

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAÏA
FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT D'INFORMATIQUE



Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel en Informatique

Option : Génie Logiciel

Thème

Conception et réalisation d'une application web pour la
gestion des emprunts et l'archivage de documents
Cas d'étude : Complexe agroalimentaire Cevital Bejaia

Réalisé par

Mlle. BOULEKBACHE Nadine

Mlle. GALOU Celina

Soutenu le 04/07/2024 devant le jury composé de :

	Nom & Prénom	Établissement
Président	Mr. MOKTEFI Mohand	Université de Bejaia
Encadrant	Mme. KHOULALENE Nadjet	Université de Bejaia
Examineur	Mr. ALLEM Khaled	Université de Bejaia
Co-Encadrant	Mr. CHIBOUTI Younes	Cevital

Année Universitaire : 2023 — 2024

Dédicaces

Je dédie cet événement marquant de ma vie :

À mon très cher père, aucune phrase ni expression, aussi éloquente soit-elle, ne saurait exprimer ma reconnaissance. Que ce travail soit le reflet de ma gratitude et de mon affection.

À ma très chère mère, la lumière de ma vie, qui a toujours été présente dans les moments les plus difficiles et qui m'a comblée d'espoir tout au long de mon parcours. Je te promets de réaliser mes plus grands rêves et de t'offrir la fierté de me voir atteindre le poste dont j'ai toujours rêvé.

À mon frère Yanis, pour sa foi inébranlable en moi.

À mes grands-parents Yahia et Malika, avec qui j'ai passé une grande partie de mon enfance et qui ont embelli chaque moment de cette période.

À mon oncle El Houas et ma tante Nassima, ma cousine Amel et mes petites cousines Sarah et Lilya, qui ont attendu ce jour avec tant d'impatience.

À ma tante Kahina et mon oncle Bouzid, ainsi qu'à mes cousines Imene et Khayra.

À mes deux merveilleuses tantes Cherifa et Rahma.

À mes amies Tayeb Cherif Zouina Sonia et Zaidi Lila, pour leur aide précieuse et leur soutien constant.

À mes meilleures amies Fairouz et Melissa, pour leur soutien indéfectible et leur amitié précieuse.

À mon kinésithérapeute, Dr Moussaoui Sofiane, que je ne remercierai jamais assez pour avoir soigné mon genou.

BOULEKBACHE Nadine

Dédicaces

Je dédie ce travail à

Mes parents, qui ont toujours été là pour moi avec leur soutien et leur amour.

Je tiens à leur exprimer ma reconnaissance pour leurs sacrifices,

leur éducation et leurs encouragements

tout au long de ces années.

Je vous dois ce que je suis aujourd'hui et ce que je serai demain et je ferai toujours de mon mieux pour rester votre fierté.

Mes frères Lyes et Salim

pour leur bienveillance, leur soutien indéfectible et leur fraternité inestimable tout au long de cette période.

Mes amis les plus proches, Selma, Sophy et Meziane avec qui j'ai partagé mes meilleures années d'études,

pour tous les bons souvenirs que nous avons pu partager ensemble.

Mes amies Sonia et Laila qui ont apporté leur aide précieuse au moment le plus sensible.

À la mémoire de mes grands-parents.

GALOU Celina

Remerciements

Nous exprimons notre gratitude à Allah le Tout-Puissant pour nous avoir accordé la force et la volonté nécessaires pour mener à bien ce projet.

Nous exprimons notre profonde gratitude à nos deux encadrants, Mme KHOULALENE Nadjat et Mr CHIBOUTI Younes, pour leurs disponibilités et leurs implications tout au long de ce projet.

Nos remerciements s'étendent également à Mr HADJAL Samir pour nous avoir aidé dans la réalisation de notre stage et à Mr OUNCER Farouk, chef de service de la direction des systèmes d'information pour son accueil tout au long de notre stage.

Nous tenons également à remercier les membres du jury d'avoir consacré une partie de leur temps à examiner ce mémoire, pour l'intérêt qu'ils ont porté à notre travail et pour leurs contributions à l'enrichir.

Nous exprimons notre gratitude envers nos parents, nos frères et nos amis.

Enfin, nous tenons à remercier toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce projet.

Table des matières

Table des figures	v
Liste des tableaux	vi
Liste des abréviations	vii
Introduction générale	1
1 Présentation de l'organisme d'accueil et recueil des besoins	3
1.1 Introduction	3
1.2 Présentation de l'organisme d'accueil	3
1.2.1 Présentation du complexe CEVITAL	3
1.2.2 Présentation de l'organigramme	4
1.2.3 Activités de l'entreprise	5
1.2.4 Les valeurs fondamentales de Cevital	6
1.3 Etude de l'existant et recueil des besoins	7
1.3.1 Analyse de l'existant	7
1.3.2 Critiques de l'existant	10
1.3.3 Solution proposée	10
1.3.4 Recueil des Besoins (Fonctionnels et non-fonctionnels)	11
1.4 Conclusion	12
2 Méthodologie de conception et Sprint de planification	13
2.1 Introduction	13
2.2 Méthodologie de développement	13
2.3 Langage de modélisation unifié	16
2.3.1 Diagramme de cas d'utilisation	17
2.3.2 Diagramme de séquence	17
2.3.3 Diagramme de classes	17
2.4 Le design Pattern Modèle-Vue-Contrôleur	17
2.5 Pilotage du projet avec la méthode Scrum	19
2.5.1 Répartition des rôles	19

2.5.2	Identification des acteurs	19
2.5.3	User-Stories	20
2.5.4	Diagramme de cas d'utilisation globale	20
2.5.5	Product Backlog	21
2.6	Charte graphique	23
2.6.1	Logotype	24
2.6.2	Palette de couleurs	24
2.6.3	Slogan	24
2.7	Langages, environnement et outils de développement	25
2.7.1	HTML CSS	25
2.7.2	JavaScript	25
2.7.3	Laravel	26
2.7.4	SQLite	26
2.7.5	HeidiSQL	26
2.7.6	Visual Studio Code	26
2.7.7	GitHub et Git	27
2.7.8	Visual Paradigm	27
2.7.9	Canva	27
2.8	Packages utilisés	27
2.8.1	Spatie/laravel-permission	27
2.8.2	Spatie/Media-Library	28
2.8.3	Admin LTE	28
2.8.4	Laravel UI	29
2.9	Conclusion	29
3	Sprint 1	30
3.1	Introduction	30
3.2	Etude du sprint 1	30
3.2.1	Espace Administrateur	30
3.2.2	Espace Archiviste	31
3.3	Diagramme de cas d'utilisation du sprint 1	31
3.4	Cas d'utilisation "S'authentifier"	32
3.4.1	Description textuelle du cas d'utilisation "s'authentifier"	32
3.4.2	Diagramme de séquence du cas d'utilisation "s'authentifier"	32
3.5	Cas d'utilisation "Ajouter compte"	33
3.5.1	Description textuelle du cas d'utilisation "Ajouter compte"	33
3.5.2	Diagramme de séquence du cas d'utilisation "Ajouter compte"	34
3.6	Cas d'utilisation "Gestion des directions"	35
3.6.1	Description textuelle des cas d'utilisation "Gestion des directions"	35

3.6.2	Diagramme de séquence du cas d'utilisation "Gestion des directions"	37
3.7	Cas d'utilisation "Ajouter département"	39
3.7.1	Description textuelle du cas d'utilisation "Ajouter département"	39
3.7.2	Diagramme de séquence du cas d'utilisation "Ajouter département"	39
3.8	Cas d'utilisation "Ajouter étagère"	40
3.8.1	Description textuelle des cas d'utilisation "Ajouter étagère"	40
3.8.2	Diagramme de séquence du cas d'utilisation "Ajouter étagère"	41
3.9	Diagramme de classe du sprint 1	42
3.10	Vue des interfaces utilisateurs du sprint1	43
3.10.1	Interface d'authentification	43
3.10.2	Interface utilisateur	44
3.10.3	Interface Ajout compte	44
3.10.4	Interface gestion des directions	45
3.10.5	Interface de gestion des départements	45
3.10.6	Interface ajout d'une étagère	46
3.11	Conclusion	46
4	Sprint 2, 3	48
4.1	Introduction	48
4.2	Etude du Sprint 2	48
4.2.1	Espace Demandeur	48
4.2.2	Diagramme de cas d'utilisation du sprint 2	48
4.3	Cas d'utilisation "Ajouter demande d'archive"	49
4.3.1	Description textuelle des cas d'utilisation "Ajouter demande d'archive"	49
4.3.2	Diagramme de séquence du cas d'utilisation "Ajouter demande d'archive"	50
4.4	Cas d'utilisation "Modifier boîte"	51
4.4.1	Description textuelle des cas d'utilisation "Modifier boîte"	51
4.4.2	Diagramme de séquence du cas d'utilisation "Modifier boîte"	52
4.5	Cas d'utilisation "Supprimer boîte"	53
4.5.1	Description textuelle des cas d'utilisation "Supprimer boîte"	53
4.5.2	Diagramme de séquence du cas d'utilisation "Supprimer boîte"	54
4.6	Cas d'utilisation "Ajouter demande emprunt"	55
4.6.1	Description textuelle des cas d'utilisation "Ajouter demande emprunt"	55
4.6.2	Diagramme de séquence du cas d'utilisation "Ajouter demande emprunt"	56
4.7	Cas d'utilisation "Envoyer demande"	57
4.7.1	Description textuelle des cas d'utilisation "Envoyer demande"	57
4.7.2	Diagramme de séquence du cas d'utilisation "Envoyer demande"	58

4.8	Diagramme de classe du sprint 2	59
4.9	Vue des interfaces utilisateurs du sprint 2	60
4.9.1	Interface ajout demande d'archive	60
4.9.2	Interface boite d'archive	60
4.9.3	Interface gestion des demandes d'archives	61
4.9.4	Interface ajout d'une demande d'emprunt	62
4.10	Etude de Sprint 3	62
4.10.1	Espace Archiviste	62
4.11	Diagramme de cas d'utilisation du sprint 3	63
4.12	Cas d'utilisation "Traitement des demandes"	63
4.12.1	Description textuelle du cas d'utilisation "Traitement demande d'archive"	63
4.12.2	Description textuelle du cas d'utilisation "Traitement demande d'emprunt"	64
4.13	Diagramme de séquence du cas d'utilisation "Traitement des demandes"	66
4.14	Diagramme de classe du sprint 3	68
4.15	Vue des interfaces utilisateurs du sprint 3	69
4.15.1	Interface traitement d'une demande d'archive	69
4.15.2	Interface traitement d'une demande d'emprunt type copie	69
4.16	Conclusion	70
5	Sprint 4, 5	71
5.1	Introduction	71
5.2	Etude du Sprint 4	71
5.2.1	Espace Demandeur	71
5.2.2	Espace Archiviste	71
5.2.3	Diagramme de cas d'utilisation du sprint 4 et 5	72
5.2.4	Cas d'utilisation "Attribuer emplacement boite"	72
5.2.5	Cas d'utilisation "Modifier emplacement"	74
5.3	Diagramme de classe du sprint 4 et 5	76
5.3.1	Vues des interfaces utilisateurs	77
5.4	Etude du Sprint 5	79
5.4.1	Espace Archiviste	79
5.4.2	Cas d'utilisation "Prêter boite"	80
5.4.3	Cas d'utilisation "Détruire boite"	82
5.4.4	Vues des interfaces utilisateurs du sprint 5	84
5.5	Conclusion	87

6	Conception de la Base de données	88
6.1	Introduction	88
6.2	Règles de gestion	88
6.3	Diagramme de classes	89
6.4	Schéma relationnel	89
6.4.1	Les règles de passage du diagramme de classe vers le modèle relationnel	89
6.5	Modèle relationnel	91
6.6	Conclusion	91
	Conclusion générale	92
	Bibliographie	93

Table des figures

1.1	Organigramme du complexe agroalimentaire CEVITAL	4
1.2	Organigramme direction système d'information	5
1.3	Quelques activités de l'entreprise Cevital	6
1.4	Interface d'authentification	7
1.5	Interface ajouter boite	8
1.6	Interface boites empruntées	8
1.7	Interface des emplacement	9
1.8	Interface des familles	9
2.1	Les principaux composants de la méthode scrum [22]	14
2.2	L'architecture MVC [23]	18
2.3	Diagramme de cas d'utilisation global	21
2.4	Logo de notre application	24
2.5	Palette de couleurs de l'application BoxFlow en hexadécimal	24
2.6	Slogan de l'application BoxFlow	25
3.1	Diagramme de cas d'utilisation du sprint 1	31
3.2	Diagramme de séquence de cas d'utilisation "S'authentifier"	33
3.3	Diagramme de séquence du cas d'utilisation "Ajouter compte"	35
3.4	Diagramme de séquence de cas d'utilisation "Gestion des directions"	38
3.5	Diagramme de séquence de cas d'utilisation "Ajouter département"	40
3.6	Diagramme de séquence du cas d'utilisation "Ajouter étagère"	42
3.7	Diagramme de classe sprint 1	43
3.8	Interface d'authentification	44
3.9	Interface dashboard demandeur	44
3.10	Interface ajouter compte	45
3.11	Interface gestion des directions	45
3.12	Interface gestion des départements	46
3.13	Interface ajouter étagère	46
4.1	Diagramme de cas d'utilisation du sprint 2	49
4.2	Diagramme de séquence du cas d'utilisation "Ajouter demande d'archive"	51

4.3	Diagramme de séquence du cas d'utilisation "Modifier boîte"	53
4.4	Diagramme de séquence du cas d'utilisation "Supprimer boîte"	55
4.5	Diagramme de séquence du cas d'utilisation "Ajouter demande emprunt"	57
4.6	Diagramme de séquence du cas d'utilisation "Envoyer demande"	59
4.7	Diagramme de classe sprint 2	59
4.8	Interface ajout demande d'archive	60
4.9	Interface d'ajout de boîte d'archive	60
4.10	Interface de gestion des boîtes d'archives à l'intérieur d'une demande	61
4.11	Interfaces des demandes d'archives	61
4.12	Interface ajout d'une demande d'emprunt	62
4.13	Diagramme de cas d'utilisation du sprint 3	63
4.14	Diagramme de séquence du cas d'utilisation "Traitement des demandes"	67
4.15	Diagramme de classe sprint 3	68
4.16	Interface traitement d'une demande d'archive	69
4.17	Interface traitement d'une demande d'emprunt type copie	70
5.1	Diagramme de cas d'utilisation du sprint 4 et 5	72
5.2	Diagramme de séquence du cas d'utilisation "Attribuer emplacement boîte"	74
5.3	Diagramme de séquence du cas d'utilisation "Modifier emplacement"	76
5.4	Diagramme de classe sprint 4 et 5	77
5.5	Interface Consulter une demande d'emprunt traitée	78
5.6	Interface attribuer emplacement boîte	78
5.7	Interface modifier emplacement boîte	79
5.8	Diagramme de séquence du cas d'utilisation "Prêter boîte"	82
5.9	Diagramme de séquence du cas d'utilisation "Détruire boîte"	84
5.10	Interface de toutes les boîtes	85
5.11	Interface prêter boîte	85
5.12	Interface boîtes empruntées	86
5.13	Interface mouvement de la boîte	86
5.14	Interface planifier la destruction d'une boîte	87
5.15	Interface des boîtes détruites	87
6.1	Diagramme de classes	89

Liste des tableaux

2.1	Répartition des rôles	19
2.2	User-Stories	20
2.3	Product Backlog	21
3.1	Description textuelle du cas d'utilisation "S'authentifier"	32
3.2	Description textuelle du cas d'utilisation "Ajouter compte"	34
3.3	Description textuelle de cas d'utilisation "Ajouter direction"	36
3.4	Descreption textuelle du cas d'utilisation "Modifier direction"	36
3.5	Descreption textuelle du cas d'utilisation "Supprimer direction"	37
3.6	Descreption textuelle du cas d'utilisation "Ajouter département"	39
3.7	Descreption textuelle du cas d'utilisation "Ajouter étagère"	41
4.1	Description textuelle des cas d'utilisation "Ajouter demande d'archive"	50
4.2	Description textuelle des cas d'utilisation "Modifier boîte"	52
4.3	Description textuelle des cas d'utilisation "Supprimer boîte"	54
4.4	Description textuelle des cas d'utilisation "Ajouter demande d'emprunt"	56
4.5	Description textuelle des cas d'utilisation "Envoyer demande"	58
4.6	Description textuelle des cas d'utilisation "Traitement demande d'archive"	64
4.7	Description textuelle des cas d'utilisation "Traitement demande d'emprunt"	64
5.1	Description textuelle des cas d'utilisation "Attribuer emplacement boîte"	73
5.2	Description textuelle des cas d'utilisation "Modifier emplacement"	75
5.3	Description textuelle des cas d'utilisation "Prêter boîte"	81
5.4	Description textuelle des cas d'utilisation "Détruire boîte"	83

Liste des abréviations

API	Application Programming Interface
BLL	Business Logic Layer
CSS	Cascading Style Sheets
DAL	Data Access Layer
DFC	Direction Finances Comptabilité
DG	Direction Générale
DRH	Direction Ressources Humaine
DSI	Direction Système d'Information
GUI	Graphical User Interface
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JS	Java Script
MVC	Model-View-Controller
ORM	Object Relational Mapping
PHP	Hypertext Preprocessor
POO	Programmation Orienté Objet
SQL	Structured Query Language
UI	User Interface
UML	Unified Modeling Language
XML	Extensible Markup Language
XP	Extreme Programming

Introduction Générale

Cevital, en tant qu'une des plus grandes entreprises en Afrique dans le domaine agroalimentaire, a dû faire face à un défi majeur lié à la gestion de ses documents. Avec ses nombreuses structures générant une variété de documents tels que les bons de commande, les factures et d'autres type de documents, il est devenu impératif de mettre en place un système d'archivage efficace. Cependant, la gestion manuelle de ces archives présentait de nombreuses difficultés, notamment en termes de stockage, d'organisation et d'accès rapide à l'information.

Pour surmonter ces obstacles, Cevital a opté pour une solution informatique et a mis en œuvre le logiciel GEODE en 2007. Ce système informatisé a permis à l'entreprise de gérer ses archives.

Cependant, en raison des progrès de l'informatique et du fait que GEODE soit une application de bureau avec de fonctionnalités limitées et dont la base de données a été perdue, l'accès lent et l'utilisation de ce type d'application ont été constatés. Par conséquent, une version améliorée de l'application, dotée de fonctionnalités plus avancées, a été recommandée. C'est ainsi que notre projet « Conception et réalisation d'une application web pour la gestion des emprunts et l'archivage de documents au sein de Cevital » a été initié, mettant l'accent sur l'optimisation de l'interaction entre le demandeur et l'archiviste pour la gestion des emprunts et de l'archivage efficace des boîtes d'archives.

Notre objectif est d'optimiser la gestion de cette tâche en rendant ces demandes d'archivage et d'emprunt de boîtes d'archives possible via l'application, tout en assurant une gestion efficace des emplacements de ces boîtes dans les différentes salles d'archivage de Cevital, à savoir les salles de Bejaia, d'El Kseur et de Lala Khedidja (Tizi Ouzou).

Pour accomplir ce travail, nous allons structurer ce mémoire en (06) chapitres :

Le premier chapitre, intitulé « Présentation de l'organisme d'accueil et recueil des besoins », a pour rôle d'introduire le projet. Nous y présentons l'organisme d'accueil,

sa structure et ses missions, tout en fournissant une description détaillée de l'étude de l'existant et une critique adaptée à cette analyse. Ce chapitre aborde également une solution aux problématiques posées et le recueil des besoins pour notre futur système.

Le deuxième chapitre, intitulé « Méthodologie de conception et sprint de planification », sera consacré à sur la présentation de la méthode Scrum, du langage de modélisation UML, ainsi que l'architecture MVC de notre système. Nous y identifions également les besoins du système, et déterminons les acteurs impliqués ainsi que les cas d'utilisations associés. Nous établissons également le diagramme de cas d'utilisations global, accompagné des users stories et le carnet de produit (Product backlog). Enfin, nous présenterons les outils de développement que nous utiliserons pour mener à bien le projet ainsi que la charte graphique qui aidera la conception visuelle de notre application.

Dans le troisième chapitre intitulé, « Sprint1 », nous mettrons l'accent sur les fonctionnalités clés de notre application tel que l'authentification et l'ajout des comptes utilisateurs. Ce sprint initial visera à établir les bases de notre application en assurant un accès sécurisé aux utilisateurs. Nous accorderons également une attention particulière à la gestion des directions et des départements de Cevital, ainsi qu'à la gestion des emplacements en développant les fonctionnalités nécessaires à leurs création, modification et suppression.

Dans le quatrième chapitre, intitulé « Sprint 2 et 3 », nous présenterons l'ajout de fonctionnalités essentielles répondant aux besoins des utilisateurs. Parmi ces fonctionnalités nous inclurons la gestion des demandes d'archivage et d'emprunt, leurs envois ainsi que leurs traitements.

Le cinquième chapitre, intitulé « Sprint 4 et 5 », sera consacré aux deux derniers sprints de développement de notre application. Cette phase finale du projet se concentrera sur l'ajout des fonctionnalités piliers de l'application tel que l'attribution des emplacements aux boîtes d'archive ainsi le processus d'emprunt des boîtes, et le processus de destruction des boites.

Le sixième chapitre et dernier chapitre « Conception de la base de données » est consacré à la présentation des règles de gestion, du diagramme de classes et du schéma relationnel de notre application.

En conclusion, nous dressons une synthèse générale du travail élaboré et nous proposons quelques perspectives d'évolution.

Chapitre 1

Présentation de l'organisme d'accueil et recueil des besoins

1.1 Introduction

Dans ce premier chapitre, nous introduirons l'entreprise d'accueil, "Cevital", en mettant en lumière ses valeurs fondamentales, ses activités clés ainsi que la structure organisationnelle de ses différentes composantes, où on a eu l'occasion d'effectuer notre stage, notamment dans le département juridique, plus précisément dans le service d'archivage. Nous explorerons également les missions et services des différentes directions, en examinant leur organigramme et leurs rôles respectifs. Ensuite nous procéderons à l'étude de l'existant et à l'analyse de l'état actuel, en critiquant le processus en place et en proposant des solutions potentielles à explorer dans le futur système. Enfin nous identifieront les besoins fonctionnels et non fonctionnels de l'application. Il est primordial de comprendre pleinement le système existant afin de formuler une évaluation précise et obtenir toutes les données nécessaires pour concevoir une solution adaptée aux besoins des utilisateurs.

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation du complexe CEVITAL

L'entreprise Cevital constitue avec certains autres groupes privés (Ifri, Hamoud Boualem, etc.) un modèle de réussite industrielle dont la notoriété dépasse les frontières nationales. C'est une entreprise familiale qui a connu, depuis sa création en avril 1998, un développement important dans plusieurs secteurs. Son complexe agroalimentaire comporte aujourd'hui plusieurs unités de production [24].

1.2.2 Présentation de l'organigramme

La figure 1.1 représente l'organigramme du complexe CEVITAL.

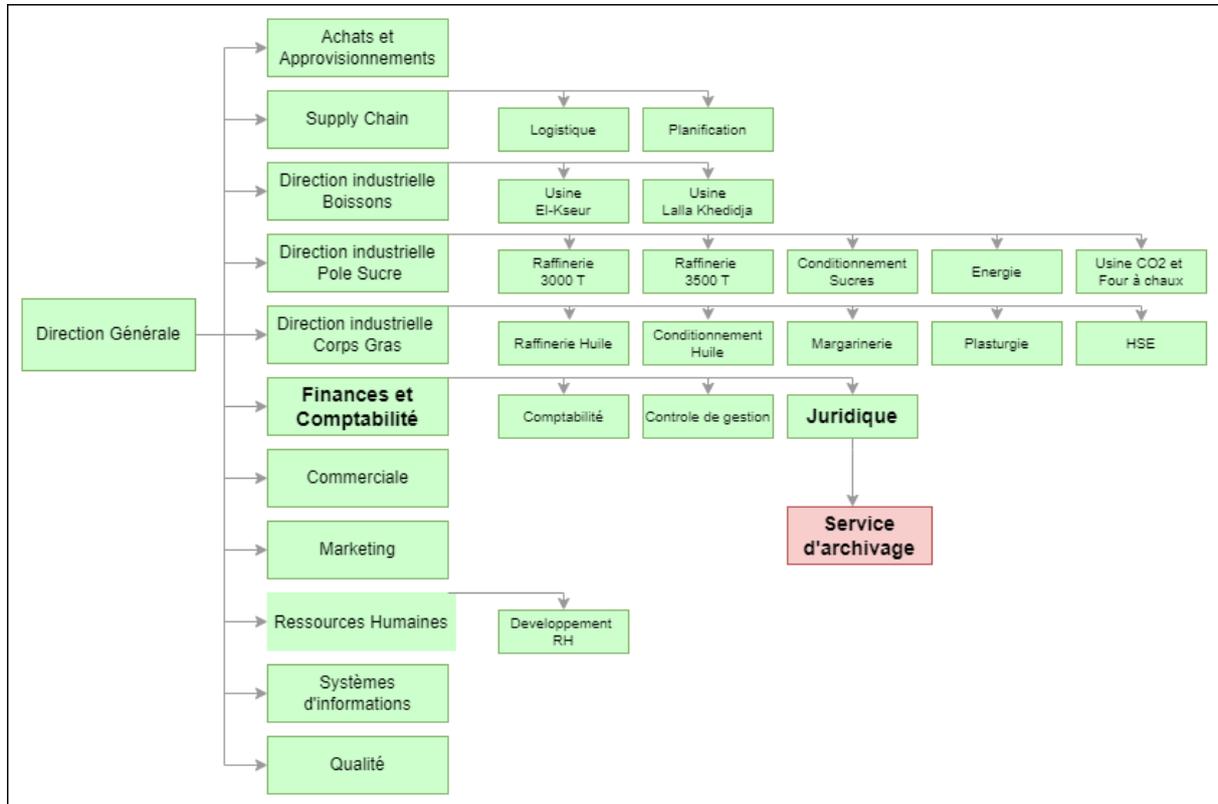


FIGURE 1.1 – Organigramme du complexe agroalimentaire CEVITAL

L'effectif actuel de " Cevital " est reparti entre les différentes directions et unités de production :

- **Direction Générale (DG)** : Chargée de veiller sur le bien être humain et matériel du complexe, de la maintenance et de coordonner les différentes directions.
- **Direction des Finances et Comptabilité(DFC)** : Cette direction est responsable de la préparation et de la mise à jour des budgets, de la tenue de la comptabilité, de la préparation des états comptables et financiers selon les normes en vigueur, ainsi que du contrôle de gestion. Elle comprend également un département juridique, avec un service d'archivage géré par des archivistes, chargés d'archiver les boîtes d'archives. Les boîtes d'archive sont stockés dans des salles, qui sont des locaux dévisés en rayons. Chaque rayon est équipé d'armoires, appelées colonnes, et chaque colonne peut contenir plusieurs étagères. Chaque étagère peut accueillir de 10 à 20 boîtes d'archives. Une boîte d'archive contient environ une centaine de documents générés par les différentes directions de cevital.
- **Direction Commerciale** : Ce service s'occupe de : l'élaboration de la politique de distribution, des stratégies face à la concurrence et gérer les campagnes de

promotions.

- **Direction Logistique** : Assure le transport de tous les produits finis, que ce soit en moyens propres (camions de CEVITAL), affrétés ou moyens de transport des clients. Et gère les stocks de produits finis dans les différents dépôts locaux (Bejaia et environs) et Régionaux (Alger, Oran, Sétif).
- **Direction Contrôle de qualité** : Elle est dotée de quatre laboratoires qui assurent le suivi permanent et continu du processus de production sous la supervision du laboratoire central qui suit la qualité microbiologique des produits.
- **Direction des Ressources Humaines (DRH)** : Elle prend en charge : la gestion de la carrière professionnelle des employés, le recrutement des effectifs, la formation et l'évolution du personnel, le règlement des salaires des employés et les relations humaines et les dossiers de la sécurité sociale des employés.
- **Direction marketing** : Elle gère la stratégie de marketing, la promotion des produits, l'analyse des marchés, la gestion de l'image de marque, et la communication avec les clients et partenaires pour renforcer la position de l'entreprise sur ses marchés cibles.
- **Direction Système Informatique** : Son rôle est d'informatiser le système administratif et la gestion des ateliers de production.

La figure 1.2 représente l'organigramme de la direction système d'information de cevital.

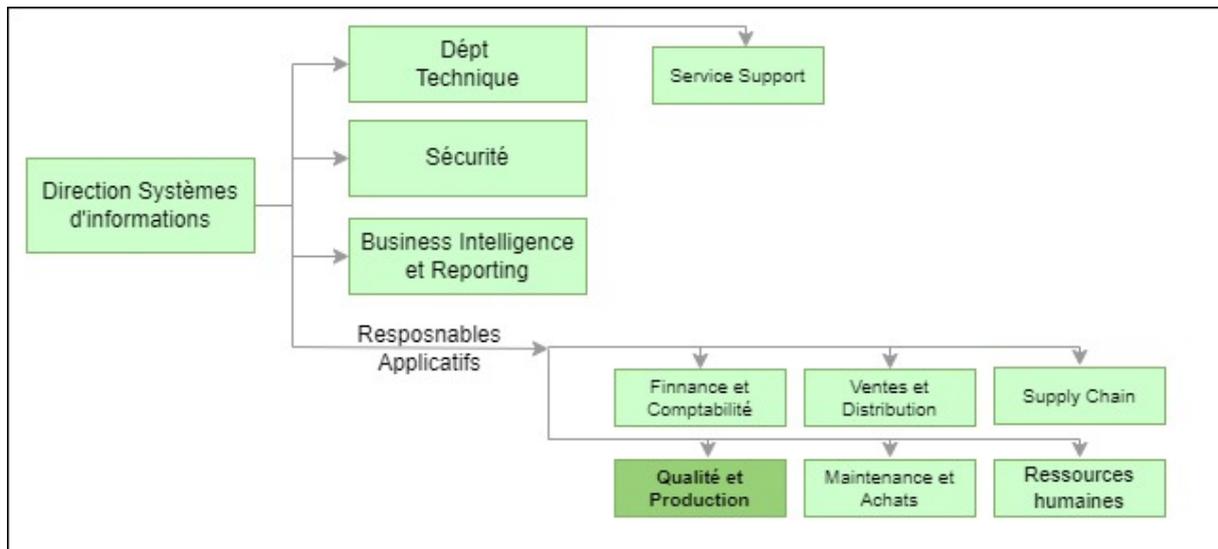


FIGURE 1.2 – Organigramme direction système d'information

1.2.3 Activités de l'entreprise

Le complexe agro-alimentaire est composé de plusieurs unités de production telles que :

- **Huiles végétales** : Unité de production qui fabrique des huiles végétales à partir de différentes matières premières telles que le soja, le tournesol, le colza.
- **Margarinerie et graisses végétales** : Unité de production qui fabrique des graisses végétales et des margarines pour une utilisation dans la cuisine et pâtisserie.
- **Sucre blanc** : Unité de production qui fabrique du sucre à partir de matières premières telles que la canne à sucre ou la betterave sucrière.
- **Sucre liquide** : Unité de production qui produit du sucre sous forme liquide à partir de la canne à sucre ou de la betterave sucrière.
- **Silos portuaires** : Les silos portuaires sont des installations de stockage de grande capacité situées dans les ports maritimes.
- **Boissons** : Unité de production qui est responsable de la fabrication de boissons non alcoolisées, telles que les boissons gazeuses, les jus de fruits, l'eau minérale, etc. La figure 1.3 représente les activités de l'entreprise cevital.



FIGURE 1.3 – Quelques activités de l'entreprise Cevital

1.2.4 Les valeurs fondamentales de Cevital

Le succès du groupe Cevital repose sur quatre valeurs qui constituent le socle de la culture d'entreprise, souvent résumées par l'acronyme I.R.I.S. :

- **Initiative** : Il s'agit de favoriser la veille proactive et l'initiative individuelle et collective.
- **Respect** : Il s'agit de respecter les personnes, les règles et l'environnement, ainsi que de traiter les clients, les partenaires commerciaux et institutionnels du Groupe avec respect.
- **Intégrité** : Il s'agit d'adopter une éthique professionnelle irréprochable, de rejeter toute forme de corruption et d'agir avec une haute rigueur intellectuelle et morale.
- **Solidarité** : Il s'agit d'assurer la solidarité entre les collaborateurs, envers le Groupe et avec les communautés environnantes.

1.3 Etude de l'existant et recueil des besoins

1.3.1 Analyse de l'existant

Le système existant est une application de bureau conçue pour l'archivage de divers documents produits par l'entreprise, tels que les factures, les bons de commande ou les bons de livraison. Avant l'installation de l'application GEODE, créée par une société française et introduite chez Cevital en 2007, l'entreprise ne disposait d'aucune composante informatique pour la gestion des archives.

En ce qui concerne la gestion physique des archives, les boîtes d'archives sont stockées dans des salles, qui sont des locaux divisés en rayons. Chaque rayon est équipé d'armoires, appelées colonnes, et chaque colonne peut contenir plusieurs étagères. Chaque étagère peut accueillir de 10 à 20 boîtes d'archives. Une boîte d'archives contient environ une centaine de documents générés par les différentes directions de Cevital. Notamment, 80% des documents proviennent des directions finances et comptabilité (DFC) et direction commercial, comprenant des documents originaux, tandis que 10% proviennent de directions des ressources humaines (DRH) et transit, ces derniers transitant obligatoirement par la DFC. Les 10% restants émanent des autres directions de l'entreprise.

Les captures d'écran suivantes représentent l'application GEODE déjà existante :

Les deux figures 1.4, 1.5 représentent respectivement l'interface d'authentification et l'interface d'ajout d'une boîte.

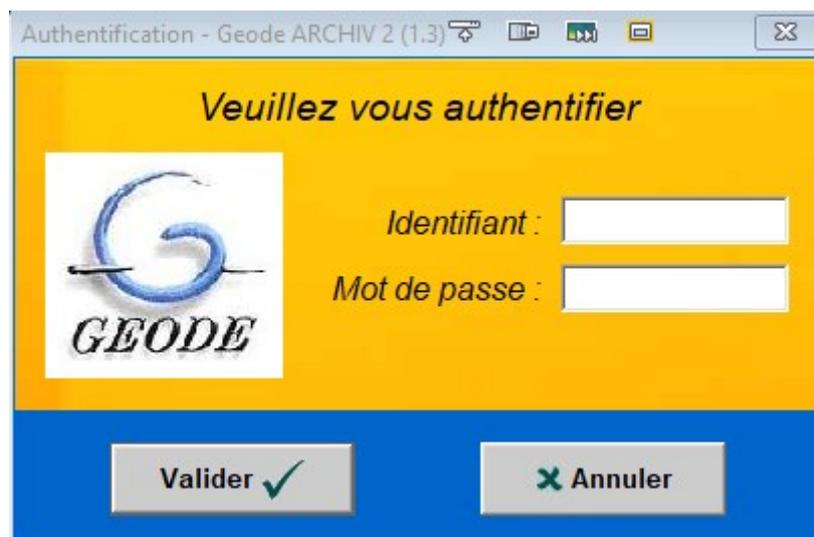


FIGURE 1.4 – Interface d'authentification

FIGURE 1.5 – Interface ajouter boîte

La figure 1.6, représente l'interface des boîtes empruntées.

N° de boîte	Famille	Sous-Famille	Contenu	Dates extrêmes	Destruction	Emplacement	Disponible
003312	Direction Transit	Sce transit	Dossiers transit Réf: (191 à 210)	2002	Non défini	BÉ JAI A 2 - R 06 - C 3 - E9	
003281	Direction Transit	Sce transit	Dossiers Matériels / N° du répertoire : 281 - 304.	2005	Prévue en : 2010	BÉ JAI A 2 - R 06 - C 2 - E9	
003334	Direction Transit	Sce transit	Dossiers tracteurs et remorques - M/V : Joliette.	2003	Non défini :	BÉ JAI A 2 - R 06 - C 3 - E5	
003405	Direction Commerci	Sce Facturation	Factures Clients / Kim Slimane / oct - nov - déc .	2006	Prévue en : 2016	LLK 2 - R1 - C 9 - E1	
003416	Direction Commerci	Sce Facturation	Factures Clients / Kim Slimane / avr - mai - juin - juill .	2006	Prévue en : 2016	LLK 2 - R1 - C 9 - E3	
003418	Direction Commerci	Sce Facturation	Factures Clients / Kim Slimane / jan - fév - mars .	2006	Prévue en : 2016	LLK 2 - R1 - C 9 - E3	
003420	Direction Commerci	Sce Facturation	Factures Clients / Kim Slimane / juill - août - sept.	2006	Prévue en : 2016	LLK 2 - R1 - C 9 - E3	
003459	Direction Commerci	Sce Facturation	Factures Clients / Djelabet Houari / août - sept - oct - nov .	2006	Prévue en : 2016	LLK 2 - R1 - C 9 - E7	
003514	Direction Commerci	Sce Facturation	Factures Clients / Khoul Md Lamine / jan à juill .	2006	Prévue en : 2016	LLK 2 - R1 - C10 - E7	
003558	Direction Commerci	Sce Facturation	Factures Clients / Djelabet Houari / jan - juill .	2006	Prévue en : 2016	LLK 2 - R1 - C11 - E4	
004101	Direction Transit	Sce transit	Dossiers transit Réf: (103 à 121).	2002	Non défini :		
004238	Direction Commerci	Sce Facturation	Factures : Boucif benouis (juin).	2007	Prévue en : 2017	LLK 2 - R2 - C11 - E2	
004241	Direction Commerci	Sce Facturation	Factures : Boucif benouis (janv-mai).	2007	Prévue en : 2017	LLK 2 - R2 - C11 - E3	
004668	Direction Commerci	Sce Clients	Dossiers clients : Sati / Eurls	2006	Illimée	LLK 2 - R3 - C 7 - E1	
004669	Direction Commerci	Sce Clients	Dossiers clients : (A.B.C.D.F.G.H)	2012	Illimée	LLK 2 - R3 - C 7 - E1	
004670	Direction Commerci	Sce Clients	Dossiers clients : (K.L.M.N.O.R.S.T.Y.Z)	2012	Illimée	LLK 2 - R3 - C 7 - E1	
004978	Direction Commerci	Sce Facturation	Factures : SARL SIMCOOPAL (Jan à juin)	2007	Prévue en : 2017	LLK 2 - R3 - C11 - E6	
005010	Direction RH	ADM	Dossiers départs complexe cevital: Ougana a/hafid - Ouzemine f	2005	Illimée	BÉ JAI A 2 - R 05 - C1 - E2	
005013	Direction RH	ADM	Dossiers départs Projet: Idoughi Youcef- Kenani Boubekeur- Khe	2006	Illimée	BÉ JAI A 2 - R 05 - C1 - E3	
005014	Direction RH	ADM	Dossiers départ projet larbaa: Aliane Mourad - Allat Mohamed -	2006	Illimée	BÉ JAI A 2 - R 05 - C1 - E3	
005022	Direction RH	ADM	Dossiers départs complexe cevital : Allouche ouahem toufik - Allc	2007	Illimée	BÉ JAI A 2 - R 05 - C1 - E4	
005027	Direction RH	ADM	Dossiers départs cevital alger: Alloua Rachid - Ben Amrouche O	2005 - 2007	Illimée	BÉ JAI A 2 - R 05 - C1 - E4	
005035	Direction RH	ADM	Dossiers départs Logistique : Medjid Ahmed - Mammeri Mahmoud	2007	Illimée	BÉ JAI A 2 - R 05 - C1 - E5	
005037	Direction RH	ADM	Dossiers départs Logistique : Abbas Nacer - Achari Bahi - Azega	2007	Illimée	BÉ JAI A 2 - R 05 - C1 - E5	

FIGURE 1.6 – Interface boites empruntées

Les deux figures 1.7, 1.8 représentent respectivement l'interface des emplacements et l'interface des familles de cevital (directions).

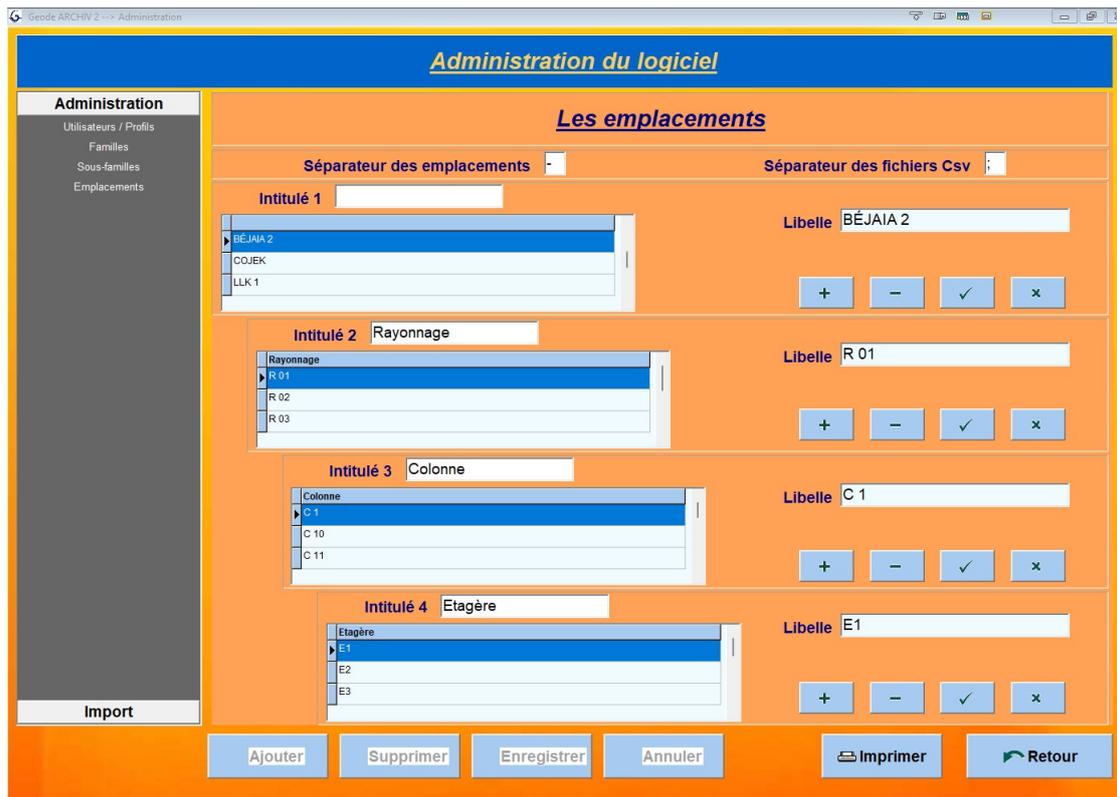


FIGURE 1.7 – Interface des emplacement



FIGURE 1.8 – Interface des familles

1.3.2 Critiques de l'existant

La critique de l'existant implique une analyse détaillée des avantages et des défauts du système en place, visant à repérer les lacunes et les inefficacités rencontrés. En scrutant les processus, les fonctionnalités et l'ergonomie de l'application, on peut identifier les aspects nécessitant des améliorations. Cette étape est cruciale dans le développement de projets informatiques, car elle permet d'établir les fondations d'une conception plus performante et mieux adaptée aux besoins des utilisateurs. Cette étude nous a permis de constater que l'application utilisée actuellement présente plusieurs insuffisances, dont :

- a) La sensibilité à la casse lors de la saisie du contenu pour la recherche de boîtes d'archives, ainsi qu'une absence de déduplication lors de l'ajout de familles existantes avec le même nom dans le système.
- b) Mauvaise gestion des emplacements car la capacité des étagères n'est pas prise en considération, ce qui oblige les archivistes à se déplacer physiquement dans le local pour trouver un emplacement vide, réduisant ainsi l'efficacité de l'application.
- c) En cas de dépassement du délai de restitution d'une boîte empruntée, aucune alerte n'est envoyée à l'archiviste.
- d) Incohérence entre le nombre total de boîtes empruntées et le nombre de boîtes non restituées (en retard) ce qui signifie une mauvaise gestion des statistiques.
- e) Plusieurs champs vides dans les tables, c'est-à-dire absence de vérification des champs lors d'ajout des boîtes.
- f) L'application présente des performances lentes, ce qui entraîne des délais importants lors du chargement de l'interface ou lors de l'ajout ou du transfert de données.
- g) La modification de l'emplacement n'est pas possible. Pour changer l'emplacement, l'archiviste doit supprimer la boîte et la recréer avec un nouvel emplacement.

1.3.3 Solution proposée

Tous les insuffisances mentionnées ci-dessus sont des inconvénients à ne pas négliger pour une entreprise qui souhaite prospérer. De ce fait, il est important de suggérer des solutions à ces problèmes pour assurer sa prospérité, notre solution consiste en un système logiciel, sous forme d'une application WEB qui doit assurer :

- a) L'amélioration de la fonction de recherche de l'emplacement des boîtes en implémentant une recherche multicritère.
- b) La cohérence de la synthèse des activités d'emprunt.
- c) Intégration d'un processus de gestion des demandes d'archivage de boîtes d'archive et de leur emprunt à travers l'application, réduisant ainsi l'interaction physique et les formulaires papiers entre l'archiviste et le demandeur

- d) Archivage des boites en tenant compte de la capacité réelle des étagères (toutes les étagères ont une capacité de 10 boites et la dernière étagère a une capacité de 20 boites).
- e) Libération automatique de l'emplacement lors de la destruction des boites.
- f) Possibilité d'archiver une boite dont le contenu est des documents numériques.
- g) La dématérialisation des archives implique la conversion des documents physiques en format numérique, ce qui facilite leur gestion, leur stockage dans des systèmes informatisés de gestion documentaire, ainsi que leur accès.

1.3.4 Recueil des Besoins (Fonctionnels et non-fonctionnels)

Dans cette section, nous aborderons les diverses exigences auxquelles l'application doit répondre.

Nous identifions deux catégories de besoins : les besoins fonctionnels et les besoins non-fonctionnels, également appelés besoins techniques.

1.3.4.1 Besoins fonctionnels

Les besoins fonctionnels du futur système sont :

- Gestion des utilisateurs par l'administrateur.
- L'authentification : accès sécurisé par un email et un mot de passe et un rôle (archiviste, administrateur, demandeur).
- La gestion des boites d'archives (archiver des boites, modifier l'emplacement des boites et destruction des boite).
- La gestion des boites empruntées et restituées.
- Le contrôle des boites en attente de destruction et des boites détruites.
- La recherche d'une boite d'archive.
- La gestion des familles et sous-familles des boites, où les familles représentent les directions de l'entreprise CEVITAL et les sous-familles désignent les départements associés à chaque direction.
- La gestion des emplacements des boites par salle, rayon, colonne et étagère.
- L'export des boîtes d'archives vers les formats Excel, PDF et CSV avec possibilité d'impression doit être pris en charge par notre système.
- Traitement des demandes d'archivage envoyées par le demandeur via des notifications.
- Possibilité de joindre des fichiers numérisés aux boites de la demande d'archive, et possibilité de les télécharger.

- Traitement des demandes d'emprunt des documents originaux et copies envoyées par le demandeur via des notifications.
- Mise à jour de la disponibilité des boites d'archive après un emprunt.
- Possibilité de joindre des fichiers numérisés lors du traitement des demandes d'emprunt de type copie, et possibilité de les télécharger.
- Prise en compte des capacités d'archivage des étagères.

1.3.4.2 Besoins non fonctionnels

En ce qui concerne les besoins non fonctionnels de notre futur système qui définissent les aspects techniques et les contraintes qui ne sont pas directement liées aux fonctionnalités spécifiques du système, nous pouvons citer :

- **La sécurité** : Protéger l'accès à l'application par l'authentification.
- L'interface de l'application doit être conviviale et facile à utiliser.
- **Performance** : Assurer des temps de réponse rapides lors de la recherche et de la manipulation des données d'archives.

1.4 Conclusion

Dans ce chapitre introductif sur le complexe agroalimentaire "Cevital", nous avons exploré en profondeur son fonctionnement, ses valeurs et sa structure organisationnelle. Nous avons analysé ses activités clés et les missions de ses différentes directions notamment la direction des finances et de la comptabilité où se trouve le service d'archivage, où nous avons effectué notre stage, afin de comprendre précisément son mode de fonctionnement. En évaluant de manière critique le système existant et en identifiant ses lacunes, nous avons posé les bases d'une solution adaptée et efficace. En recueillant les besoins fonctionnels et non fonctionnels, nous avons tracé la voie vers le développement d'une solution répondant aux besoins de Cevital, marquant ainsi le début d'un processus itératif vers une solution concrète et performante.

Dans le chapitre suivant, nous aborderons la méthodologie agile utilisée lors de la conception de notre solution, le pilotage du projet avec la méthode Scrum, ainsi que les différents outils utilisés lors de son développement.

Chapitre 2

Méthodologie de conception et Sprint de planification

2.1 Introduction

Dans ce chapitre, nous présenterons la méthodologie de conception et les phases de planification à l'aide de la méthode SCRUM. Nous commencerons par définir la méthode agile Scrum et ses principaux composants. Ensuite, nous aborderons l'utilisation du langage de modélisation UML, pour représenter les différentes composantes de notre système, suivie d'une présentation de l'architecture MVC utilisée dans notre application. Par la suite, nous examinerons en détail le fonctionnement de notre équipe Scrum, clarifiant les rôles et les responsabilités de chaque membre pour assurer une collaboration efficace et une répartition optimales des tâches. Enfin nous définirons la charte graphique de notre application ainsi que les outils et les lanages utilisés lors du développement de notre application Web.

2.2 Méthodologie de développement

Dans la gestion de projet informatique, on distingue deux approches très importantes et différentes l'une de l'autre [1] :

- Les méthodes dites traditionnelles.
- Les méthodes dites agiles.

Pour faire simple, dans les méthodes dites traditionnelles, le projet suit un processus séquentiel. Le client fournira un cahier des charges qui détaille le projet qu'il souhaite mettre en place. Après plusieurs échanges avec le client, pour être certain que le besoin est clair, le prestataire va pouvoir intervenir, réaliser le projet et le livrer au client à la fin. Et si le besoin du client a changé pendant cette phase de réalisation du projet, c'est compliqué de faire machine arrière : le prestataire s'engage sur un périmètre défini (une

liste de fonctionnalités bien détaillée), c'est donc plus difficile quand il s'agit d'enlever une brique complète ou bien de la refaire autrement.

Pour les méthodes dites **agiles**, on va au contraire se trouver dans un principe de projet itératif, c'est-à-dire qu'on va construire l'application au fur et à mesure et créer des livraisons intermédiaires. Pour ce genre de méthode, il faut évidemment que le client soit bien présent. En travaillant en parallèle avec le prestataire, il aura beaucoup plus de libertés s'il souhaite changer une fonctionnalité ou demander une évolution non pensée au début du projet. L'idée globale de l'agilité sur un projet informatique c'est de développer de manière itérative un produit, en créant des lots qui sont fonctionnels et qui ont un intérêt [1].

Les méthodes agiles populaires comprennent Kanban [1], Extreme Programming (XP) [1] et SCRUM que nous avons choisi pour la gestion de notre projet.

- **Méthode Scrum** : Scrum est une méthode de développement agile itérative et incrémentale qui met l'accent sur la gestion de projets et la gestion d'équipes. Elle est adaptée pour les projet de moyenne à longue durée comme c'est le cas de notre projet d'où notre choix pour cette méthode. Scrum fournit un cadre de gestions de projet avec des rôles, des réunions, des artefacts, des règles de gestion et un cycle itératif de développement. Le cadre de travail Scrum est simple, transparent et pragmatique. Pour résumer simplement, on définit le contenu d'une itération (ou « sprint scrum ») en termes de fonctionnalités, elles sont développées, puis validées à l'issue du sprint. Un bilan du sprint écoulé est réalisé avant de passer au sprint suivant [2].

La figure 2.1 représente les principaux composants de la méthode scrum.

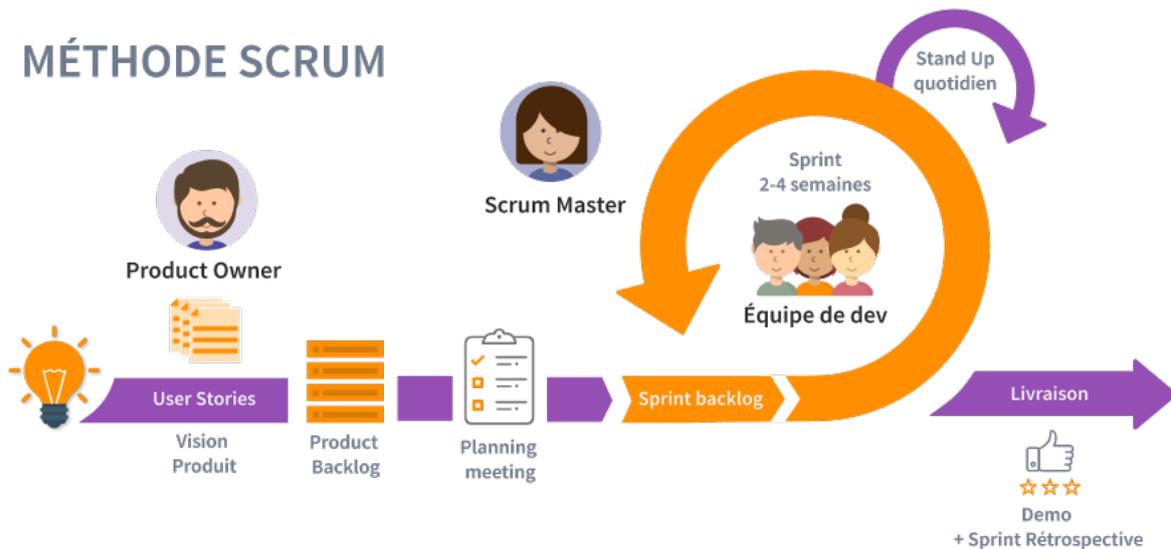


FIGURE 2.1 – Les principaux composants de la méthode scrum [22]

- 1) **Sprint** : est une période de temps fixe, généralement de 1 à 4 semaines,

pendant laquelle l'équipe de développement travaille à la réalisation d'un ensemble de fonctionnalités sélectionnées à partir du backlog produit. Les fonctionnalités à réaliser sont définies lors de la planification du sprint, et l'équipe s'engage à les compléter dans la durée du sprint. À la fin du sprint, une version fonctionnelle du produit est prête à être évaluée ou livrée au client.

- 2) **Artefacts** : Les artefacts Agile Scrum sont des éléments utilisés par une équipe Scrum et les parties prenantes utilisent pour détailler le produit en cours de développement, les actions pour le produire et les actions effectuées au cours du projet. [3].

Les principaux artefacts Scrum agiles sont le backlog de produit, le backlog de sprint et les incréments.

- a) **Product backlog** : ou également appelé carnet de produit est un artefact dynamique qui est constamment mis à jour en fonction de nouvelles informations disponibles, il contient une liste de nouvelles fonctionnalités, améliorations, corrections de bogues, tâches ou exigences de travail nécessaires à la création d'un produit [3].
 - b) **Sprint backlog** : (carnet de sprint) est un ensemble de tâches du backlog de produit qui ont été promues pour être développées lors de la prochaine incrémentation de produit. Les backlogs de sprint sont créés par les équipes de développement pour planifier les livrables des incréments futurs et détailler le travail requis pour créer l'incrément [6].
 - c) **Incrément** : Un incrément de produit correspond aux livrables client qui ont été produits en accomplissant les tâches du backlog de produit au cours d'un sprint [3].
- 3) **Membres d'une équipe scrum** : une équipe scrum regroupe trois acteurs qui travaille d'une manière unie et autonome en se concentrant sur un objectif commun :
- a) **Scrum Master** : il a pour mission de tout mettre en œuvre pour que l'équipe travaille dans de bonnes conditions et se concentre sur l'objectif du projet.
 - b) **Product Owner** : est la personne responsable de la vision de produit. Il représente les clients et utilisateurs qui vont l'exploiter après la livraison.
 - c) **Equipe de développement** : une équipe de développement est responsable de la conception, développement, test, administration et livraison des incréments de produit.

- 4) **Cérémonies** : les cérémonies scrum sont un ensemble d'événements qui se

déroulent au cours de chaque sprint afin d'aider l'équipe scrum à planifier, exécuter et revoir son travail de manière itérative et collaborative. On distingue quatre cérémonies scrum.

- a) **Planification du sprint** : Au début d'un sprint, le propriétaire du produit aura un backlog de produit prioritaire . Ils discutent de chaque élément avec l'équipe de développement et le groupe estime collectivement l'effort impliqué. L'équipe de développement établira ensuite une prévision de sprint décrivant la quantité de travail que l'équipe peut accomplir à partir du backlog produit. Cet ensemble de travaux devient alors le backlog du sprint [3].
- b) **Réunion quotidienne** : est une courte réunion quotidienne de 15 minutes (ou moins) pour discuter des progrès et identifier les bloqueurs. Les participants sont invités à participer debout pour que la réunion soit courte [3].
- c) **Revue de sprint** : à la fin de chaque sprint l'équipe de développement montre quels éléments du backlog sont « terminés » aux parties prenantes qui peuvent ensuite donner leur avis [3].
- d) **Rétrospective du sprint** : est une réunion permettant de passer en revue ce qui a réussi pendant le sprint et ce qui peut être amélioré pour le prochain sprint [3].
- 5) **Release** : est une livraison planifiée de fonctionnalités fonctionnelles et testées du produit, qui a pour but de répondre aux besoins des utilisateurs finaux ou du client. Elle est le fruit du travail de l'équipe Scrum durant plusieurs itérations. En outre, la méthode Scrum offre une grande flexibilité pour gérer les changements et les imprévus qui peuvent survenir au cours du projet.

2.3 Langage de modélisation unifié

Le langage UML (Unified Modeling Language) est constitué de diagrammes intégrés utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système. Le langage de modélisation peut servir de modèle pour un projet et garantir une architecture d'information structurée ; il peut également aider les développeurs à présenter leur description d'un système d'une manière compréhensible pour les spécialistes externes. UML est principalement utilisé dans le développement de logiciels orientés objet. Les améliorations apportées à la norme dans la version 2.0 la rendent également adaptée à la représentation des processus de gestion [4].

Parmi les diagrammes UML utilisés dans la conception de notre système, on cite :

2.3.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est une représentation simple entre un utilisateur et les différents cas d'utilisation auxquels il peut participer. Ce diagramme permet d'identifier les potentiels utilisateurs d'un système et offre une vue simplifiée de celui-ci, ce qui est utile notamment lors des communications avec un client [5].

2.3.2 Diagramme de séquence

Un diagramme de séquence représente les interactions entre les différents objets qui composent le système, illustrant ainsi le flux d'exécution d'un scénario donné. Il offre une vue chronologique des interactions et facilite la compréhension des échanges entre les différents éléments du système.

2.3.3 Diagramme de classes

Le diagramme de classe est l'outil de conception le plus utilisé dans le cadre de la programmation orienté objet (POO). Il permet de représenter les différentes classes, avec leurs attributs et méthodes, ainsi que les relations entre celles-ci (héritage, implémentation d'interfaces, appartenance) [5].

2.4 Le design Pattern Modèle-Vue-Contrôleur

Le modèle MVC, connu sous le nom de Modèle-Vue-Contrôleur, est un modèle d'architecture largement utilisé pour structurer une application. Laravel, un framework PHP populaire, adopte nativement ce modèle. Il consiste à séparer l'application en trois couches interdépendantes : la couche de présentation (UI : User Interface), la couche métier (BLL : Business Logic Layer) et la couche d'accès aux données (DAL : Data Access Layer). Cette architecture facilite la réutilisation du code et le développement parallèle, en assurant que les modifications apportées à une couche n'affectent pas les autres [6].

La figure 2.2 représente le design pattern Model vue controlleur.

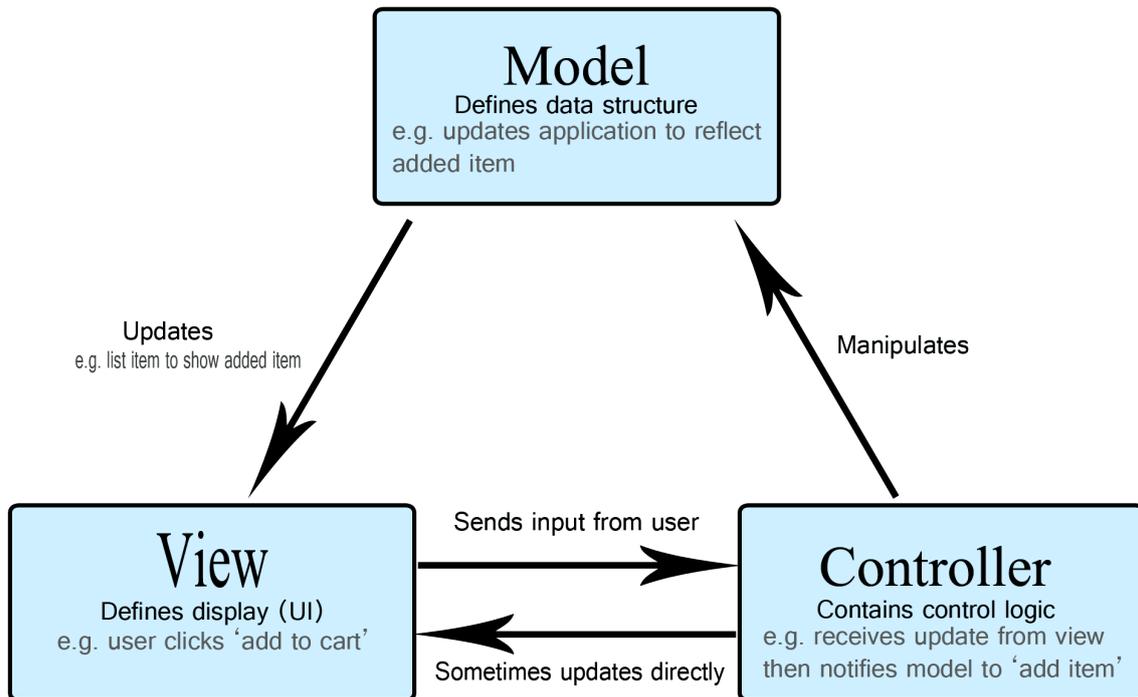


FIGURE 2.2 – L’architecture MVC [23]

- **Couche modèle** : Cette couche représente la structure des données de l’application, également appelée ”back-end”. Elle contient la logique métier, les règles de traitement des données et les opérations de persistance. Le modèle est responsable de la gestion des données et de la manipulation des informations indépendamment de l’interface utilisateur ou de l’affichage. Il peut interagir avec la couche d’accès aux données pour récupérer et stocker les données nécessaires.
- **Couche vue** : Cette couche correspond à l’interface utilisateur, également appelée ”front end” ou ”GUI” (Graphical User Interface). Elle est chargée de présenter les données au client sous une forme visuelle ou graphique. La vue utilise les informations fournies par le modèle pour générer une représentation visuelle de l’application, telle qu’une page HTML, une interface utilisateur graphique ou un résultat XML. La vue est généralement passive et n’effectue pas de traitement complexe des données.
- **Couche contrôleur** : Le contrôleur agit comme un intermédiaire entre la vue et le modèle. Il reçoit les requêtes des utilisateurs à partir de l’interface utilisateur (vue) et les traite en conséquence. Le contrôleur est responsable de la coordination des

actions de l'application, de l'interaction avec le modèle pour récupérer les données nécessaires et de l'ordonnancement de la vue pour afficher les résultats au client. Il gère la logique de l'application et les règles métier, en manipulant les données provenant du modèle et en les adaptant pour répondre aux besoins spécifiques de l'interface utilisateur.

2.5 Pilotage du projet avec la méthode Scrum

2.5.1 Répartition des rôles

La réussite d'un projet Scrum dépend largement de la collaboration efficace entre les membres de l'équipe. Il est essentiel que chaque personne comprenne parfaitement son rôle et ses responsabilités pour assurer un déroulement sans heurts du processus. Pour notre projet, les rôles sont répartis dans le tableau 2.1 ci-dessous :

TABLE 2.1 – Répartition des rôles

Rôles SCRUM	Personne(s) assignée(s)
Product Owner	Mr CHIBOUTI Younes
Scrum Master	Mme KHOULALENE Nadjet
Équipe de Développement	Mlle GALOU Celina Mlle BOULEKBACHE Nadine

2.5.2 Identification des acteurs

Un acteur représente un rôle joué par une personne, un processus ou une entité externe interagissant avec le système. En d'autres termes, un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages pouvant contenir des données. Dans le cadre de notre projet, nous avons identifié quatre acteurs qui interagissent directement avec le système étudié [25].

- **Administrateur** : personne responsable de l'administration et la gestion de l'application prenant en charge tout ce qui concerne la gestion des comptes utilisateurs, et la gestion des droits d'accès, ainsi que la gestion des directions et des départements de cevital. Il veille au bon fonctionnement du serveur de données et sa sécurité.
- **Archiviste** : personne responsable de la gestion des emplacements des boîtes d'archives, et de la supervision du traitement des demandes d'archivage et d'emprunt, dispose d'un compte dans notre application.

- **Demandeur** : personne affilié à une direction, possédant un compte dans l'application et ayant le droit de faire une demande d'archivage ou d'emprunt de boîtes d'archive.

2.5.3 User-Stories

Une user story est une explication générale et informelle d'une fonctionnalité logicielle écrite du point de vue de l'utilisateur final. Son objectif est d'expliquer comment une fonctionnalité logicielle apportera de la valeur au client [3].

Le tableau 2.2 ci-dessous représente la user-story de notre projet :

TABLE 2.2 – User-Stories

<p>En tant qu'archiviste Je veux gérer les salles, rayons, colonnes, étagères pour placer des boîtes d'archives.</p>	<p>En tant qu'archiviste Je veux gérer les demandes d'archives pour pouvoir attribuer un emplacement à une boîte, ainsi que les demandes d'emprunt pour pouvoir prêter une boîte d'archive ou ses documents.</p>
<p>En tant que demandeur Je veux faire une demande pour archiver une boîte d'archive, joindre des des documents numérisés à la demande, gérer cette demande, l'envoyer à l'archiviste et suivre l'état de son traitement.</p>	<p>En tant que demandeur Je veux faire une demande pour emprunter une boîte d'archive ou un document, gérer cette demande, l'envoyer à l'archiviste et suivre l'état de son traitement.</p>
<p>En tant qu'administrateur Je veux gérer les comptes des utilisateurs et leurs droits d'accès.</p>	<p>En tant qu'administrateur Je veux gérer les directions et leurs départements.</p>

2.5.4 Diagramme de cas d'utilisation globale

Les cas d'utilisations identifiés sont illustrés dans le diagramme de cas d'utilisation globale de la figure 2.3, qui sera détaillé ultérieurement.

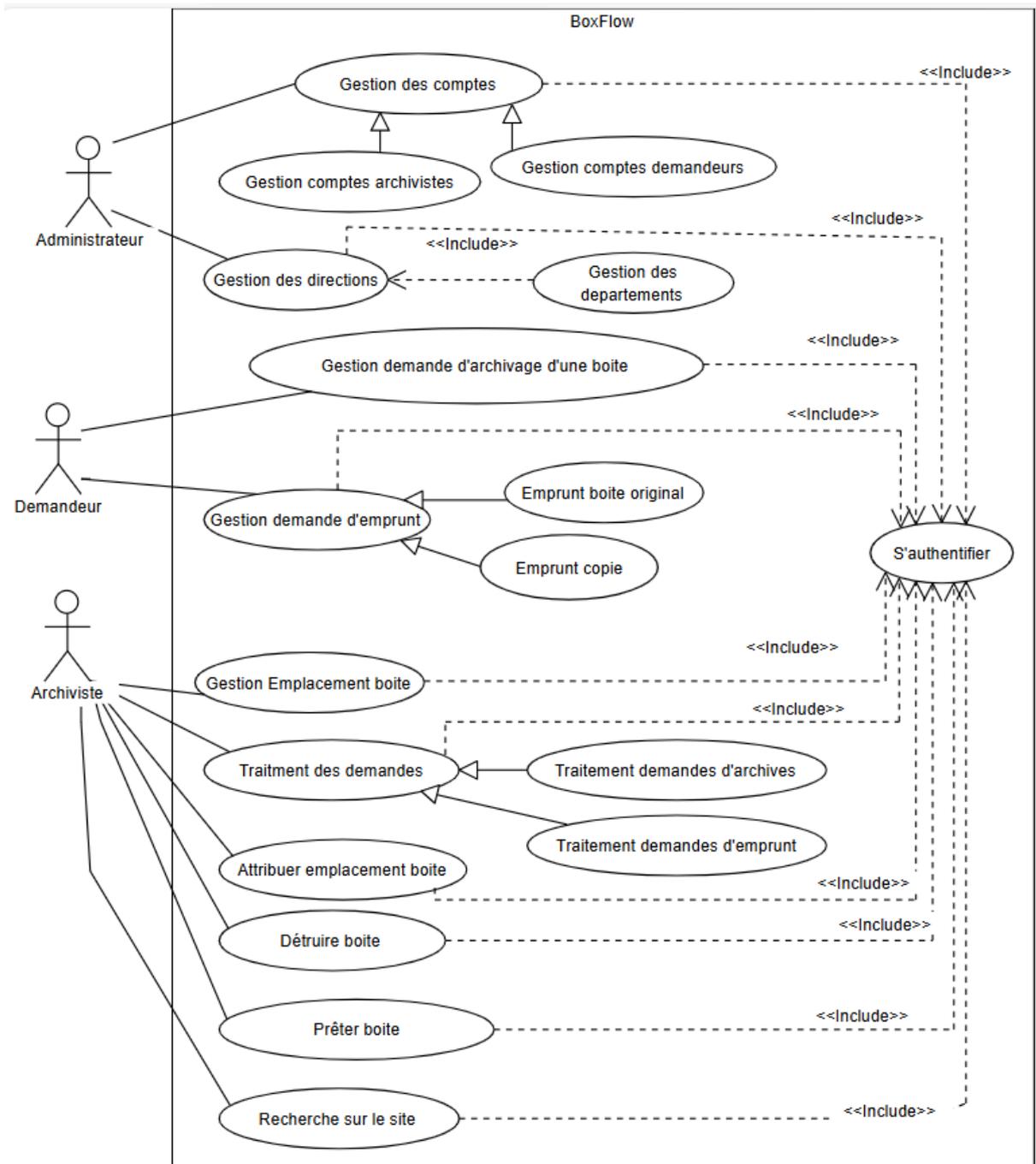


FIGURE 2.3 – Diagramme de cas d'utilisation global

2.5.5 Product Backlog

Chaque user-story est caractérisée par une priorité définie par le Product Owner. Le backlog de notre futur système est présentée dans le tableau 2.3 ci-dessous :

TABLE 2.3 – Product Backlog

Sprint	Items	En tant que	Je veux	Priorité
	Authentification	Administrateur	M'authentifier	1

Table 2.3 – suite

Sprint	Items	En tant que	Je veux	Priorité	
1		Archiviste	M'authentifier		
		Demandeur	M'authentifier		
	Gestion des comptes	Administrateur	Ajouter, modifier, supprimer, désactiver compte archiviste /demandeur, consulter liste comptes		
	Gestion des directions	Administrateur	Ajouter, modifier, supprimer direction		
	Gestion des départements	Administrateur	Ajouter, modifier, supprimer département		
	Gestion des salles	Archiviste	Ajouter, modifier, supprimer salle		2
	Gestion des rayons	Archiviste	Ajouter, modifier, supprimer rayon		
2	Gestion demande versement d'une boîte d'archive	Demandeur	Ajouter, modifier, supprimer colonne	3	
			Ajouter, modifier, supprimer étagère		
	Gestion demande d'emprunt d'une boîte d'archive	Demandeur	Créer une demande d'archive, ajouter boîtes, joindre documents, modifier contenu boîte, supprimer contenu boîte ;		
			Créer une demande d'emprunt original/copie, modifier demande d'emprunt, supprimer demande d'emprunt		
3	Traitement des demandes d'archive/Traitement des demandes emprunt original	Archiviste	Créer une demande d'emprunt original/copie, modifier demande d'emprunt, supprimer demande d'emprunt	3	
			Envoyer demande d'archive et d'emprunt par email et par notification	4	
3	Traitement des demandes d'archive/Traitement des demandes emprunt original	Archiviste	Accepter demande/refuser demande, envoyer réponse par email et par notification	5	

Table 2.3 – suite

Sprint	Items	En tant que	Je veux	Priorité
	Traitement des demandes emprunt copie		Accepter/ refuser demande, joindre copie document, envoyer réponse par email et par notification	5
	Rechercher		Rechercher direction/département /emplacement/boite/demande	6
4	Consulter demande d'archive/emprunt original traitée	Demandeur	Consulter demande acceptée/refusée, afficher motif de refus, refaire demande	7
	Consulter demande emprunt copie traitée		Consulter demande acceptée/refusée, afficher motif de refus, refaire demande	
	Consulter demande emprunt copie traitée		Consulter demande acceptée/refusée, afficher motif de refus, refaire demande afficher document, télécharger document	
	Archivage boite	Archiviste	Attribuer salle, rayon, colonne, étagère, modifier emplacement boite	
5	Prêter boite	Archiviste	Prêter boite/gérer boites empruntées/empruntées en retard, notifier demandeur du retard	8
	Restituer boite		Remettre boite en place	
	Destruction boite		Détruire boite/libérer emplacement /planifier destruction /gérer boite en attente de destruction	

2.6 Charte graphique

La charte graphique joue un rôle essentiel dans la création de l'identité visuelle d'une application. Elle comprend les éléments tels que le logo, les couleurs, la typographie et d'autres éléments, assurant une cohérence visuelle sur tous les supports de communication.

2.6.1 Logotype

Un logo est une représentation graphique qui a pour but d'identifier de manière unique et directe une marque, une entreprise ou une application.

La figure 2.4 illustre le logo de notre application, représentant des boîtes d'archives avec deux flèches ajoutées, accompagnées du mot 'BoxFlow' en dessous. Ce logo souligne le processus d'interaction entre le demandeur et l'archiviste pour l'archivage ou l'emprunt des boîtes.

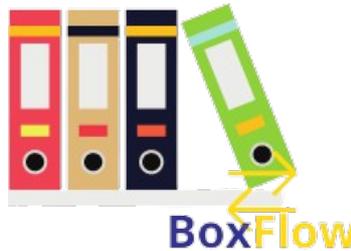


FIGURE 2.4 – Logo de notre application

2.6.2 Palette de couleurs

La palette de couleurs est un ensemble spécifique de couleurs sélectionnées pour être utilisées de manière cohérente dans l'application. Elle assure une harmonie visuelle et contribue à l'identité visuelle en définissant les couleurs principales et secondaires.

La figure 2.5 présente la palette de couleurs que nous avons choisie, où le code couleur est en hexadécimal. Les trois premières couleurs – le bleu, le jaune et le blanc – représentent les couleurs de l'entreprise d'accueil, Cevital. Les deux autres couleurs, le vert et le rouge, apportent un dynamisme à notre application en favorisant l'interaction et l'échange de boîtes entre le demandeur et l'archiviste.

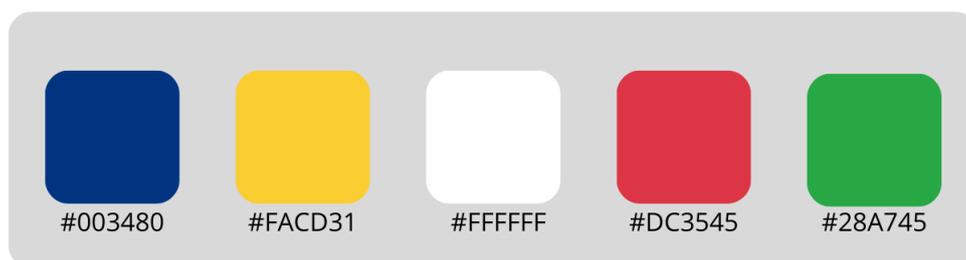


FIGURE 2.5 – Palette de couleurs de l'application BoxFlow en hexadécimal

2.6.3 Slogan

Un slogan est une phrase courte et mémorable utilisée pour représenter et promouvoir une entreprise, un produit, ou même une application web. Il vise à capter l'attention des

utilisateurs, transmettre un message clé ou les valeurs de l'application, et reste facilement ancré dans l'esprit des utilisateurs.

La figure 2.6 présente le slogan de BoxFlow, comprenant :

- **Archive smarter** : pour un archivage efficace adapté à la capacité des étagères.
- **Borrow easier** : pour un emprunt facile des boîtes, surtout pour les copies, permettant au demandeur de ne pas avoir à se déplacer.
- **Interact faster** : pour mettre en avant une interaction rapide entre le demandeur et l'archiviste.

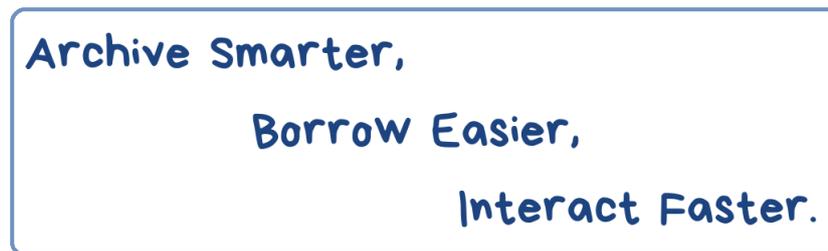


FIGURE 2.6 – Slogan de l'application BoxFlow

2.7 Langages, environnement et outils de développement

Cette section est dédiée à la présentation des différentes technologies utilisées pour le développement de notre système.

2.7.1 HTML CSS

Nous avons utilisé HTML (HyperText Markup Language) pour fournir la structure de la page et CSS (Cascading Style Sheets) pour gérer la présentation visuelle. HTML et CSS, en plus des graphiques et des scripts, ont constitué la base de notre création de pages et d'applications web.[7].

2.7.2 JavaScript

JavaScript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Nous l'avons surtout utilisé pour charger les différents départements d'une direction de Cevital, mais aussi pour charger les différents rayons d'un site précis, les différentes colonnes d'un rayon, et enfin les différentes étagères d'une colonne précise. [8].

2.7.3 Laravel

Nous avons utilisé Laravel, un framework open source de développement d'applications web en PHP, reconnu pour sa syntaxe expressive et élégante. Nous avons choisi Laravel car il suit le modèle de conception MVC (Modèle-Vue-Contrôleur) pour structurer le code et utilise l'ORM (Object Relational Mapping) pour simplifier l'interaction avec les bases de données. Laravel simplifie le processus de développement en fournissant une structure organisée et des fonctionnalités puissantes telles qu'une injection approfondie de dépendances, une couche d'abstraction de base de données expressive, des files d'attente et des tâches planifiées, ainsi que des tests unitaires et d'intégration. L'une des raisons principales pour lesquelles nous avons choisi Laravel est qu'il propose une documentation complète.[9].

2.7.4 SQLite

Nous avons choisi SQLite comme système de gestion de base de données car il est léger et rapide, ainsi que pour sa simplicité, sa fiabilité et sa flexibilité. Il offre un stockage de données efficace sans nécessiter de configuration serveur complexe, car il ne requiert aucun processus serveur ; il est directement intégré dans l'application. SQLite prend en charge la plupart des fonctionnalités standard des bases de données relationnelles, y compris les transactions ACID, les clés étrangères, et les requêtes SQL avancées. [21].

2.7.5 HeidiSQL

HeidiSQL est un outil d'administration gratuit et open source, conçu pour les développeurs Web utilisant les serveurs populaires comme MariaDB, MySQL, Microsoft SQL Server ou PostgreSQL. Il permet de parcourir et de modifier des données, de créer et de modifier des tables, des vues, des procédures, des déclencheurs et des événements planifiés. De plus, vous pouvez exporter la structure et les données vers un fichier SQL, le presse-papiers ou vers d'autres serveurs [10]. Dans notre cas, nous avons utilisé HeidiSQL pour visualiser notre base de données existante SQLite et y exécuter des requêtes SQL.

2.7.6 Visual Studio Code

VSCoDe10 (Visual Studio Code) est un éditeur de code source léger et puissant développé par Microsoft. Nous l'avons choisi comme environnement de développement car il offre une interface utilisateur conviviale et une prise en charge étendue de nombreux langages de programmation. Grâce à sa flexibilité et à son écosystème d'extensions riche, VS Code nous a permis de personnaliser notre environnement de développement selon nos besoins [11].

2.7.7 GitHub et Git

GitHub est une plate-forme basée sur le cloud sur laquelle on peut stocker, partager et collaborer pour écrire du code. Stocker notre code dans un « référentiel » sur GitHub nous a permis de : partager notre travail, suivre et gérer les modifications apportées à notre code au fil du temps, et collaborer sur un projet partagé. Le travail collaboratif, l'une des fonctionnalités fondamentales de GitHub, est rendu possible par le logiciel open source Git, sur lequel GitHub est construit. Git est un système de contrôle de version qui suit intelligemment les modifications apportées aux fichiers. Git nous a été particulièrement utile lorsque on apporte des modifications aux mêmes fichiers en même temps. [12].

2.7.8 Visual Paradigm

Visual Paradigm est une plateforme qui offre un large éventail d'outils de modélisation. Nous l'avons utilisée pour modéliser différents diagrammes UML, tels que les diagrammes de séquences, les diagrammes de cas d'utilisation et les diagrammes de classes [13].

2.7.9 Canva

Canva est une plateforme de conception graphique qui fournit des outils pour créer des graphiques, des présentations, des produits promotionnels et des sites Web pour les réseaux sociaux. Nous l'avons particulièrement utilisée pour le logo, le slogan et la page d'accueil. [14].

2.8 Packages utilisés

Dans ce qui suit, nous allons présenter les différents packages implémentés dans notre application :

2.8.1 Spatie/laravel-permission

Ce package fournit un système de gestion des rôles et des permissions pour les applications Laravel. Il nous a permis de définir des rôles, d'assigner des permissions aux rôles et d'autoriser les actions des utilisateurs en fonction de leurs rôles et permissions [15].

Voici quelques fonctionnalités clés fournies par la package "Spatie/laravel-permission" :

- **Gestion des rôles et des permissions** Elle nous a permis de créer et de gérer les rôles, ainsi que de définir des permissions associées à chaque rôle. Cela permet

de déterminer précisément les actions autorisées pour chaque rôle dans une application.

- **Contrôle d'accès basé sur les rôles et les permissions** Le package fournit des méthodes pour vérifier facilement les autorisations d'un utilisateur en fonction de ses rôles et de ses permissions. On peut contrôler l'accès à certaines fonctionnalités ou à certaines parties d'une application en utilisant ces vérifications d'autorisation.

L'installation de ce package se fait à l'aide de composer en exécutant la commande suivante :

```
composer require spatie/laravel-permission
```

2.8.2 Spatie/Media-Library

Ce package nous a permis d'associer des fichiers aux modèles Eloquent. Il fournit une API simple et fluide avec laquelle travailler [16]. Parmi ses principales fonctionnalités :

- **Association de fichiers** : On peut associer des fichiers à n'importe quel modèle Eloquent. Que ce soit des images, des documents, ou tout autre type de fichier.
- **Conversions d'images** : On peut définir des conversions d'images pour générer des versions de différentes tailles ou avec des effets appliqués.
- **Collections de médias** : On peut organiser les fichiers en collections pour mieux les gérer.
- **Optimisation des images** : Le package peut optimiser les images pour une meilleure performance.
- **Facilité d'utilisation** : Une API simple et fluide pour gérer les fichiers.

L'installation de ce package se fait à l'aide de composer en exécutant la commande suivante :

```
composer require spatie/laravel-medialibrary
```

2.8.3 Admin LTE

AdminLTE 3 est un template d'interface utilisateur open-source basé sur Bootstrap 4. Nous l'avons utilisé car il offre une variété de fonctionnalités telles que des graphiques, des formulaires, des tables et des notifications pour faciliter le développement rapide d'interfaces utilisateur riches et réactives. Nous l'avons surtout utilisé car ce package fournit un système de configuration de menu avancé. [17].

```
composer require jeroennoten/laravel-adminlte php artisan adminlte :install
```

2.8.4 Laravel UI

Laravel UI est un package pour Laravel qui fournit des implémentations front-end pour l'authentification, incluant des vues, des styles CSS et des composants JavaScript. Il nous a permis de configurer rapidement les fonctionnalités d'authentification en utilisant des outils front-end populaires, dans notre cas Bootstrap [18].

L'installation de ce package se fait à l'aide de composer en exécutant la commande suivante :

```
composer require laravel/ui
```

2.9 Conclusion

Au cours de ce chapitre, nous avons présenté un aperçu de la méthode scrum ainsi que ses différents composants. Nous avons introduit brièvement les notions de UML et l'architecture MVC. Ensuite, Nous avons procédé à la planification de notre projet en définissant les rôles de l'équipe, en identifiant les acteurs et en établissant les cas d'utilisation qui constituent notre Product Backlog, ainsi que les user-stories des acteurs et le diagramme de cas d'utilisation. Nous avons également organisé notre travail en plusieurs sprints afin de répondre au besoin d'utilisateurs de manière itérative. Enfin, nous avons présenté l'identité visuelle de notre application ainsi que les outils utilisés lors du développement de notre application. Dans le prochain chapitre, nous aborderons le sprint 1, ou nous commencerons à concrétiser notre application en mettant en œuvre les fonctionnalités prioritaires du Product Backlog.

Chapitre 3

Sprint 1

3.1 Introduction

Le présent chapitre a pour objectif de détailler le premier lancement initial du projet, incluant le premier sprint d'une durée de quatre semaines. Ce sprint couvre un ensemble de cas d'utilisation représentant les fonctionnalités du système accompagnées de leurs descriptions textuelles. Par la suite, nous présenterons les diagrammes de séquence associés à ce sprint, ce qui nous permettra de visualiser le fonctionnement de notre application. Enfin, nous montrerons quelques interfaces résultant de ce sprint.

3.2 Etude du sprint 1

3.2.1 Espace Administrateur

Il comprend trois items réalisés en 2 semaine :

- **Authentification** : Est le processus de vérification de l'identité d'un utilisateur ou d'un système afin de garantir que l'accès aux ressources ou aux informations est accordé uniquement aux personnes ou aux entités légitimes.
- **Gestion des comptes** : Fait référence à l'ensemble des actions et fonctionnalités permettant de gérer les comptes d'utilisateurs. Cela inclut les tâches suivantes : Création de compte pour l'archiviste et le demandeur, modification de comptes, gestion des droits d'accès, suppression et désactivation des comptes.
- **Gestion des directions et des département** : Chaque département est associé à une direction spécifique, donc pour gérer un département (ajout, modification, suppression), il faut d'abord gérer les directions correspondantes (ajout, modification, suppression).

3.2.2 Espace Archiviste

Il comprend un item réalisé en 2 semaines :

- **Gestion des emplacements** : Cela comprend la gestion des salles et des rayons correspondants aux salles, puis la gestion des colonnes correspondantes au rayons, et enfin la gestion des étagères correspondantes aux colonnes.

3.3 Diagramme de cas d'utilisation du sprint 1

Le diagramme de cas d'utilisation de la figure 3.1 décrit d'une façon globale les cas d'utilisations constituant les items du premier sprint :

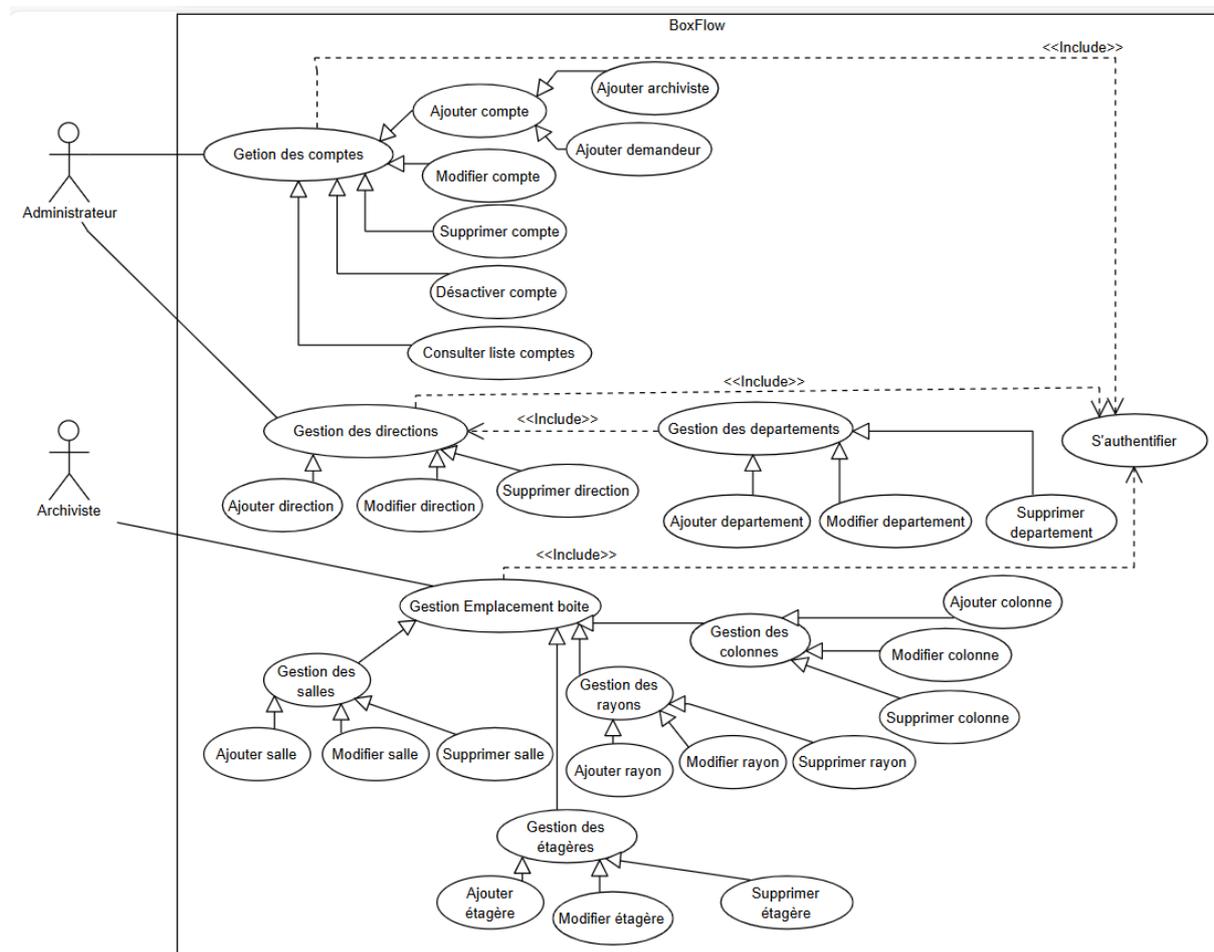


FIGURE 3.1 – Diagramme de cas d'utilisation du sprint 1

3.4 Cas d'utilisation "S'authentifier"

3.4.1 Description textuelle du cas d'utilisation "s'authentifier"

Le tableau 3.1 suivant représente la description textuelle du cas d'utilisation "s'authentifier" pour l'administrateur, c'est le même fonctionnement pour l'authentification de l'archiviste et le demandeur.

TABLE 3.1 – Description textuelle du cas d'utilisation "S'authentifier"

CU	S'authentifier
BUT	Ce cas permet à l'administrateur du système de se connecter.
Acteur Principal	Administrateur
Acteur Secondaire	/
Pré-condition	L'administrateur possède déjà un compte.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur accède au site. 2. Le système affiche le formulaire de connexion. 3. L'administrateur saisit son email et son mot de passe. 4. L'administrateur valide la connexion. 5. Le système vérifie les champs du formulaire. 6. Le système affiche le tableau de bord de l'utilisateur.
Enchaînement alternatif	5.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 2.
Post-condition	L'administrateur accède à son espace avec succès.

3.4.2 Diagramme de séquence du cas d'utilisation "s'authentifier"

La figure 3.2 représente le diagramme de séquence du cas d'utilisation s'authentifier pour l'administrateur. C'est le même traitement pour l'authentification de l'archiviste et le demandeur [19].

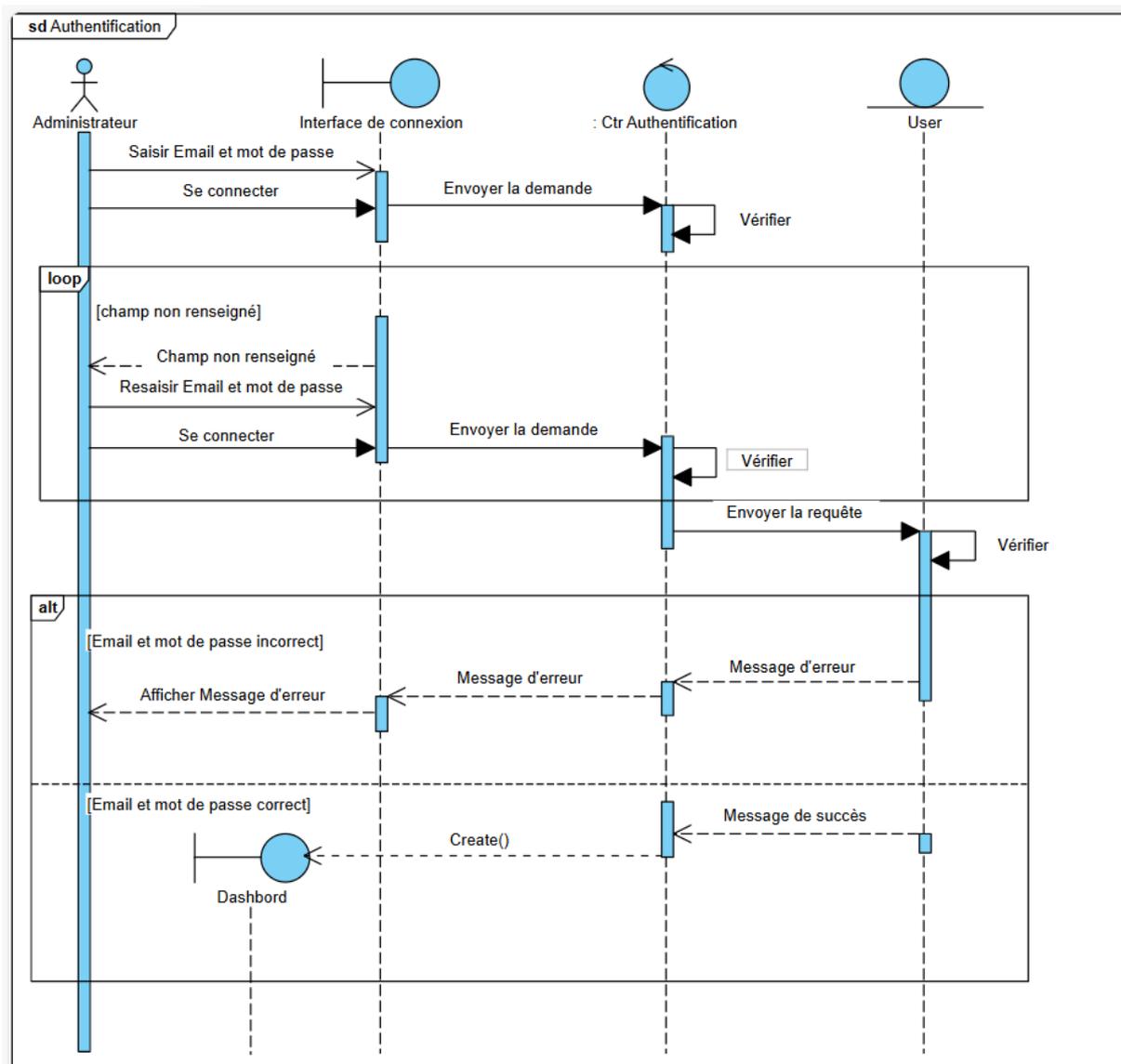


FIGURE 3.2 – Diagramme de séquence de cas d'utilisation "S'authentifier"

3.5 Cas d'utilisation "Ajouter compte"

3.5.1 Description textuelle du cas d'utilisation "Ajouter compte"

Le tableau 3.2 suivant représente la description textuelle du cas d'utilisation "Ajouter compte".

TABLE 3.2 – Description textuelle du cas d'utilisation "Ajouter compte"

CU	Ajouter compte
BUT	Ce cas permet à l'administrateur de créer un compte pour les utilisateurs de l'application (archiviste, demandeur).
Acteur Principal	Administrateur
Acteur Secondaire	/
Pré-condition	L'administrateur est déjà connecté à son espace. Rôle déjà ajouté.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à son espace. 2. L'administrateur demande le formulaire de création de compte. 3. Le système affiche le formulaire. 4. L'administrateur remplit le formulaire. 4.a. L'administrateur sélectionne le rôle de l'utilisateur (rôle déjà ajouté avec des permissions attribuées). 5. L'administrateur valide le formulaire. 6. Le système vérifie les champs du formulaire. 7. Le système crée le nouveau compte.
Enchaînement alternatif	6.a. Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3.
Post-condition	L'administrateur accède à son espace avec succès.

3.5.2 Diagramme de séquence du cas d'utilisation "Ajouter compte"

La figure 3.3 suivante représente le diagramme de séquence de cas d'utilisation "Ajouter compte" pour l'administrateur.

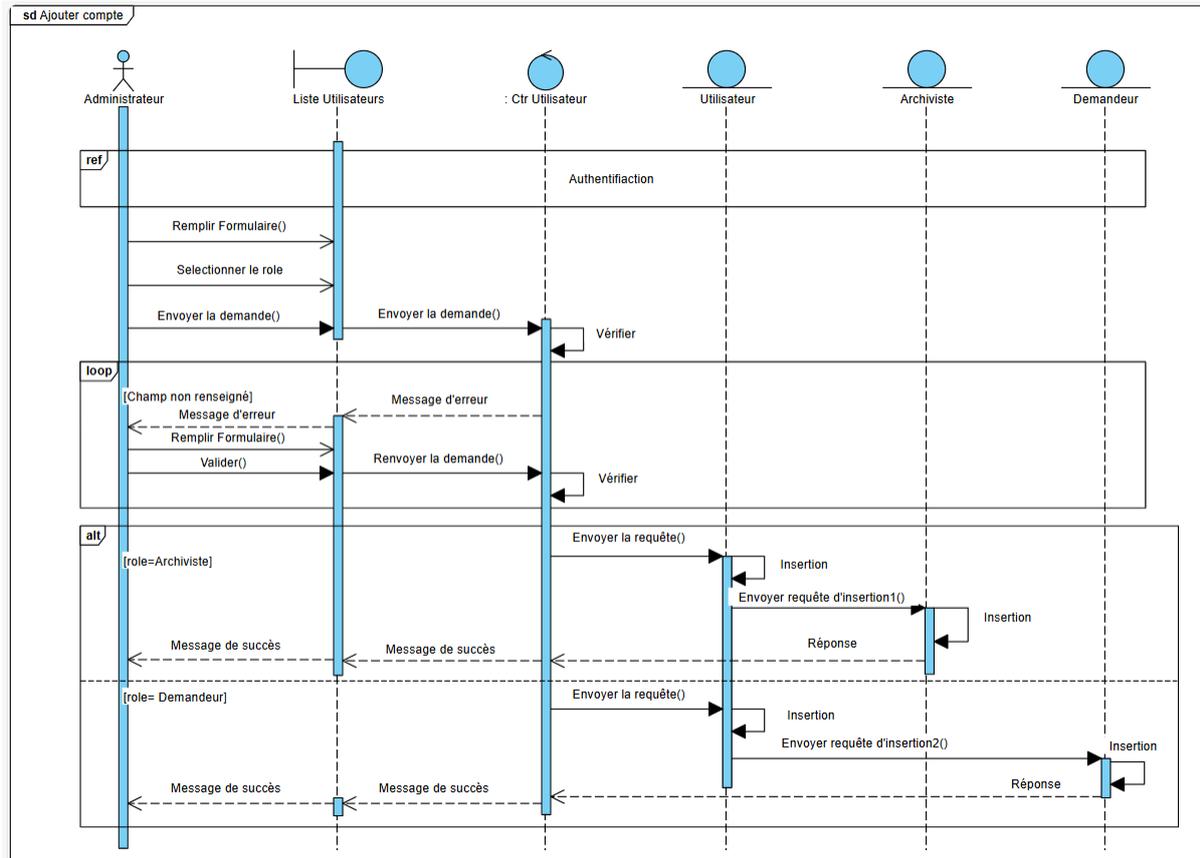


FIGURE 3.3 – Diagramme de séquence du cas d'utilisation "Ajouter compte"

3.6 Cas d'utilisation "Gestion des directions"

3.6.1 Description textuelle des cas d'utilisation "Gestion des directions"

Les tableaux 3.3, 3.4, 3.7 représentent respectivement les descriptions textuelles des cas d'utilisations "Ajouter/Modifier/supprimer direction" pour l'administrateur.

TABLE 3.3 – Description textuelle de cas d'utilisation "Ajouter direction"

CU	Ajouter direction
BUT	Ce cas permet à l'administrateur d'ajouter une direction.
Acteur Principal	Administrateur
Acteur Secondaire	/
Pré-condition	L'administrateur est déjà connecté à son espace.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à son espace. 2. L'administrateur demande le formulaire de création d'une direction. 3. Le système affiche le formulaire. 4. L'administrateur remplit le formulaire. 5. L'administrateur valide le formulaire. 6. Le système vérifie les champs du formulaire. 7. Le système ajoute une nouvelle direction.
Enchaînement alternatif	6.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3.
Post-condition	La direction est ajoutée avec succès.

TABLE 3.4 – Description textuelle du cas d'utilisation "Modifier direction"

CU	Modifier direction
BUT	Ce cas permet à l'administrateur de modifier une direction.
Acteur Principal	Administrateur
Acteur Secondaire	/
Pré-condition	<p>L'administrateur est déjà connecté à son espace.</p> <p>Direction déjà ajoutée.</p>
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à l'interface des liste directions. 2. L'administrateur choisi la modification d'une direction identifiée. 3. Le système affiche le formulaire des informations de direction. 4. L'administrateur saisit les modifications. 5. L'administrateur valide le formulaire. 6. Le système vérifier les champs du formulaire. 7. Le système modifie la direction.
Enchaînement alternatif	6.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3.
Post-condition	La direction est modifiée avec succès.

TABLE 3.5 – Description textuelle du cas d'utilisation "Supprimer direction"

CU	Supprimer direction
BUT	Ce cas permet à l'administrateur de supprimer une direction.
Acteur Principal	Administrateur
Acteur Secondaire	/
Pré-condition	L'administrateur est déjà connecté à son espace. Direction déjà ajoutée.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à l'interface des listes directions. 2. L'administrateur choisi la suppression d'une direction identifiée. 3. Le système affiche une fenêtre pour demander la confirmation de suppression. 4. L'administrateur confirme sa décision de suppression. 5. L'administrateur valide le formulaire. 6. Le système supprime la direction et ne sera plus affichée.
Enchaînement alternatif	3.a Si l'administrateur annule sa décision de suppression, le système désactive la fenêtre de confirmation de suppression.
Post-condition	La direction est supprimée avec succès.

3.6.2 Diagramme de séquence du cas d'utilisation "Gestion des directions"

La figure 3.4 suivante représente le diagramme de séquence du cas d'utilisation "Gestion des directions" pour l'administrateur.

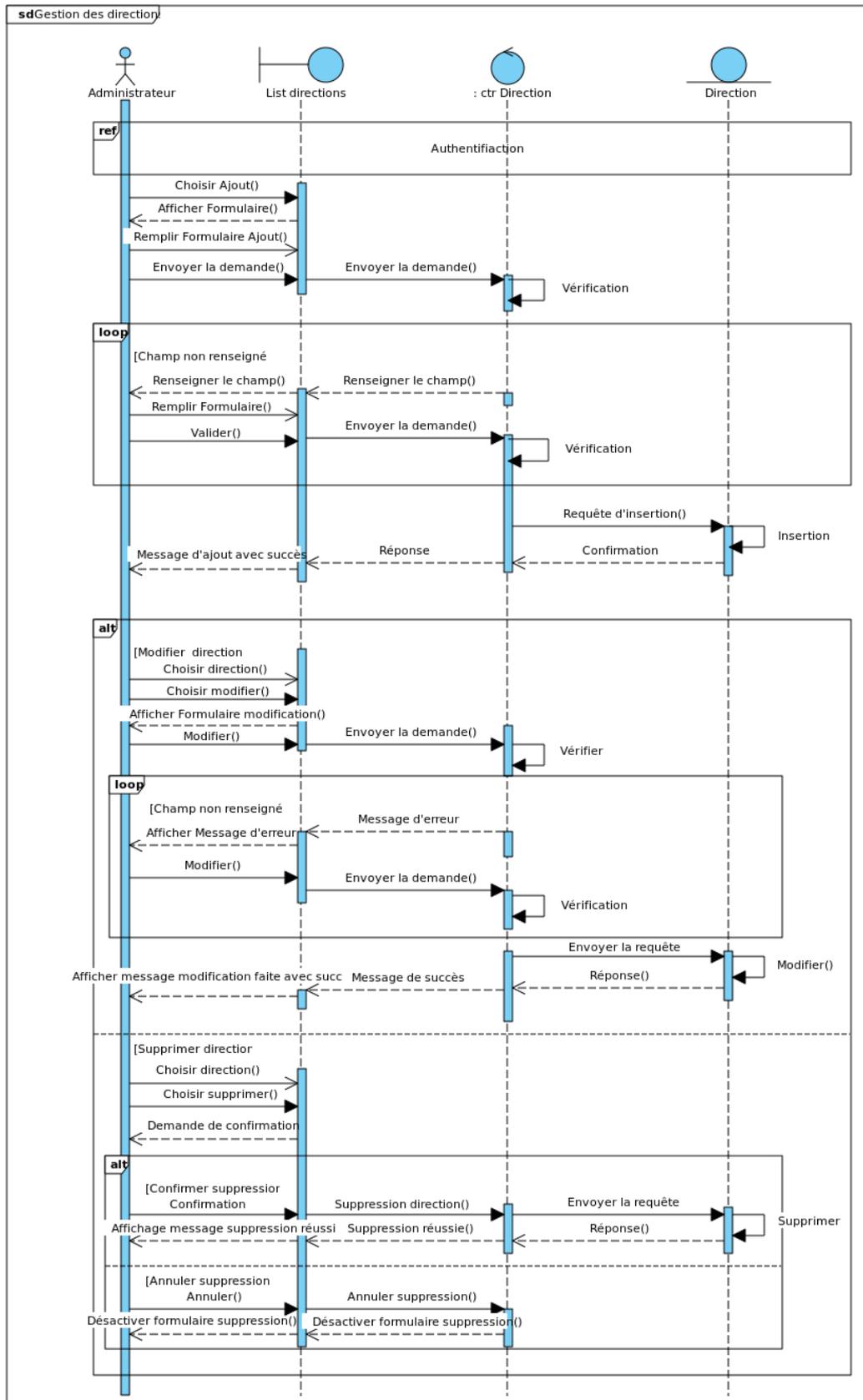


FIGURE 3.4 – Diagramme de séquence de cas d'utilisation "Gestion des directions"

3.7 Cas d'utilisation "Ajouter département"

3.7.1 Description textuelle du cas d'utilisation "Ajouter département"

Le tableau 3.6 représente la description textuelle du cas d'utilisation "Ajouter département" pour l'administrateur.

TABLE 3.6 – Description textuelle du cas d'utilisation "Ajouter département"

CU	Ajouter département
BUT	Ce cas permet à l'administrateur d'ajouter un département.
Acteur Principal	Administrateur
Acteur Secondaire	/
Pré-condition	L'administrateur est déjà connecté à son espace. Direction déjà ajoutée.
Scénario nominal	L'administrateur accède à son espace. 2. L'administrateur demande le formulaire de création d'un département. 3. Le système affiche le formulaire . 4. L'administrateur sélectionne la direction. 4.a L'administrateur remplit le formulaire (nom département). 5. L'administrateur valide le formulaire. 6. Le système vérifie les champs du formulaire. 7. Le système ajoute un nouveau département.
Enchaînement alternatif	6.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3.
Post-condition	Le département est ajouté avec succès.

3.7.2 Diagramme de séquence du cas d'utilisation "Ajouter département"

La figure 3.5 suivante représente le diagramme de séquence de cas d'utilisation "Ajouter département" pour l'administrateur.

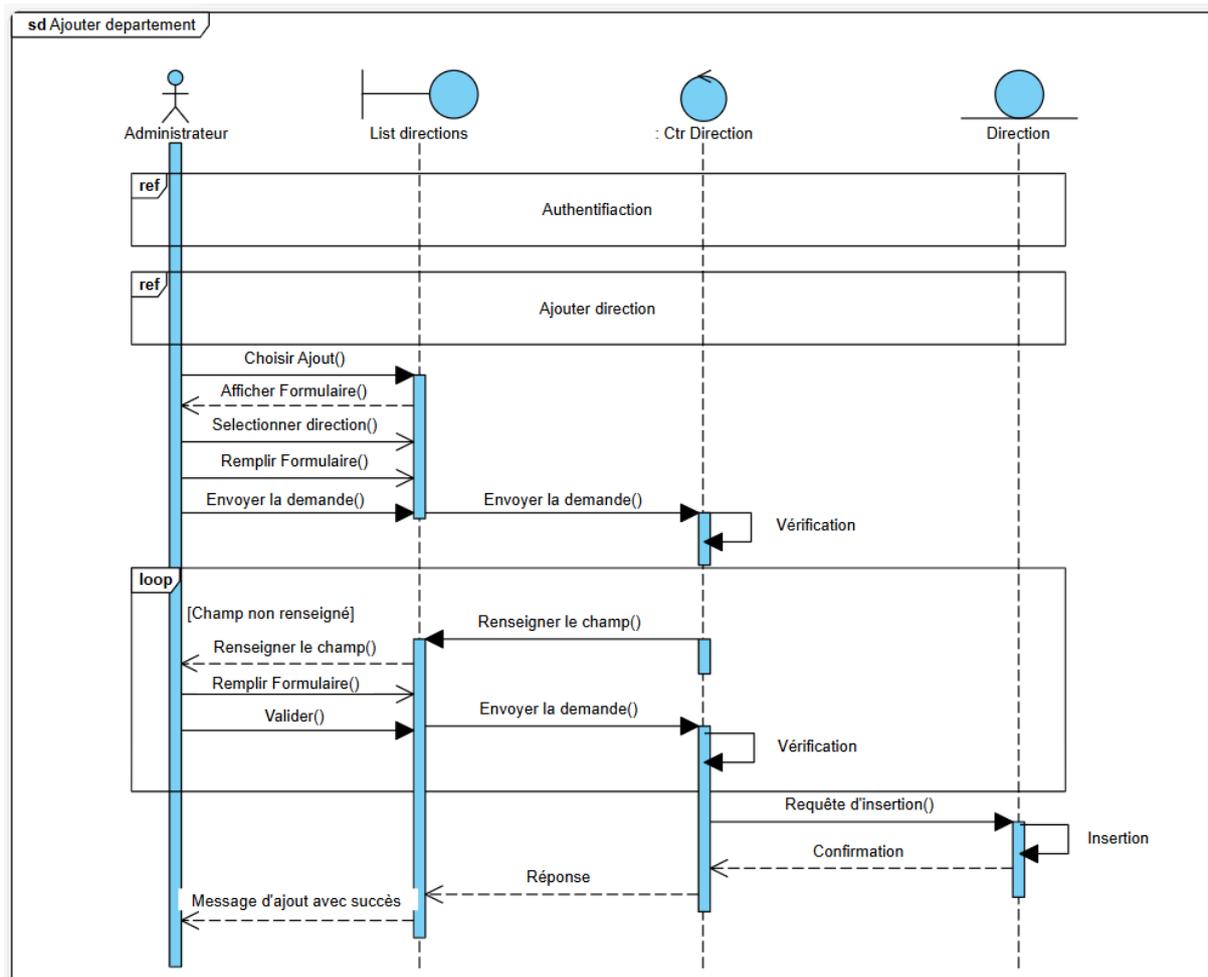


FIGURE 3.5 – Diagramme de séquence de cas d'utilisation "Ajouter département"

3.8 Cas d'utilisation "Ajouter étagère"

3.8.1 Description textuelle des cas d'utilisation "Ajouter étagère"

Le tableau 3.7 représente la description textuelle du cas d'utilisation "Ajouter étagère" pour l'archiviste.

TABLE 3.7 – Description textuelle du cas d'utilisation "Ajouter étagère"

CU	Ajouter étagère
BUT	Ce cas permet à l'archiviste d'ajouter une étagères.
Acteur Principal	Archiviste
Acteur Secondaire	/
Pré-condition	L'archiviste est déjà connecté à son espace. Salles déjà ajoutées, Rayons déjà ajoutés, Colonnes déjà ajoutées.
Scénario nominal	<ol style="list-style-type: none"> 1.L'archiviste accède à son espace. 2. L'archiviste demande le formulaire de création d'une étagère. 3.Le système affiche le formulaire. 4. L'archiviste sélectionne la salle. 5.Le système retourne la liste des rayons correspondants à la salle sélectionnée. 6. L'archiviste sélectionne le rayon. 7.Le système retourne la liste des colonnes correspondantes au rayon sélectionné. 8.L'archiviste sélectionne la colonne. . 9.L'archiviste remplit le reste du formulaire (nom étagère) et choisit la capacité de l'étagère (10 boites ou 20 boites). 10.L'archiviste valide le formulaire. 11.Le système vérifie les champs du formulaire. 12.Le système ajoute une nouvelle étagère.
Enchaînement alternatif	11.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3.
Post-condition	: L'étagère est ajoutée avec succès.

3.8.2 Diagramme de séquence du cas d'utilisation "Ajouter étagère"

La figure 3.6 suivante représente le diagramme de séquence du cas d'utilisation "Ajouter étagère" pour l'archiviste [19].

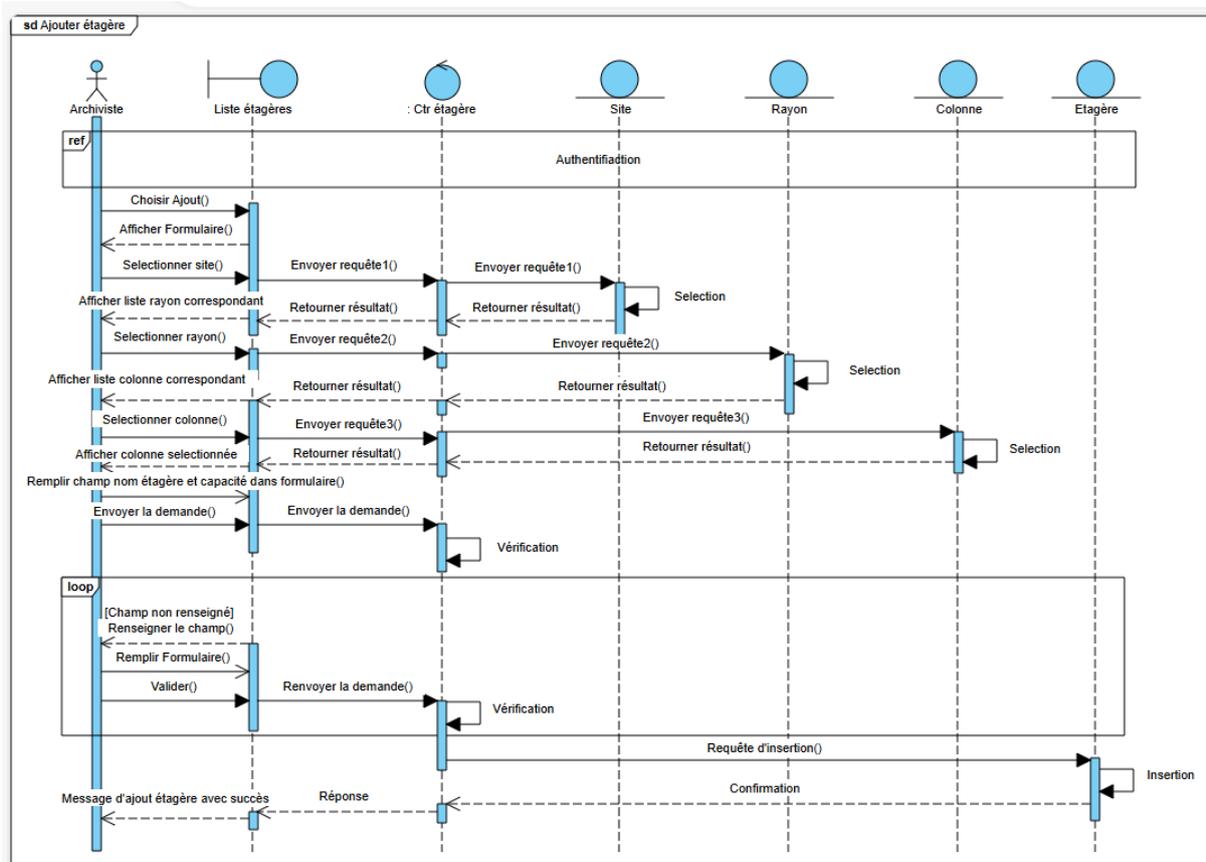


FIGURE 3.6 – Diagramme de séquence du cas d'utilisation "Ajouter étagère"

3.9 Diagramme de classe du sprint 1

La figure 3.7 représente le diagramme de classe qui comporte les classes associées aux fonctionnalités du sprint 1 :

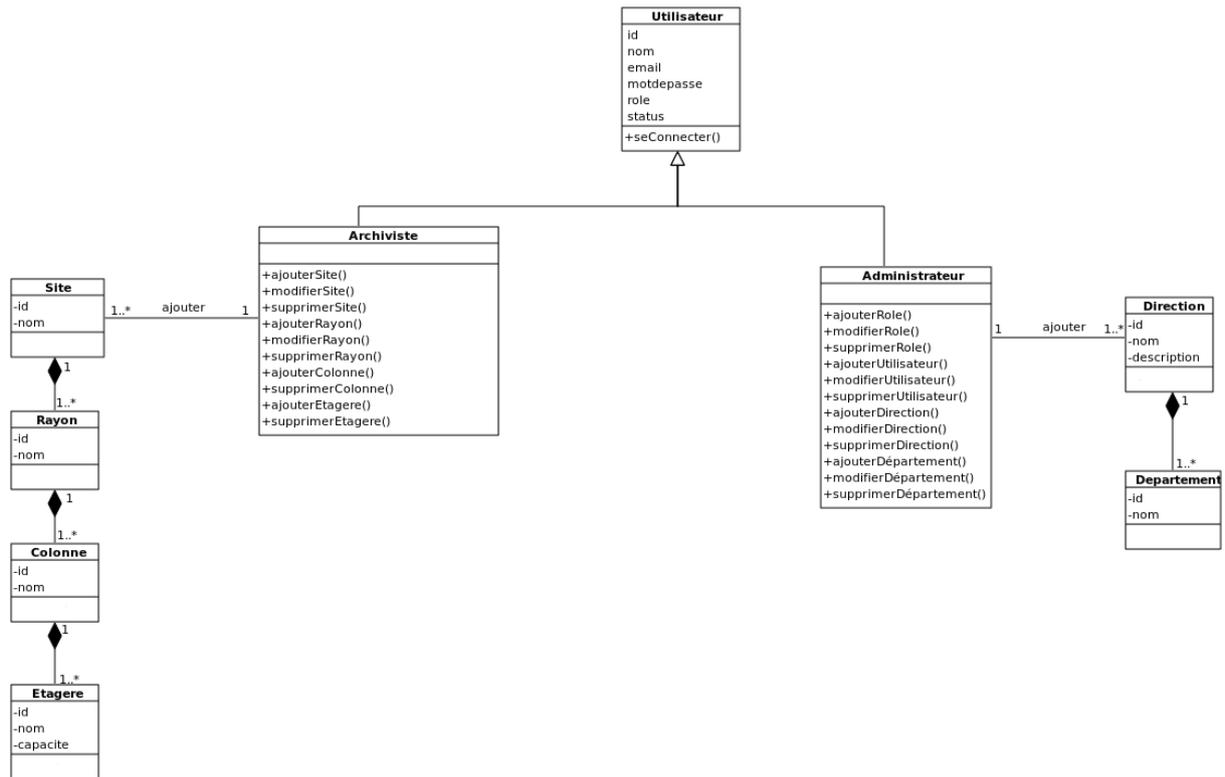


FIGURE 3.7 – Diagramme de classe sprint 1

3.10 Vue des interfaces utilisateurs du sprint1

3.10.1 Interface d'authentification

L'interface d'authentification, présentée par la figure 3.8 suivante, permet aux utilisateurs de l'application (Administrateur, archiviste, demandeur) de s'authentifier chacun avec son email et son mot de passe ce qui leur permettra ensuite d'accéder à leurs propres espaces.

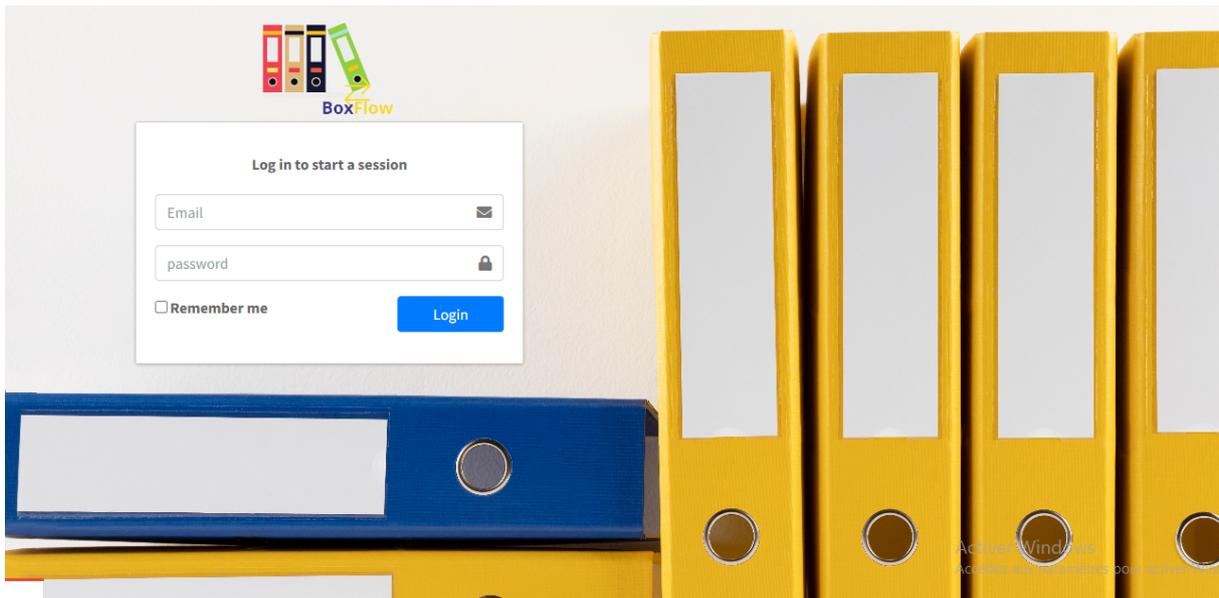


FIGURE 3.8 – Interface d'authentification

3.10.2 Interface utilisateur

L'interface du tableau de bord utilisateur, présentée par la figure 3.9, désigne l'interface vers laquelle un demandeur sera redirigé après son authentification.

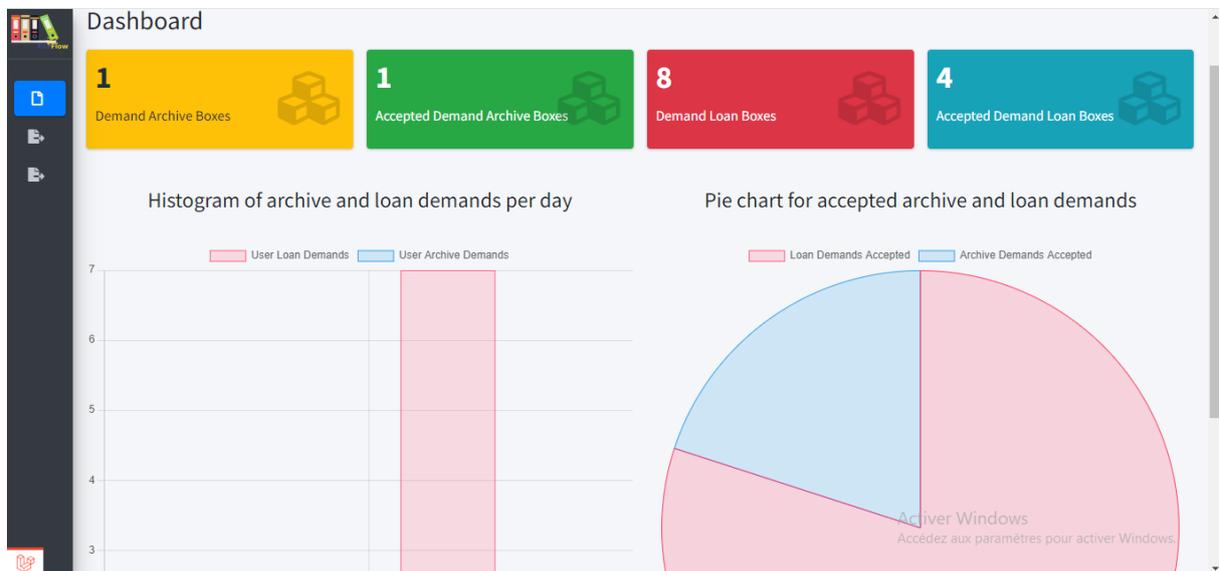


FIGURE 3.9 – Interface dashbord demandeur

3.10.3 Interface Ajout compte

L'interface d'ajout compte, présentée par la figure 3.10, permet à l'administrateur d'ajouter des demandeurs et des archivistes avec des rôles spécifiques (Demandeur ou Archiviste).

The screenshot shows a web interface titled "Users lists" with a blue "Add user" button at the top. Below it is a "Back" button. The form contains the following fields:

- Username: *** (text input)
- Email: *** (text input)
- Password: *** (password input)
- Confirm password: *** (password input)
- Status user** (dropdown menu with "active" selected)
- User authority** (dropdown menu with "Admin", "Archiviste", and "applicant" options)

A red "Submit" button is located at the bottom center. In the bottom right corner, there is a message: "Activer Windows. Accédez aux paramètres pour activer Windows."

FIGURE 3.10 – Interface ajouter compte

3.10.4 Interface gestion des directions

L'interface de gestion de directions, présentée par la figure 3.11, permet à l'administrateur d'ajouter, modifier, supprimer une direction ainsi que d'afficher la liste des directions.

The screenshot shows a web interface titled "Directions lists" with a blue "Add Direction" button at the top. Below it is a search bar and a "Show 10 rows" dropdown. The table below has the following data:

#	name	description	Actions
1	DSI	Direction système d'information	[edit] [delete]
2	Transit	Direction Transit	[edit] [delete]
3	Commercial	Direction Commercial	[edit] [delete]
4	RH	Direction RH	[edit] [delete]
5	Finance et Comptabilite	Direction des Finances et Comptabilites	[edit] [delete]

At the bottom left, it says "Showing 1 to 5 of 5 entries". At the bottom right, there are "Previous", "1", and "Next" pagination buttons. In the bottom right corner, there is a message: "Activer Windows. Accédez aux paramètres pour activer Windows."

FIGURE 3.11 – Interface gestion des directions

3.10.5 Interface de gestion des départements

L'interface de gestion des départements, présentée par la figure 3.12, permet à l'administrateur d'ajouter, modifier ou supprimer un département.

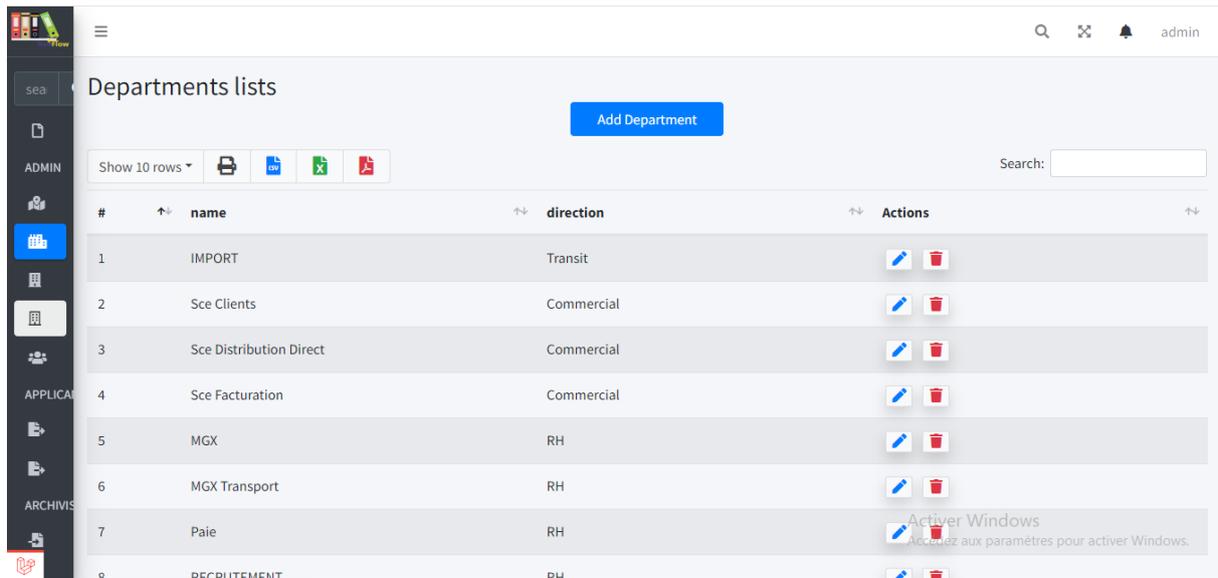


FIGURE 3.12 – Interface gestion des départements

3.10.6 Interface ajout d'une étagère

L'interface d'ajout d'une étagère, présentée par la figure 3.13, permet à l'archiviste d'ajouter une étagère dépendants d'une colonne d'un rayon d'une salle déjà ajoutée.

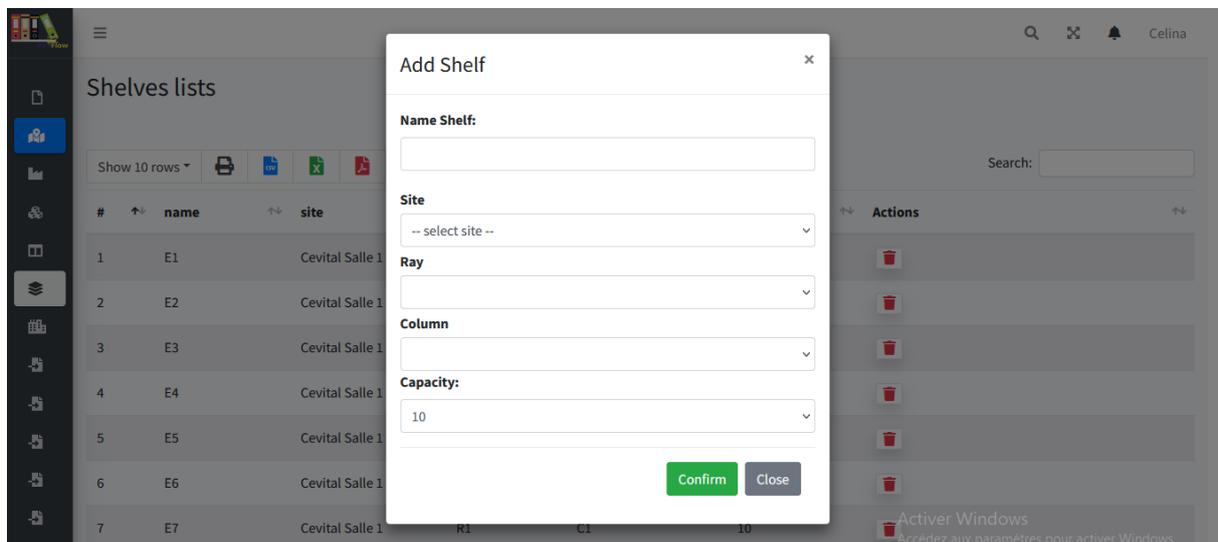


FIGURE 3.13 – Interface ajouter étagère

3.11 Conclusion

Dans ce chapitre, nous avons présenté le premier sprint de notre projet. Nous avons travaillé sur des items de l'espace administrateur et de l'espace archiviste. À la fin du sprint, nous avons fait une revue de sprint durant laquelle nous avons montré aux parties prenantes quels éléments du backlog étaient terminés. Dans le prochain chapitre, nous

détailleront le déroulement des sprints 2 et 3, en continuons à développer les fonctionnalités du système et en consolidant progressivement son évolution.

Chapitre 4

Sprint 2, 3

4.1 Introduction

Dans ce chapitre nous, allons poursuivre le développement de notre application en examinant les sprints 2 et 3. Pour chaque sprint, nous présenteront des diagrammes de cas d'utilisation, des descriptions textuelles et des diagrammes de séquence pour illustrer les différentes fonctionnalités et interactions du système, et détailler le fonctionnement de l'application, après on montrera quelques interfaces du résultat obtenue dans ce sprint.

4.2 Etude du Sprint 2

4.2.1 Espace Demandeur

Il comprend deux items réalisés en 4 semaines :

- **Gestion demande d'archivage t d'une boite d'archive** : Cela comprend la création d'une demande de versement d'une boite d'archive, ajout de boites avec des documents scannés, modification du contenu de la boite, suppression du contenu de la boite, modification de la demande, suppression de la demande et envoie de la demande par email et par notification.
- **Gestion demande d'emprunt** : Cela comprend la création d'une demande d'emprunt d'une boite de type original ou copie, modification de la demande, suppression de la demande et envoie de la demande par email et par notification.

4.2.2 Diagramme de cas d'utilisation du sprint 2

Le diagramme de cas d'utilisation de la figure 4.1 décrit d'une façon globale les cas d'utilisations constituant les items du deuxième sprint :

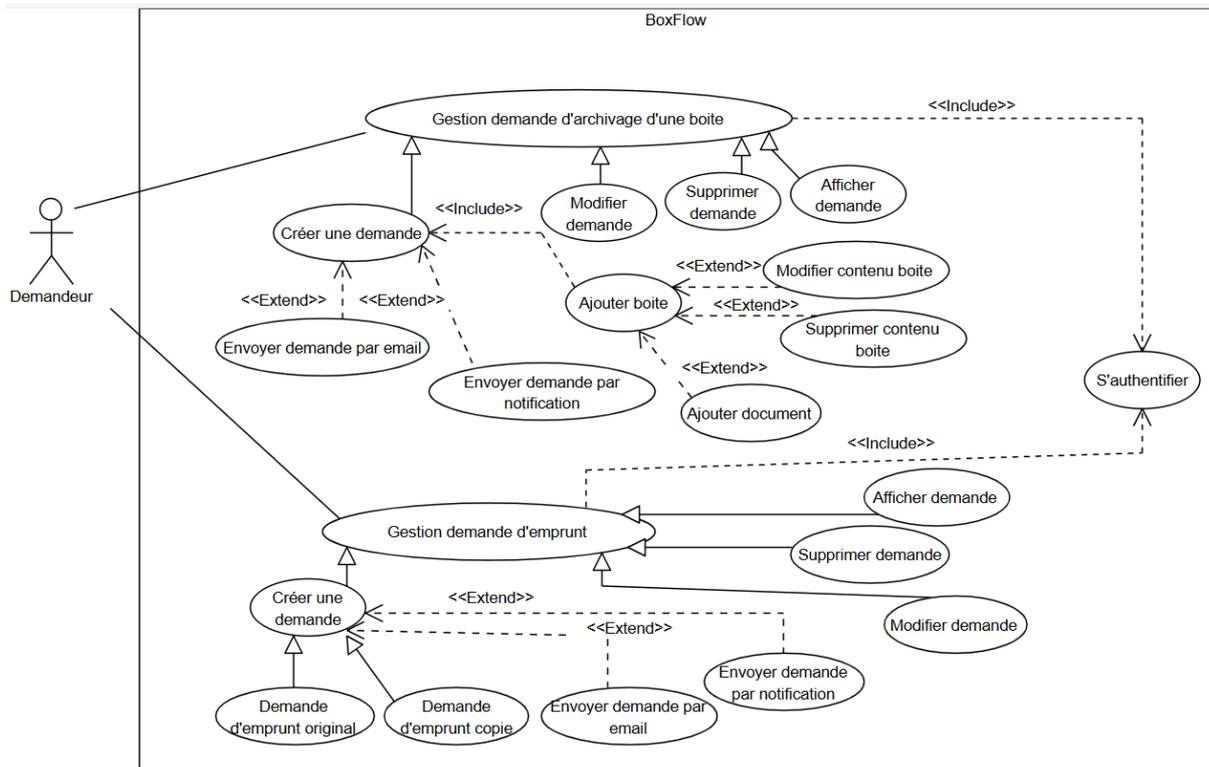


FIGURE 4.1 – Diagramme de cas d'utilisation du sprint 2

4.3 Cas d'utilisation "Ajouter demande d'archive"

4.3.1 Description textuelle des cas d'utilisation "Ajouter demande d'archive"

Le tableau 4.1 suivant représente la description textuelle du cas d'utilisation "Ajouter demande d'archive" pour le demandeur.

TABLE 4.1 – Description textuelle des cas d'utilisation "Ajouter demande d'archive"

CU	Ajouter demande d'archive
BUT	Ce cas permet au demandeur d'ajouter une demande d'archivage.
Acteur Principal	Demandeur
Acteur Secondaire	/
Pré-condition	Le demandeur est déjà connecté à son espace.
Scénario nominal	<ol style="list-style-type: none"> 1. Le demandeur accède à son espace. 2. Le demandeur demande le formulaire de création d'une demande d'archive. 3. Le système affiche le formulaire. 4. Le demandeur remplit le formulaire. 5. Le demandeur valide le formulaire. 6. Le système vérifie les champs du formulaire. 7. Le système affiche les informations de la demande saisie en dessus de l'interface d'ajout boîte. 8. Le demandeur valide l'ajout de la boîte à la demande. 9. Le système affiche le formulaire. 10. Le demandeur remplit le formulaire d'ajout boîte et ajoute les documents nécessaires. 11. Le demandeur valide le formulaire d'ajout boîte. 12. Le système vérifie les champs du formulaire. 13. Le système ajoute la boîte. 14. Le système ajoute la demande de versement d'archive.
Enchaînement alternatif	<ol style="list-style-type: none"> 6.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3. 8.a Si aucune boîte est insérée, le système affiche un message d'erreur, indiquant qu'il faut insérer au moins une boîte. Reprendre à partir de 9. 12.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 9.
Post-condition	La demande d'archivage est ajoutée avec succès.

4.3.2 Diagramme de séquence du cas d'utilisation "Ajouter demande d'archive"

La figure 4.2 représente le diagramme de séquence de cas d'utilisation "Ajouter demande d'archive" pour le demandeur.

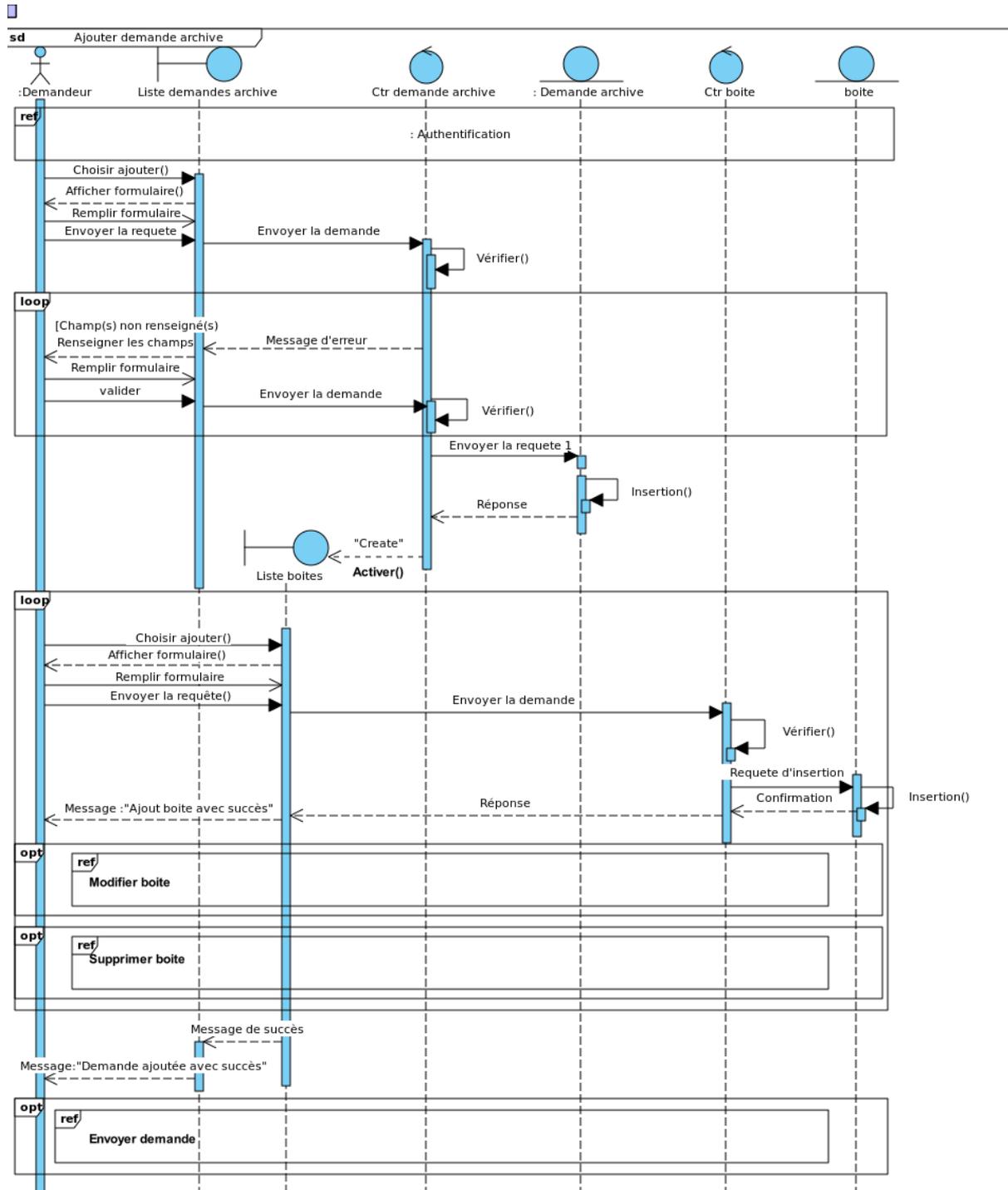


FIGURE 4.2 – Diagramme de séquence du cas d'utilisation "Ajouter demande d'archive"

4.4 Cas d'utilisation "Modifier boite"

4.4.1 Description textuelle des cas d'utilisation "Modifier boite"

Le tableau 4.2 suivant représente la description textuelle du cas d'utilisation "Modifier boite" pour le demandeur.

TABLE 4.2 – Description textuelle des cas d'utilisation "Modifier boite"

CU	Modifier boite
BUT	Ce cas permet au demandeur de modifier le contenu d'une boite d'une demande d'archive.
Acteur Principal	Demandeur
Acteur Secondaire	/
Pré-condition	Le demandeur est déjà connecté à son espace. La demande d'archive est déjà ajoutée. La demande n'est pas encore envoyée au traitement.
Scénario nominal	<ol style="list-style-type: none"> 1. Le demandeur accède à l'interface des listes demandes d'archives. 2. Le demandeur clique sur le bouton show d'une demande identifiée. 3. Le système affiche l'interface détails de la demande sélectionnée. 4. Le demandeur choisit la modification d'une boite identifiée. 5. Le système affiche le formulaire des informations de la boite. 6. Le demandeur saisit les modifications. 7. Le demandeur valide le formulaire. 8. Le système vérifie les champs du formulaire. 9. Le système modifie la boite.
Enchaînement alternatif	8.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 5.
Post-condition	La boite d'archive est modifiée avec succès.

4.4.2 Diagramme de séquence du cas d'utilisation "Modifier boite"

La figure 4.3 représente le diagramme de séquence de cas d'utilisation "Modifier boite" pour le demandeur.

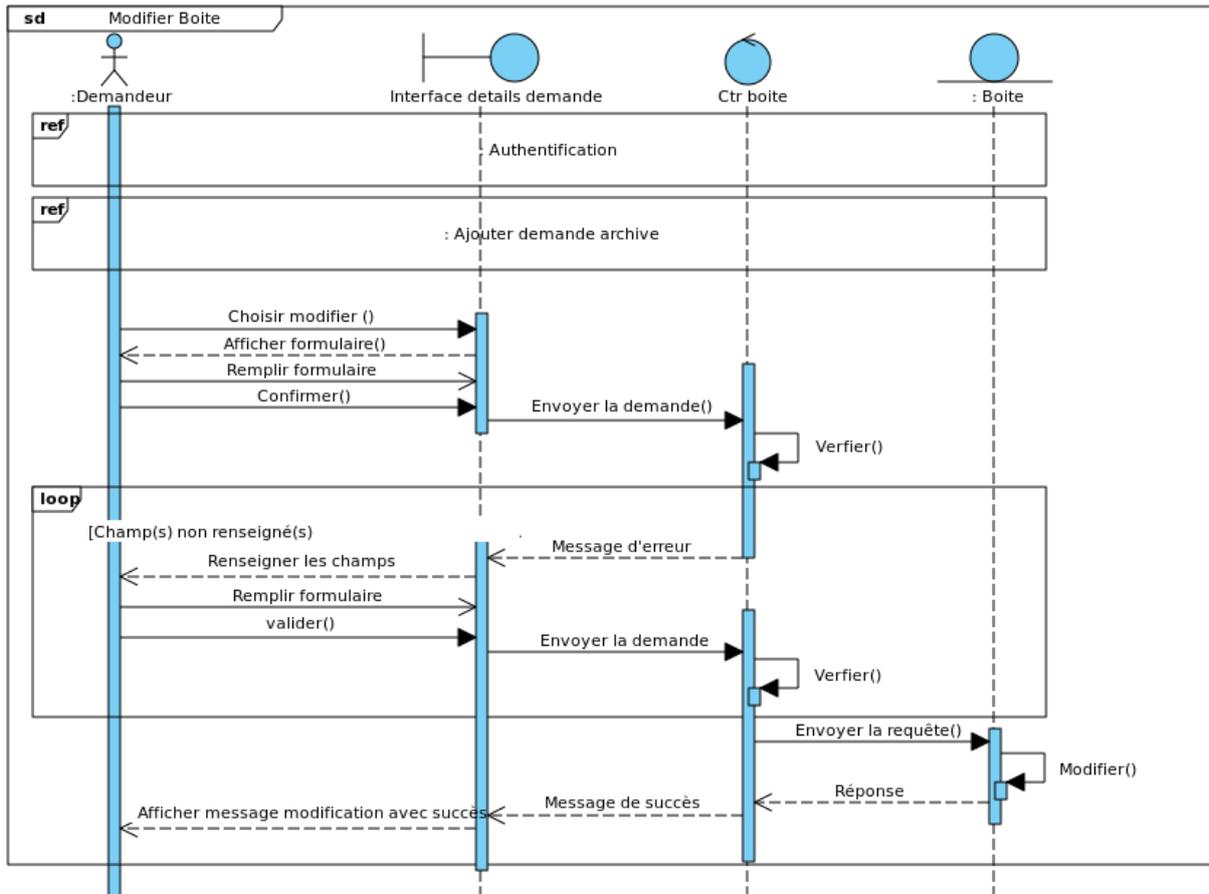


FIGURE 4.3 – Diagramme de séquence du cas d'utilisation "Modifier boite"

4.5 Cas d'utilisation "Supprimer boite"

4.5.1 Description textuelle des cas d'utilisation "Supprimer boite"

Le tableau 4.3 représente la description textuelle du cas d'utilisation "Supprimer boite" pour le demandeur.

TABLE 4.3 – Description textuelle des cas d'utilisation "Supprimer boite"

CU	Supprimer boite
BUT	Ce cas permet au demandeur de supprimer le contenu d'une boite d'une demande d'archive.
Acteur Principal	Demandeur
Acteur Secondaire	/
Pré-condition	Le demandeur est déjà connecté à son espace. La demande d'archive est déjà ajoutée. La demande n'est pas encore envoyée au traitement.
Scénario nominal	1. Le demandeur accède à l'interface des listes demandes d'archives. 2. Le demandeur clique sur le bouton show d'une demande identifiée. 3. Le système affiche l'interface détails de la demande sélectionnée. 4. Le demandeur choisit la suppression d'une boite identifiée. 5. Le système affiche une fenêtre pour demander la confirmation de suppression. 6. Le demandeur confirme sa décision de suppression. 7. Le système supprime la boite d'archive et ne sera plus affichée.
Enchaînement alternatif	5.a Si le demandeur annule sa décision de suppression, le système désactive la fenêtre de confirmation de suppression. Reprendre à partir de 3.
Post-condition	La boite d'archive est supprimée avec succès.

4.5.2 Diagramme de séquence du cas d'utilisation "Supprimer boite"

La figure 4.4 représente le diagramme de séquence de cas d'utilisation "Supprimer boite" pour le demandeur

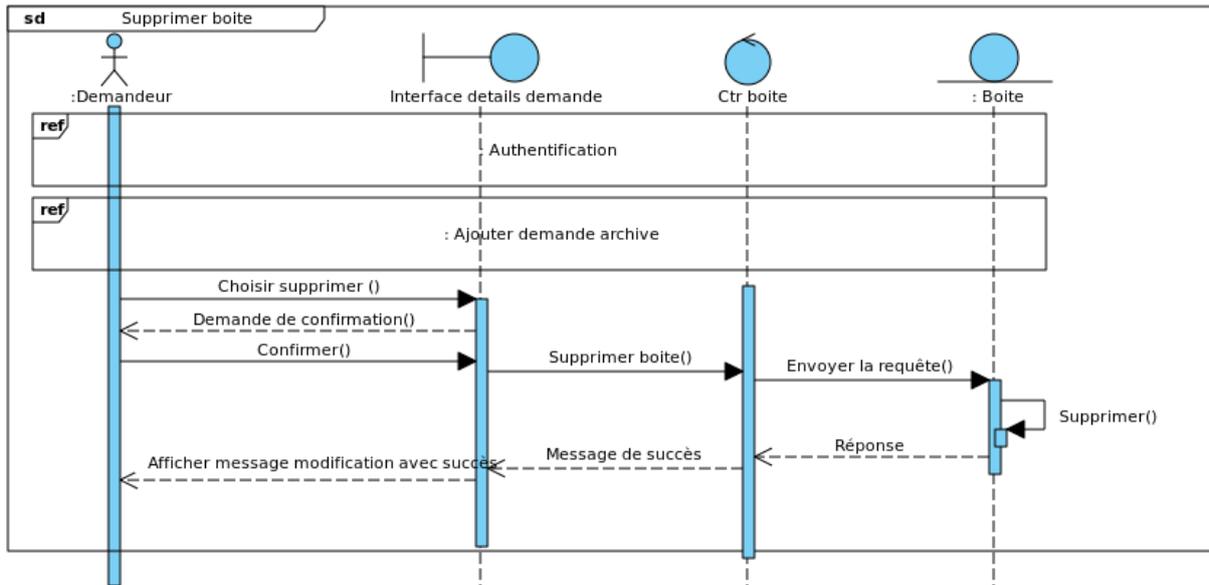


FIGURE 4.4 – Diagramme de séquence du cas d'utilisation "Supprimer boîte"

4.6 Cas d'utilisation "Ajouter demande emprunt"

4.6.1 Description textuelle des cas d'utilisation "Ajouter demande emprunt"

Le tableau 4.4 représente la description textuelle du cas d'utilisation "Ajouter demande emprunt" pour le demandeur.

TABLE 4.4 – Description textuelle des cas d'utilisation "Ajouter demande d'emprunt"

CU	Ajouter demande d'emprunt
BUT	Ce cas permet au demandeur d'ajouter une demande d'emprunt.
Acteur Principal	Demandeur
Acteur Secondaire	/
Pré-condition	Le demandeur est déjà connecté à son espace.
Scénario nominal	<ol style="list-style-type: none"> 1. Le demandeur accède à son espace. 2. Le demandeur demande le formulaire de création d'une demande d'emprunt. 3. Le système affiche le formulaire. 4. Le demandeur remplit le formulaire. 5. Le demandeur sélectionne le type d'emprunt (boite originale ou copie document). 6. Le demandeur valide le formulaire. 7. Le système vérifie les champs du formulaire. 8. Le système ajoute la demande d'emprunt.
Enchaînement alternatif	7.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3.
Post-condition	La demande d'emprunt est ajoutée avec succès.

4.6.2 Diagramme de séquence du cas d'utilisation "Ajouter demande emprunt"

La figure 4.5 représente le diagramme de séquence de cas d'utilisation "Ajouter demande emprunt" pour le demandeur.

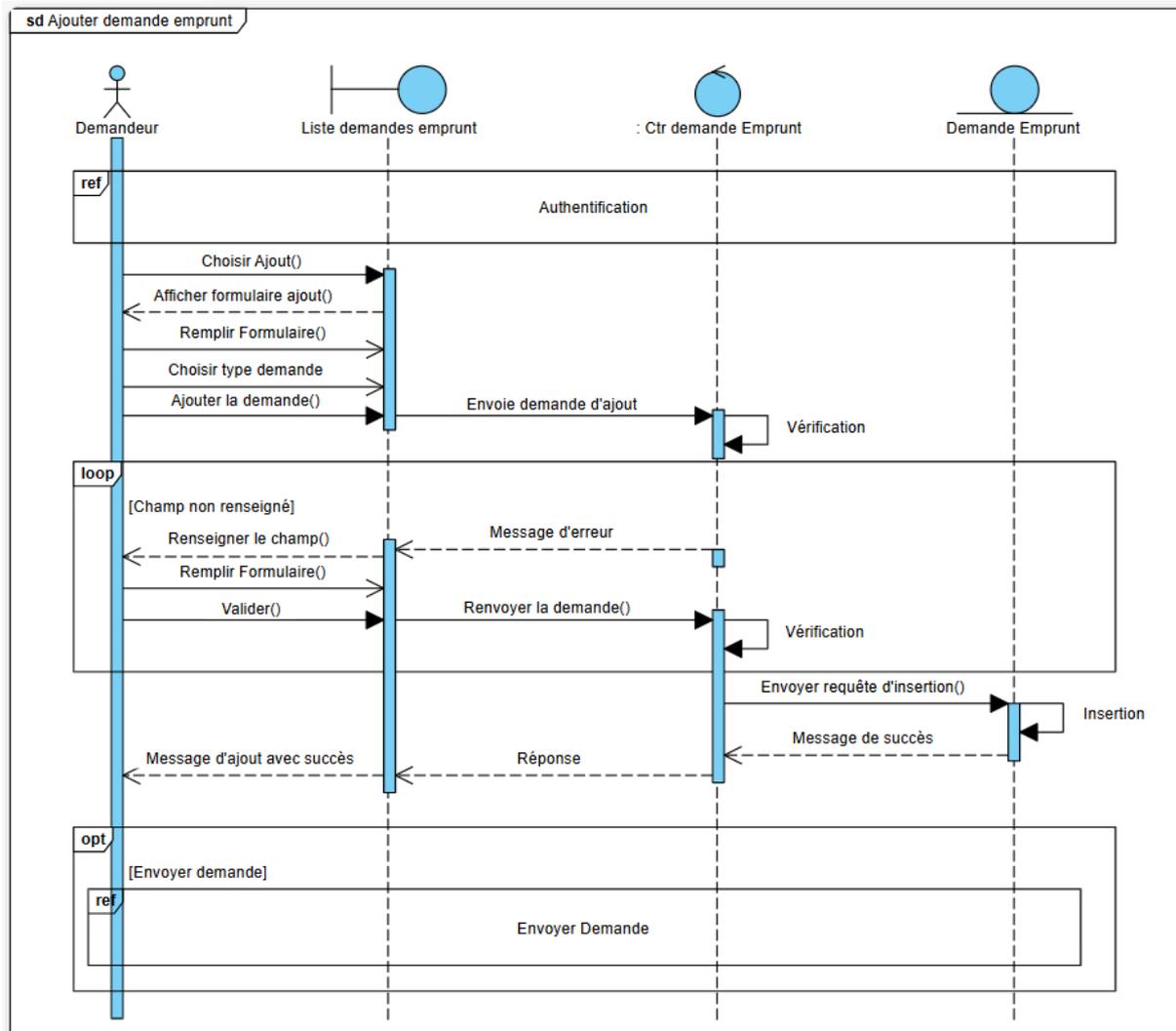


FIGURE 4.5 – Diagramme de séquence du cas d'utilisation "Ajouter demande emprunt"

4.7 Cas d'utilisation "Envoyer demande"

4.7.1 Description textuelle des cas d'utilisation "Envoyer demande"

Le tableau 4.5 représente la description textuelle du cas d'utilisation "Envoyer demande" pour le demandeur.

TABLE 4.5 – Description textuelle des cas d'utilisation "Envoyer demande"

CU	Envoyer demande
BUT	Ce cas permet au demandeur d'envoyer sa demande d'archive ou d'emprunt après l'avoir créée.
Acteur Principal	Demandeur
Acteur Secondaire	/
Pré-condition	Le demandeur est déjà connecté à son espace. La demande est déjà créée ou la demande est refusée.
Scénario nominal	<ol style="list-style-type: none"> 1. Le demandeur accède à la liste de ses demandes d'archive/d'emprunt dont il possède une demande d'archive/d'emprunt avec un statut égal à « created » ou « refused ». 2. Le demandeur choisit d'envoyer la demande. 3. Le système envoie la demande par notification à la base de données des notifications et au mail. 4. Le système ajoute une nouvelle notification. 5. Le système envoie en notification la demande à l'espace d'archiviste destiné aux notifications (icône de notification dans la navbar du dashboard) et à son email. 6. Le système affiche un message indiquant que l'envoi est réussi, le statut de la demande devient « sent ».
Enchaînement alternatif	5.a Si une défaillance de connexion se produit, le système affiche un message indiquant que la connexion n'a pas pu être établie. Reprendre à partir de 1.
Post-condition	Demande envoyée avec succès.

4.7.2 Diagramme de séquence du cas d'utilisation "Envoyer demande"

La figure 4.6 représente le diagramme de séquence de cas d'utilisation "Envoyer demande" pour le demandeur.

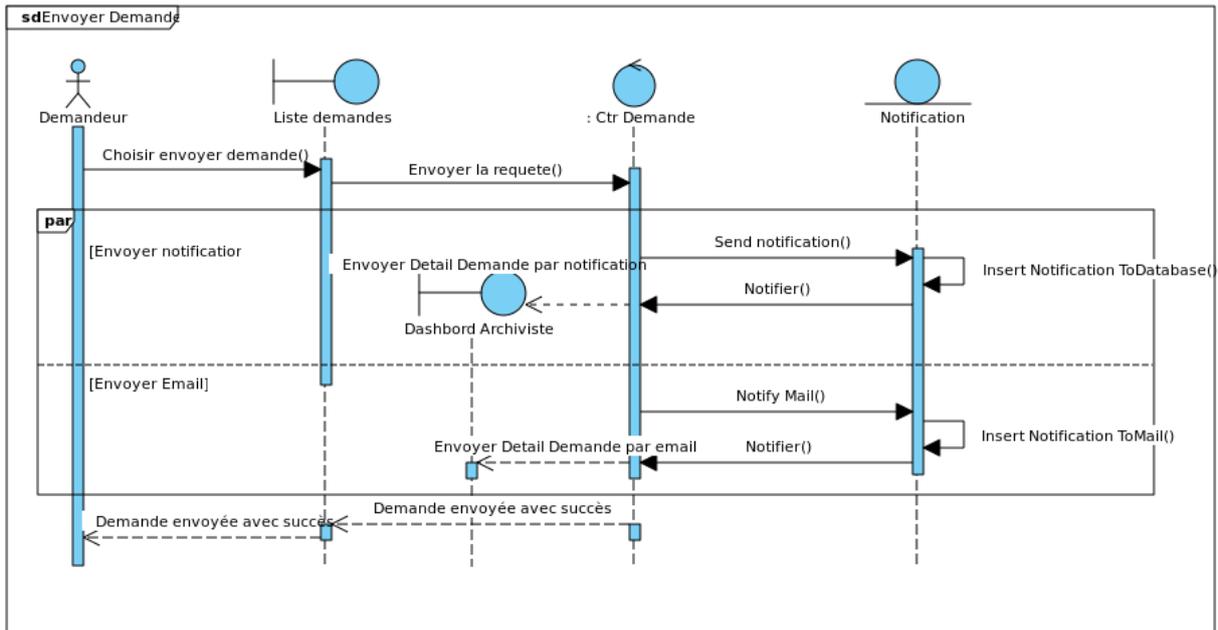


FIGURE 4.6 – Diagramme de séquence du cas d'utilisation "Envoyer demande"

4.8 Diagramme de classe du sprint 2

La figure 4.7 représente le diagramme de classe qui comporte les classes associées aux fonctionnalités du sprint 2 :

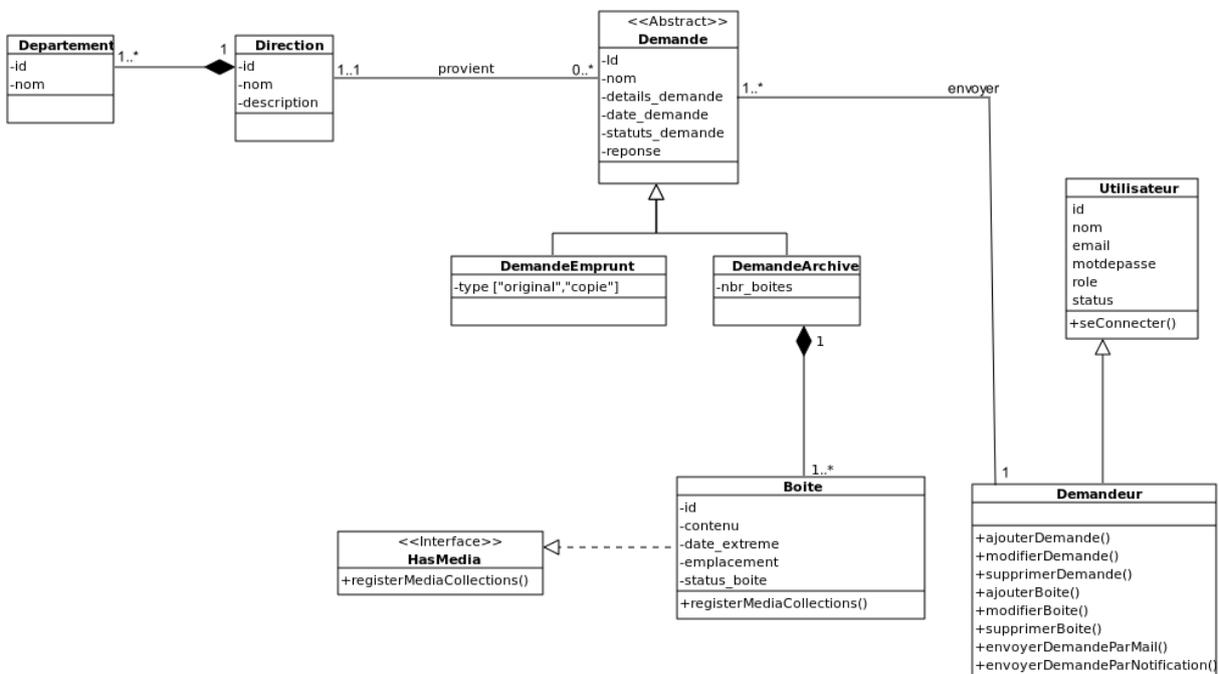


FIGURE 4.7 – Diagramme de classe sprint 2

4.9 Vue des interfaces utilisateurs du sprint 2

4.9.1 Interface ajout demande d'archive

L'interface d'ajout d'une demande d'archive, présentée par la figure 4.8, permet au demandeur appartenant à une direction spécifique d'ajouter une demande d'archive.

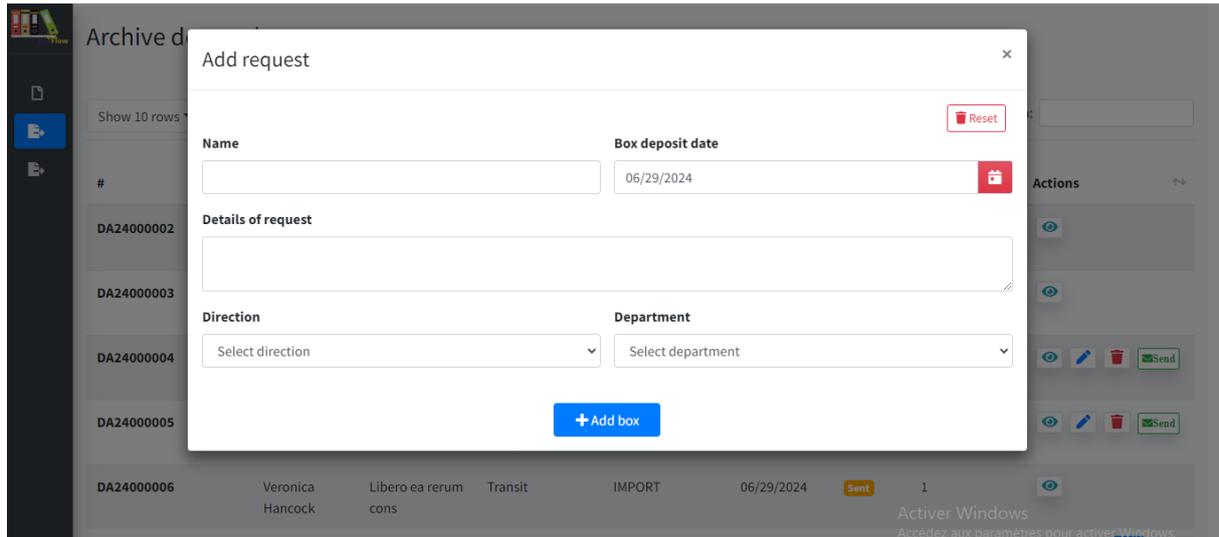


FIGURE 4.8 – Interface ajout demande d'archive

4.9.2 Interface boîte d'archive

L'interface d'ajout d'une boîte d'archive, présentée par la figure 4.9, permet au demandeur d'ajouter des boîtes dans sa demande d'archive, avec la possibilité d'insertion des documents numériques.

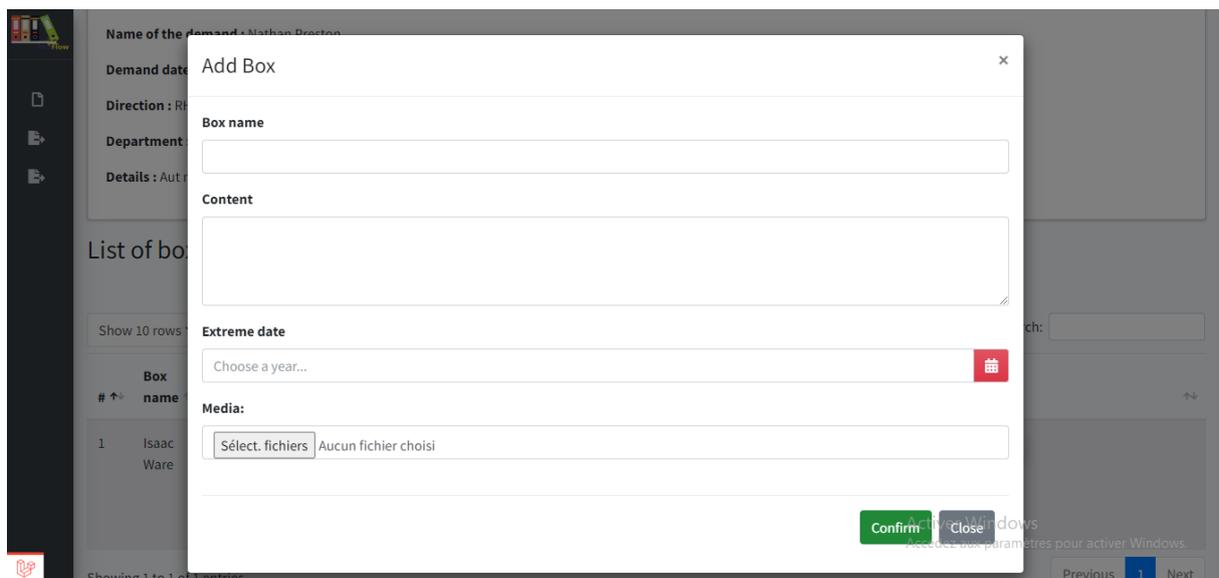


FIGURE 4.9 – Interface d'ajout de boîte d'archive

4.9.2.1 Interface gestion des boites d'archives

L'interface gestion des boites d'archives, présentée par la figure 4.10, permet au demandeur de gérer des boites (ajouter, modifier, supprimer boite) se trouvant dans sa demande d'archive.

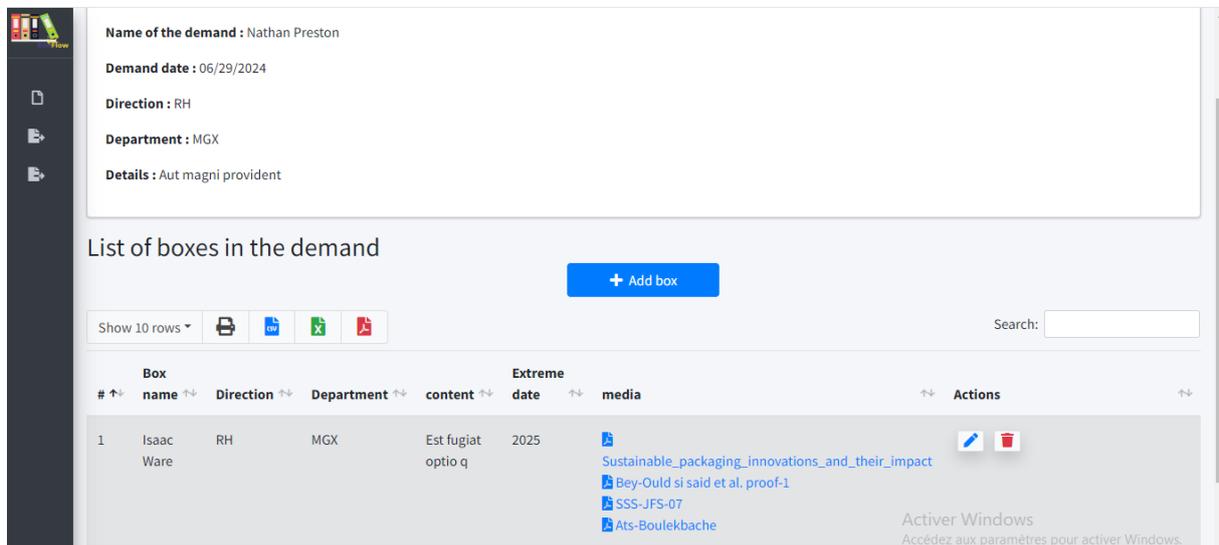


FIGURE 4.10 – Interface de gestion des boites d'archives à l'intérieur d'une demande

4.9.3 Interface gestion des demandes d'archives

L'interface gestion des demandes d'archives, présentée par la figure 4.11, permet au demandeur de gérer sa demande d'archive (modifier, supprimer demande) avec la possibilité d'envoyer sa demande à l'archiviste pour le traitement.

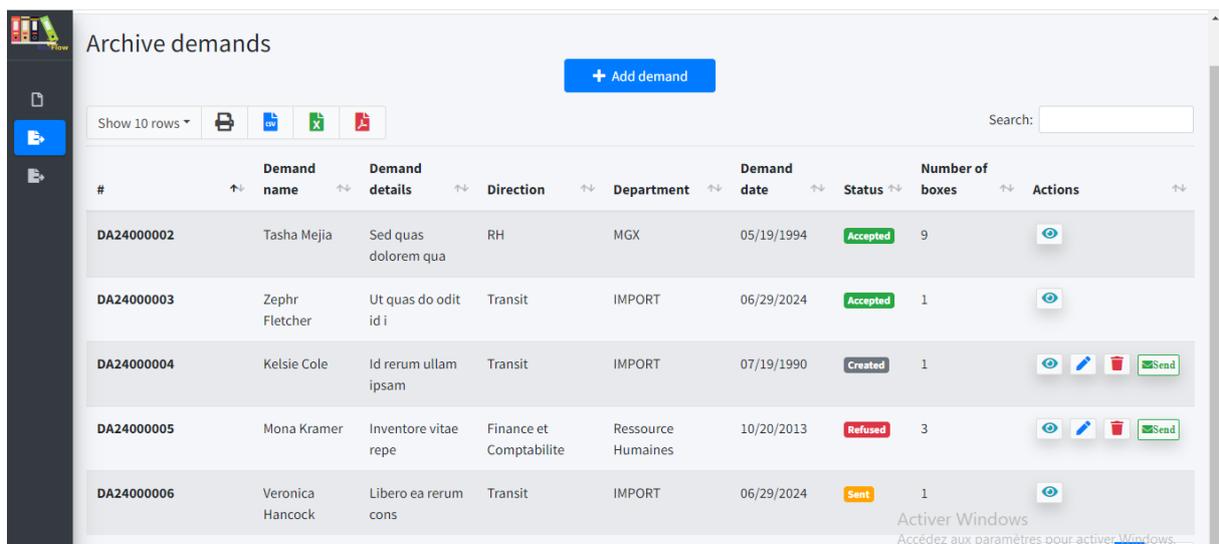


FIGURE 4.11 – Interfaces des demandes d'archives

4.9.4 Interface ajout d'une demande d'emprunt

L'interface ajout d'une demande d'emprunt, présentée par la figure 4.12, permet au demandeur de créer une demande d'emprunt tout en précisant son type (emprunt boîte original ou emprunt d'une copie) qui pourra ensuite être modifiée, supprimée et envoyée à l'archiviste pour le traitement.

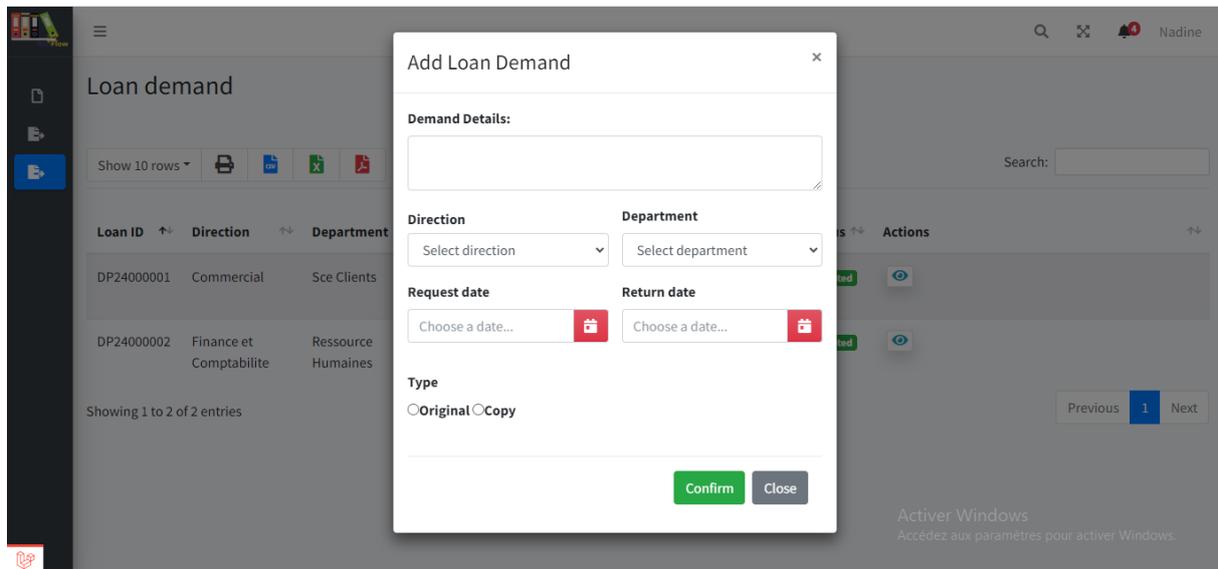


FIGURE 4.12 – Interface ajout d'une demande d'emprunt

4.10 Etude de Sprint 3

4.10.1 Espace Archiviste

Il comprend quatre items réalisés en 2 semaines :

- **Traitement des demandes d'archive** : cela comprend l'acceptation d'une demande d'archivage, le refus de la demande en attribuant un motif de refus au demandeur et l'envoi de la réponse par email et par notification.
- **Traitement des demandes d'emprunt original** : cela comprend l'acceptation d'une demande d'emprunt original, le refus de la demande en attribuant un motif de refus au demandeur de la demande et l'envoi de la réponse par email et par notification.
- **Traitement des demandes d'emprunt d'une copie des documents archivés** : cela comprend l'acceptation d'une demande d'emprunt copie, jonction des documents à la demande acceptée, suppression des documents, le refus de la demande en attribuant un motif de refus au demandeur et l'envoi de la réponse par email et par notification.

- **Recherche sur l'interface archiviste** : Cela comprend la recherche des directions, départements, emplacements (salle, rayon, colonne, étagère), recherche des demandes après traitement par leur statut (created, sent, accepted, rejected), et la recherche des boites contenues dans les demandes par leurs libellés et autres champs se trouvant dans la demande.

4.11 Diagramme de cas d'utilisation du sprint 3

Le diagramme de cas d'utilisation de la figure 4.13 décrit d'une façon globale les cas d'utilisations constituant les items du troisième sprint :

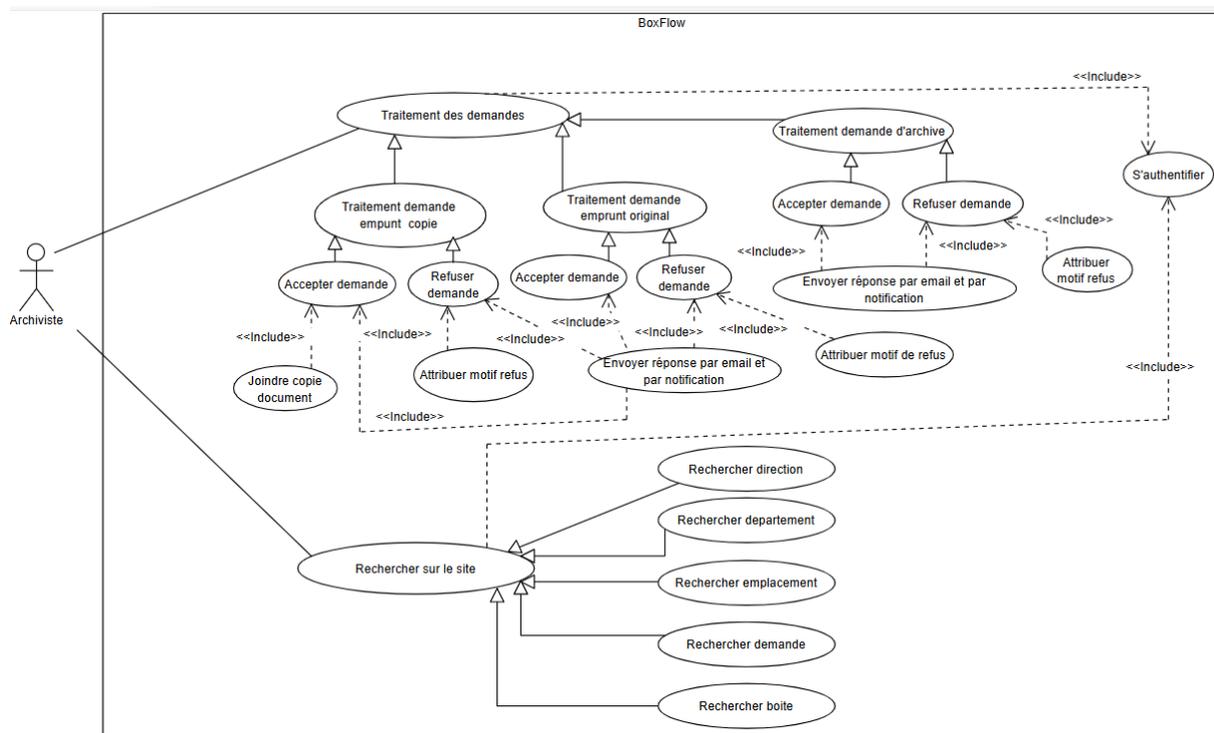


FIGURE 4.13 – Diagramme de cas d'utilisation du sprint 3

4.12 Cas d'utilisation "Traitement des demandes"

4.12.1 Description textuelle du cas d'utilisation "Traitement demande d'archive"

Le tableau 4.6 représente la description textuelle du cas d'utilisation "Traitement demande d'archive" pour l'archiviste.

TABLE 4.6 – Description textuelle des cas d'utilisation "Traitement demande d'archive"

CU	Traitement demande d'archive
BUT	Ce cas permet à l'archiviste de traiter une demande d'archive.
Acteur Principal	Archiviste
Acteur Secondaire	/
Pré-condition	L'archiviste est déjà connecté à son espace. La demande d'archive est déjà ajoutée et envoyée par notification et par mail.
Scénario nominal	<ol style="list-style-type: none"> 1. L'archiviste accède à l'icône de notification de son espace ou bien à son compte Gmail. 2. L'archiviste clique sur la notification reçue. 3. Le système affiche la demande envoyée. 4.1 L'archiviste choisit d'accepter la demande. <ol style="list-style-type: none"> 4.1.1. Le système met à jour le statut de la demande à "accepted". 4.1.2. L'archiviste procède à l'archivage de la boîte en lui attribuant un emplacement. 4.2. L'archiviste choisit de refuser la demande. <ol style="list-style-type: none"> 4.2.1. Le système affiche un formulaire pour attribuer un motif de refus. 4.2.2. L'archiviste remplit le formulaire en attribuant un motif de refus. 4.2.3. L'archiviste valide le formulaire. 4.2.4. Le système vérifie les champs du formulaire. 4.2.5. Le système met à jour le statut de la demande à "rejected" avec ajout du motif de refus à la demande. 5. Le système notifie le demandeur de la réponse à sa demande par une notification sur son compte mail et sur l'interface de l'application.
Enchaînement alternatif	4.2.4.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 4.2.1.
Post-condition	La demande d'archivage est acceptée. OU La demande d'archivage est refusée.

4.12.2 Description textuelle du cas d'utilisation "Traitement demande d'emprunt"

Le tableau 4.7 représente la description textuelle du cas d'utilisation "Traitement demande d'emprunt pour l'archiviste.

TABLE 4.7 – Description textuelle des cas d'utilisation "Traitement demande d'emprunt"

CU	Traitement demande d'emprunt
-----------	------------------------------

BUT	Ce cas permet à l'archiviste de traiter une demande d'emprunt type original ou copie.
Acteur Principal	Archiviste
Acteur Secondaire	/
Pré-condition	L'archiviste est déjà connecté à son espace. La demande d'emprunt est déjà ajoutée et envoyée par une notification et par un email.
Scénario nominal	<ol style="list-style-type: none"> 1. L'archiviste accède à l'icône de notification de son espace ou bien à son compte mail. 2. L'archiviste clique sur la notification reçue. 3. Le système affiche la demande envoyée. 4.1. L'archiviste choisit de refuser la demande. <ol style="list-style-type: none"> 4.1.1. Le système affiche un formulaire pour attribuer un motif de refus. 4.1.2. L'archiviste remplit le formulaire en attribuant un motif de refus. 4.1.3. L'archiviste valide le formulaire. 4.1.4. Le système vérifie les champs du formulaire. 4.1.5. Le système met à jour le statut de la demande à "rejected" avec ajout de motif de refus à la demande. 4.2. L'archiviste choisit d'accepter la demande. 4.3. Le système vérifie le type de la demande. <ol style="list-style-type: none"> 4.3.1. Si la demande est de type original : <ol style="list-style-type: none"> 4.3.1.1 Le système met à jour le statut de la demande à "accepted". 4.3.2. Si la demande est de type copie : <ol style="list-style-type: none"> 4.3.2.1. Le système affiche un onglet pour joindre des documents à la demande acceptée. 4.3.2.2. L'archiviste choisit ajouter document et ajoute les documents nécessaires. 4.3.2.3. L'archiviste valide le formulaire. 4.3.2.4. Le système vérifie les champs du formulaire. 4.3.2.5. Le système ajoute les documents à la demande acceptée. 5. Le système notifie le demandeur de la réponse à sa demande par une notification sur son compte mail et sur son interface.

Enchaînement alternatif	4.3.2.4.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 4.3.2.1.
Post-condition	La demande d'emprunt est traitée (acceptée ou refusée).

4.13 Diagramme de séquence du cas d'utilisation "Traitement des demandes"

La figure [4.14](#) représente le diagramme de séquence de cas d'utilisation "Traitement des demandes" pour l'archiviste.

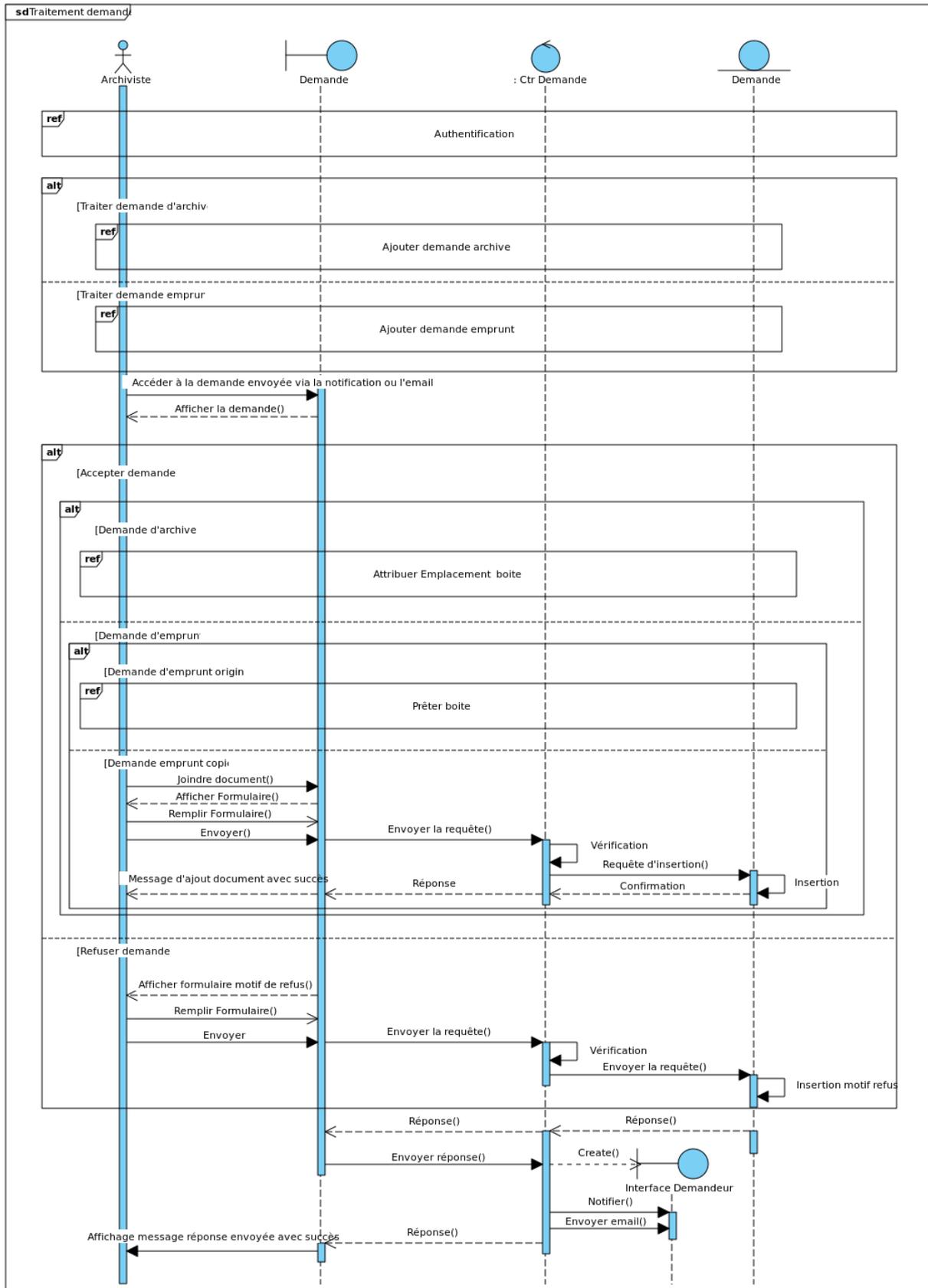


FIGURE 4.14 – Diagramme de séquence du cas d'utilisation "Traitement des demandes"

4.14 Diagramme de classe du sprint 3

La figure 4.15 représente le diagramme de classe qui comporte les classes associées aux fonctionnalités du sprint 3 :

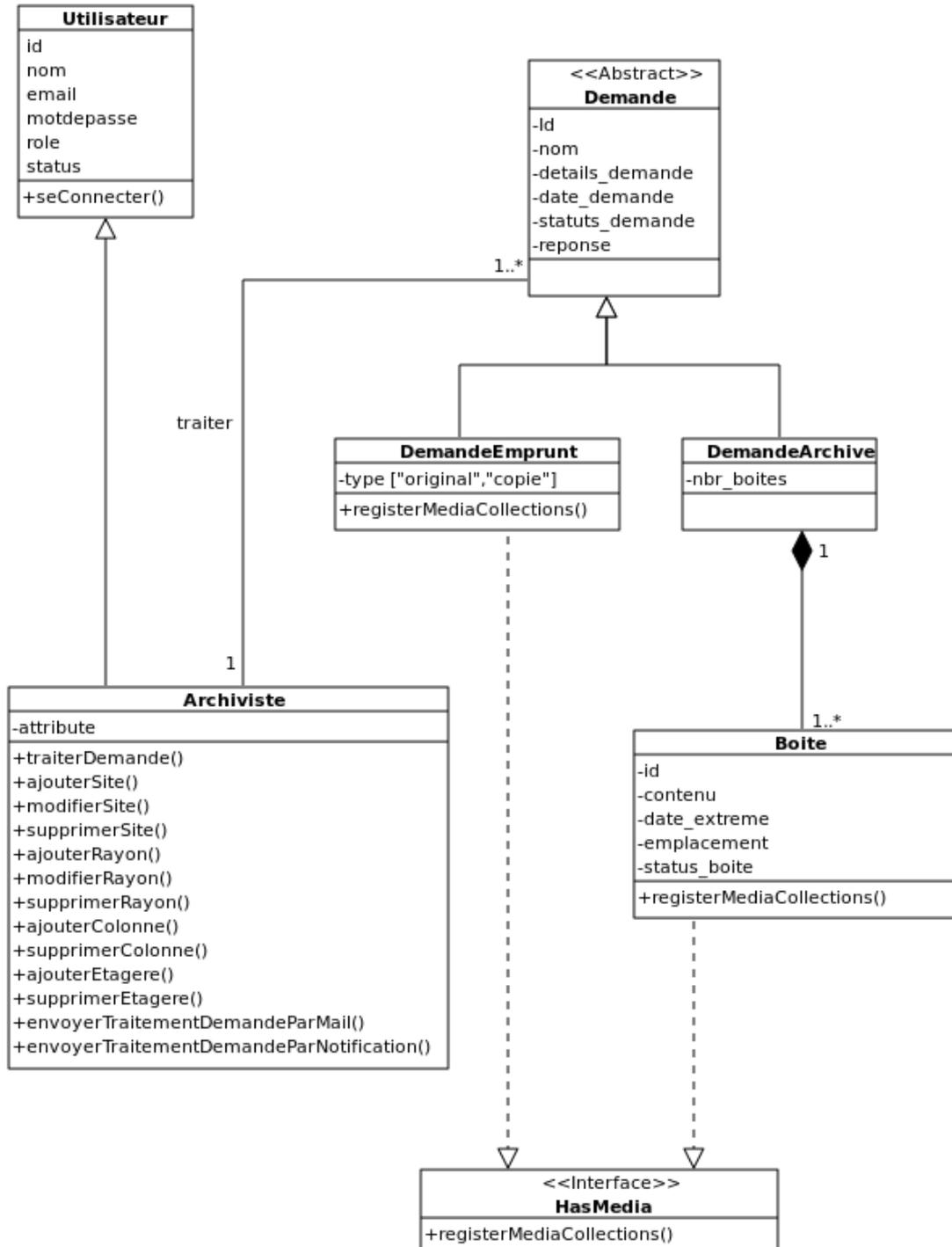


FIGURE 4.15 – Diagramme de classe sprint 3

4.15 Vue des interfaces utilisateurs du sprint 3

4.15.1 Interface traitement d'une demande d'archive

L'interface de traitement des demandes d'archivage, illustrée par la figure 4.16, permet à l'archiviste de gérer les demandes reçues dans son espace et notifiées par courrier électronique. Il peut ainsi accepter ou refuser une demande, en fournissant un motif de refus si la demande est refusée.

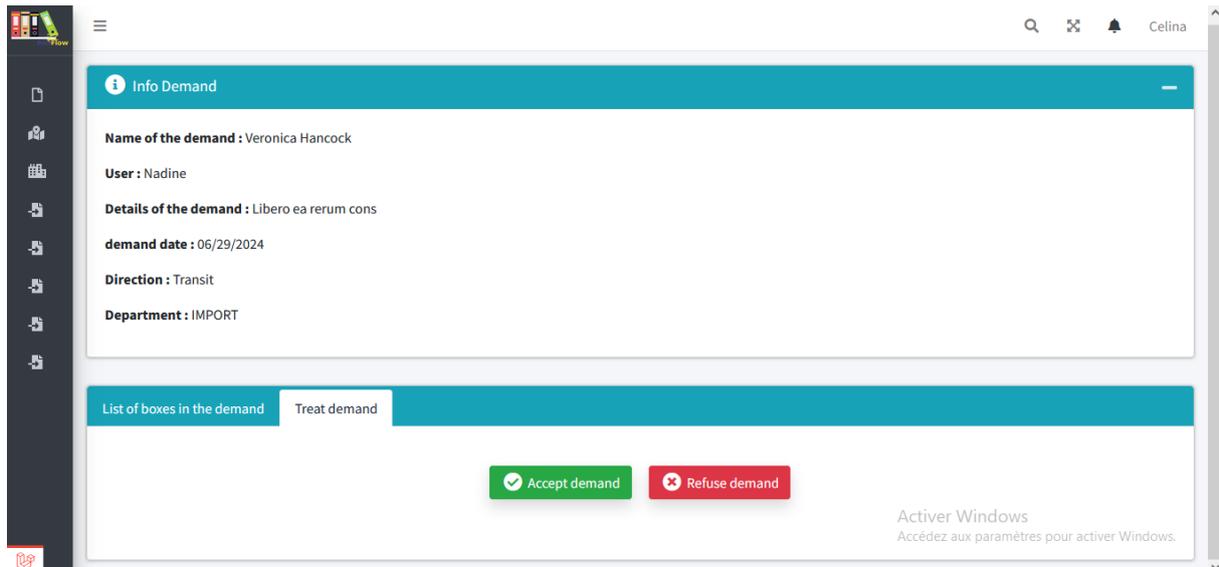


FIGURE 4.16 – Interface traitement d'une demande d'archive

4.15.2 Interface traitement d'une demande d'emprunt type copie

L'interface de traitement d'une demande d'emprunt copie, présentée par la figure 4.17, permet à l'archiviste de traiter une demande d'emprunt reçue dans son espace et notifiée par courrier électronique. Il peut alors accepter ou refuser la demande, en ajoutant éventuellement un motif de refus en cas de refus, et en insérant des fichiers téléchargeables en cas d'acceptation.

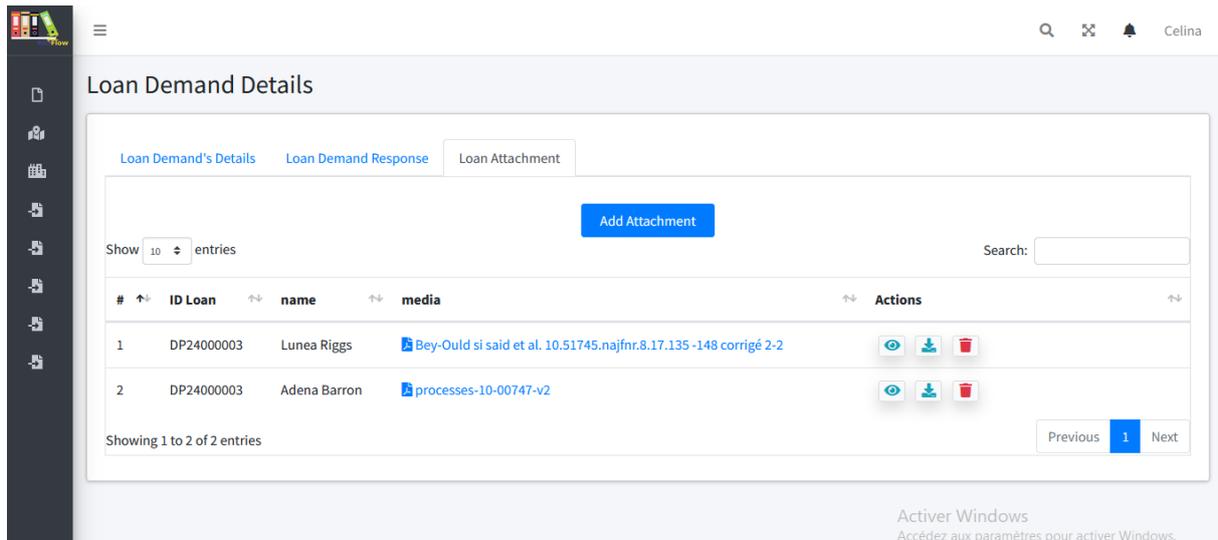


FIGURE 4.17 – Interface traitement d'une demande d'emprunt type copie

4.16 Conclusion

Dans ce chapitre, nous avons présenté en détail les deux sprints qui ont constitué une release soumise aux parties prenantes pour revue et validation. Nous avons détaillé les fonctionnalités de chaque sprint, représentant les deux espaces : archiviste et demandeur. Dans le prochain chapitre, nous détailleront le déroulement des sprint 4 et 5, en continuons à développer les fonctionnalités du système et en consolidant progressivement son évolution.

Chapitre 5

Sprint 4, 5

5.1 Introduction

Dans ce chapitre nous allons continuer le développement de notre application en explorant les sprints 4 et 5. Pour chaque sprint, nous présenteront des diagrammes de cas d'utilisation, des descriptions textuelles et des diagrammes de séquence afin d'illustrer les divers fonctionnalités et interactions du système, tout en détaillant le fonctionnement de l'application. Enfin, nous présenterons quelques interfaces du résultat obtenue dans ces sprints.

5.2 Etude du Sprint 4

5.2.1 Espace Demandeur

Il comprend un item réalisé en 1 semaines à savoir :

La consultation des demandes traitées. Cela inclut la réception de la demande traitée via une notification dans l'espace demandeur et une notification sur son compte mail. En cas de refus de la demande, le motif de refus sera affiché, et il sera possible de soumettre une nouvelle demande pour remédier au problème causant le refus. Pour une demande acceptée, le demandeur pourra visualiser l'acceptation et afficher ou télécharger les documents reçus dans le cas d'une demande d'emprunt de type copie.

5.2.2 Espace Archiviste

Il comprend également un seul item réalisé en 1 semaine qui est :

L'archivage d'une boîte. Cela inclut l'attribution d'emplacement pour une boîte d'archive sur une étagère, dans une colonne d'un rayon, et dans une salle précise, tout en respectant la capacité des étagères, ainsi qu'une possibilité de modification de cet emplacement après l'avoir attribué.

5.2.3 Diagramme de cas d'utilisation du sprint 4 et 5

Le diagramme de cas d'utilisation de la figure 5.1 décrit d'une façon globale les cas d'utilisations constituant les items du quatrième et cinquième sprint :

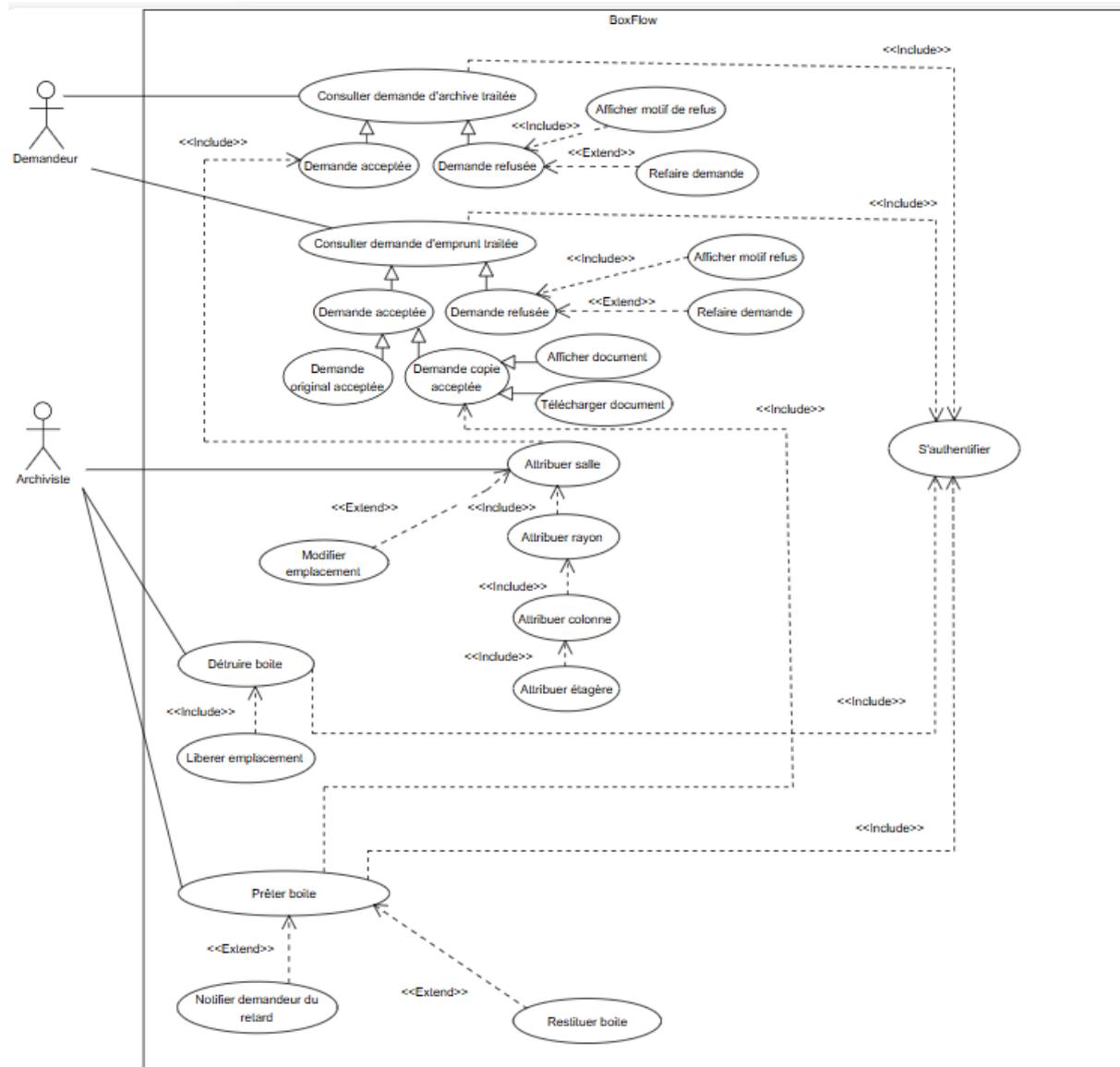


FIGURE 5.1 – Diagramme de cas d'utilisation du sprint 4 et 5

5.2.4 Cas d'utilisation "Attribuer emplacement boîte"

5.2.4.1 Description textuelle des cas d'utilisation "Attribuer emplacement boîte"

Le tableau 5.1 représente la description textuelle du cas d'utilisation "Attribuer emplacement boîte" pour l'archiviste.

TABLE 5.1 – Description textuelle des cas d'utilisation "Attribuer emplacement boîte"

CU	Attribuer emplacement boîte
BUT	Ce cas permet à l'archiviste d'archiver une boîte d'archive en lui attribuant un emplacement.
Acteur Principal	Archiviste
Acteur Secondaire	/
Pré-condition	Le demandeur est déjà connecté à son espace. La demande d'archivage est déjà acceptée.
Scénario nominal	<ol style="list-style-type: none"> 1. L'archiviste accède à son espace. 2. L'archiviste accède à l'interface des boîtes correspondantes aux demandes acceptées et choisit le bouton "attribuer emplacement boîte". 3. Le système affiche le formulaire. 4. L'archiviste sélectionne la salle. 5. Le système retourne la liste des rayons correspondants à la salle sélectionnée. 6. L'archiviste sélectionne le rayon. 7. Le système retourne la liste des colonnes correspondantes au rayon sélectionné. 8. L'archiviste sélectionne la colonne. 9. Le système retourne la liste des étagères correspondantes à la colonne sélectionnée. 10. L'archiviste sélectionne l'étagère. 11. L'archiviste valide le formulaire. 12. Le système vérifie la capacité de l'étagère sélectionnée. 13. Le système ajoute l'emplacement pour la boîte.
Enchaînement alternatif	<ol style="list-style-type: none"> 12.a Si l'étagère est pleine, reprendre à partir de 9. 12.b Si toutes les étagères sont pleines alors reprendre à partir de 7. 12.c Si toutes les colonnes sont pleines alors reprendre à partir de 6. 12.d Si tous les rayons sont pleins alors reprendre à partir de 4.
Post-condition	L'emplacement est ajouté avec succès.

5.2.4.2 Diagramme de séquence du cas d'utilisation "Attribuer emplacement boîte"

La figure 5.2 représente le diagramme de séquence de cas d'utilisation "Attribuer emplacement boîte" pour l'archiviste.

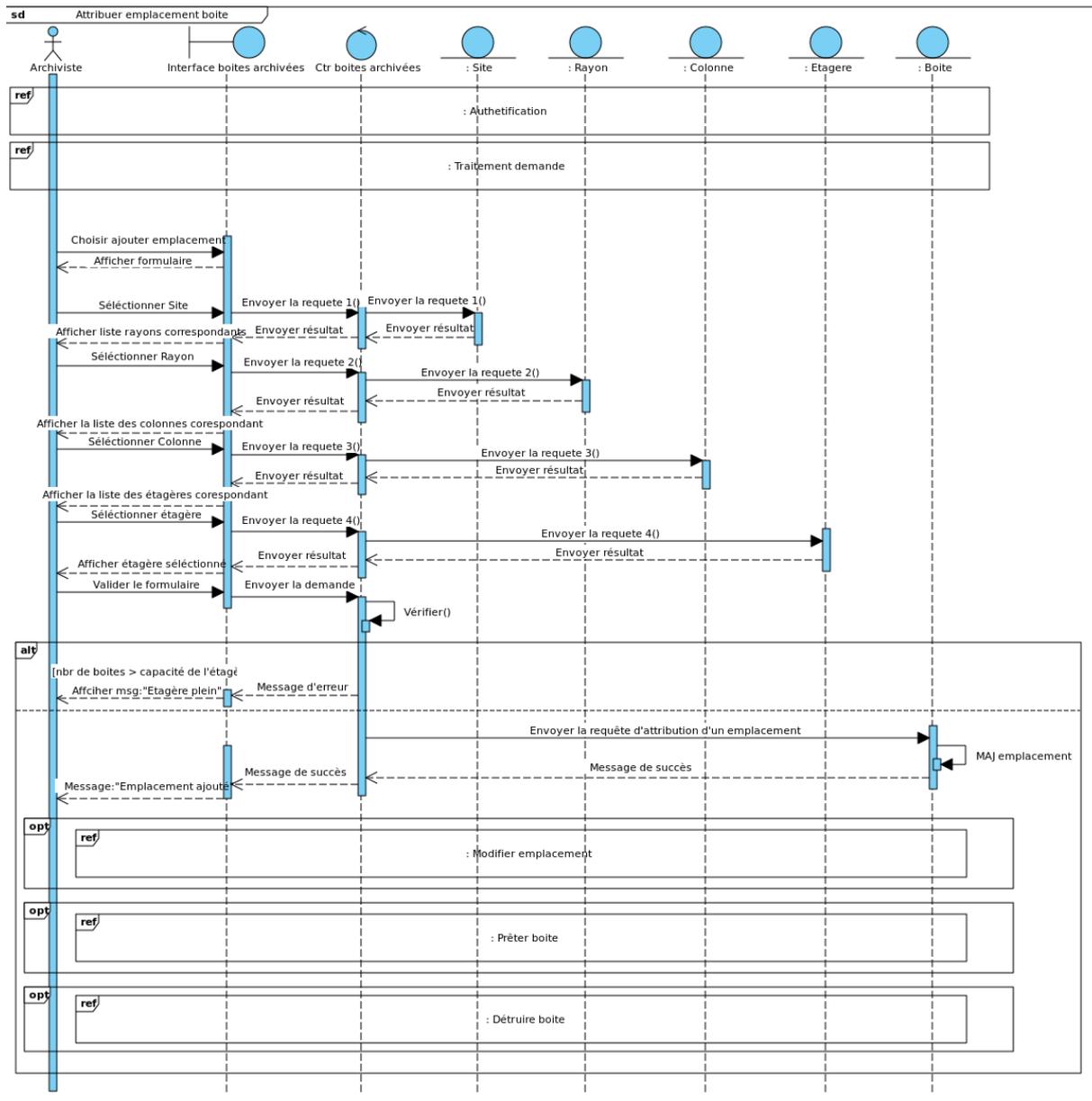


FIGURE 5.2 – Diagramme de séquence du cas d'utilisation "Attribuer emplacement boîte"

5.2.5 Cas d'utilisation "Modifier emplacement"

5.2.5.1 Description textuelle des cas d'utilisation "Modifier emplacement"

Le tableau 5.2 représente la description textuelle du cas d'utilisation "Modifier emplacement" pour l'archiviste.

TABLE 5.2 – Description textuelle des cas d'utilisation "Modifier emplacement"

CU	Modifier emplacement
BUT	Ce cas permet à l'archiviste de modifier l'emplacement d'une boîte déjà archivée.
Acteur Principal	Archiviste
Acteur Secondaire	/
Pré-condition	L'archiviste est déjà connecté à son espace. La boîte possède déjà un emplacement.
Scénario nominal	<ol style="list-style-type: none"> 1. L'archiviste accède à son espace. 2. L'archiviste accède à l'interface des boîtes archivées et choisit le bouton "Modifier emplacement". 3. Le système affiche le formulaire tout en récupérant l'ancien emplacement choisi. 4. L'archiviste sélectionne la salle. 5. Le système retourne la liste des rayons correspondants à la salle sélectionnée. 6. L'archiviste sélectionne le rayon. 7. Le système retourne la liste des colonnes correspondantes au rayon sélectionné. 8. L'archiviste sélectionne la colonne. 9. Le système retourne la liste des étagères correspondantes à la colonne sélectionnée. 10. L'archiviste sélectionne l'étagère. 11. L'archiviste valide le formulaire. 12. Le système vérifie la capacité de l'étagère sélectionnée. 13. Le système modifie l'emplacement pour la boîte.
Enchaînement alternatif	<ol style="list-style-type: none"> 12.a Si l'étagère est pleine, reprendre à partir de 9. 12.b Si toutes les étagères sont pleines alors reprendre à partir de 7. 12.c Si toutes les colonnes sont pleines alors reprendre à partir de 6. 12.d Si tous les rayons sont pleins alors reprendre à partir de 4.
Post-condition	L'emplacement est modifié avec succès.

5.2.5.2 Diagramme de séquence du cas d'utilisation "Modifier emplacement"

La figure 5.3 représente le diagramme de séquence du cas d'utilisation "Modifier emplacement" par l'archiviste, en mettant en évidence uniquement les éléments supplémentaires par rapport à l'attribution de l'emplacement, à savoir le chargement de l'ancien emplacement.

Ci-dessous se trouve la définition des requêtes présentes dans le diagramme de séquence :

Requête 1() : récupérer l'id de la boîte sélectionnée.

Requête 2() : récupérer le site sélectionné lors de l'attribution d'un emplacement.

Requête 3() : récupérer les rayons correspondants au site sélectionné lors de l'attribution d'un emplacement.

Requête 4() : récupérer les colonnes correspondantes au rayon sélectionné lors de l'attribution d'un emplacement.

Requête 5() : récupérer les étagères correspondantes à la colonne sélectionnée lors de l'attribution d'un emplacement.

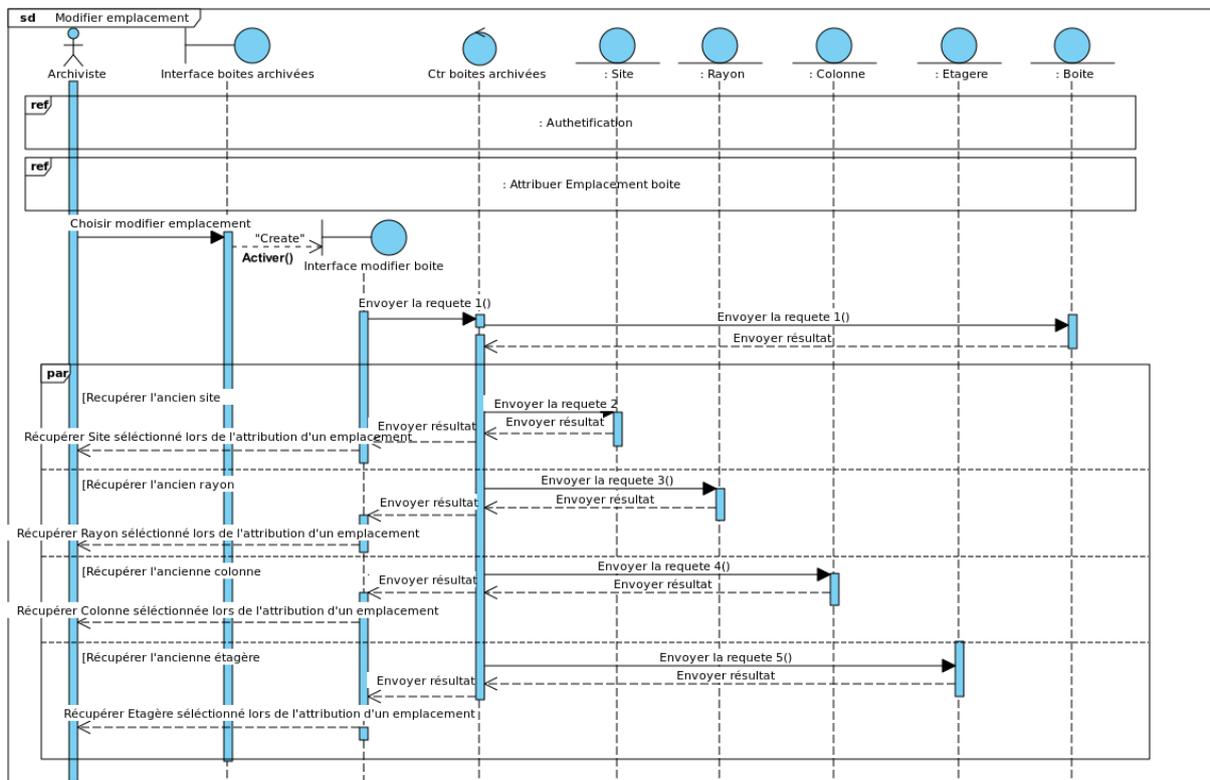


FIGURE 5.3 – Diagramme de séquence du cas d'utilisation "Modifier emplacement"

5.3 Diagramme de classe du sprint 4 et 5

La figure 5.4 représente le diagramme de classe qui comporte les classes associées aux fonctionnalités du sprint 4 et 5 :

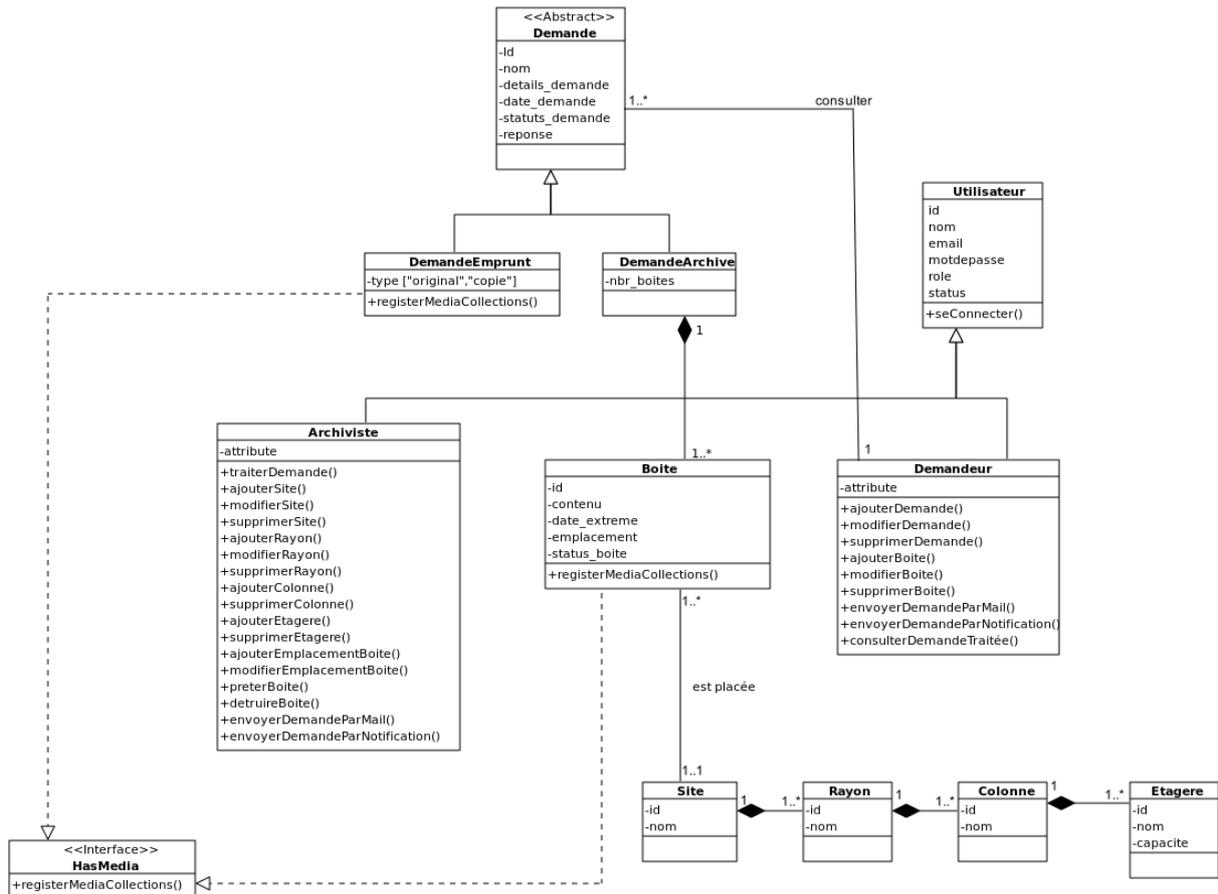


FIGURE 5.4 – Diagramme de classe sprint 4 et 5

5.3.1 Vues des interfaces utilisateurs

5.3.1.1 Interface Consulter une demande d'emprunt traitée

L'interface de consultation d'une demande d'emprunt traitée, présentée par la figure 5.5, permet au demandeur de consulter sa demande traitée par l'archiviste, reçue par email et notifiée dans son espace. Il a également la possibilité de refaire sa demande en cas de refus.

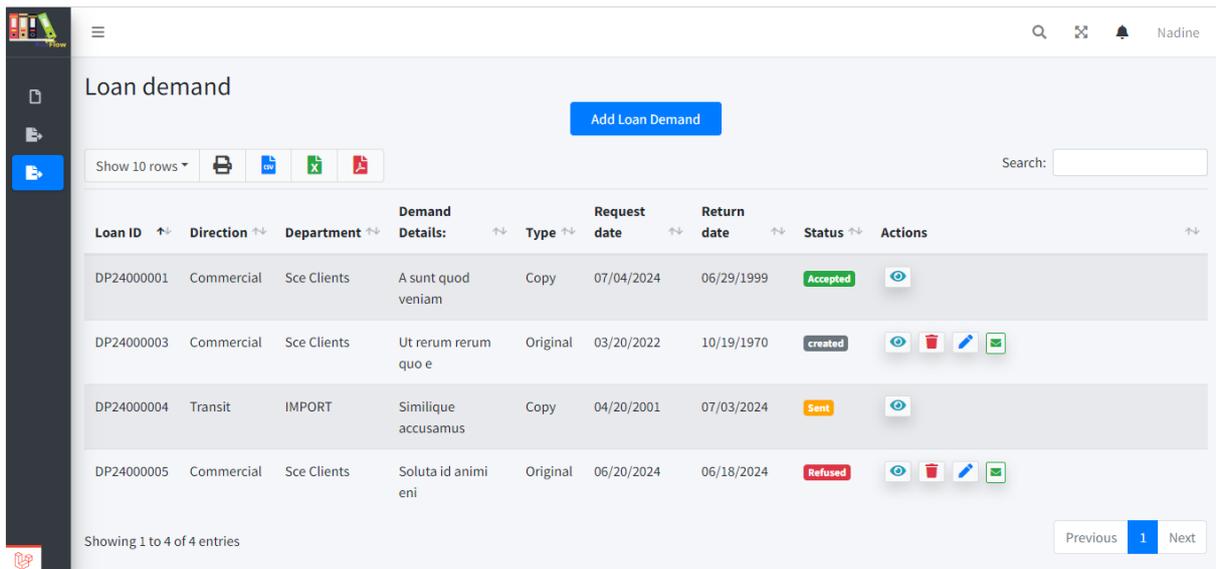


FIGURE 5.5 – Interface Consulter une demande d'emprunt traitée

5.3.1.2 Interface attribuer emplacement boite

L'interface attribuer emplacement boite, présentée par la figure 5.6 suivante, permet à l'administrateur d'archiver une boite dans une étagère d'un rayon d'une colonne d'une salle déjà ajoutée.

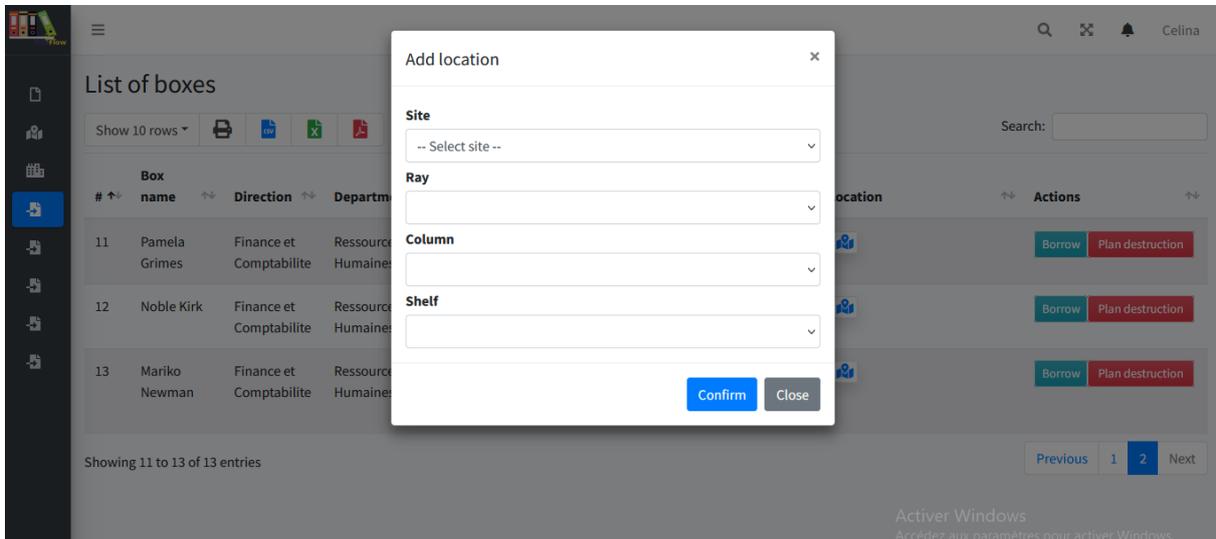


FIGURE 5.6 – Interface attribuer emplacement boite

5.3.1.3 Interface modifier emplacement boite

L'interface modifier emplacement boite, présentée par la figure 5.7, permet à l'administrateur de modifier l'emplacement d'une boite déjà archivée.

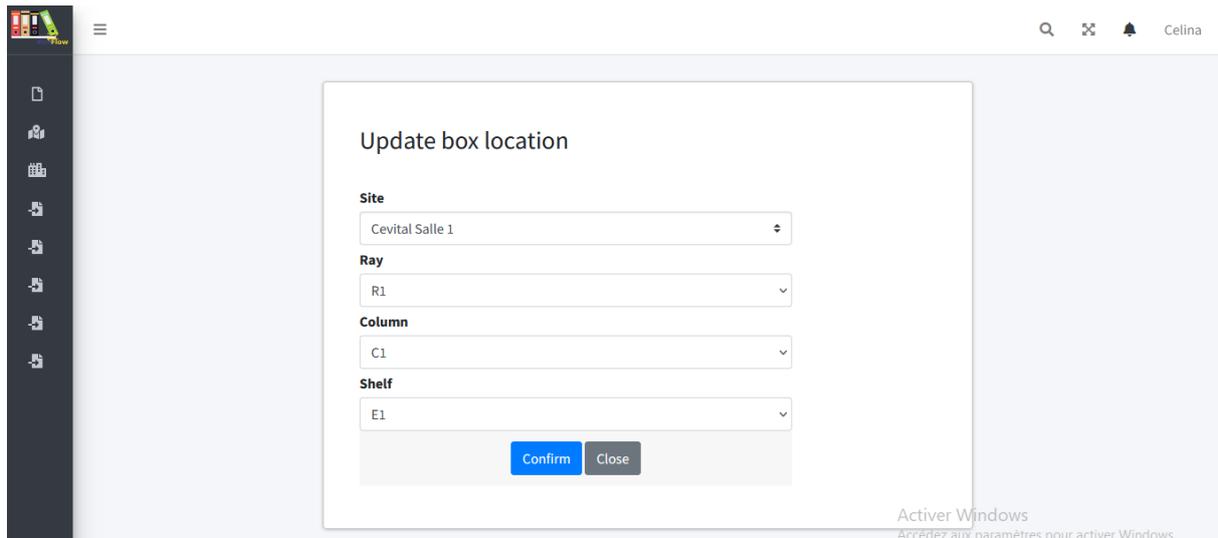


FIGURE 5.7 – Interface modifier emplacement boîte

5.4 Etude du Sprint 5

5.4.1 Espace Archiviste

Il comprend quatre items réalisés en 2 semaines :

- **Prêter boîte** : Cela inclut le processus de prêt d'une boîte à un demandeur après acceptation de sa demande d'emprunt d'une boîte originale, tout en gardant la trace des transactions effectuées (libellé boîte, date d'emprunt, date retour prévu).
- **Gestion boîtes empruntées et en retard** : lorsque le demandeur tarde à remettre la boîte empruntée, une notification d'alerte lui sera envoyée.
- **Gestion des boîtes restituées** : lorsque le demandeur restitue la boîte qu'il a empruntée, l'archiviste procède à la remise en place de la boîte, qui sera alors disponible pour un nouvel emprunt. La trace de la transaction sera mise à jour avec la date de retour réelle de la boîte.
- **Gestion des boîtes détruites** : Cela comprend la planification de la destruction des boîtes ayant déjà une date de destruction illimitée, lesquelles seront déplacées vers la liste des boîtes en attente de destruction, puis détruites. Leur emplacement sera alors libéré pour recevoir de nouvelles boîtes.

5.4.2 Cas d'utilisation "Prêter boîte"

5.4.2.1 Description textuelle des cas d'utilisation "Prêter boîte"

Le tableau [5.3](#) représente la description textuelle du cas d'utilisation "Prêter boîte" pour l'archiviste.

TABLE 5.3 – Description textuelle des cas d'utilisation "Prêter boîte"

CU	Prêter boîte
BUT	Ce cas permet à l'archiviste de prêter une boîte d'archive.
Acteur Principal	Archiviste
Acteur Secondaire	/
Pré-condition	L'archiviste est déjà connecté à son espace. La boîte possède déjà un emplacement. La demande d'emprunt d'une boîte original déjà acceptée.
Scénario nominal	<ol style="list-style-type: none"> 1. L'archiviste accède à son espace. 2. L'archiviste accède à l'interface des boîtes archivées et choisit de Prêter la boîte. 3. Le système affiche le formulaire. 4. L'archiviste sélectionne la demande d'emprunt original acceptée. 5. Le système affiche la date d'emprunt et restitution correspondante à la demande sélectionnée. 6. L'archiviste valide le formulaire. 7. Le système vérifie les champs du formulaire. 8. Le système affiche le message "emprunt effectué avec succès" et le statut de la boîte passe à "non disponible". 9. Le système ajoute la boîte à la liste des boîtes empruntées. 10. Le système insère les détails de la demande (libellé, date emprunt, date restitution prévue) dans l'interface "mouvement boîte".
Scénario nominal (restitution)	<ol style="list-style-type: none"> 1. Le demandeur restitue la boîte empruntée. 2. L'archiviste accède à l'interface des boîtes empruntées et choisit de restituer la boîte pour enregistrer la restitution. 3. Le système affiche un formulaire de date restitution réelle. 4. L'archiviste saisit la date de retour réelle. 5. Le système ajoute la date de retour réelle au mouvement de la boîte et met à jour la boîte. 6. Le système supprime la boîte de l'interface des liste boîtes empruntées. 7. Le système change le statut de la boîte à "disponible" et affiche le bouton prêter à nouveau.
Enchaînement alternatif	7.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3.
Enchaînement alternatif (restitution)	Le système vérifie régulièrement si la date de retour prévue pour la demande est dépassée. Si oui, alors l'archiviste envoie une notification d'alerte au demandeur pour restituer la boîte.
Post-condition	La boîte est prêtée avec succès.

5.4.2.2 Diagramme de séquence du cas d'utilisation "Prêter boîte"

La figure 5.8 représente le diagramme de séquence de cas d'utilisation "Prêter boîte" pour l'archiviste.

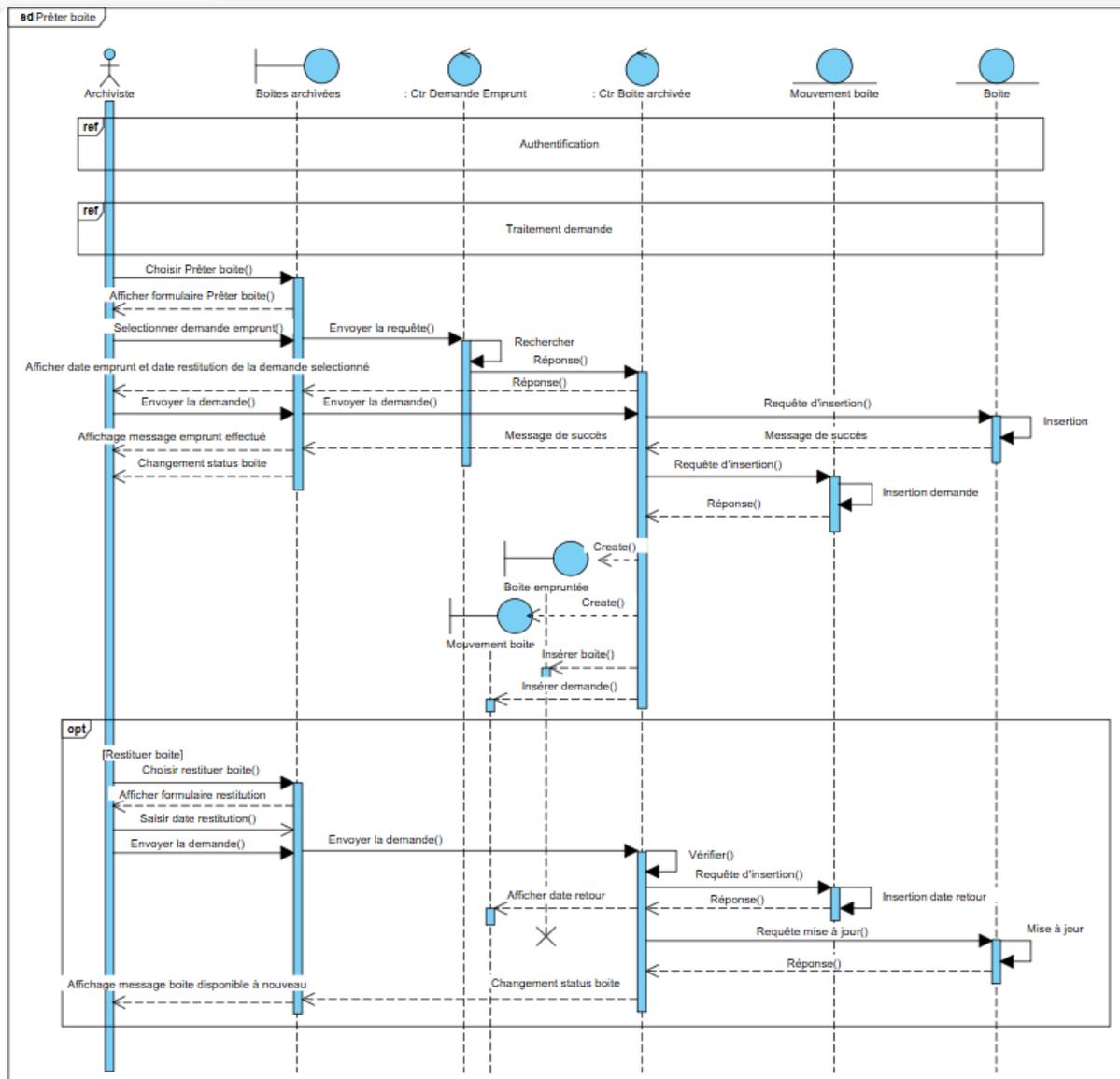


FIGURE 5.8 – Diagramme de séquence du cas d'utilisation "Prêter boîte"

5.4.3 Cas d'utilisation "Détruire boîte"

5.4.3.1 Description textuelle des cas d'utilisation "Détruire boîte"

Le tableau 5.4 représente la description textuelle du cas d'utilisation "Détruire boîte" pour l'archiviste.

TABLE 5.4 – Description textuelle des cas d'utilisation "Détruire boîte"

CU	Détruire boîte
BUT	Ce cas permet à l'archiviste de détruire une boîte d'archive.
Acteur Principal	Archiviste
Acteur Secondaire	/
Pré-condition	L'archiviste est déjà connecté à son espace. La boîte possède déjà un emplacement.
Scénario nominal	<ol style="list-style-type: none"> 1. L'archiviste accède à son espace. 2. L'archiviste accède à l'interface des boîtes archivées et choisit de Détruire la boîte. 3. Le système affiche le formulaire. 4. L'archiviste planifie la date de destruction de la boîte en choisissant une année de destruction dans le calendrier des dates qui apparaît. 5. L'archiviste valide le formulaire. 6. Le système vérifie les champs du formulaire. 7. Le système met à jour la date de destruction de la boîte et copie la boîte dans l'interface des boîtes en attente de destruction. 8. Le système affiche le message "Boîte marquée pour destruction avec succès". 9. L'archiviste accède à l'interface des boîtes en attente de destruction et choisit le bouton "Détruire". 10. Le système supprime la boîte d'archive et libère son emplacement, qui redevient disponible. 11. Le système affiche la boîte supprimée dans l'interface des boîtes détruites avec un champ "deleted_at" montrant sa date de destruction.
Enchaînement alternatif	6.a Si une erreur est produite, le système affiche un message d'erreur. Reprendre à partir de 3.
Post-condition	La boîte est détruite avec succès.

5.4.3.2 Diagramme de séquence du cas d'utilisation "Détruire boîte"

La figure 5.9 représente le diagramme de séquence de cas d'utilisation "Détruire boîte" pour l'archiviste.

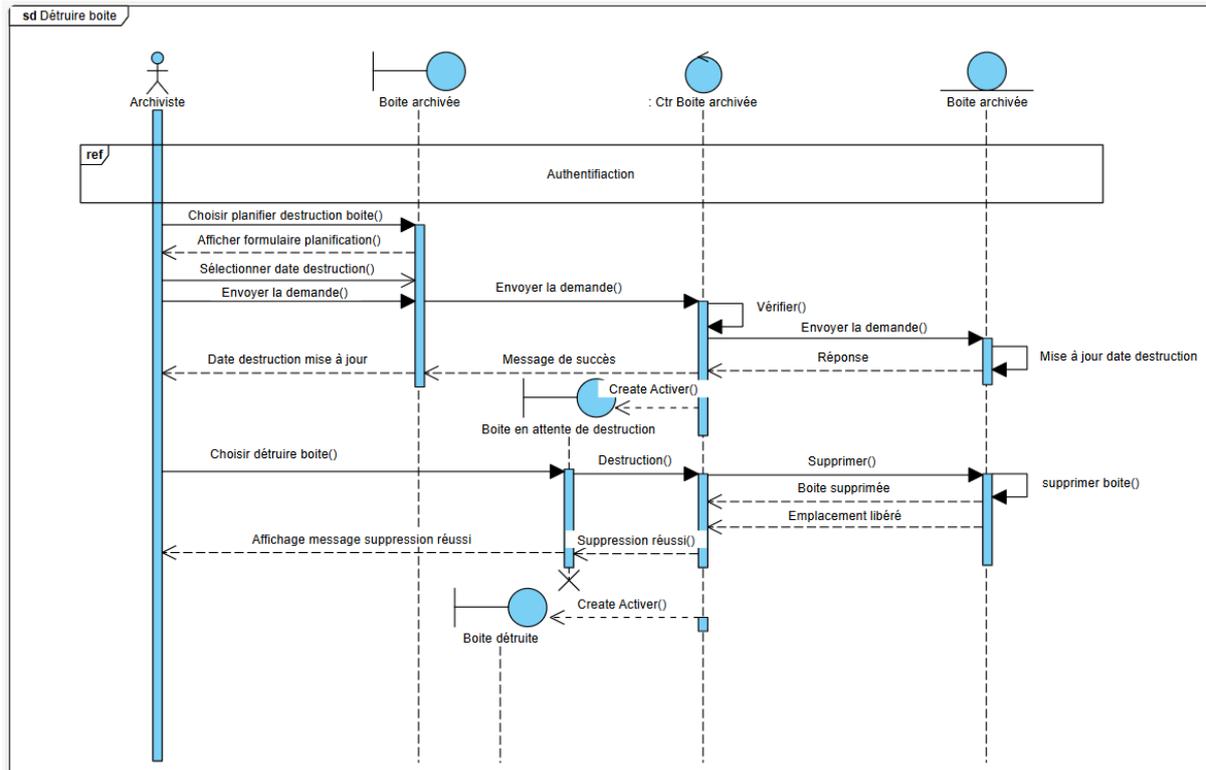


FIGURE 5.9 – Diagramme de séquence du cas d'utilisation "Détruire boîte"

5.4.4 Vues des interfaces utilisateurs du sprint 5

5.4.4.1 Interface de la liste des boites

L'interface de toutes les boites existantes, présentée par la figure 5.10, permet à l'archiviste de récupérer toutes les boites dont la demande d'archivage est acceptée. Sur cette interface, il est possible d'enregistrer un prêt d'une boite d'archive et de planifier sa destruction.

#	Box name	Direction	Department	content	Extreme date	Destruction	Status	Location	Actions
1	Bert Graves	RH	MGX	Id exercitacion eos	2009	UNLIMITED	Available	Cevital Salle 1-R1-C1-E1	Borrow Plan destruction
2	Reese Tyson	RH	MGX	Excepturi ut placeat	2002	UNLIMITED	Available	Cevital Salle 1-R1-C1-E1	Borrow Plan destruction
3	Meghan Spence	RH	MGX	Est ut eius et aut v	2021	UNLIMITED	Available	Cevital Salle 1-R1-C1-E1	Borrow Plan destruction
4	Brock Stephenson	RH	MGX	Explicabo Nisi Nam	2021	UNLIMITED	Available	Cevital Salle 1-R1-C1-E1	Borrow Plan destruction
5	Mary Mcrray	RH	MGX	Dolores dicta omnis	2026	UNLIMITED	Available	Cevital Salle 1-R1-C1-E1	Borrow Plan destruction

FIGURE 5.10 – Interface de toutes les boites

5.4.4.2 Interface prêter boite

L'interface de prêter boite, présentée par la figure 5.11, permet à l'archiviste de prêter une boite d'archive à un demandeur en sélectionnant la demande parmi les demandes emprunt original acceptées.

Borrow archive box [X]

Loan demand number

Transmission date
 [Calendar icon]

Expected return date
 [Calendar icon]

FIGURE 5.11 – Interface prêter boite

5.4.4.3 Interface des boites empruntées

L'interface des boites empruntées, présentée par la figure 5.12, permet à l'archiviste de récupérer toutes les boites empruntées avec la possibilité de restitution des boites.

#	Box name	Department	Status	Location	Loan demand number	Transmission date	Expected return date	Overdue	Actions
1	Bert Graves	MGX	Not available	Cevital Salle 1-R1-C1-E1	DP24000006	06/20/2024	06/18/2024	Overdue	Return
2	Reese Tyson	MGX	Not available	Cevital Salle 1-R1-C1-E1	DP24000007	07/05/2024	06/19/2024	Overdue	Return

FIGURE 5.12 – Interface boites empruntées

5.4.4.4 Interface mouvement de la boite

L’interface mouvement de la boite, présentée par la figure 5.13, permet à l’archiviste de garder la trace des emprunts effectués.

#	Request Number	Transmission Date	Return Date	Real Return Date	Status
30	DP24000006	06/20/2024	06/18/2024		Not available
31	DP24000007	07/05/2024	06/19/2024		Not available
32	DP24000006	06/20/2024	06/18/2024	06/06/2024	Available
33	DP24000007	07/05/2024	06/19/2024	06/07/2024	Available

FIGURE 5.13 – Interface mouvement de la boite

5.4.4.5 Interface planifier la destruction d’une boite

L’interface planifier la destruction d’une boite, présentée par la figure 5.14, permet à l’archiviste de planifier la destruction d’une boite dont la date de destruction est illimitée.

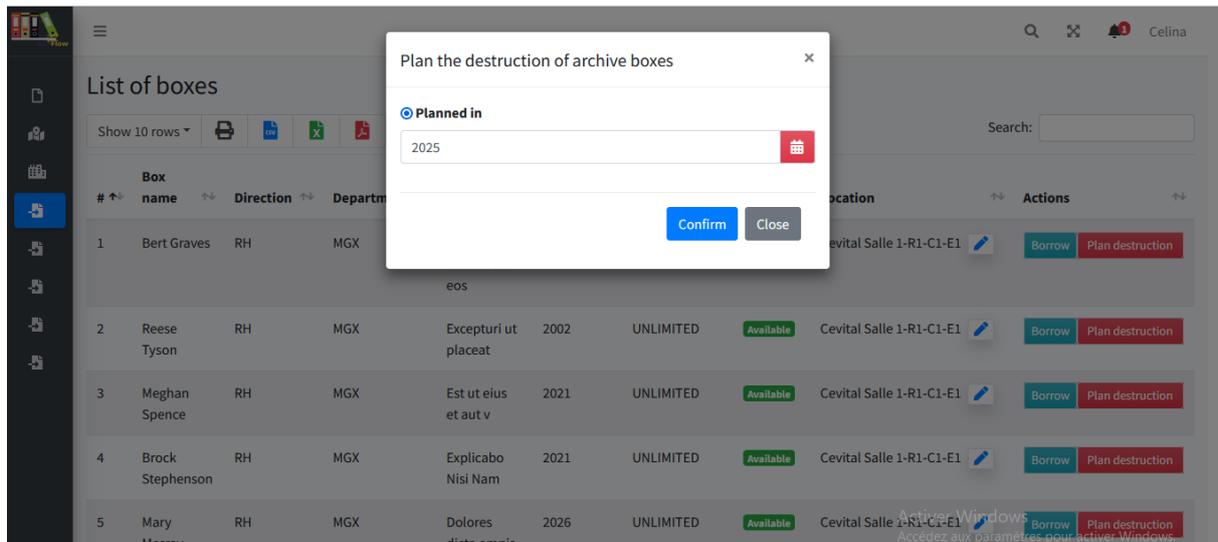


FIGURE 5.14 – Interface planifier la destruction d’une boîte

5.4.4.6 Interface des boîtes détruites

L’interface des boîtes détruites, présentée par la figure 5.15, permet à l’archiviste de visualiser les boîtes détruites avec leurs dates de destruction.

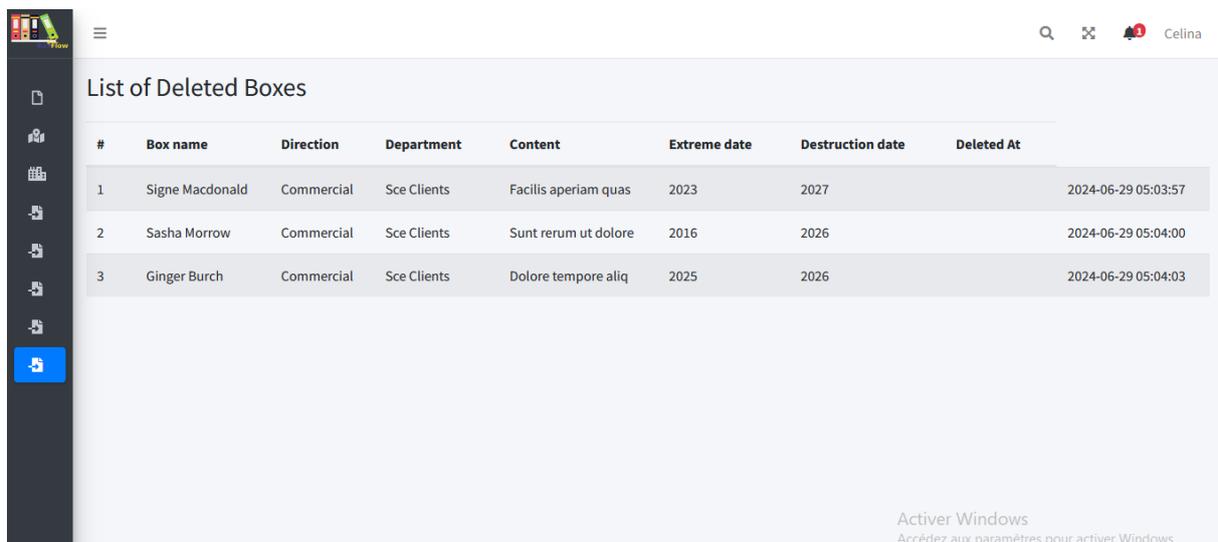


FIGURE 5.15 – Interface des boîtes détruites

5.5 Conclusion

Dans ce chapitre, nous avons présenté en détail les deux sprints qui ont constitué une release soumise aux parties prenantes pour revue et validation. Après avoir réalisé les sprints et développé les fonctionnalités correspondantes, nous avons réussi à produire un incrément répondant aux besoins du client.

Le chapitre suivant sera consacré à la conception complète de la base de données.

Chapitre 6

Conception de la Base de données

6.1 Introduction

Dans ce chapitre, nous allons présenter le diagramme de classe. À partir de ce dernier, nous allons élaborer le modèle relationnel en appliquant les règles de gestion correspondantes. Le diagramme de classe constitue une représentation visuelle des différentes classes du système ainsi que des relations qui existent entre elles. Il permet de structurer et d'organiser les données de manière cohérente avant de les transposer en un modèle relationnel. En appliquant rigoureusement les règles de gestion, nous assurerons que le modèle relationnel résultant est à la fois précis et fonctionnel, facilitant ainsi la conception et la mise en œuvre de la base de données.

6.2 Règles de gestion

Ci-dessous les règles de gestion que nous avons suivies :

- Un utilisateur peut être un demandeur, un archiviste ou un administrateur.
- On a deux types de demande : Une demande d'archivage et une demande d'emprunt.
- Une demande d'archivage est composée d'une ou plusieurs boites.
- Une boite est placée dans un et un seul site.
- Un site est composé de plusieurs rayons.
- Un rayon est composé de plusieurs colonnes.
- Une colonne est composée de plusieurs étagères.
- Un demandeur envoie une ou plusieurs demandes.
- Un archiviste traite une ou plusieurs demandes.
- Une demande provient d'une et une seule direction.
- Une direction peut faire plusieurs demandes.
- Une direction est composée de plusieurs départements.

de la base de données.

- **Classe vers Table** : Toute classe du diagramme de classe est généralement représentée par une table dans le modèle relationnel. Le nom de la table porte généralement le même que celui de la classe.
- **Attributs vers Colonnes** : Tout attribut d'une classe est généralement représenté par une colonne dans la table correspondante. Le nom de la colonne porte généralement le même que celui de l'attribut.
- **Clé primaire** : L'attribution d'une clé primaire est indispensable pour chaque table. La clé primaire d'une classe dans le diagramme de classe correspond à la clé primaire de la table correspondante dans le modèle relationnel.
- **Relations un-à-un** : Si une relation un-à-un existe entre deux classes dans le diagramme de classe, cela peut être représenté en incluant la clé primaire d'une classe comme clé étrangère dans l'autre classe. Cela établit une relation entre les deux tables correspondantes.
- **Relations un-à-plusieurs** : Si une relation un-à-plusieurs existe entre deux classes, cela peut être représenté en incluant la clé primaire de la classe "un" comme clé étrangère dans la classe "plusieurs". Cela permet à la classe "plusieurs" de faire référence à la classe "un".
- **Relations plusieurs-à-plusieurs** : Si une relation plusieurs-à-plusieurs existe entre deux classes, cela nécessite généralement une table de jointure supplémentaire. Cette table de jointure contient les clés primaires des deux classes en tant que clés étrangères, ce qui permet d'établir une relation entre les enregistrements des deux tables[20].

- **Transformation de l'héritage**

Trois décompositions sont possibles dont :

- **Décomposition par distinction** : Les classes mères et filles sont représentées par des relations. La clé primaire de la classe mère est utilisée pour identifier chacune de ses classes filles : cette clé étant pour chaque classe fille à la fois la clé primaire et une clé étrangère vers la classe mère.
- **Décomposition descendante (push-down)** : On ne représente que les classes fille par des relations. Tous les attributs de la classe mère sont répétés au niveau de chaque classe fille. La clé primaire de la classe mère est utilisée pour identifier chacune de ses classes filles.
- **Décomposition ascendante (push-up)** : On ne représente que la classe mère en intégrant les attributs de chaque classe fille au niveau de la classe mère [20].
- **Transformation de la composition** : La classe composite et la classe partie sont toutes deux représentées par des relations dans le modèle relationnel. La clé

primaire de la classe composite est à la fois une clé primaire et une clé étrangère dans la classe partie [20].

6.5 Modèle relationnel

Dans le contexte de notre conception, la décomposition choisie pour l'héritage est ascendante pour la classe mère Utilisateur. Il est à noter que l'attribut 'role' est de type énumération avec les valeurs possibles suivantes : administrateur, archiviste ou demandeur. Quant à la classe mère Demande, nous avons plutôt opté pour une décomposition descendante. Nous avons obtenu le schéma relationnel suivant :

Utilisateur (idUser, nom, email, motDePasse, role, statut)

DemandeArchive (idDemande, nomDemande, detailsDemande, dateDemande, statutDemande, reponse, nbrBoites, #idUser, #idDirection)

DemandeEmprunt (idDemande, nomDemande, detailsDemande, dateDemande, statusDemande, reponse, type, #idUser, #idDirection)

Boite (idBoite, #idDemande, contenu, dateExtreme, emplacement, statusBoite, #idSite)

Site (idSite, nomSite, #idUser)

Rayon (idRayon, #idSite, nomRayon)

Colonne (idColonne, #idRayon, nomColonne)

Etagere (idEtagere, #idColonne, nomEtagere)

Direction (idDirection, nomDirection, description, #idUser)

Département (idDepartement, #idDirection, nomDepartement)

6.6 Conclusion

En résumé, les règles de gestion, le diagramme de classes et le modèle relationnel ont joué un rôle crucial en facilitant la création de notre application. Ces outils assurent la cohérence et la conformité aux exigences spécifiées.

Conclusion générale

Notre projet de fin de cycle a abouti à la création d'une application web appelée "Box Flow" pour la gestion des boîtes d'archives au sein du complexe agroalimentaire Cevital de Bejaia. Pour mener à bien notre projet, nous avons adopté une approche méthodique et structurée. Nous avons choisi d'utiliser UML et le pattern de conception MVC. En termes de méthodologie de développement, SCRUM a été adopté, nous permettant d'organiser notre travail en sprints successifs et de nous adapter aux besoins changements du projet. L'application a été développée en utilisant le framework Laravel (PHP), le template AdminLTE et le package Spatie. Pour la gestion des données, SQLite a été choisi comme système de gestion de bases de données..

La solution réalisée permet aux demandeurs de différentes directions de Cevital d'emprunter un document ou une boîte d'archive, ainsi que d'archiver les boîtes de leurs directions via notre application en éliminant le besoin de déplacement physique lors de l'emprunt de copies des documents. Elle offre également aux archivistes la possibilité de répondre aux besoins des demandeurs en archivant leurs boîtes par l'attribution d'emplacements ou en prêtant des boîtes et des documents, permettant un accès rapide à ces informations en cas de besoin. Enfin, "BoxFlow" facilite le processus d'interaction complet entre demandeur et archiviste grâce à la double notification par email et sur "BoxFlow".

Bien que nous ayons atteint les objectifs initiaux, il existe des perspectives pour enrichir davantage notre application. Parmi celles-ci, on peut mentionner :

- Possibilité de prendre en charge les fichiers volumineux.
- Ajouter un système de traitement par lot des données.
- Notifier le demandeur en cas de retard de restitution d'une boîte empruntée.
- Implémentation d'un système de sécurité des fichiers pour garantir la confidentialité des données.

Bibliographie

- [1] Picard, F. (2018). Gestion de projet Agile : Avec Scrum, Lean, Extreme Programming. Dunod.
- [2] Schwaber, K., & Sutherland, J. (2017). Scrum Guide : The Definitive Guide to Scrum : The Rules of the Game. Scrum.org.
- [3] Atlassian. (2024). Scrum Artifacts. Consulté le 20 mai 2024, à l'adresse <https://www.atlassian.com/agile/scrum/artifacts/>.
- [4] Ionos. (2024). UML : Un langage de modélisation pour la programmation orientée objet. Consulté le 20 mai 2024, à l'adresse <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/uml-un-langage-de-modelisation-pour-la-programmation-orientee-objet/>.
- [5] Bruel, J.-M. (2016). Mobile Modeling 2016. Consulté le 20 mai 2024, à l'adresse <https://jmbruel.github.io/MobileModeling2016-1/>.
- [6] Hernandez, R. (2021). The Model View Controller Pattern : MVC Architecture and Frameworks Explained. Consulté le 30 mai 2024, à l'adresse <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>.
- [7] Codecademy. (2024). HTML & CSS. Consulté le 30 mai 2024, à l'adresse <https://www.codecademy.com/catalog/language/html-css/>.
- [8] MDN Web Docs. (2024). JavaScript. Consulté le 30 mai 2024, à l'adresse <https://developer.mozilla.org/fr/docs/Web/JavaScript/>.
- [9] Laravel. (2024). Documentation Laravel 11.x. Consulté le 15 juin 2024, à l'adresse <https://laravel.com/docs/11.x/>.
- [10] StackShare. (2024). HeidiSQL. Consulté le 15 juin 2024, à l'adresse <https://stackshare.io/heidisql/>.
- [11] Visual Studio Code. (2024). Documentation. Consulté le 15 juin 2024, à l'adresse <https://code.visualstudio.com/docs/>.
- [12] GitHub. (2024). About GitHub and Git. Consulté le 26 juin 2024, à l'adresse <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git/>.

-
- [13] Visual Paradigm. (2024). User Guide. Consulté le 26 juin 2024, à l'adresse https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html/.
- [14] Canva. (2024). Canva Home Page. Consulté le 16 juin 2024, à l'adresse https://www.canva.com/fr_fr/.
- [15] Spatie. (2024). Laravel Permission. Consulté le 16 juin 2024, à l'adresse <https://spatie.be/docs/laravel-permission/v6/introduction/>.
- [16] Spatie. (2024). Laravel Media Library. Consulté le 16 juin 2024, à l'adresse <https://spatie.be/docs/laravel-medialibrary/v11/introduction/>.
- [17] Noten, J. (2024). Laravel AdminLTE Installation. Consulté le 16 juin 2024, à l'adresse <https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Installation/>.
- [18] Packagist. (2024). Laravel UI. Consulté le 16 juin 2024, à l'adresse <https://packagist.org/packages/laravel/ui/>.
- [19] Roques, P. (2006). UML 2 par la pratique (N11770, 236 pages). Collections Cahier du programmeur. Consulté le 17 juin 2024.
- [20] Crozat, S. (2024). Transformation de l'héritage en relationnel [Cours universitaire]. Université de Technologie de Compiègne. Consulté le 17 juin 2024, à l'adresse <https://stph.scenari-community.org/bdd/rel3-prs/co/rel3.html>.
- [21] SQLite. (2024). SQLite Home Page. Consulté le 30 juin 2024, à l'adresse <https://www.sqlite.org/>.
- [22] Tiilt. (2024). Gestion des tâches : méthodes et outils. Consulté le 2 juillet 2024, à l'adresse <https://www.tiilt.io/gestion-des-taches-methodes-et-outils/>.
- [23] WeLoveDevs. (2024). MVC : Définition et exemples. Consulté le 2 juillet 2024, à l'adresse <https://welovedevs.com/fr/articles/mvc/>.
- [24] Cevital. (Consulté le 2 juillet 2024). Cevital Agro-Industrie. Disponible à l'adresse <https://www.cevital.com/cevital-agro-industrie/>.
- [25] Roques, P. (2010). *UML 2 par la pratique : Etudes de cas et exercices corrigés*. Eyrolles.

Résumé

Ce mémoire a été rédigé en vue de l'obtention du diplôme de fin de cycle master en informatique, parcours génie logiciel. Il traite les problèmes posés par l'ancien système de gestion d'archivage utilisé au sein de l'entreprise Cevital à Bejaia. La solution proposée consiste à développer une application web qui numérise l'interaction entre le demandeur et l'archiviste pour les demandes d'emprunt et d'archivage, ainsi qu'à assurer une gestion efficace des emplacements des boîtes d'archives. Pour mettre en œuvre notre solution, nous avons choisi d'utiliser UML et le pattern de conception MVC. En termes de méthodologie de développement, SCRUM a été adopté, nous permettant d'organiser notre travail en sprints successifs et de nous adapter aux besoins changeants du projet. L'application a été développée en utilisant le framework Laravel (PHP), le template AdminLTE et le package Spatie. Pour la gestion des données, SQLite a été choisi comme système de gestion de bases de données.

Mots-clés : Archivage et emprunt de boîtes d'archive, Cevital, UML, MVC, SCRUM, Laravel, AdminLTE, Spatie, SQLite.

Abstract

This thesis was written with the aim of obtaining a Master's degree in Software Engineering. It addresses the issues posed by the former archival management system used within the company Cevital in Bejaia. The proposed solution involves developing a web application that digitizes the interaction between the applicant and the archivist for borrowing and archiving requests, as well as ensuring efficient management of archive box locations. To implement our solution, we chose to use UML and the MVC design pattern. In terms of development methodology, we adopted SCRUM, allowing us to organize our work into successive sprints and adapt to changing project needs. The application was developed using the Laravel framework (PHP), AdminLTE template, and Spatie package. For data management, SQLite was selected as the database management system.

Keywords : Archiving and borrowing archive boxes, Cevital, UML, MVC, SCRUM, Laravel, AdminLTE, Spatie, SQLite.