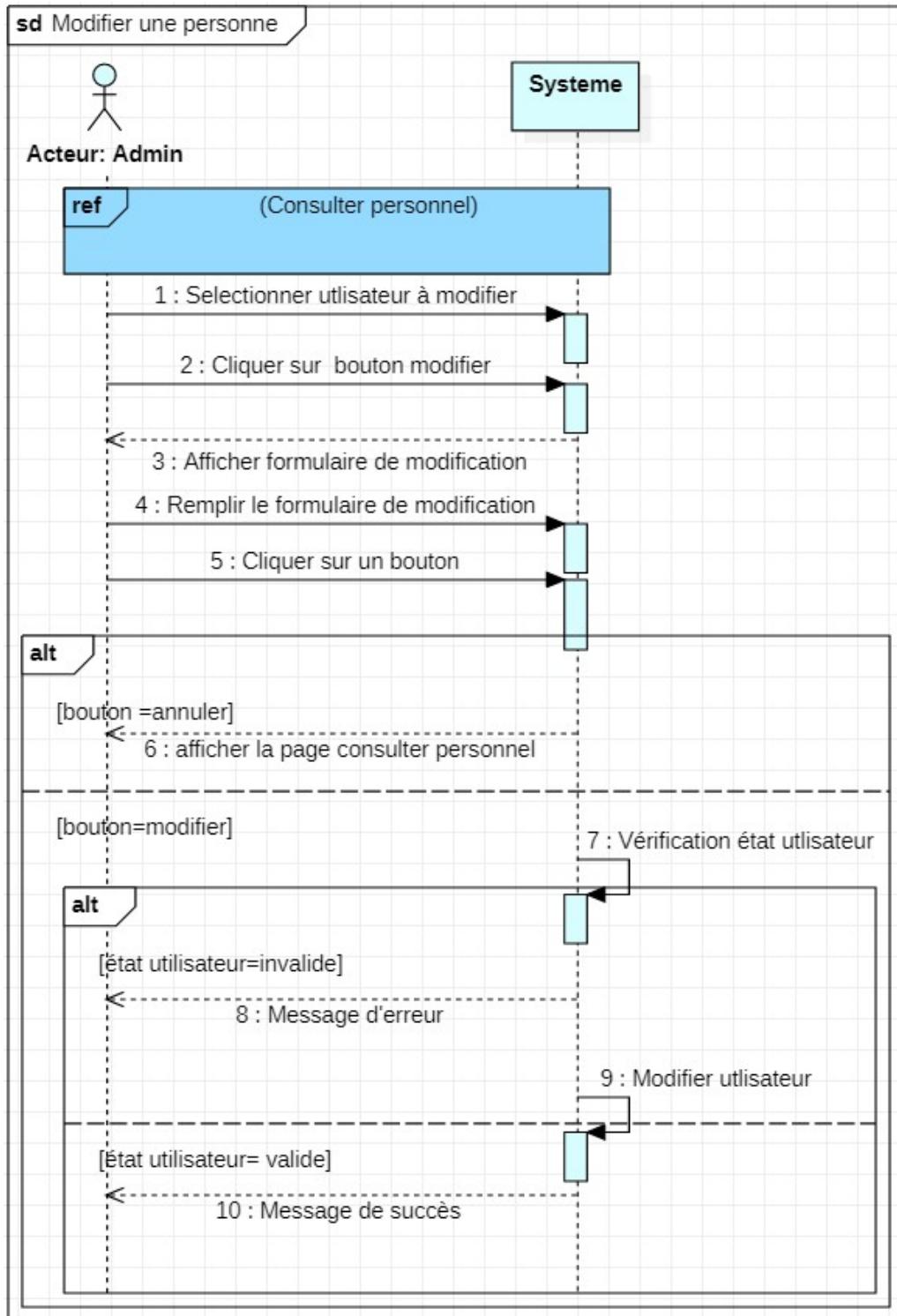


FIGURE 4.8 – Diagramme de séquences système du sous-cas d'utilisation « Modifier une personne »

4.2.2.6 Cas d'utilisation « Rechercher personnel »

Dans ce qui suit le diagramme de séquences système du sous-cas d'utilisation « Rechercher personnel ».

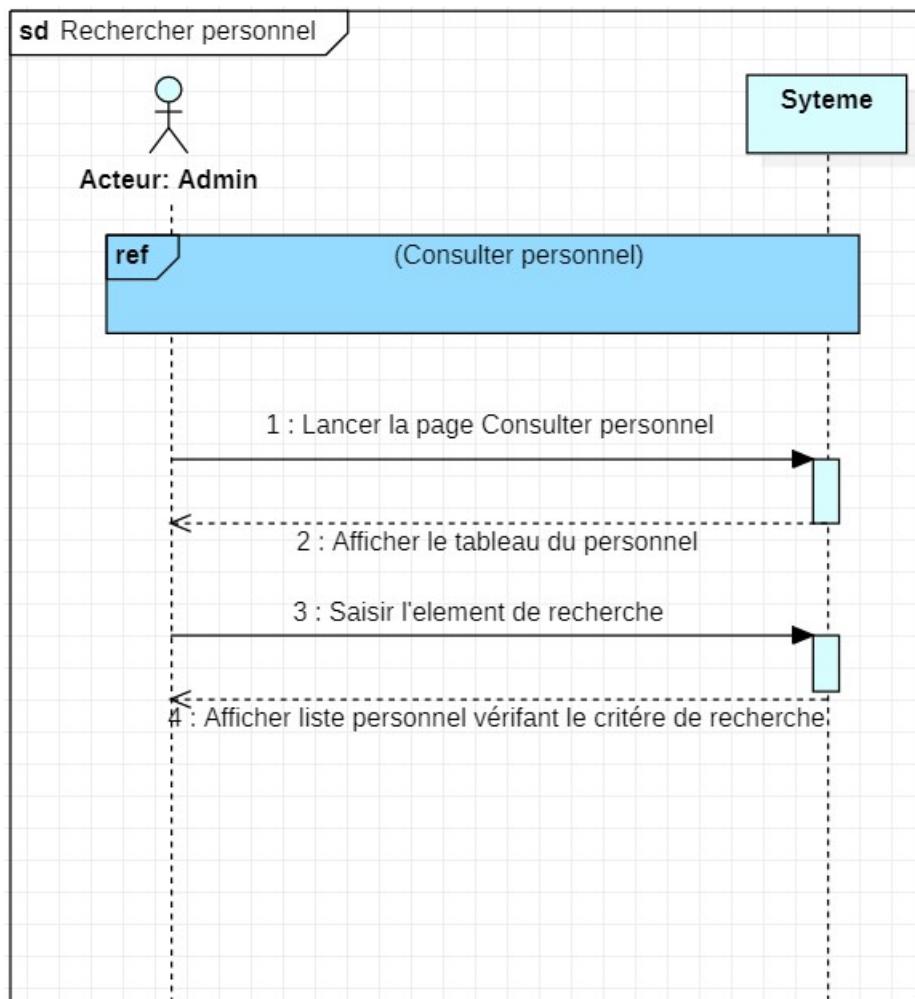


FIGURE 4.9 – Diagramme de séquences système du sous-cas d'utilisation « Rechercher personnel »

4.2.2.7 Cas d'utilisation « Consulter tickets »

Dans ce qui suit le diagramme de séquences système du sous-cas d'utilisation « Consulter tickets ».

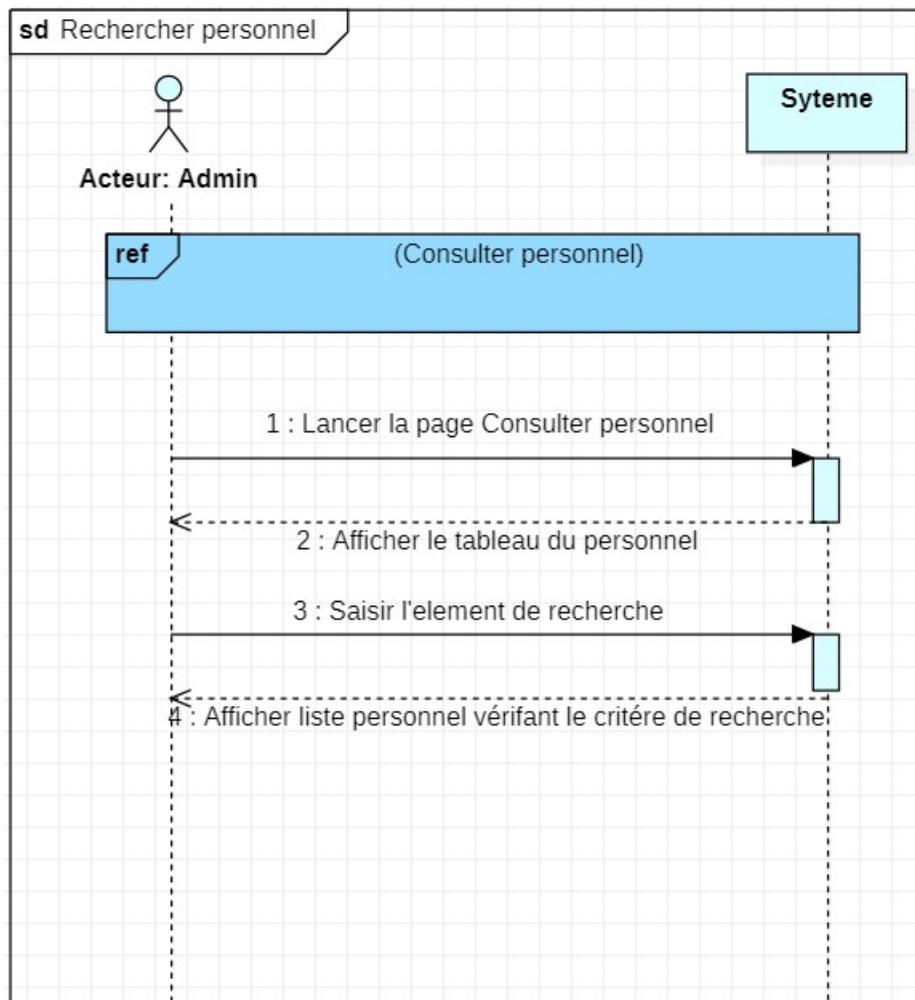


FIGURE 4.10 – Diagramme de séquences système du sous-cas d'utilisation « Consulter tickets »

4.2.2.8 Cas d'utilisation « Consulter tickets approuvés »

Dans ce qui suit le diagramme de séquences système du sous-cas d'utilisation « Consulter tickets approuvés ».

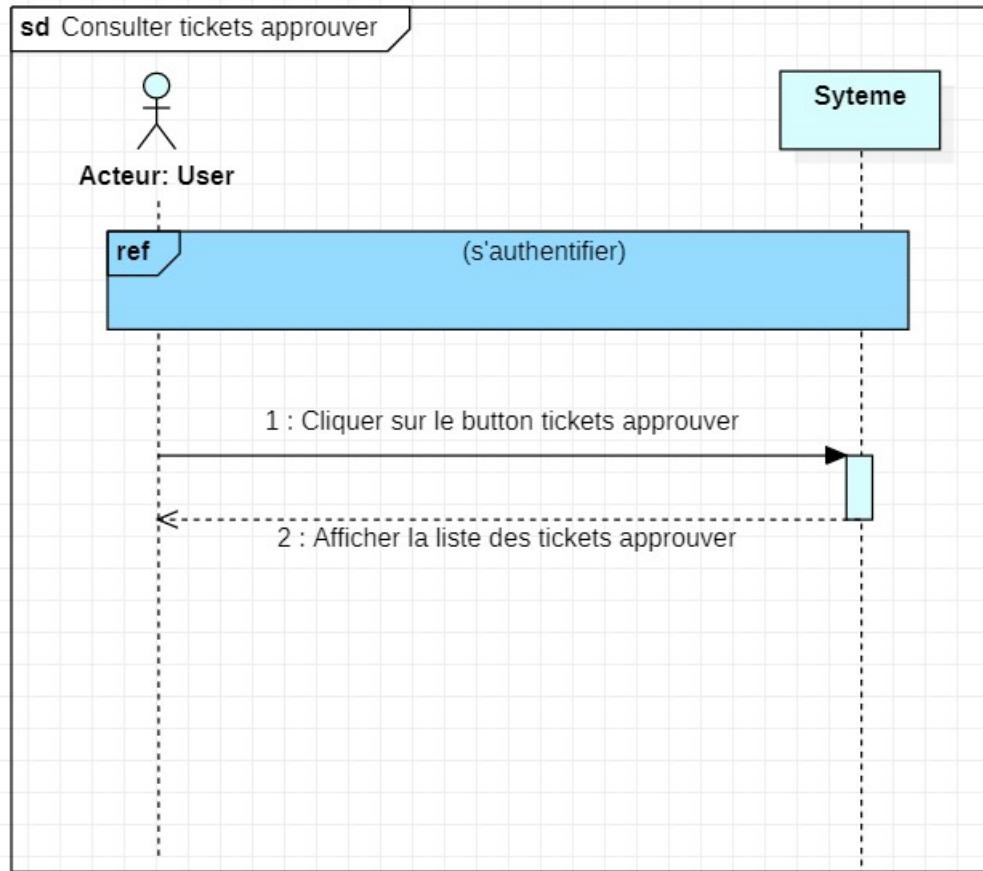


FIGURE 4.11 – Diagramme de séquences système du sous-cas d'utilisation « Consulter tickets approuvés »

4.3 Conception

La conception présente une entrée majeure pour les activités d'implémentation et de test. Elle se traduit, dans mon cas, par les diagrammes d'activités et les diagramme de séquences détaillés.

4.3.1 Diagrammes de séquences détaillés

4.3.1.1 Cas d'utilisation « Ajouter une personne »

Dans ce qui suit le diagramme de séquences détaillés du sous cas d'utilisation « Ajouter une personne ».

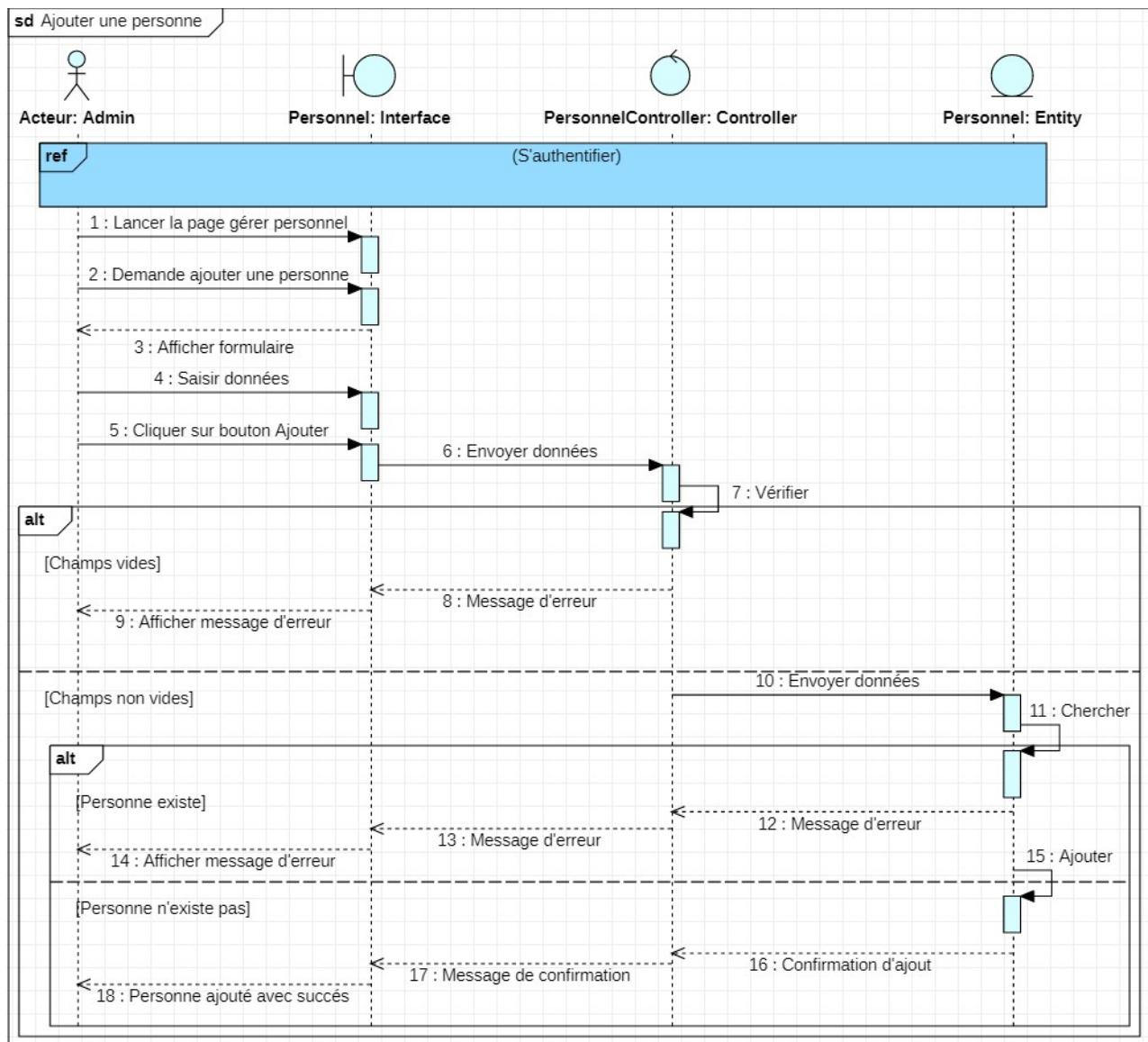


FIGURE 4.12 – Diagramme de séquences détaillé du sous-cas d'utilisation « Ajouter une personne »

4.3.1.2 Cas d'utilisation « Consulter personnel »

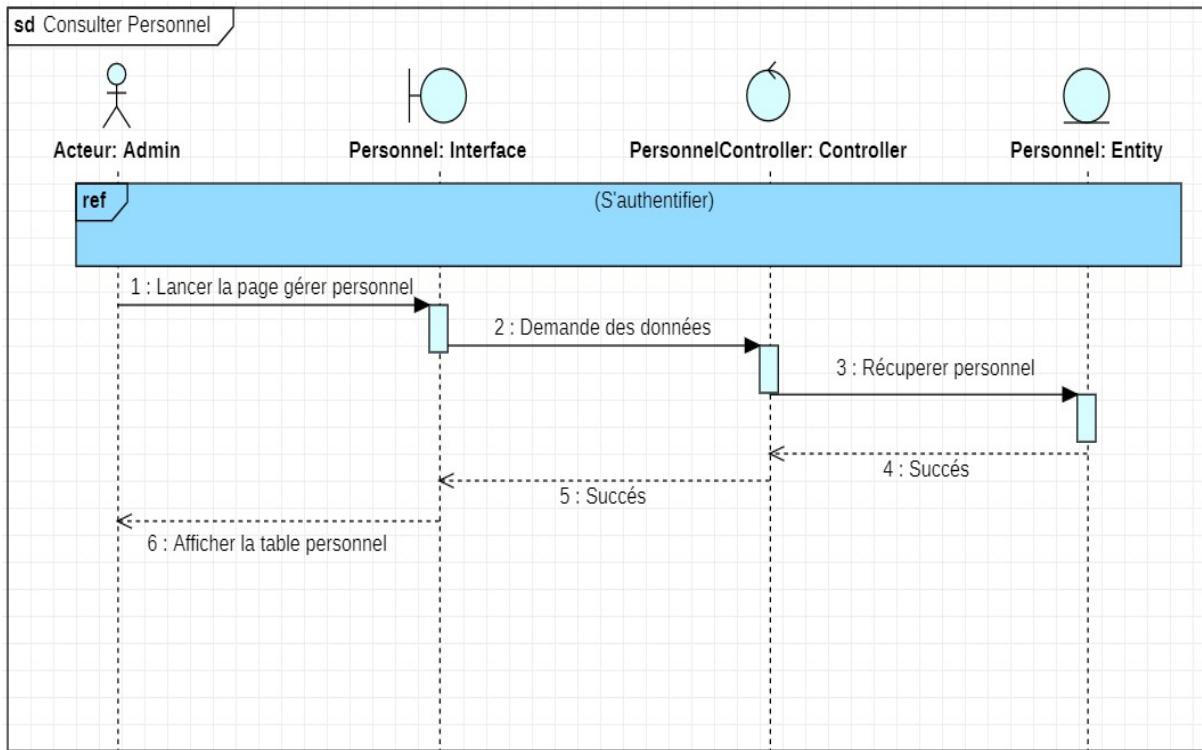


FIGURE 4.13 – Diagramme de séquences détaillé du sous-cas d'utilisation « Consulter personnel »

4.3.1.3 Cas d'utilisation « Supprimer une personne »

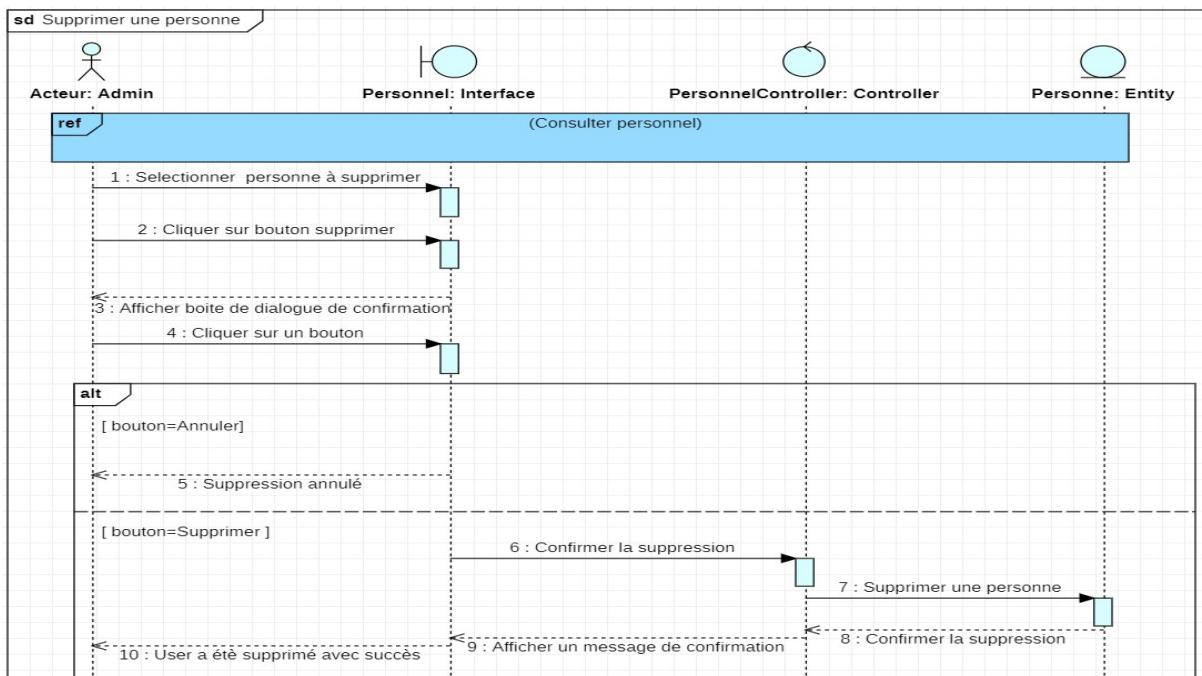


FIGURE 4.14 – Diagramme de séquences détaillé du sous-cas d'utilisation « Supprimer une personne »

4.3.1.4 Cas d'utilisation « Consulter une personne »

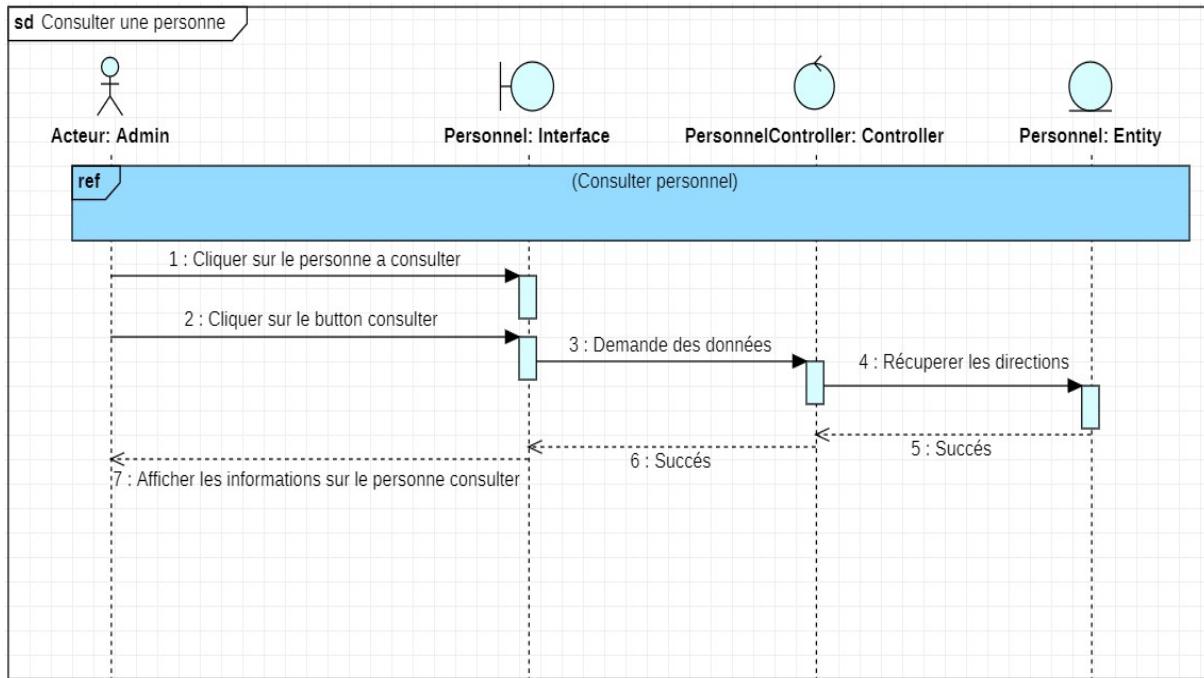


FIGURE 4.15 – Diagramme de séquences détaillé du sous-cas d'utilisation « Consulter une personne »

4.3.1.5 Cas d'utilisation « Consulter tickets »

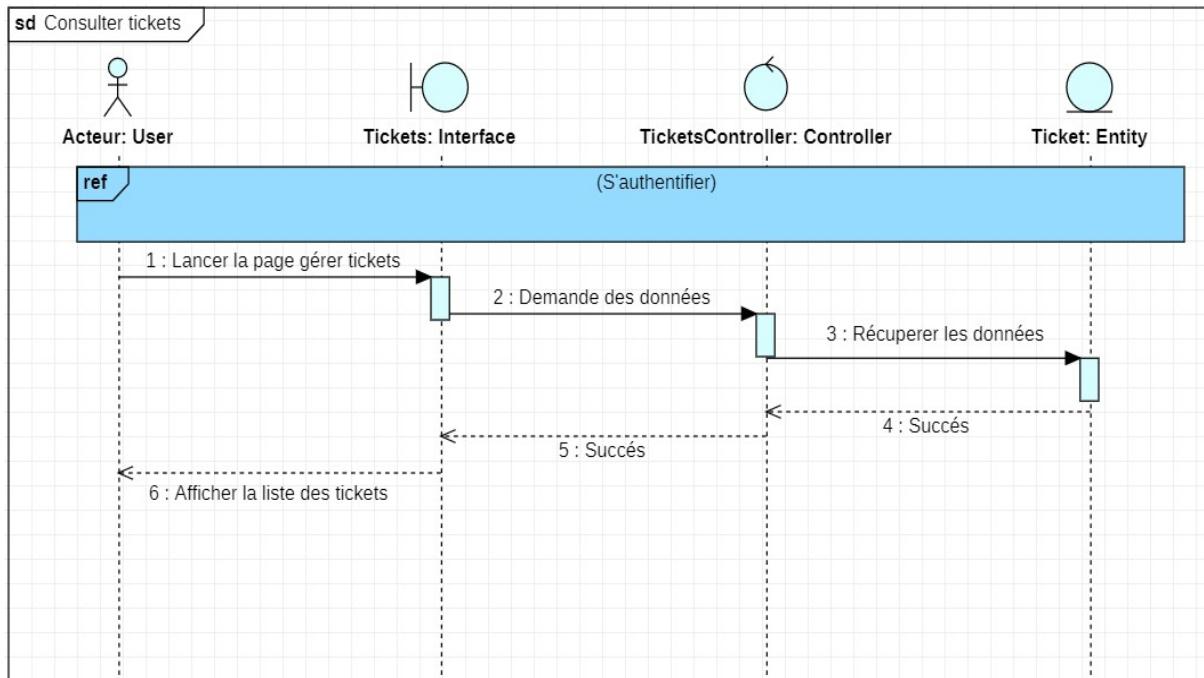


FIGURE 4.16 – Diagramme de séquences détaillé du sous-cas d'utilisation « Consulter tickets »

4.3.1.6 Cas d'utilisation « Modifier une personne »

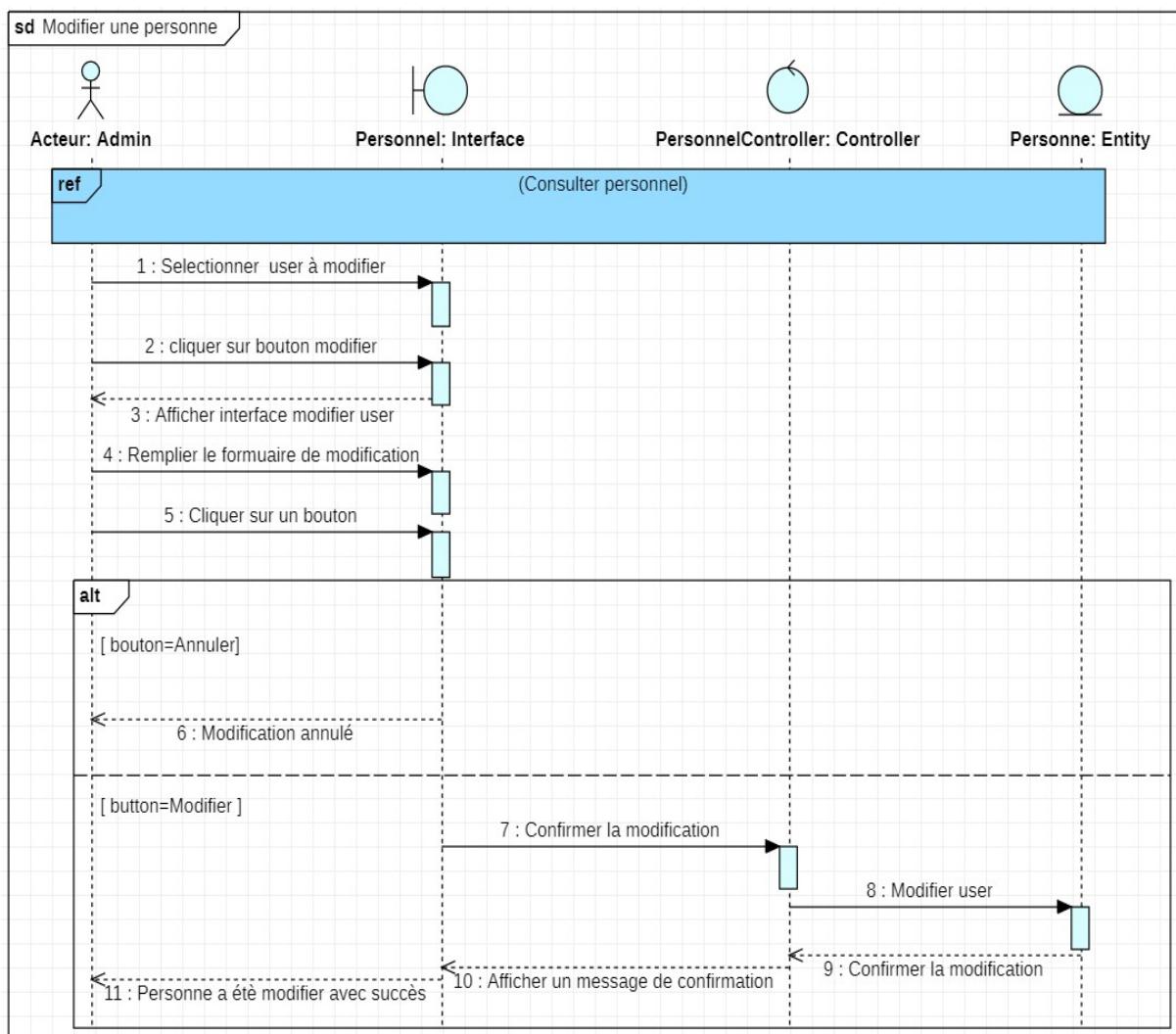


FIGURE 4.17 – Diagramme de séquences détaillé du sous-cas d'utilisation « Modifier une personne »

4.3.1.7 Cas d'utilisation « Consulter tickets approuver »

Dans ce qui suit le diagramme de séquences détaillés du sous-cas d'utilisation « Consulter tickets approuvés ».

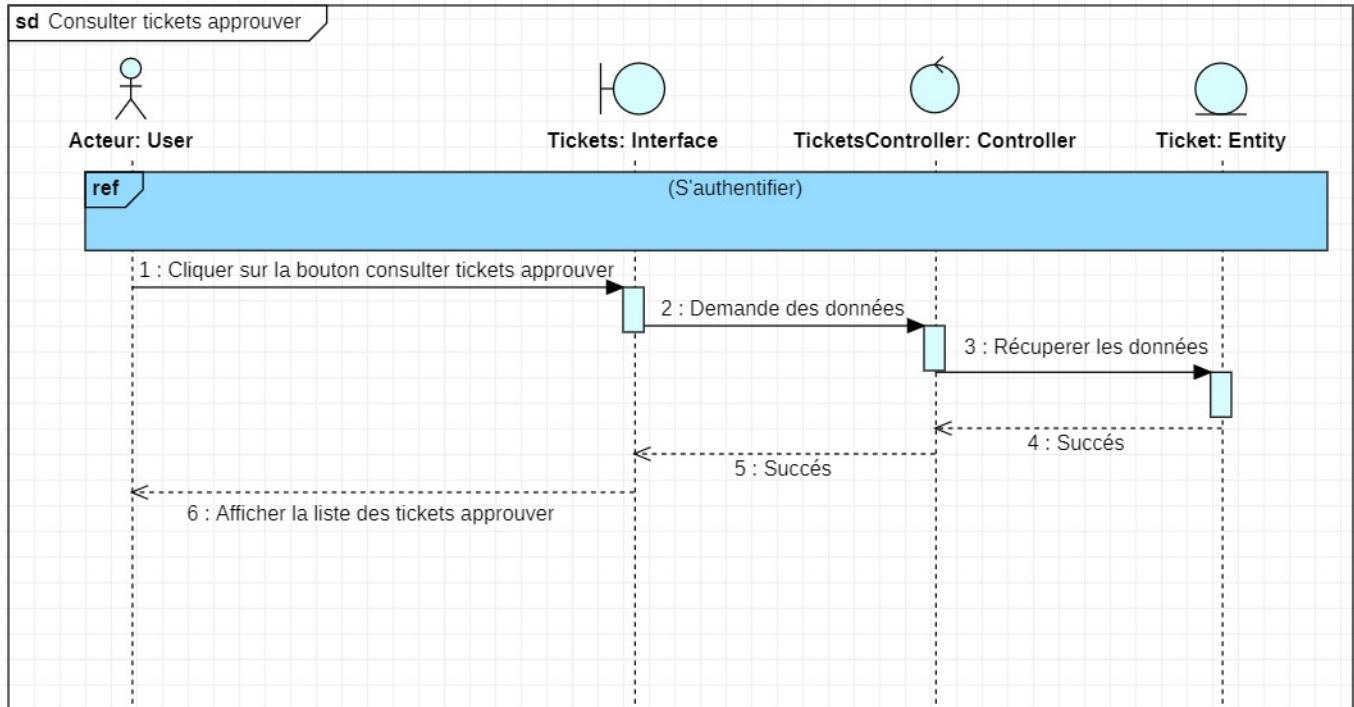


FIGURE 4.18 – Diagramme de séquences détaillés du sous-cas d'utilisation « Consulter tickets approuvés »

4.3.2 Diagrammes d'activités

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation [Fou12].

4.3.2.1 Cas d'utilisation « Modifier une personne »

Dans ce qui suit le diagramme d'activité du sous-cas d'utilisation « Modifier une personne ».

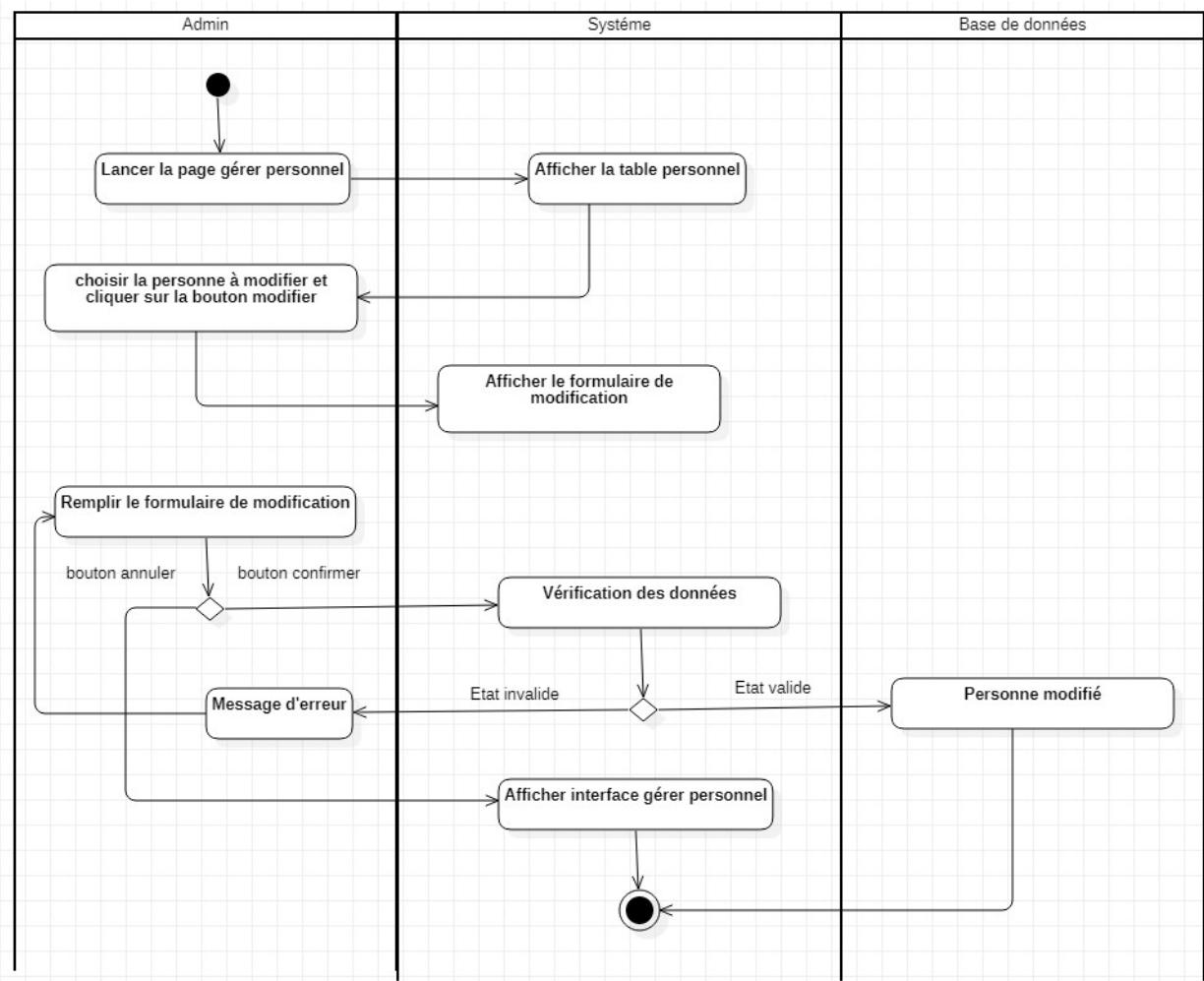


FIGURE 4.19 – Diagramme d'activité du sous-cas d'utilisation « Modifier une personne »

4.3.2.2 Cas d'utilisation « Rechercher personnel »

Dans ce qui suit le diagramme d'activité du sous-cas d'utilisation « Rechercher personnel ».

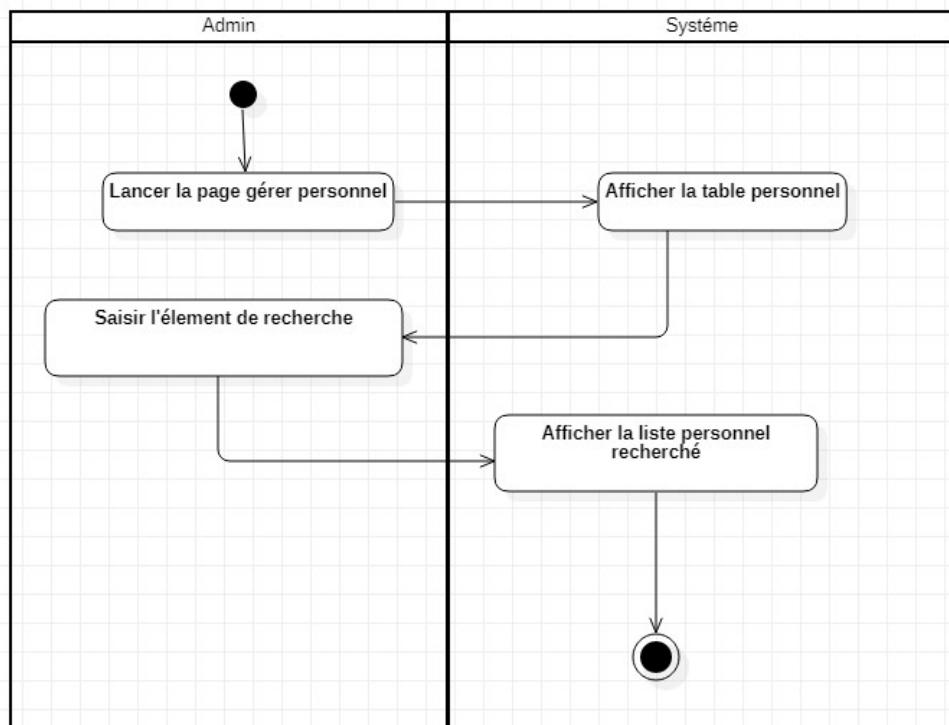


FIGURE 4.20 – Diagramme d'activité du sous-cas d'utilisation « Rechercher personnel »

4.4 Réalisation

- Interface : Consulter personnel

NOM	PRENOM	MATRICULE	DIRECTION			
alaa	khouja	azerty	FIN			
mootez	dridi	6536T	APV			
zied	khouja	jshdkqjlsdh	FIN			
admin	admin1	admin1	ADMIN			
admin	admin1	admin2	ADMIN			

FIGURE 4.21 – Interface : Consulter personnel

- Interface : Ajouter une personne

Ajouter User

Nom

Prenom

Email

Matricule

Direction

Responsable

Mot de passe

Role

Directeur

FIGURE 4.22 – Interface : Ajouter une personne

- Interface : Consulter une personne

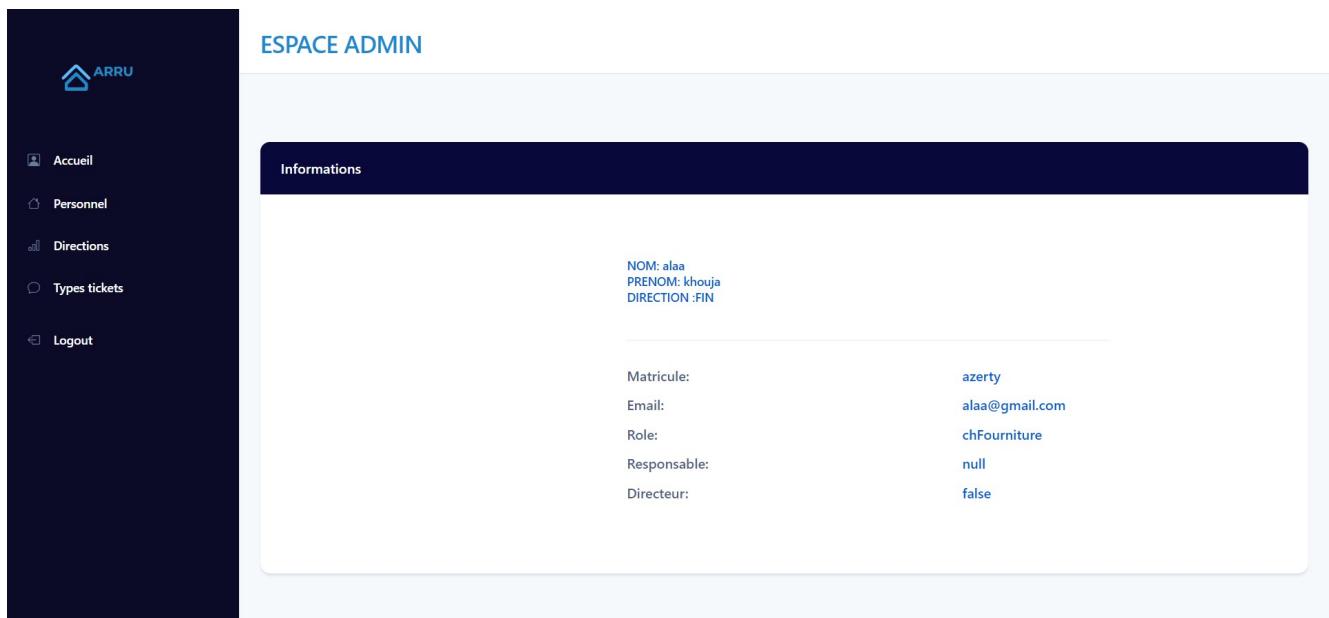


FIGURE 4.23 – Interface : Consulter une personne

- Interface : Modifier une personne

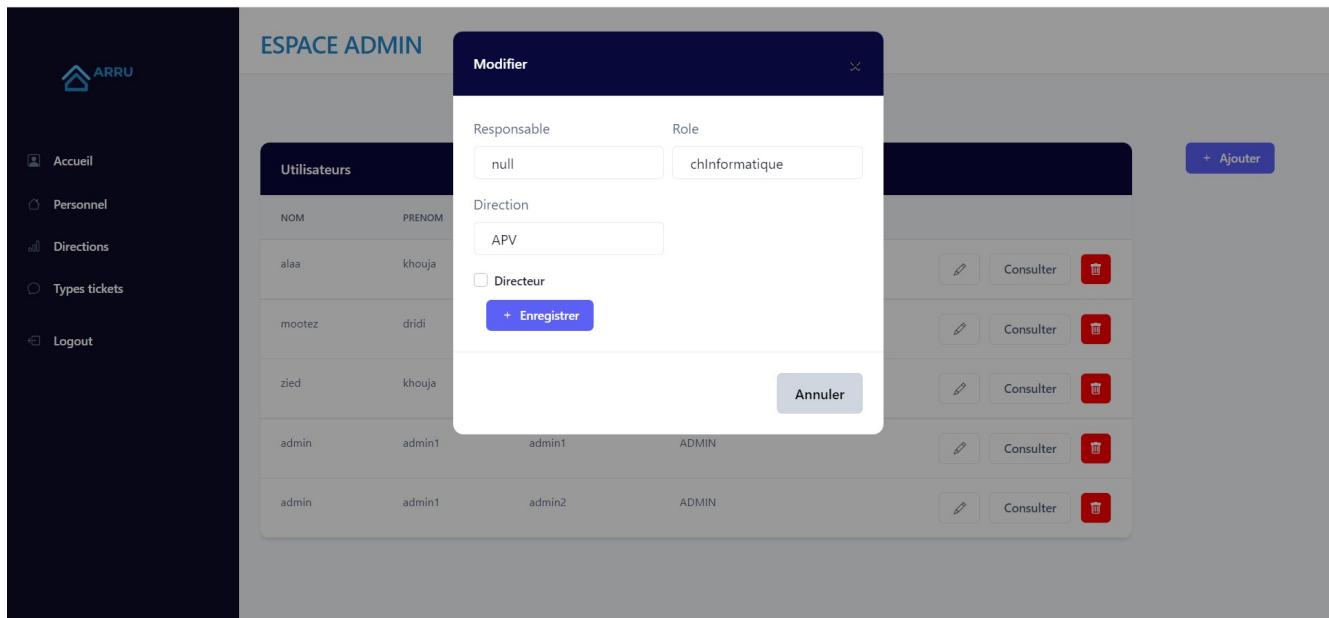


FIGURE 4.24 – Interface : Modifier une personne

- Interface : Consulter tickets

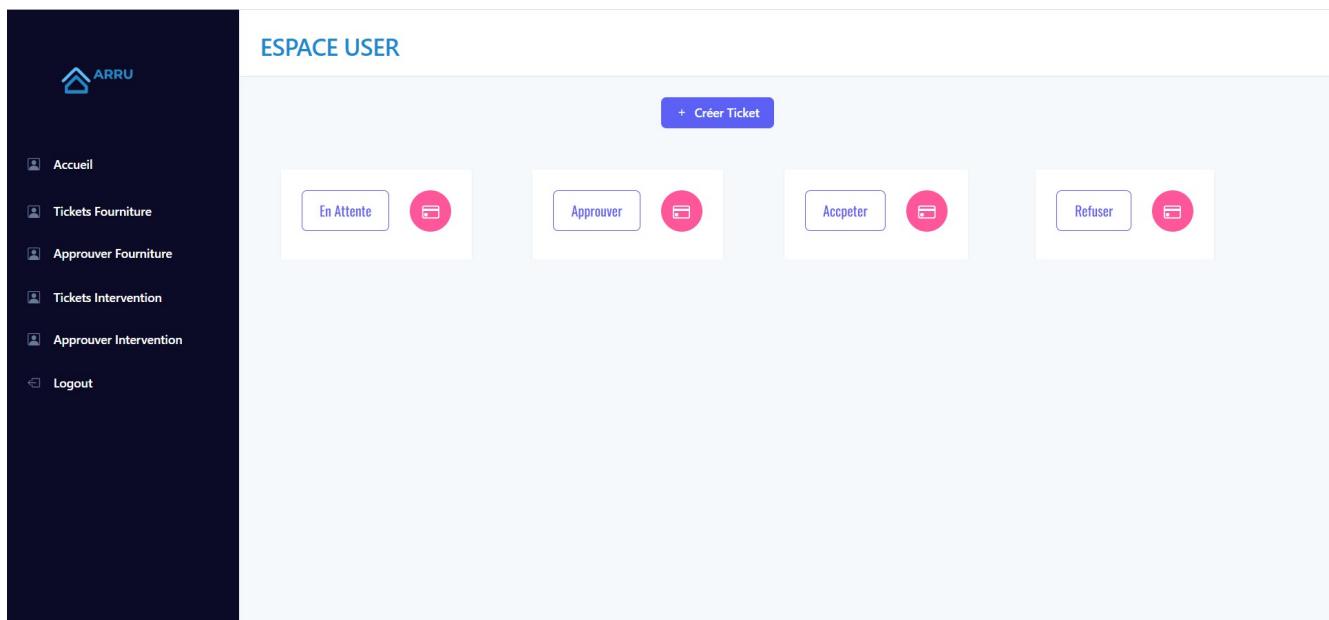


FIGURE 4.25 – Interface : Consulter tickets

- Interface : Consulter tickets approuvés

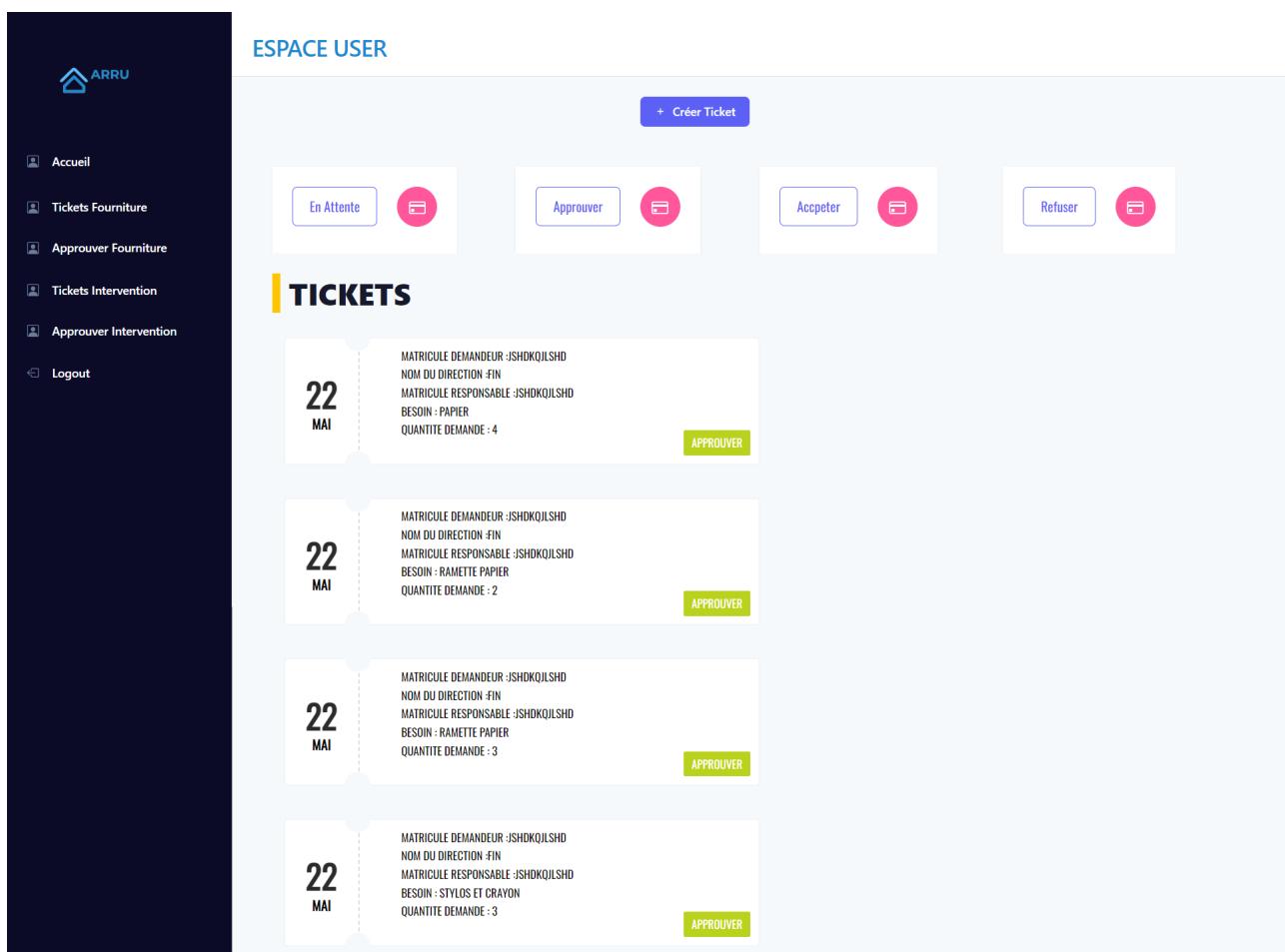


FIGURE 4.26 – Interface : Consulter tickets

- Interface : Consulter tickets refusés

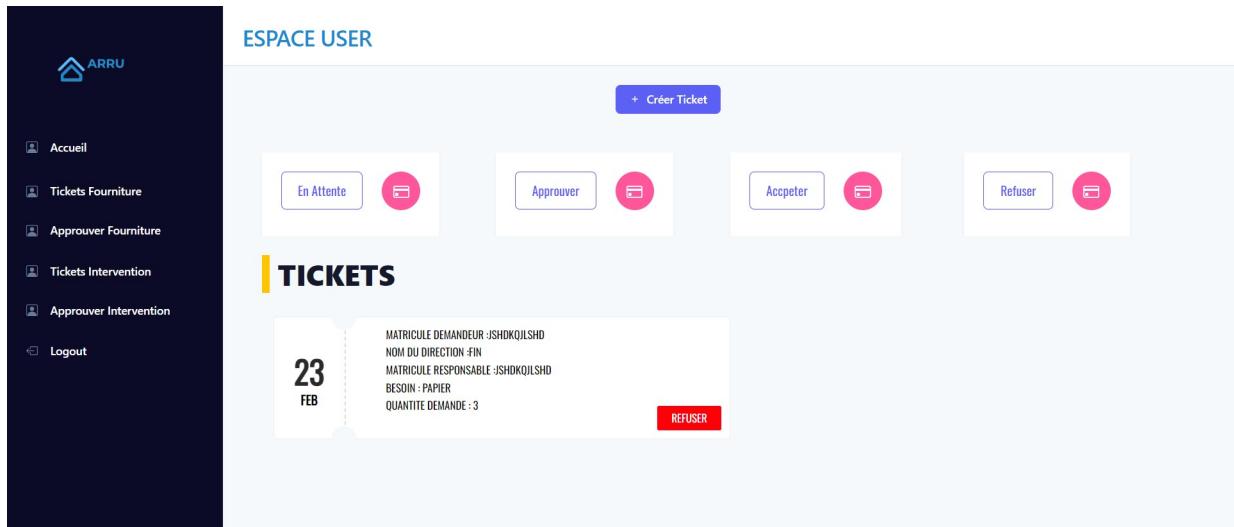


FIGURE 4.27 – Interface : Consulter refusés

4.5 Tests et validation

J'ai terminé ce sprint par une étape de validation et tests réalisés avec le product owner qui a testé toutes les fonctionnalités du sprint 2 et les a validées.

Fonctionnalité	Résultat
Gérer personnel	Conforme
Gérer tickets	Conforme

TABLE 4.14 – Tests et validation

Conclusion

Dans le présent chapitre, j'ai détaillé toutes les étapes de mise en place des fonctionnalités du sprint 2 de l'analyse des besoins jusqu'aux tests et validation. Le prochain chapitre sera consacré au sprint 3 d'où la finalisation de mon application.

Chapitre 5

Sprint 3

Sommaire

Introduction	85
5.1 Spécification des besoins	85
5.1.1 Backlog du sprint 3	85
5.1.2 Diagramme de cas d'utilisation du sprint 3	86
5.2 Analyse	86
5.2.1 Raffinement des cas d'utilisation	86
5.2.2 Diagrammes de séquences système	91
5.3 Conception	94
5.3.1 Diagrammes de séquences détaillés	94
5.3.2 Diagrammes d'activités	97
5.4 Réalisation	99
5.5 Tests et validation	101
Conclusion	101

Introduction

Ce chapitre est consacré aux travaux effectués durant le troisième et dernier sprint. À la fin de ce sprint le backlog produit sera clôturé, pour avoir un projet complet.

5.1 Spécification des besoins

Dans cette partie, je présente le backlog du sprint suivi par le diagramme de cas d'utilisation.

5.1.1 Backlog du sprint 3

Au niveau de cette partie, je présente le backlog du sprint 3 à travers la Table 5.1.

TABLE 5.1 – Le Backlog du sprint 3

ID	User Stories	ID	Type	Tache	Complexité
1	En tant que responsable, je peux approuver ou refuser les tickets en attente	1.1	Back-End	Créer un web service 'ApprouverTicketAPI' pour approuver les tickets en attente.	1
		1.2	Front-End	Créer un service web front.	2
		1.3	Front-End	Créer l'interface "Approuver tickets".	2
		-	Test	Test et validation.	-
		-	Conception	Conception des diagrammes.	-
2	En tant que chargé tickets, je peux accepter ou refuser les tickets approuvés	2.1	Back-End	Créer un web service 'AccepterTicketAPI' pour Accepter les tickets approuver.	1
		2.2	Front-End	Créer un service web front.	2
		2.3	Front-End	Créer l'interface "Gérer ticket".	2
		-	Test	Test et validation.	-
		-	Conception	Conception des diagrammes.	-

5.1.2 Diagramme de cas d'utilisation du sprint 3

La figure représente le diagramme de cas d'utilisation du sprint 3 :

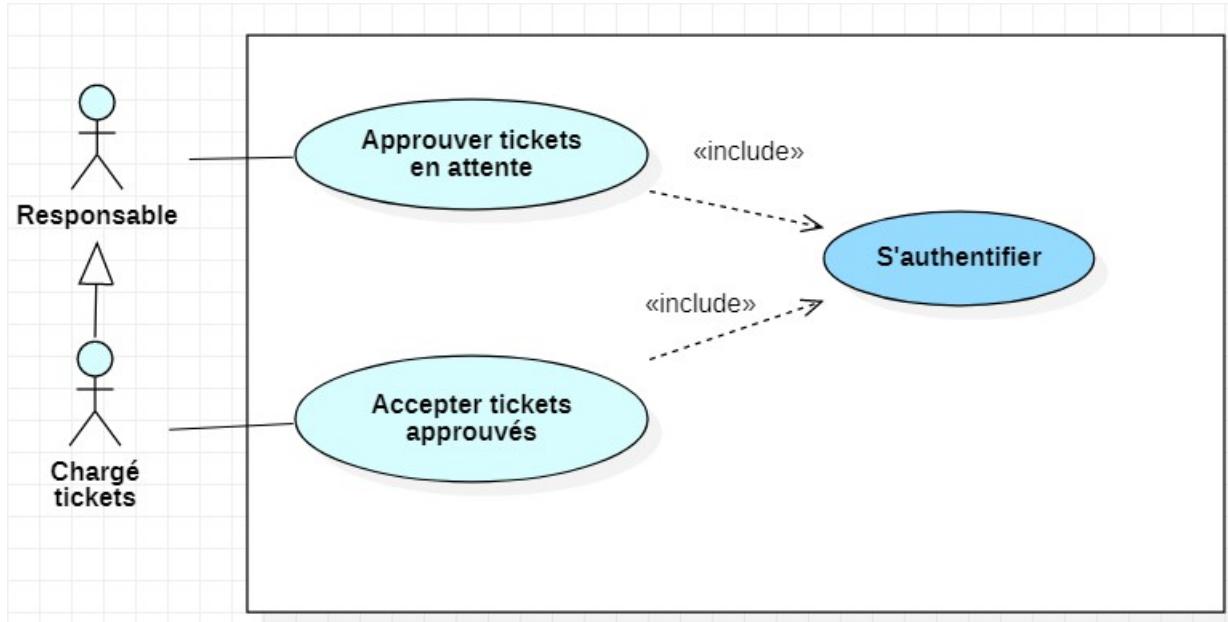


FIGURE 5.1 – Diagramme de cas d'utilisation du sprint 3

5.2 Analyse

Afin de mieux assimiler les cas d'utilisation, j'établis leurs raffinements pour livrer une description sur les différents scénarios possibles.

5.2.1 Raffinement des cas d'utilisation

Dans cette sous-section, je présente les raffinements des diagrammes de cas d'utilisation liés au quatrième sprint, ainsi qu'une description textuelle des principaux cas d'utilisation.

5.2.1.1 Ca d'utilisation « Approuver tickets en attente »

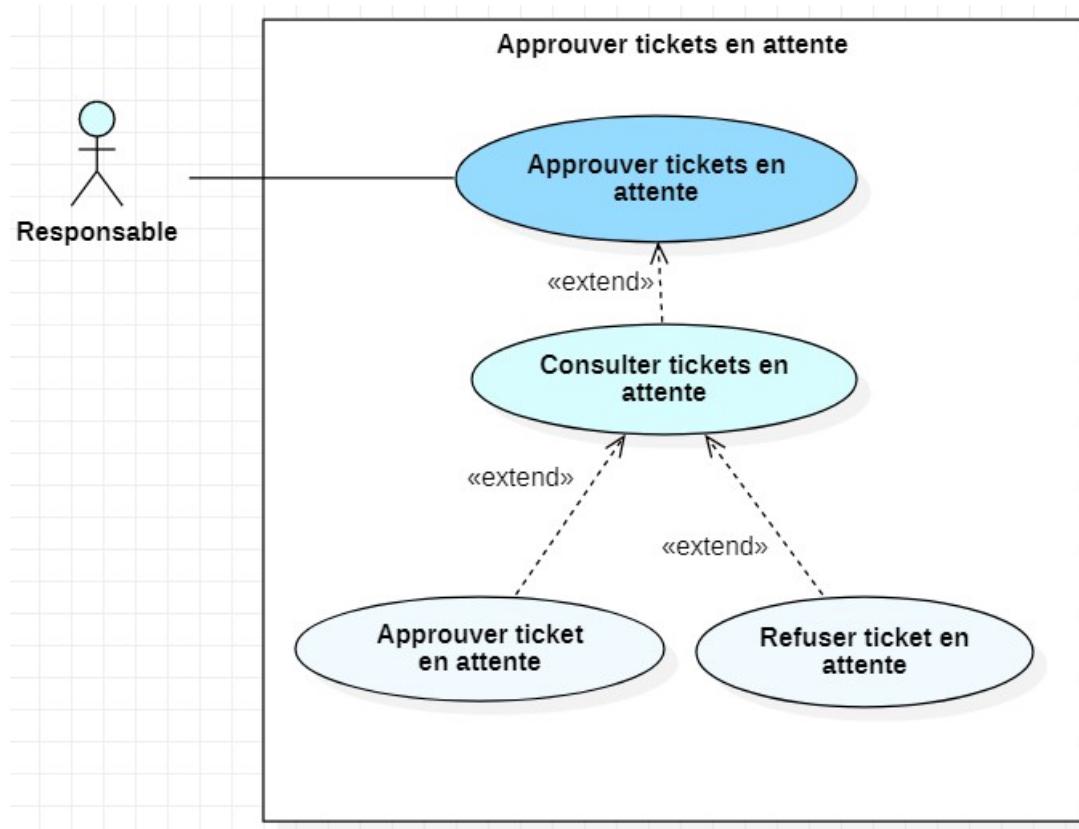


FIGURE 5.2 – Diagramme de cas d'utilisation « Approuver tickets en attente ».

TABLE 5.2 – Description textuelle du sous-cas d'utilisation « Consulter tickets en attente »

Cas d'utilisation	Consulter tickets en attente
Acteurs	Responsable / Chargé tickets
Pré-condition	Acteur authentifié
Scénario Nominal	1. Le responsable ouvre l'interface approuver tickets en attente. 2. La liste des tickets en attente s'affiche.
Post-condition	La liste des tickets en attente est affichée.
Exception	Problème de connexion à la base de données.

TABLE 5.3 – Description textuelle du sous-cas d'utilisation « Approuver ticket en attente »

Cas d'utilisation	Approuver ticket en attente
Acteurs	Responsable / Chargé tickets
Pré-condition	Acteur authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. Afficher la liste des tickets en attente. 2. Le responsable choisit le ticket en attente à approuver et clique sur l'icone approuver. 3. Le responsable clique sur le bouton « Confirmer » . 4. Le système affiche un message de succès d'approbation du ticket en attente. 5. Le système affiche la nouvelle liste des tickets en attente.
Post-condition	Un ticket en attente approuvé.
Exception	Problème de connexion à la base de données.

TABLE 5.4 – Description textuelle du sous-cas d'utilisation « Refuser ticket en attente »

Cas d'utilisation	Refuser ticket en attente
Acteurs	Responsable
Pré-condition	Responsable authentifié
Scénario Nominal	<ol style="list-style-type: none"> 1. Afficher la liste des tickets en attente. 2. Le responsable choisit le ticket à refuser et clique sur l'icone refuser. 3. Le responsable clique sur le bouton « Confirmer » . 4. Le système affiche un message de succès de refus du ticket en attente. 5. Le système affiche la nouvelle liste des tickets en attente.
Post-condition	Un ticket en attente refuser.
Exception	Problème de connexion à la base de données.

5.2.1.2 Cas d'utilisation « Accepter tickets approuvés »

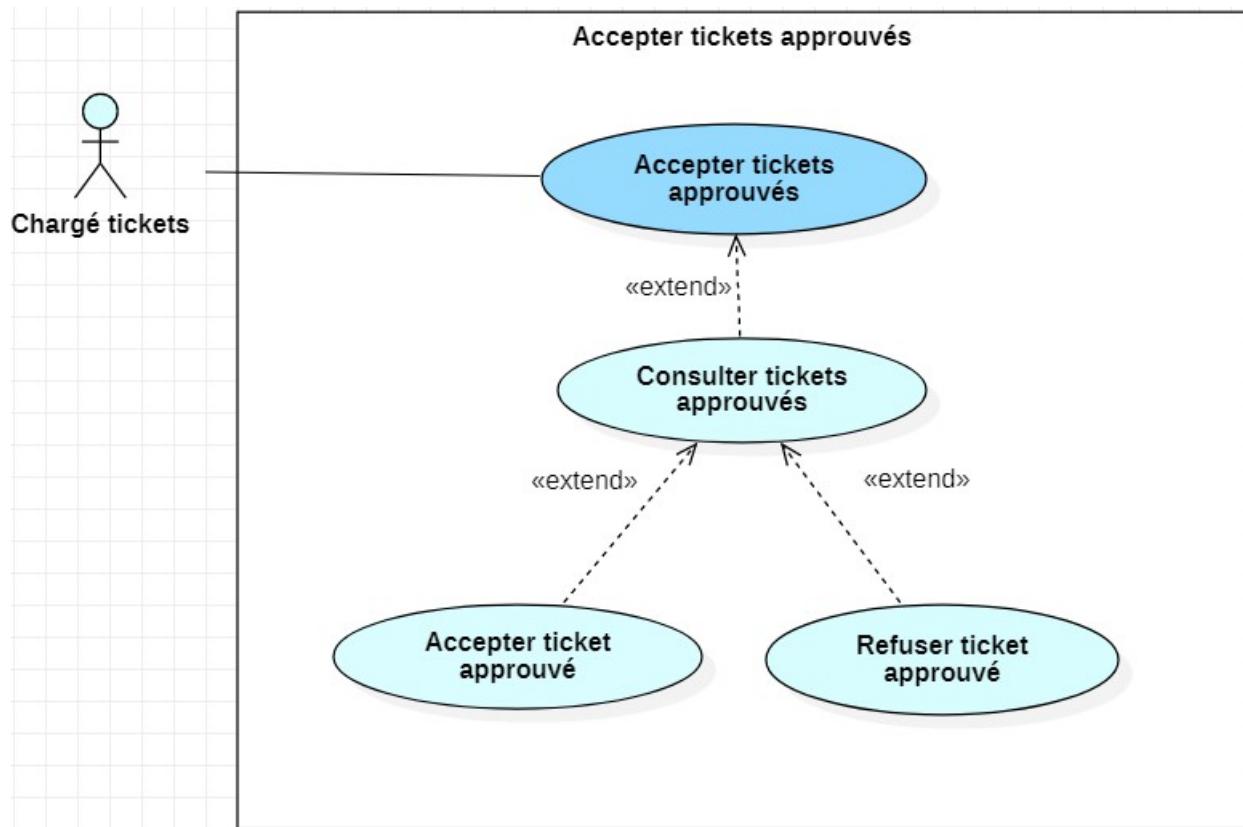


FIGURE 5.3 – Diagramme de cas d'utilisation « Accepter tickets approuvés »

TABLE 5.5 – Description textuelle du sous-cas d'utilisation « Consulter tickets approuvés »

Cas d'utilisation	Consulter tickets approuvés
Acteurs	Chargé tickets
Pré-condition	Chargé tickets authentifié
Scénario Nominal	1. Le chargé tickets ouvre l'interface accepter tickets approuvés . 2. La liste des tickets approuvés s'affiche.
Post-condition	La liste des tickets approuvés est affichée.
Exception	Problème de connexion à la base de données.

TABLE 5.6 – Description textuelle du sous-cas d'utilisation « Accepter ticket approuvé »

Cas d'utilisation	Accepter ticket approuvé
Acteurs	Chargé tickets
Pré-condition	Chargé tickets authentifié
Scénario Nominal	1. Afficher la liste des tickets approuvés.
	2. Le responsable choisit le ticket approuvé à accepter et clique sur l'icone accepter.
	3. Chargé tickets clique sur le bouton «Confirmer» .
	4. Le système affiche un message de succès d'acceptation du ticket approuvé.
	5. Le système affiche la nouvelle liste des tickets approuvés.
Post-condition	Un ticket approuvé est accepté.
Exception	Problème de connexion à la base de données.

TABLE 5.7 – Description textuelle du sous-cas d'utilisation « Refuser ticket approuvé »

Cas d'utilisation	Refuser ticket approuvé
Acteurs	Chargé tickets
Pré-condition	Chargé tickets authentifié
Scénario Nominal	1. Afficher la liste des tickets approuver.
	2. Le responsabel choisir le ticket à accepter et clique sur l'icone refuser.
	3. Chargé tickets clique sur le bouton «Confirmer» .
	4. Le système affiche un message de succès de refus du ticket approuvé.
	5. Le système affiche la nouvelle liste des tickets approuvés.
Post-condition	Un ticket approuvé est refusé.
Exception	Problème de connexion à la base de données.

5.2.2 Diagrammes de séquences système

5.2.2.1 Cas d'utilisation « Consulter tickets en attente »

La Figure suivante présente le diagramme de séquences système du sous-cas d'utilisation « Consulter tickets en attente ».

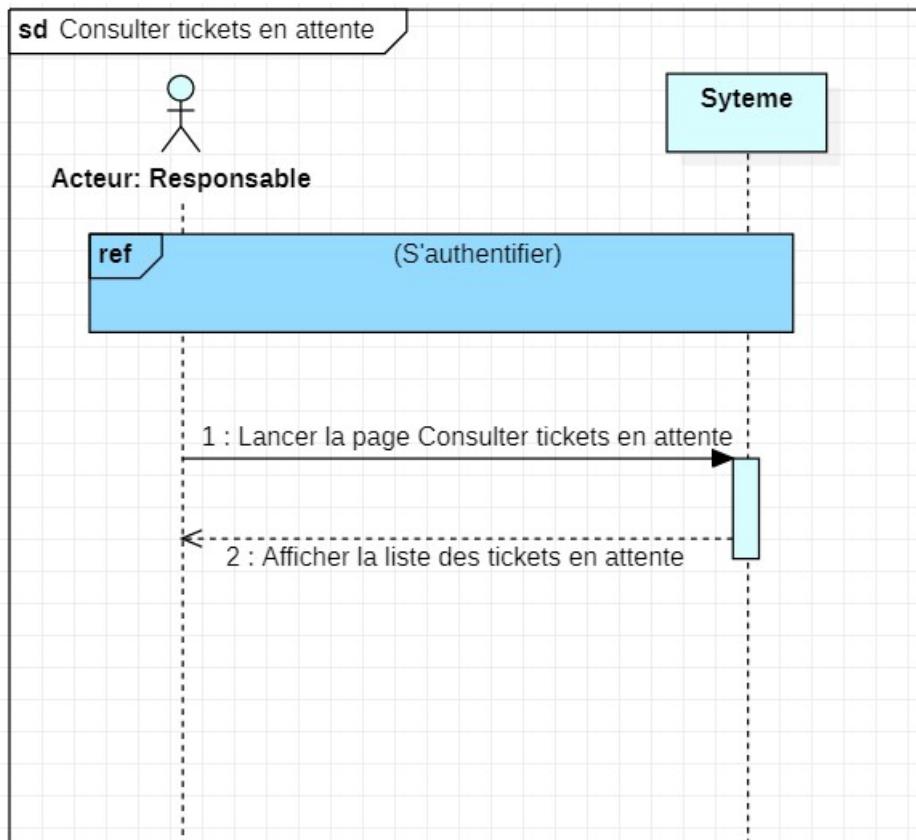


FIGURE 5.4 – Diagramme de séquences système du sous-cas d'utilisation « Consulter tickets en attente ».

5.2.2.2 Cas d'utilisation « Approuver tickets en attente »

La Figure ci-après visualise le diagramme de séquences système du sous-cas d'utilisation « Approuver ticket en attente ».

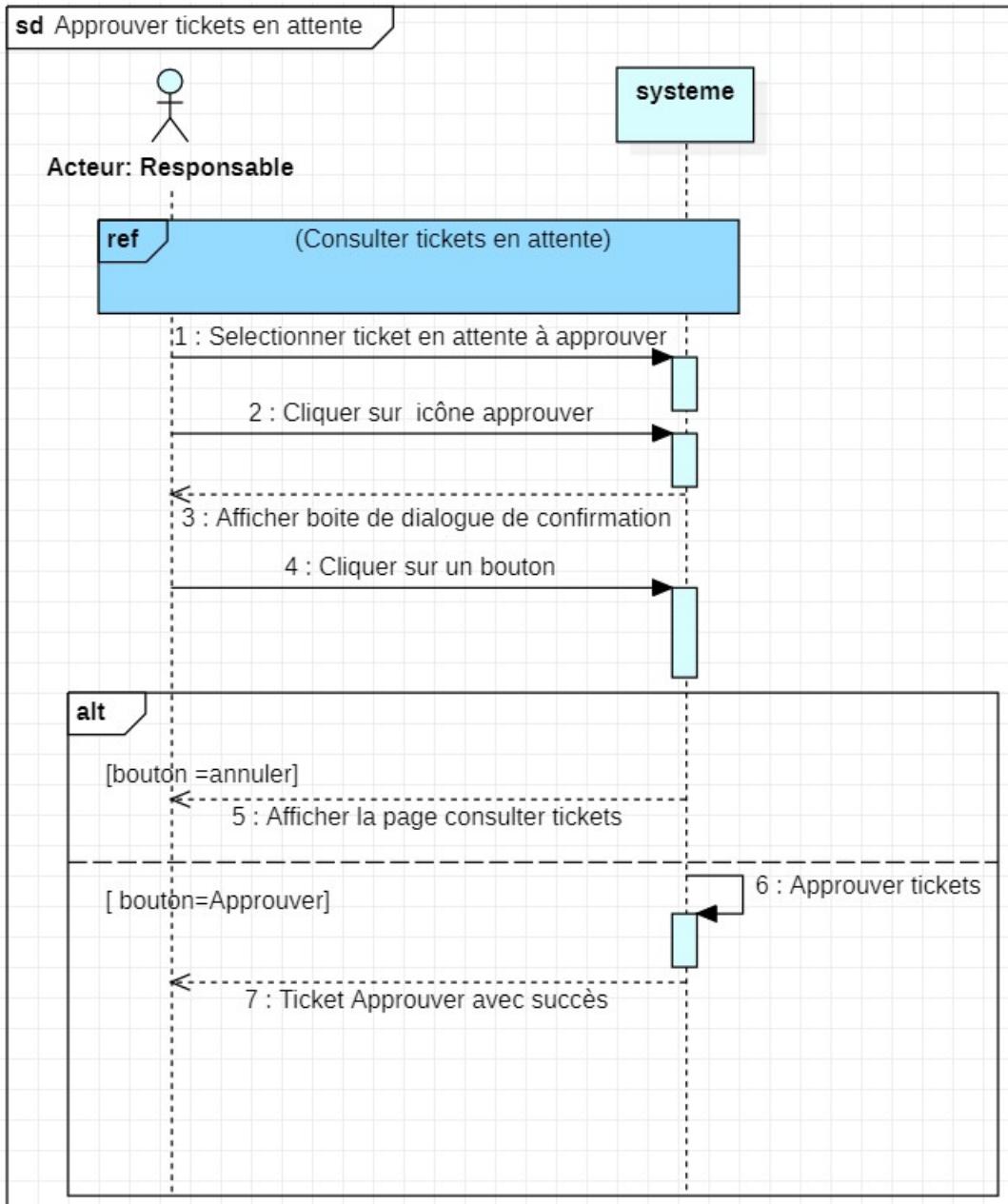


FIGURE 5.5 – Diagramme de séquences système du sous-cas d'utilisation « Approuver ticket en attente ».

5.2.2.3 Cas d'utilisation « Refuser ticket en attente »

La Figure ci-dessous présente le diagramme de séquences système du sous-cas d'utilisation « Refuser ticket en attente ».

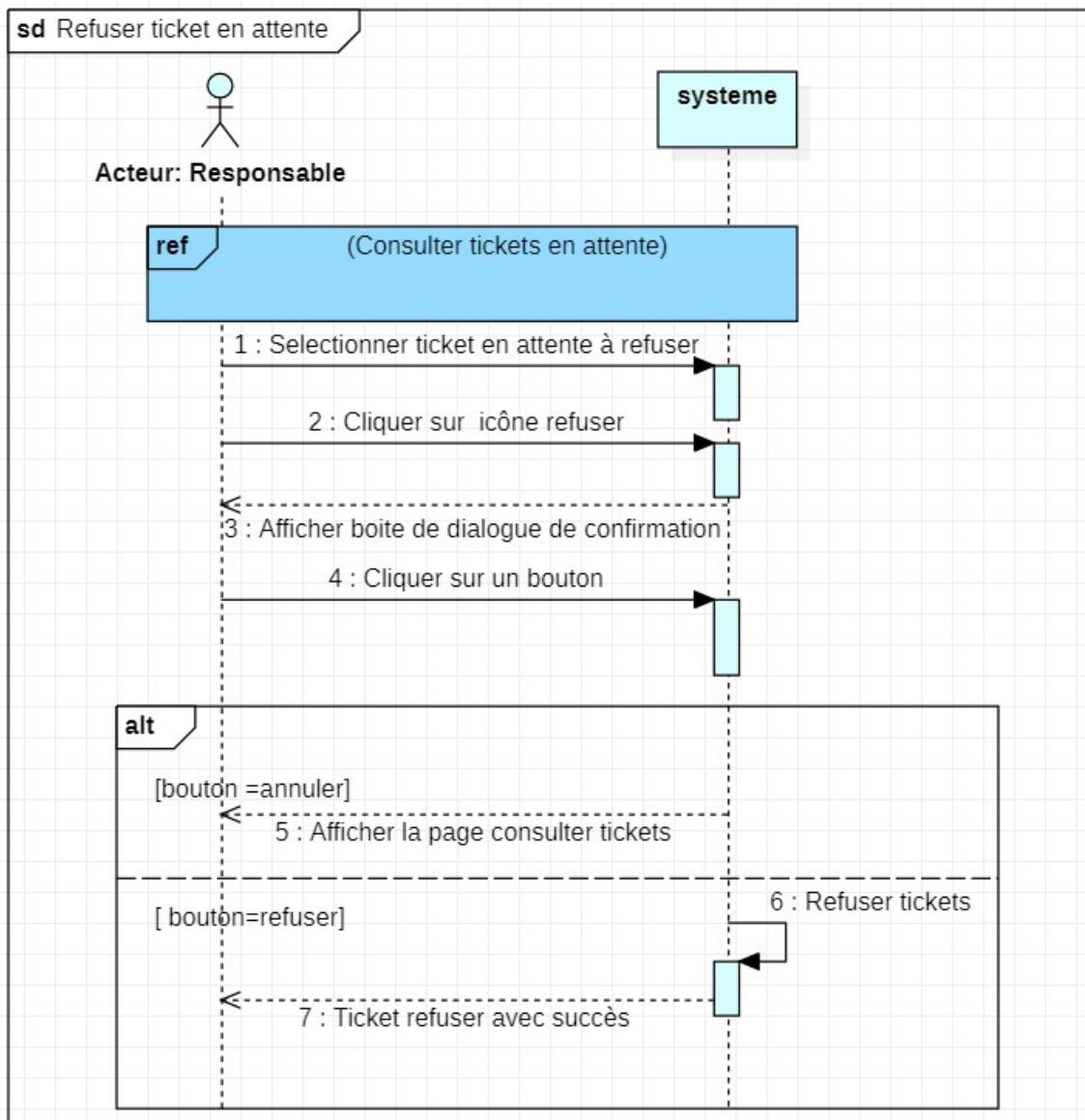


FIGURE 5.6 – Diagramme de séquences système du sous-cas d'utilisation « Refuser ticket en attente »

5.3 Conception

Dans cette section je présenterai les diagrammes de séquences détaillés et d'activité du sprint 3.

5.3.1 Diagrammes de séquences détaillés

5.3.1.1 Cas d'utilisation « Consulter tickets en attente »

La Figure suivante présente le diagramme de séquences détaillés du sous-cas d'utilisation « Consulter tickets en attente ».

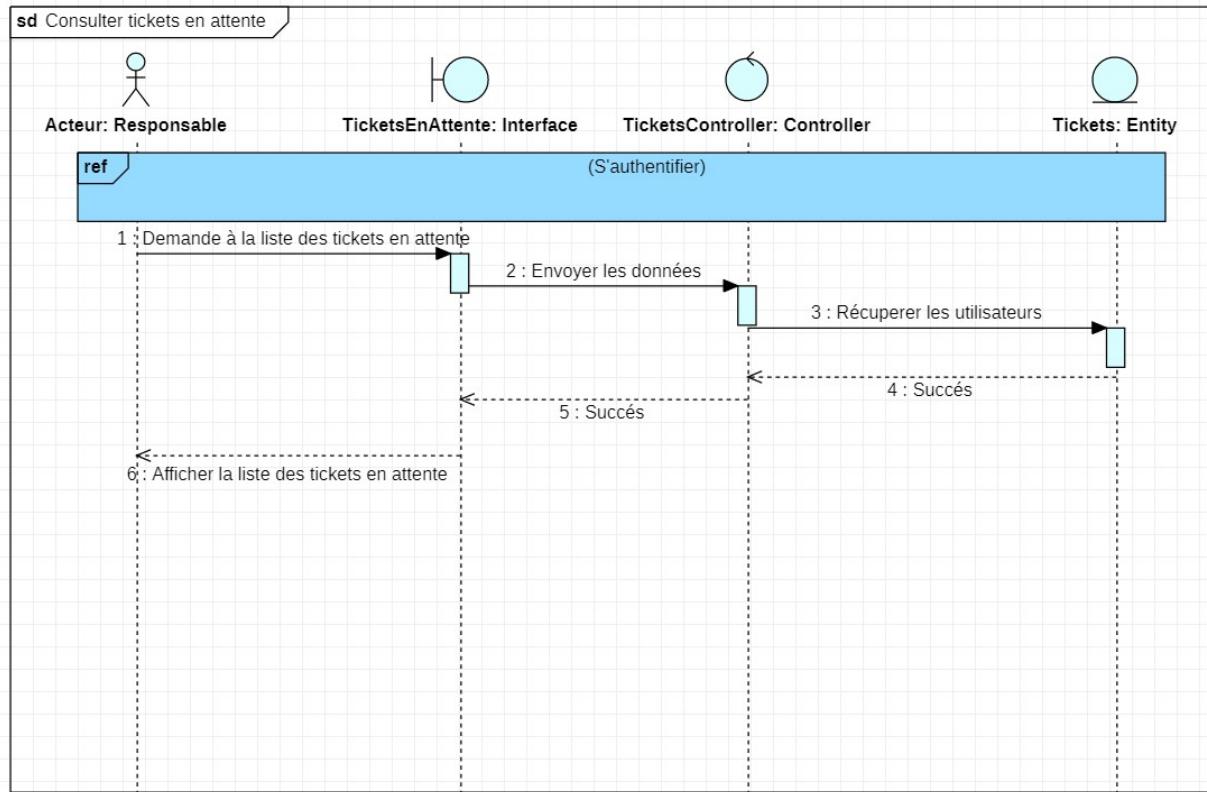


FIGURE 5.7 – Diagramme de séquences détaillé du cas d'utilisation « Consulter tickets en attente »

5.3.1.2 Cas d'utilisation « Approuver ticket en attente »

La Figure ci-dessous présente le diagramme de séquences détaillé du sous-cas d'utilisation « Approuver ticket en attente ».

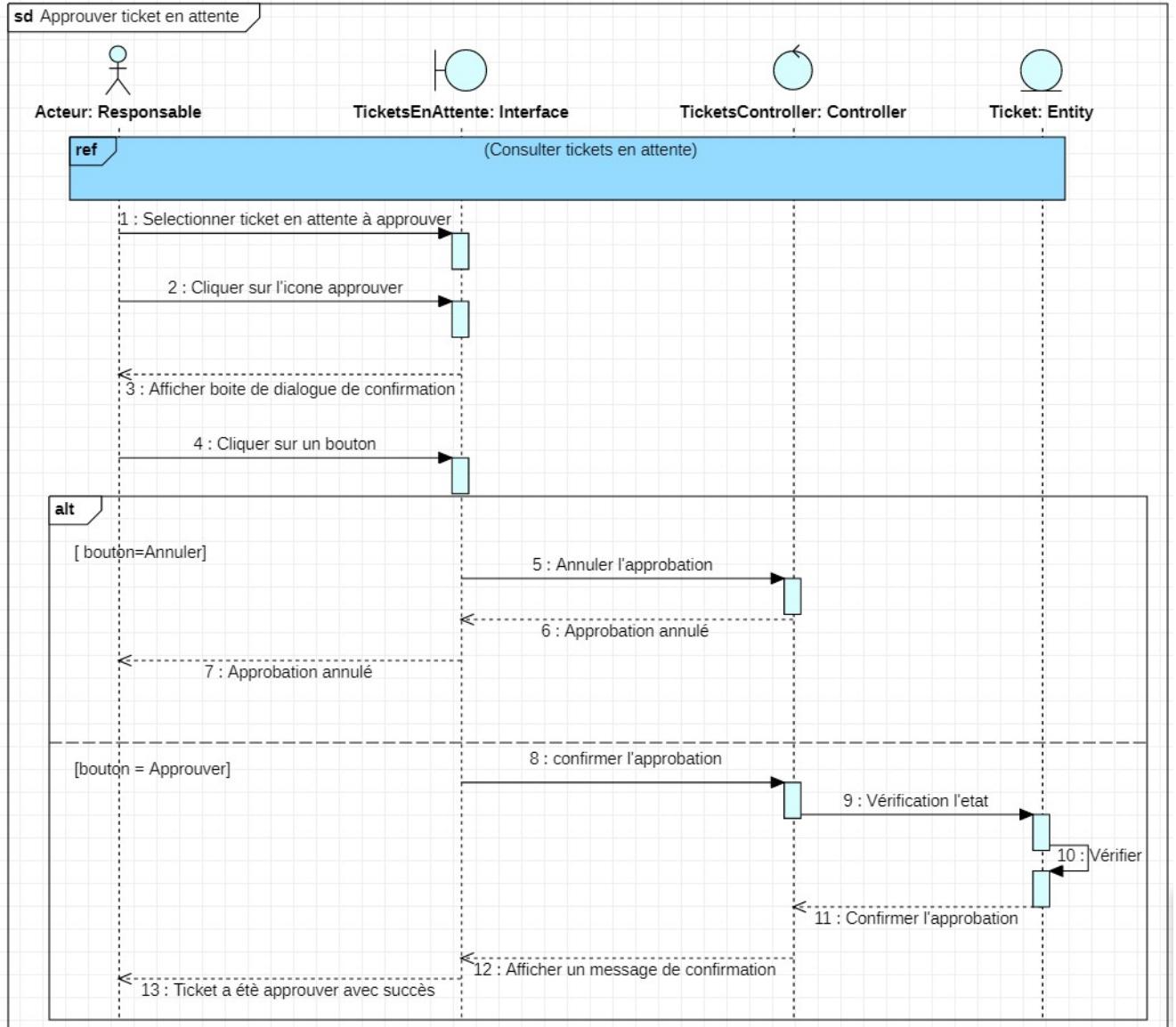


FIGURE 5.8 – Diagramme de séquences détaillé du sous-cas d'utilisation « Approuver ticket en attente »

5.3.1.3 Cas d'utilisation « Refuser ticket en attente »

La Figure suivante présente le diagramme de séquences détaillés du sous-cas d'utilisation « Refuser ticket en attente ».

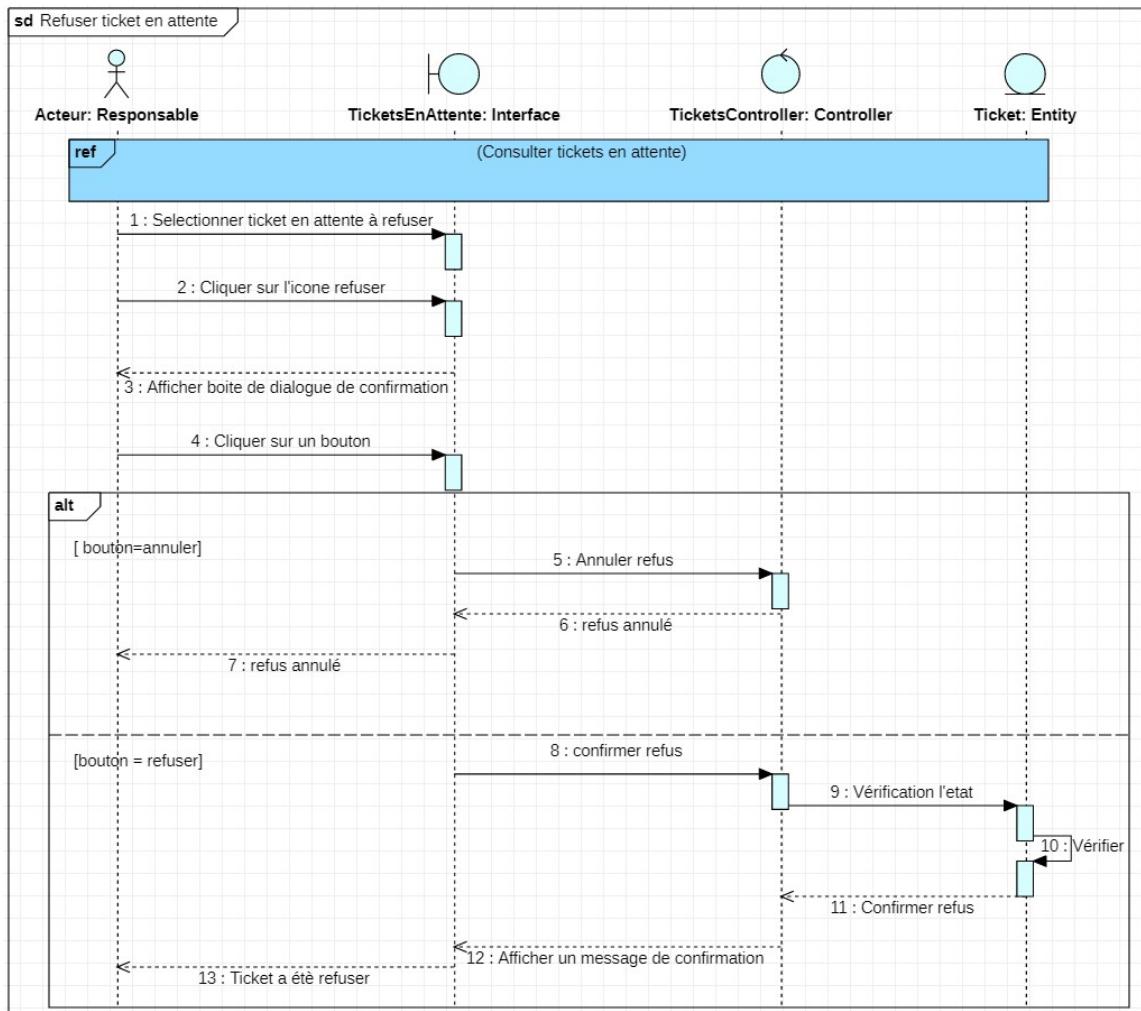


FIGURE 5.9 – Diagramme de séquences détaillé du sous-cas d'utilisation « Approuver ticket en attente ».

5.3.2 Diagrammes d'activités

5.3.2.1 Cas d'utilisation « Approuver tickets en attente »

La Figure suivante présente le diagramme d'activité du sous-cas d'utilisation « Approuver ticket en attente ».

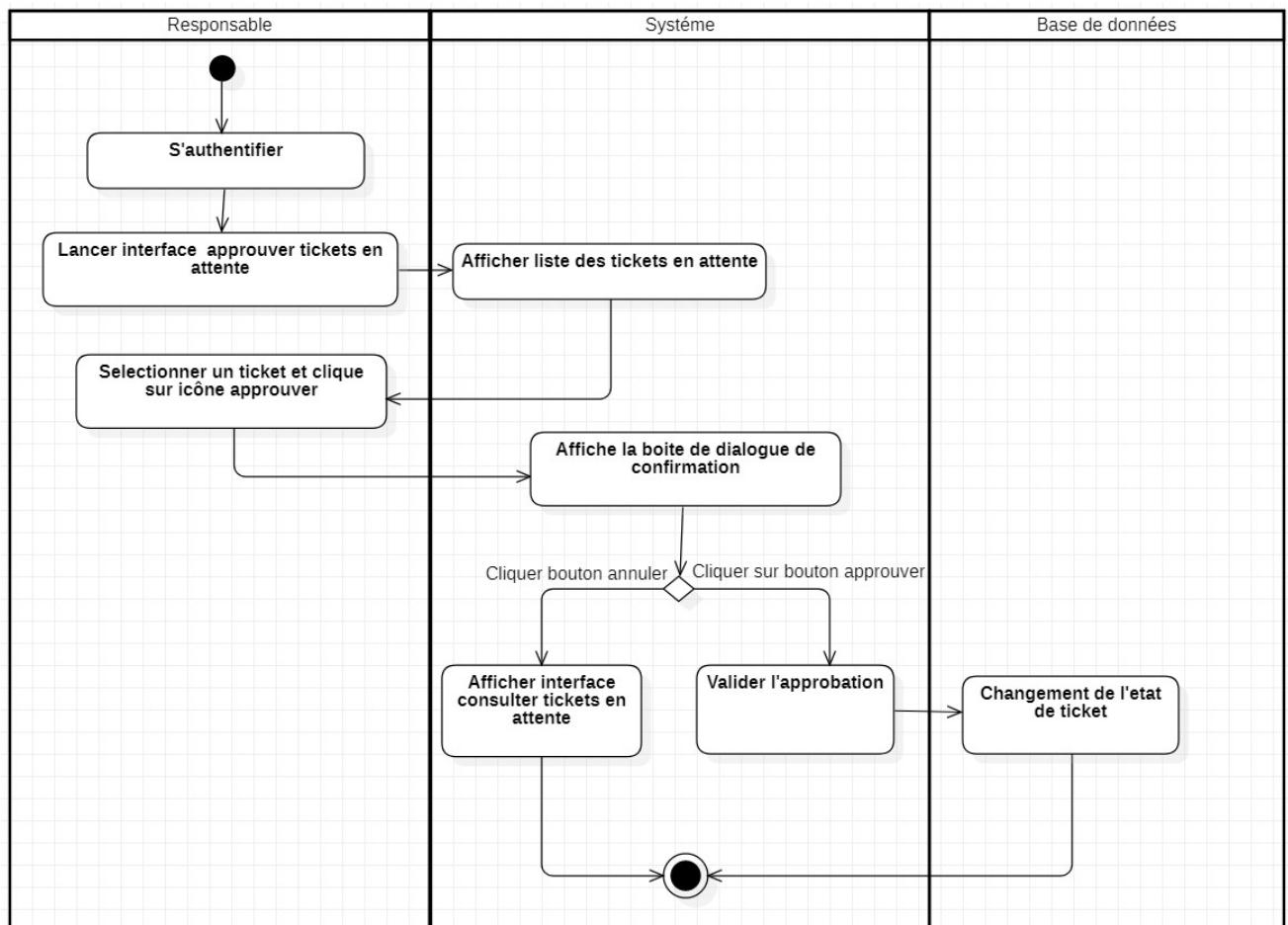


FIGURE 5.10 – Diagramme d'activité du sous-cas d'utilisation « Approuver ticket en attente ».

5.3.2.2 Cas d'utilisation « Refuser ticket en attente »

La Figure suivante présente le diagramme d'activité du sous-cas d'utilisation « Refuser ticket en attente ».

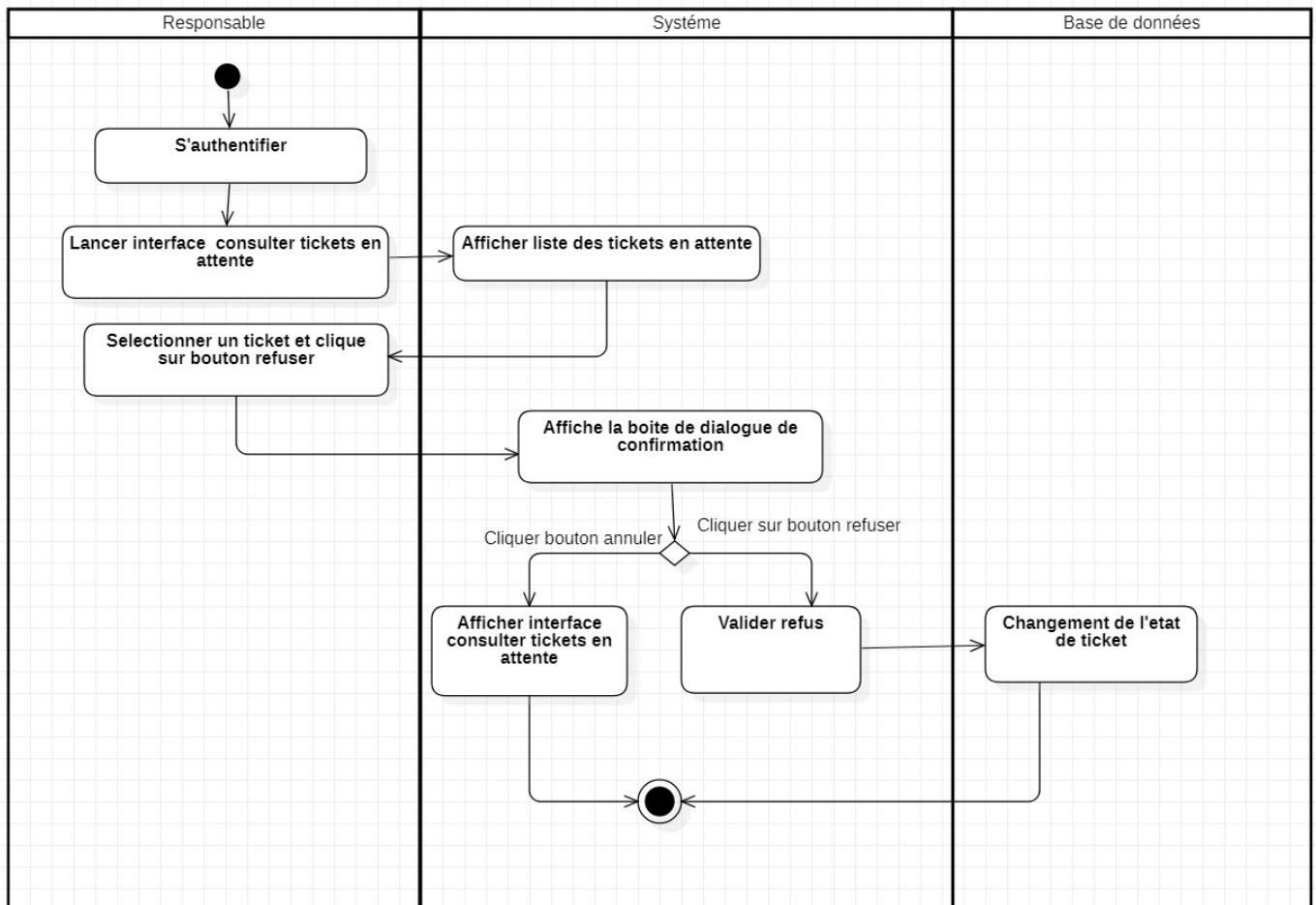


FIGURE 5.11 – Diagramme d'activité du sous-cas d'utilisation « Refuser ticket en attente ».

5.4 Réalisation

- Interface : Consulter tickets en attente

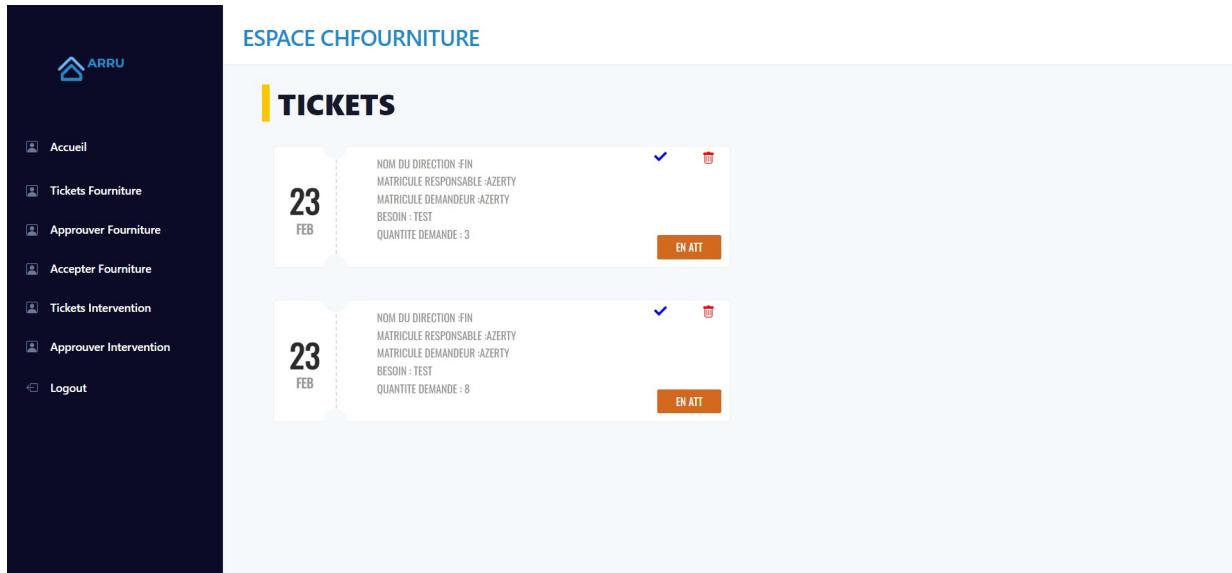


FIGURE 5.12 – Interface : Consulter tickets en attente

- Interface : Approuver ticket en attente

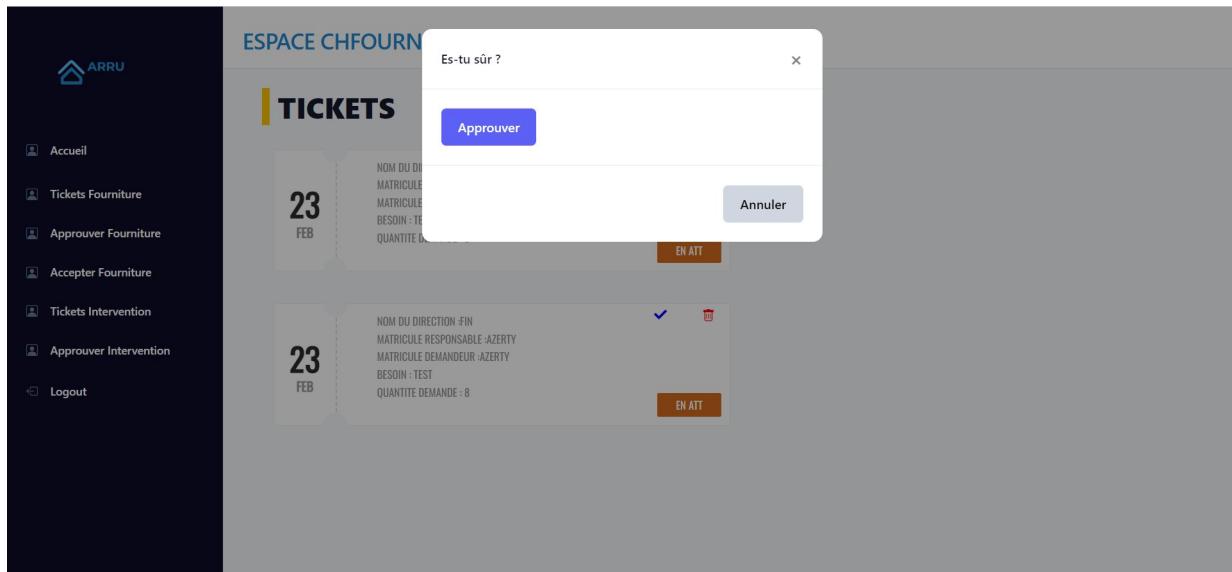


FIGURE 5.13 – Interface : Approuver ticket en attente

- Interface : Refuser ticket en attente

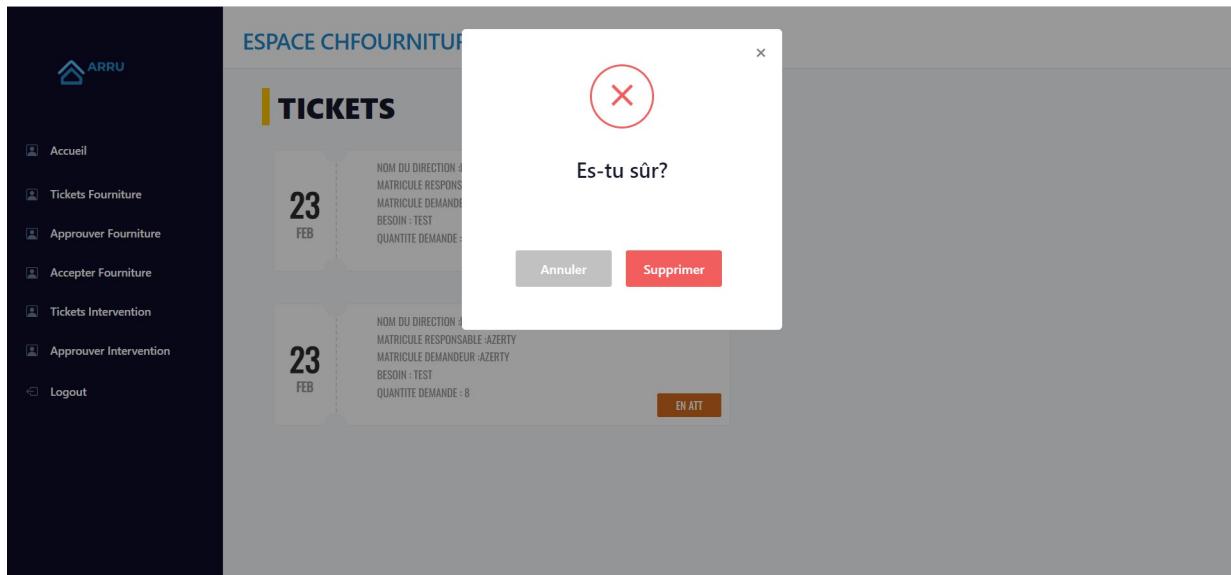


FIGURE 5.14 – Interface : Refuser ticket en attente

- Interface : Consulter tickets en approuvés

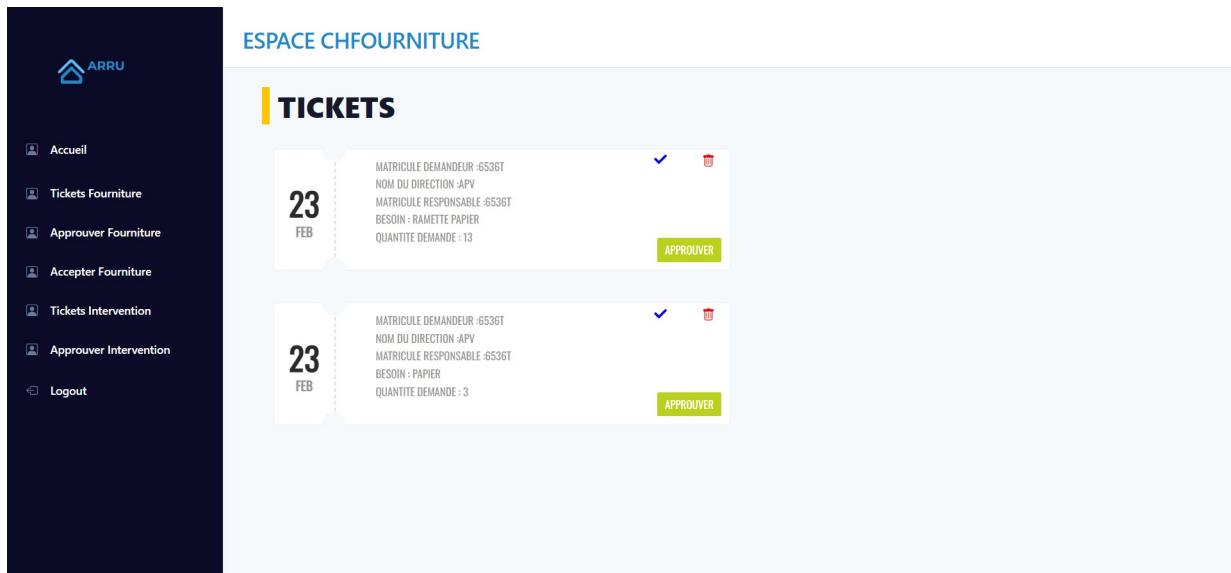


FIGURE 5.15 – Interface : Consulter tickets en approuvés

5.5 Tests et validation

Les tests des fonctionnalités ont été réalisés en collaboration avec le product owner. Avec cette étape je clôture la mise en place de mon application qui a été faite avec succès.

Fonctionnalité	Résultat
Approuver tickets en attente	Conforme
Accpeter tickets approuvés	Conforme

TABLE 5.8 – Liste des tests

Conclusion

A ce stade, j'ai réussi à développer le dernier sprint de ma solution pour arriver à un produit complet et fonctionnel. Ma solution est donc prête à être exploitée en offrant aux acteurs la possibilité de gérer convenablement toutes les fonctionnalités du système.

Conclusion et perspectives

Dans le cadre de mon projet de fin d'études, j'ai conçu et développé une application web de HelpDesk permettant d'automatiser la gestion des demandes. Cette application sera un système fiable utilisé par tout le personnel de l'A.R.R.U participant dans le traitement d'une demande dès son remplissage jusqu'à la dernière signature.

En se basant sur la formation acquise durant mon cursus universitaire et sur les nouvelles notions que j'ai pu assimiler au sein de l'entreprise dans laquelle j'ai effectué mon stage, les objectifs que j'ai fixés au départ sont en grande partie atteints.

Ce projet de fin d'études m'a été très bénéfique et enrichissant puisqu'il m'a permis, d'une part, de me familiariser avec la vie professionnelle et, d'autre part, de découvrir de nouvelles approches de développement et d'utiliser des technologies récentes (Angular, Node.js, la base de données MongoDB, Postman, Overleaf, Vs Code..) ainsi que les principes de fonctionnement des HelpDesk. Mais cela n'empêche que j'ai eu quelques difficultés, vu que c'était ma première expérience avec SCRUM, j'ai consacré beaucoup de temps pour apprendre et maîtriser cette méthodologie et malgré toutes les difficultés rencontrées et les contraintes de temps, j'ai réussi à améliorer mes performances et mes connaissances de la conception jusqu'à la réalisation en pratiquant mes acquis et en dépassant toutes sortes de problèmes.

Mon travail ne s'arrête pas à ce niveau, il est loin d'être le temps d'arrêter le développement de cette application du HelpDesk, en effet la solution développée reste extensible à tout type d'amélioration.

Après avoir fini cette partie du projet, j'ai consacré encore du temps, après la période du stage, pour développer la partie mobile de cette application afin de simplifier la communication entreprise-clients. On sait tous que, de nos jours, nul ne peut se passer de son smartphone, donc cela facilitera aux employés de l'A.R.R.U de passer les demandes et tickets à travers leurs téléphones mobiles, qui leurs fera bien évidemment gagner beaucoup plus de temps.

CONCLUSION ET PERSPECTIVES

Pour conclure, à travers ce stage, j'ai eu la chance d'améliorer mes aptitudes à communiquer, à travailler en collaboration et à savoir être responsable, ce qui va me permettre de bien s'intégrer dans le milieu professionnel.

Bibliographie

- [Ang] ANGULAR. « Angular ». In : *Official Angular website*. (). URL : <https://angular.io/docs>.
- [Aub11] Claude AUBRY. « Scrum ». In : *Le guide pratique de la méthode Agile la plus populaire (2nd ed)*. Malakoff, France: Dunod Editions (2011).
- [Bas16] Arindam BASU. « How to write using rich text format and markdown in latex and overleaf ». In : *Universidad de Canterbury* (2016).
- [Bha15] Snig BHAUMIK. *Bootstrap essentials*. Packt Publishing Ltd, 2015.
- [CS16] Alain COLLIGNON et Joachim SCHÖPFEL. « Méthodologie de gestion agile d'un projet. Scrum—les principes de base ». In : *I2D Information, donnees documents* 53.2 (2016), p. 12-15.
- [DV16] Laurent DEBRAUWER et Fien VAN DER HEYDE. « UML 2.5 ». In : *Barcelona: Editions ENI*. Obtenido de <https://books.google.com.ec/books> (2016).
- [Fou12] Damien FOURES. « Transformation des diagrammes d'activités SysML1. 2 vers les réseaux de Petri dans un cadre MDE ». In : (2012).
- [Guy18] Marine GUYOT. « Méthode Agile vs Méthode Classique, quelle est la meilleure façon de gérer son projet ? » In : *marine-guyot* (2018).
- [HI12] Lamia Ben HIBA et Mohammed Abdou Janati IDRISI. « Tendances des méthodes de gestion des projets informatiques ». In : *Revu* (2012), p. 7.
- [JSO] JSON. « JSON ». In : *Official JSON website*. (). URL : <https://www.json.org/json-en.html>.
- [Mey06] Eric A MEYER. *CSS: The Definitive Guide: The Definitive Guide*. ” O'Reilly Media, Inc.”, 2006.
- [MG00] Pierre-Alain MULLER et Nathalie GAERTNER. *Modélisation objet avec UML*. T. 514. Eyrolles Paris, 2000.
- [npm] Inc NPM. « npm ». In : *Official npm, Inc website*. (). URL : <https://www.npmjs.com/about>.
- [Pos] Inc POSTMAN. « Postman ». In : *Official Postman, Inc website*. (). URL : <https://learning.postman.com/docs/getting-started/introduction/>.
- [RLJ+99] Dave RAGGETT, Arnaud LE HORS, Ian JACOBS et al. « HTML 4.01 Specification ». In : *W3C recommendation* 24 (1999).
- [Typ] TYPESCRIPT. « TypeScript ». In : *Official TypeScript website*. (). URL : <https://www.typescriptlang.org/>.

BIBLIOGRAPHIE
