

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 Master
professionnel en informatique

Option : Génie Logiciel

Conception et réalisation d'une application web de gestion d'une crèche

Cas d'étude : Kenza Crèche

Réalisé par :

Mlle. LOULOU Malika
Mlle. MAIOU Fadila

Encadré par :

M. MIR Foudil

Soutenu le **01 Juillet 2025**

Devant le jury composé de :

M.	OUZEGANE Redouane	- Président	(Univ. A. Mira Béjaïa)
M.	BENAI Yani	- Examineur	(Univ. A. Mira Béjaïa)
Mme.	TAHAKOURT Zineb	- Examineur	(Univ. A. Mira Béjaïa)
Mme.	KHALED Hayette	- Examineur	(Univ. A. Mira Béjaïa)

Année universitaire : 2024/2025

Remerciements

Nous remercions Dieu de nous avoir donné la force, la patience et la persévérance pour mener à bien ce travail.

Nous adressons nos sincères remerciements à notre encadrant, M. **Mir Foudil** pour son soutien, ses conseils précieux et sa disponibilité tout au long de ce projet.

Nous remercions également les membres du jury pour le temps qu'ils ont consacré à lire et à évaluer notre mémoire.

Un grand merci à notre famille, en particulier à nos parents, pour leur amour, leur soutien et leur présence à chaque étape de notre parcours.

Enfin, à toutes les personnes qui nous ont aidés, de près ou de loin, nous exprimons toute notre gratitude..

Malika & Fadila

Dédicace

Je tiens à dédier ce travail à mes parents, pour leur soutien constant, leur patience et leur amour inestimable tout au long de mes études.

À ma famille, qui m'a toujours encouragée, motivée et portée dans les moments de doute.

À mes amis, pour leur présence réconfortante, leurs mots bienveillants et leur précieuse aide au fil du temps.

À mes professeurs et encadrants, pour leur disponibilité, leurs conseils avisés et leur accompagnement tout au long de cette aventure.

Et enfin, à toutes les personnes qui, de près ou de loin, ont contribué à la réussite de ce projet.

Malika

Dédicace

Je dédie ce mémoire à mes chers parents, Rahmani Baya et Salah Maiou, pour leur amour inconditionnel, leurs sacrifices constants, leur sagesse et leur soutien sans faille tout au long de mon parcours.

À mes frères et sœurs : Abdelghani, Lilia, Djawida et Sihem, pour leur présence bienveillante, leurs encouragements et leur affection.

À mes petits anges, mes nièces chéries : Lamis, Meriem et Létissia, qui illuminent ma vie par leur innocence et leur joie de vivre.

À ma précieuse binôme, Malika Loulou, pour sa complicité, son soutien indéfectible, et tous les moments de partage, de travail et de rires.

À mes amies fidèles : Youva Aiche, Sihem Belaid, Tassadit Baderdine, Dihya, et Chaïma Kadri, pour leur amitié sincère, leur écoute et leur présence tout au long de ce chemin.

À mon cher cousin Rahmani Massinisa, en hommage affectueux. Parti trop tôt, mais toujours présent dans nos pensées. Ce travail est aussi pour toi.

À Madame Java, une femme exceptionnelle, qui m'a tendu la main, fait confiance, encouragée, et surtout appris à aimer l'informatique. Grâce à son regard bienveillant, ses conseils, et l'opportunité qu'elle m'a offerte, je suis devenue ce que je suis aujourd'hui. Après Allah, c'est à elle que je dois une grande partie de mon cheminement.

Et enfin, à moi-même, pour ma persévérance, ma résilience et ma capacité à croire en mes rêves, malgré les obstacles.

Fadila

Table des matières

Table des matières	i
Table des figures	iv
Liste des tableaux	vi
Liste des sigles et acronymes	vii
Introduction générale	1
1 Applications informatiques et Ergonomie web	3
1.1 Introduction	3
1.2 Application informatique	3
1.2.1 Application mobile	3
1.2.2 Application desktop	4
1.2.3 Application web	4
1.3 Ergonomie web	5
1.3.1 Principes fondamentaux de l’ergonomie web	5
1.3.2 Objectifs de l’ergonomie web	6
1.3.3 Règles de l’ergonomie web	6
1.4 Interface utilisateur et Expérience utilisateur	7
1.4.1 Interface utilisateur	7
1.4.2 Expérience utilisateur	7
1.5 Conclusion	8
2 Présentation de l’organisme d’accueil et analyse des solutions existantes	9
2.1 Introduction	9
2.2 Présentation de l’organisme d’accueil	9
2.2.1 Organisation structurelle de la crèche	10
2.2.2 Équipe de Kenza Crèche	10
2.2.3 Activités de Kenza Crèche	11
2.2.4 Organigramme de Kenza Crèche	11
2.3 Description du métier	12
2.3.1 Principales responsabilités	12
2.3.2 Compétences et qualités requises	12
2.3.3 Formation et qualifications nécessaires	13
2.3.4 Type de crèche existante	13
2.4 Étude des solutions existantes	14

2.4.1	Clever Nursery (Algérie)	14
2.4.2	Créacreche (Algérie)	14
2.4.3	LutinRouge (International)	15
2.4.4	Procure (International)	16
2.4.5	Tableau comparatif des solutions existantes	17
2.5	Problématique	17
2.6	Solution proposée	18
2.7	Conclusion	19
3	Langages de modélisation et méthodologies de développement	20
3.1	Introduction	20
3.2	Méthodologie de développement	20
3.2.1	Méthodes classiques	20
3.2.2	Méthodes orientées objet	21
3.2.3	Méthodologie Agile	22
3.3	Méthode de développement choisie	23
3.3.1	Scrum	23
3.4	Langage UML	25
3.4.1	Diagrammes de structure (ou statiques)	25
3.4.2	Diagrammes de comportement	26
3.4.3	Choix des diagrammes à utiliser	28
3.5	Architecture MVC	28
3.6	Conclusion	29
4	Conception et développement	30
4.1	Introduction	30
4.2	Sprint Zéro	30
4.2.1	Spécification des besoins	30
4.2.2	Capture des besoins fonctionnels	32
4.2.3	Capture des besoins non fonctionnels	33
4.2.4	Backlog produit du sprint zéro	34
4.2.5	Planification des sprints	35
4.2.6	Présentation du logo de la plate-forme	35
4.3	Sprint Un	36
4.3.1	Backlog produit du sprint Un	36
4.3.2	Spécifications des besoins	37
4.3.3	Diagramme de cas d'utilisation du sprint un	37
4.3.4	Description textuelle	39
4.3.5	Diagrammes de séquences détaillés	48
4.4	Sprint deux	58
4.4.1	Backlog produit du sprint deux	58
4.4.2	Spécification des besoins	60
4.4.3	Diagramme de cas d'utilisation du sprint deux	60
4.4.4	Description textuelle	64
4.4.5	Diagrammes de séquences détaillés	71
4.5	Sprint trois	77
4.5.1	Backlog produit du sprint trois	77
4.5.2	Spécifications des besoins	78
4.5.3	Diagramme de cas d'utilisation du sprint trois	78

4.5.4	Description textuelle	79
4.5.5	Diagrammes de séquences détaillés	83
4.5.6	Implémentation	86
4.6	Conclusion	90
5	Réalisation	91
5.1	Introduction	91
5.2	Outils et environnement de développement	91
5.2.1	Environnement de développement	91
5.2.2	Technologies web utilisées	93
5.2.3	Autres Outils	94
5.3	Principales interfaces	94
5.3.1	Accueil	94
5.3.2	Authentification	95
5.3.3	S'inscrire	96
5.3.4	Demandes d'inscription reçues	96
5.3.5	Liste des inscriptions effectuées par le parent	98
5.3.6	Soumission d'une demande d'inscription par le parent	98
5.3.7	Compléter le dossier d'inscription	100
5.3.8	Prestation	102
5.3.9	Facture	102
5.3.10	Facture générer	103
5.3.11	Engagement	104
5.3.12	Fiche d'inscription	105
5.3.13	Autorisation parentale	107
5.3.14	Messagerie	109
5.4	Conclusion	109
	Conclusion générale et perspectives	110
	Bibliographie	110
	Annexes	115
	Annexe A : Dictionnaire de Données	115

Table des figures

1.1	UI et UX [14].	7
2.1	Organigramme de Kenza Crèche.	12
2.2	Clever Nursery[19].	14
2.3	Créacreche[20].	15
2.4	LutinRouge[21].	16
2.5	Procare[22].	16
3.1	Méthodologie Scrum [29].	23
3.2	L'architecture MVC [38].	28
4.1	Diagramme de Contexte".	31
4.2	Logo de la plate-forme.	35
4.3	Diagramme du cas d'utilisation du sprint un	38
4.4	Diagramme de séquence détaillé "S'authentifier".	49
4.5	Rechercher un objet.	50
4.6	Diagramme de séquence détaillé "Rechercher une saison".	51
4.7	Diagramme de séquence détaillé "Ajouter un employé".	52
4.8	Diagramme de séquence détaillé "Modifier une catégorie d'abonnement".	53
4.9	Diagramme de séquence détaillé "Supprimer une section".	54
4.10	Diagramme de séquence détaillé "Activer/Désactiver un supplément".	55
4.11	Diagramme de séquence détaillé "Exporter la liste d'un groupe".	56
4.12	Diagramme d'activité "S'authentifier".	57
4.13	Diagramme de cas d'utilisation "directeur".	61
4.14	Diagramme de cas d'utilisation "Éducateur".	62
4.15	Diagramme de cas d'utilisation "Parent".	63
4.16	Diagramme de séquence détaillé "Demander une inscription ou une réinscription".	72
4.17	Diagramme de séquence détaillé "Ajouter un planning pédagogique".	73
4.18	Diagramme de séquence détaillé "Modifier une séance".	74
4.19	Diagramme de séquence détaillé "Modifier l'état d'une inscription".	75
4.20	Diagramme de séquence détaillé "Accepter définitivement l'inscription".	76
4.21	Diagramme de cas d'utilisation du sprint trois.	79
4.22	Diagramme de séquence détaillé "Ajouter une actualité".	84
4.23	Diagramme de séquence détaillé "Gérer la messagerie".	85
4.24	Diagramme de séquence détaillé "Modifier un évènement".	86
4.25	Diagramme de classe	87
5.1	Logo de Laravel [42].	91

5.2	Logo de Bootstrap [44].	92
5.3	Logo de VS Code [46].	92
5.4	Logo de WAMP Server [48].	92
5.5	Logo de GitHub [50]	92
5.6	Logo de HTML [52]	93
5.7	Logo de CSS [54]	93
5.8	Logo de JavaScript [56]	93
5.9	Logo de PHP [58].	93
5.10	Logo de SQL [60].	93
5.11	Logo de Visual Paradigm [62].	94
5.12	Logo de MySQL [64].	94
5.13	Logo de Git [66].	94
5.14	Logo de Composer [68].	94
5.15	Interface d'accueil	95
5.16	Interface d'authentification	95
5.17	Interface d'inscription	96
5.18	Interface complète des demandes d'inscription envoyées au directeur .	97
5.19	Interface des inscriptions effectuées par le parent	98
5.20	Interface de soumission d'une demande d'inscription par le parent . .	99
5.21	Interface de saisie des responsables légaux	100
5.22	Interface de complétion du dossier d'inscription	101
5.23	Espace de signature électronique du représentant légal	102
5.24	Interface de prestation	102
5.25	Interface de facture	103
5.26	Interface de facture générer	103
5.27	Interface d'engagement	104
5.28	Première partie du document de fiche d'inscription générer par la plateforme	105
5.29	Deuxième partie du document de fiche d'inscription générer par la plateforme	106
5.30	Troisième partie du document de fiche d'inscription générer par la plateforme	107
5.31	Document de l'autorisation parentale générer par la plateforme	108
5.32	Interface de la messagerie	109

Liste des tableaux

2.1	Tableau comparatif des solutions existantes de gestion de crèche . . .	17
4.1	Messages échangés entre le système et les utilisateurs	32
4.2	Cas d'utilisation par acteur.	33
4.3	Le Backlog produit du sprint zéro avec ordre de priorité.	35
4.4	Règles de gestion : S'authentifier.	35
4.5	Backlog produit du sprint un	37
4.6	Règles de gestion : S'authentifier.	39
4.7	Description textuelle du CU : S'authentifier.	39
4.8	Description textuelle du CU : Recherche côté client.	40
4.9	Description textuelle du CU : Gérer les catégories d'abonnement. . .	40
4.10	Description textuelle du CU : Gérer les suppléments.	41
4.11	Description textuelle du CU : Gérer les sections.	43
4.12	Description textuelle du CU : Gérer les groupes.	44
4.13	Description textuelle du CU : Gérer les employés.	46
4.14	Description textuelle du CU : Exporter en PDF ou Excel.	47
4.15	Description textuelle du CU : Activer et désactiver une saison.	48
4.16	Backlog du produit du sprint deux	60
4.17	Description textuelle du CU : Gérer les matières.	64
4.18	Description textuelle du CU : Gérer les séances.	65
4.19	Description textuelle du CU : Gérer les plannings de cours.	66
4.20	Description textuelle du CU : Gérer les évaluations.	67
4.21	Description textuelle du CU : Gestion des inscriptions.	68
4.22	Description textuelle du CU : Compléter le dossier d'inscription. . . .	69
4.23	Description textuelle du CU : Traitement des inscriptions.	70
4.24	Description textuelle du CU : Valider définitivement l'inscription. . .	70
4.24	Description textuelle du CU : Valider définitivement l'inscription. . .	71
4.25	Le Backlog du produit du sprint trois	77
4.26	Description textuelle du CU : Gérer les actualités.	80
4.27	Description textuelle du CU : Gérer la galerie.	81
4.28	Description textuelle du CU : Gérer la messagerie.	82
4.29	Description textuelle du CU : Gérer les évènements.	82
1	Dictionnaire de données des entités du système	115

Liste des sigles et acronymes

2TUP	Two Track Unified Process
CAP	Certificat d'Aptitude Professionnelle
CD	Continuous Delivery
CI	Continuous Integration
CMS	Content Management System
CSS	Cascading Style Sheets
CU	Cas d'Utilisation
DOM	Document Object Model
GPS	Global Positioning System
HTML	HyperText Markup Language
IBM	International Business Machines
iOS	iPhone Operating System
MVC	Modèle-Vue-Contrôleur
OMT	Object Modeling Technique
OOSE	Object-Oriented Software Engineering
PHP	Hypertext Preprocessor
PWA	Progressive Web App
RG	Règle de Gestion
RUP	Rational Unified Process
UP	Unified Process
US	User Story
UX/UI	User eXperience / User Interface (Expérience utilisateur / Interface utilisateur)
UML	Unified Modeling Language
WAMP	Windows, Apache, MySQL, PHP
XP	Extreme Programming

Introduction générale

La petite enfance est une période essentielle dans le développement d'un enfant. C'est une étape durant laquelle il a besoin d'un environnement sûr, stimulant et bienveillant pour grandir dans de bonnes conditions. Les crèches jouent un rôle central dans cet accompagnement en accueillant les enfants au quotidien, tout en apportant un soutien précieux aux familles. Elles assurent non seulement la garde, mais aussi une partie de l'éveil et de l'apprentissage des tout-petits.

En Algérie, les crèches connaissent un développement croissant, porté par une demande de plus en plus forte de la part des familles, notamment dans les zones urbaines. Ces structures d'accueil sont majoritairement privées, bien que certaines soient gérées par des institutions publiques ou associatives. Leur mission principale est d'offrir un cadre sécurisé et pédagogique aux enfants âgés de 3 mois à 6 ans. Toutefois, la gestion quotidienne de ces établissements reste souvent artisanale, avec peu de solutions numériques adaptées aux besoins spécifiques du secteur. Dans la majorité des cas, l'administration repose encore sur des outils bureautiques classiques (Word, Excel, dossiers papier), ce qui complique le suivi des enfants, la gestion des inscriptions ou encore la communication avec les parents. Il en résulte une charge de travail importante pour les responsables, ainsi que des risques d'erreurs ou d'informations perdues.

Ces difficultés touchent également les parents, qui rencontrent souvent des obstacles lors de l'inscription ou de la réinscription de leurs enfants. Les démarches sont généralement longues, nécessitant des déplacements répétés à la crèche pour déposer des documents ou obtenir des informations. Il est également difficile pour eux de suivre l'état d'avancement du dossier de leur enfant, de vérifier si une pièce est manquante ou si l'inscription est validée. De plus, il n'existe que très rarement un espace dédié permettant aux parents de consulter les activités quotidiennes, les absences, ou encore de suivre les progrès de leur enfant. L'absence de plateforme centralisée limite la transparence et la fluidité de la communication entre les familles et la crèche.

La gestion d'une crèche implique en effet un grand nombre de tâches administratives et organisationnelles : suivi des enfants, planification des activités, gestion des inscriptions, organisation des repas, gestion des plannings, et communication avec les parents. À cela s'ajoute la coordination du personnel, composé notamment du directeur, du gérant, des éducateurs, du chauffeur et du cuisinier. Dans beaucoup de crèches, ces tâches sont encore gérées manuellement à l'aide de documents papier ou de fichiers informatiques simples. Cela peut entraîner des erreurs, des pertes d'informations ou un manque d'efficacité dans le fonctionnement quotidien.

Face à ces difficultés, nous avons choisi de concevoir une application web dédiée à la gestion d'une crèche. L'objectif de ce projet est d'apporter une solution simple et pratique qui permet de centraliser les données importantes : les informations sur les enfants, la répartition par groupe d'âge, les emplois du temps des cours, les abonnements, ainsi que les échanges entre la crèche et les parents. Accessible à partir d'un ordinateur ou d'un téléphone connecté à Internet, cette application vise à alléger la charge de travail des responsables, à faciliter les démarches des parents, et à améliorer l'organisation globale de la crèche. Elle est pensée pour être facile à utiliser, même pour des utilisateurs peu expérimentés en informatique, et adaptée aux besoins concrets d'une seule crèche.

Ce mémoire retrace toutes les étapes de notre projet. Le premier chapitre présente les principes de l'ergonomie web et les types d'applications informatiques, afin de situer notre solution dans son contexte. Le deuxième chapitre est dédié à la présentation de la crèche partenaire et à l'analyse de son fonctionnement actuel. Le troisième aborde les outils de modélisation utilisés ainsi que la méthode agile Scrum adoptée pour le développement. Le quatrième détaille la conception de l'application et les différents sprints réalisés. Enfin, le cinquième chapitre présente la réalisation finale avec les principales fonctionnalités mises en place.

Ce projet de fin d'études a pour objectif de répondre de manière concrète aux besoins réels d'une crèche, tout en mettant en pratique les compétences acquises durant notre formation.

Chapitre 1

Applications informatiques et Ergonomie web

1.1 Introduction

Pour concevoir une application web efficace et adaptée aux attentes des utilisateurs, il est important de bien comprendre les éléments qui permettent de la rendre utile, simple à utiliser, et agréable pour les utilisateurs.

Dans ce chapitre, nous allons voir pourquoi ces aspects sont essentiels dans la création d'une solution numérique. Nous parlerons d'abord des applications informatiques, pour mieux comprendre leur rôle dans notre vie quotidienne et dans différents domaines professionnels. Ensuite, nous aborderons l'ergonomie web, qui permet de rendre une interface claire et facile à prendre en main. Enfin, nous verrons comment le design UX/UI (User eXperience/User Interface) aide à penser l'application du point de vue de l'utilisateur.

Tous ces éléments sont importants pour concevoir une application efficace et agréable à utiliser.

1.2 Application informatique

Une application informatique, souvent appelée application ou logiciel applicatif, est un programme conçu pour aider l'utilisateur à accomplir une ou plusieurs tâches spécifiques sur un ordinateur, un appareil mobile ou d'autres appareils numériques.

Elle peut être utilisée pour des besoins personnels, professionnels ou industriels. Par exemple : applications bureautiques, jeux vidéo, applications mobiles, systèmes de gestion.

1.2.1 Application mobile

Les applications mobiles sont des programmes créés pour fonctionner sur des téléphones ou des tablettes. On peut les télécharger depuis des boutiques comme l'App Store ou Google Play. Elles peuvent être faites pour un seul type d'appareil (comme Android ou iOS – iPhone Operating System) ou pour plusieurs à la fois.

Elles utilisent les éléments du téléphone (comme le GPS – Global Positioning System, ou la caméra) pour offrir des fonctions utiles et personnalisées [1].

1.2.1.1 Différents types d'application mobiles

- **Application native :** Une application native est une application créée spécialement pour un seul type d'appareil, soit Android ou iOS. On utilise des langages propres à chaque système : Swift pour iOS, Java ou Kotlin pour Android.

Ces applications sont rapides, agréables à utiliser, et peuvent accéder facilement à la caméra, au GPS, ou envoyer des notifications. Elles peuvent aussi fonctionner sans Internet. Mais leur principal inconvénient, c'est qu'il faut créer une version différente pour chaque système, ce qui prend plus de temps et d'argent [2].

- **Application hybride :** Une application hybride est une application qu'on peut utiliser à la fois sur Android et iOS. Elle est faite avec un seul code, ce qui permet de gagner du temps et de réduire les coûts. On utilise pour cela des outils comme React Native, Ionic ou Cordova.

Ces applications peuvent aussi utiliser des fonctions du téléphone comme le GPS, l'appareil photo ou le micro. Mais elles sont parfois moins rapides que les applications natives, et l'expérience utilisateur peut être moins fluide. Il peut aussi être plus compliqué de bien l'adapter à chaque système [3].

- **Progressive Web app :** PWA (une Progressive Web App) est une application web qui fonctionne comme une application mobile, mais qu'on installe depuis un navigateur, pas depuis un store (comme Google Play). Elle est créée avec des langages web comme HTML (HyperText Markup Language), CSS (Cascading Style Sheets) et React.

Les PWA sont rapides, légères, fonctionnent même sans connexion, et se mettent à jour toutes seules. Elles prennent peu de place sur le téléphone et sont faciles à trouver sur le web.

De grandes entreprises comme Spotify, Twitter ou Uber utilisent ce type d'applications. Par contre, elles ont moins de fonctions sur iOS (comme les notifications push), et consomment plus de batterie quand on les utilise en déplacement[4].

1.2.2 Application desktop

Une application desktop, c'est un logiciel qu'on installe sur un ordinateur (portable ou fixe). Elle permet de faire des tâches précises comme écrire un document ou retoucher une photo.

Elle utilise les ressources de l'ordinateur (processeur, mémoire, stockage) et peut souvent fonctionner sans Internet, ce qui est un avantage par rapport aux applications web. Certaines peuvent aussi être partagées entre plusieurs utilisateurs dans un réseau [5].

1.2.3 Application web

Une application web est un logiciel qu'on utilise directement en ligne, à travers un navigateur (comme Chrome ou Firefox), sans rien installer. Elle fonctionne sur tous les appareils connectés à Internet : téléphone, tablette, ou ordinateur.

Son interface (frontend) est faite avec HTML, CSS et JavaScript. Son fonctionnement interne (backend) avec des langages comme PHP (Hypertext Preprocessor),

Python ou Ruby. Elle ressemble à une application mobile quand on l'utilise sur un smartphone, mais elle reste accessible depuis n'importe quel appareil.

1.2.3.1 Différents types d'application web

- **Application web statique** : Ce sont des pages web simples avec peu de contenu et sans interactions. Elles sont conçues en HTML et CSS. Modifier leur contenu est plus difficile, car il faut changer le code manuellement sur le serveur [6].
- **Application web dynamique** : Contrairement aux statiques, ces applications utilisent une base de données. Elles réagissent aux actions de l'utilisateur et affichent des informations à jour. Elles sont souvent faites avec un CMS (Content Management System) [7].
- **Application e-commerce** : Ce sont des boutiques en ligne permettant la vente de produits ou de services sur Internet comme Amazon. Elles doivent permettre des paiements sécurisés (carte, PayPal, etc.), gérer les commandes et le stock, etc. Leur création est plus délicate et demande souvent un développeur professionnel [8].
- **Application portail** : Il s'agit d'une application centralisant plusieurs services accessibles depuis une page d'accueil unique. Elle peut intégrer des forums, une messagerie électronique, un système de discussion instantanée, des actualités, ou encore des espaces de travail collaboratif. Ce type d'application vise à offrir une interface unifiée permettant aux utilisateurs d'accéder facilement à l'ensemble des fonctionnalités proposées [9].
- **Application web animée** : Ce sont des applications avec des effets visuels et du mouvement pour rendre l'utilisation plus agréable. Elles sont faites avec HTML5, CSS3 et JavaScript. Elles sont modernes, fonctionnent sans flash, mais demandent plus de travail pour bien marcher sur tous les appareils.

1.3 Ergonomie web

Alain Wisner, médecin et fondateur de l'ergonomie française, la définissait ainsi : « L'ensemble des connaissances scientifiques relatives à l'Homme nécessaires pour concevoir des outils, des machines et des dispositifs qui puissent être utilisés avec le maximum de confort, de sécurité et d'efficacité » [10].

L'ergonomie web fait référence à l'ensemble des méthodes et techniques utilisées pour faire un site bien construit, plaisant à utiliser, compréhensible, où l'on trouve facilement ce qu'on cherche, et qui charge rapidement.

En intégrant ces principes dès la conception, les développeurs et les designers peuvent améliorer l'expérience utilisateur d'une manière significative.

1.3.1 Principes fondamentaux de l'ergonomie web

- **Cohérence** : La cohérence de la structure et du design (notamment les couleurs, polices et styles) sur l'ensemble des pages du site permet d'offrir une expérience fluide et intuitive aux utilisateurs. Par exemple, un menu de navigation clair permet aux utilisateurs de trouver facilement ce qu'ils cherchent [11].
- **Simplicité** : La simplicité du design est également cruciale pour éviter toute confusion [11].

- **Feedback immédiat** : Une bonne ergonomie web peut considérablement améliorer la satisfaction des utilisateurs. Lorsqu'ils ont une expérience positive, ils sont plus susceptibles de revenir sur le site. En revanche, un site mal conçu peut entraîner des abandons rapides et nuire à la réputation de l'entreprise [11].
- **Performance** : Les pages doivent se charger rapidement, même sur des appareils avec des ressources limitées [11].
- **Responsive design** : Le responsive design est une approche de conception web qui adapte automatiquement l'affichage des pages en fonction de la taille et de la résolution de l'écran utilisé, assurant ainsi une expérience utilisateur optimale sur divers appareils tels que les smartphones, les tablettes et ordinateurs[11].

1.3.2 Objectifs de l'ergonomie web

- **Utilisabilité** : Le visiteur doit être en mesure de trouver facilement l'information qu'il est venu chercher, sans se perdre au cours de la navigation [12].
- **Lisibilité** : Le contenu des pages doit être hiérarchisé avec les informations importantes en priorité, les couleurs et polices doivent rester simples et claires pour une visualisation fluide [12].
- **Rapidité** : Un bon développeur veille à réduire les temps de chargement des pages. Une interface rapide permet de capter l'attention de l'utilisateur dès les premières secondes, évite la frustration et augmente les chances qu'il reste sur le site [12].
- **Accessibilité** : Un site web ergonomique est inclusif, accessible sur tous les terminaux et navigateurs, et conçu pour répondre aux besoins des personnes en situation de handicap visuel [12].
- **Interactivité** : Des liens clairs et bien placés permettent aux utilisateurs de naviguer facilement, avec la possibilité de commenter ou de partager le contenu en quelques clics [12].

1.3.3 Règles de l'ergonomie web

- **Une organisation visuelle esthétique et cohérente de vos pages web** : Il faut éviter le trop-plein d'informations sur vos pages web car cela peut gêner l'utilisateur pendant sa navigation sur votre site Internet [13].
- **Site web doit informer l'utilisateur et lui répondre** : Votre site web doit réagir aux actions de l'utilisateur et l'informer correctement au bon moment[13].
- **Site web doit prévoir les erreurs de vos utilisateurs** : Le site web doit pouvoir anticiper les erreurs commises par l'utilisateur lors de sa navigation web [13].
- **Votre site web facilite l'accessibilité pour tous** : Tout le monde doit avoir accès à votre site web assez facilement [13].
- **Un affichage rapide** : Réduire la taille des pages permet de les charger plus vite et d'offrir une meilleure expérience aux utilisateurs [13].
- **Des textes lisibles** : Pour faciliter l'expérience de votre audience, il est très important d'avoir un texte bien lisible [13].

1.4 Interface utilisateur et Expérience utilisateur

L'interface utilisateur et l'expérience utilisateur sont deux concepts essentiels en ergonomie web qui se complètent pour garantir une interface utilisateur performante et une expérience utilisateur optimale.

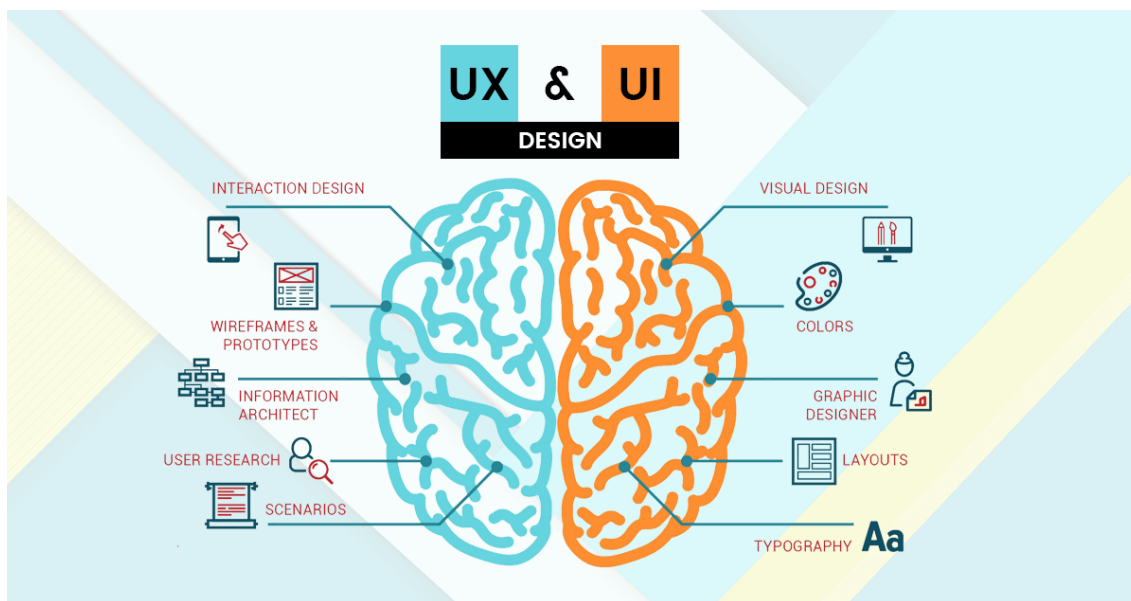


FIGURE 1.1 – UI et UX [14].

1.4.1 Interface utilisateur

L'interface utilisateur design sert quant à lui à améliorer l'interaction d'un utilisateur avec un produit. L'UI intervient dans toutes les étapes du développement de votre site et/ou application mobile, sauf celle de la recherche [15].

Responsabilités importantes :

- **Esthétique et cohérence visuelle :** Utilisation harmonieuse des couleurs, polices et éléments graphiques.
- **Disposition des éléments :** Organisation des boutons, menus et champs pour un accès rapide et intuitif.
- **Interaction fluide :** Assurer que les clics, transitions et animations rapides et faciles à comprendre.
- **Adaptabilité :** Conception responsive pour une utilisation sur tous les appareils (ordinateurs, tablettes, smartphones).

1.4.2 Expérience utilisateur

L'expérience utilisateur se réfère à l'étude des attentes et besoins de l'utilisateur pour la création d'un site web et/ou d'une application mobile. Il s'agit donc de prendre en compte le ressenti d'un utilisateur quand il navigue sur votre sup-

port digital. L'UX design vise à rendre le parcours de l'utilisateur plus simple et agréable[15].

Responsabilités importantes :

- **Facilité d'utilisation** : Le site doit permettre d'atteindre les objectifs rapidement (exemple : inscription, recherche d'information).
- **Navigation intuitive** : Les utilisateurs doivent pouvoir naviguer facilement sur un site ou une application, avec des parcours clairs, bien organisés et adaptés à leurs attentes.
- **Feedback utilisateur** : Fournir des confirmations, alertes ou corrections immédiates pour guider l'utilisateur.
- **Émotion et engagement** : Créer une expérience qui suscite la confiance, le plaisir et la satisfaction.

1.5 Conclusion

Les applications informatiques, l'ergonomie web et le design UX/UI jouent un rôle important dans la création d'une application efficace et agréable à utiliser. Ces éléments permettent de mieux répondre aux attentes des utilisateurs et d'assurer une bonne expérience.

Dans le chapitre suivant, nous allons présenter l'organisme d'accueil Kenza Crèche, et faire une analyse des solutions existante afin de bien comprendre les besoins réels du projet.

Chapitre 2

Présentation de l'organisme d'accueil et analyse des solutions existantes

2.1 Introduction

Avant de commencer le développement de notre application, il est essentiel de bien comprendre l'environnement dans lequel elle sera utilisée. Ce chapitre a pour objectif de présenter le fonctionnement réel d'une crèche, à travers l'exemple de **Kenza Crèche**, l'organisme d'accueil de notre projet. Cela nous permettra d'identifier clairement les besoins à prendre en compte.

Nous présenterons d'abord l'organisation de la crèche et le rôle de son personnel, puis nous analyserons les plateformes existantes pour en dégager les limites. Cette réflexion mènera à la formulation de la problématique et à la proposition d'une application web adaptée aux besoins de Kenza Crèche.

2.2 Présentation de l'organisme d'accueil

Kenza Crèche, c'est un lieu d'accueil privé destiné aux jeunes enfants, situé à Akkal Aberkan dans la commune d'El Kseur, qui se consacre à offrir un environnement chaleureux et sécurisé. Elle propose une variété d'activités éducatives et ludiques afin de combiner l'apprentissage et le plaisir, offrant ainsi aux enfants la possibilité de s'épanouir tout en s'amusant. Elle s'est dotée d'un personnel compétent, composé de professionnels formés à l'éducation de la petite enfance, garantissant un suivi en fonction des besoins des enfants et un encadrement de haute qualité.

Kenza Crèche a acquis une solide réputation dans ce domaine, ceci grâce à son travail, son attitude positive envers les enfants, et son engagement envers les familles.

Son objectif principal consiste à soutenir les parents dans l'équilibre entre leur vie professionnelle et familiale en leur offrant une solution de garde fiable pour leurs enfants. De plus, elle vise à développer leurs capacités cognitives, sociales et émotionnelles, dans le but de les préparer au mieux aux défis de l'avenir.

2.2.1 Organisation structurelle de la crèche

Les enfants à Kenza crèche sont divisés en trois sections selon leur âge afin d'assurer une assistance optimale et une prise en charge adaptée à leurs besoins et leurs rythmes d'apprentissage :

- **Petite section** : Spécialisée dans les enfants âgés de 3 à 4 ans, où l'accent est mis sur l'acquisition des mouvements fondamentaux, ainsi que l'interaction avec les autres enfants pour favoriser le partage et la coopération.
- **Moyenne section** : Pour les enfants de 4 à 5 ans, avec des activités stimulant l'indépendance, la créativité et le développement social.
- **Grande section (préscolaire)** : Dans cette section, les enfants âgés de 5 à 6 ans apprennent la lecture et les mathématiques afin de se préparer à l'école, ainsi que des compétences sociales avancées.

Afin de garantir un suivi pédagogique efficace, chaque section est partagée en deux groupes au maximum comptant chacun de 12 à 15 enfants, en fonction de leur mois de naissance, tout en proposant une prise en charge individuelle, en petits groupes facilitant l'apprentissage et le développement social. Les groupes sont structurés comme suit :

- **Groupe A** : Les enfants nés entre janvier et juin sont accueillis dans le groupe A de janvier jusqu'à juin.
- **Groupe B** : Les enfants nés entre juillet et décembre sont accueillis dans le groupe B de juillet jusqu'à décembre.

2.2.2 Équipe de Kenza Crèche

L'équipe de Kenza Crèche forme une véritable famille, où chaque membre travaille ensemble pour créer un cadre d'apprentissage stimulant et parfait pour les enfants. Cela inclut :

- **Directrice** : C'est le propriétaire. Assure la gestion quotidienne de la crèche, la supervision de l'équipe et la gestion budgétaire. Elle crée les plannings des cours et des repas, prend les décisions finales et maintient le lien avec les parents. En cas d'urgence, elle gère l'hospitalisation des enfants et du personnel; en son absence, la gérante prend le relais.
- **Gérante** : La gérante, en collaboration avec la directrice, assure le bon déroulement des journées à la crèche. Elle est un véritable pilier pour les éducatrices, veille au respect des protocoles et au bien-être des enfants. Toujours à l'écoute des parents. Elle leur donne les nouvelles des progrès de leurs enfants. Elle s'occupe aussi de la gestion des ressources matérielles pour que chaque activité se déroule dans les meilleures conditions.
- **Éducatrices de jeunes enfants** : Elles assurent l'encadrement quotidien des enfants, la planification des activités pédagogiques et l'accompagnement personnalisé.

À Kenza Crèche, le niveau du diplôme ne suffit pas pour recruter des éducatrices. C'est pourquoi une formation interne est assurée par la directrice et la gérante pour compléter leurs compétences afin de garantir une prise en charge de qualité. Elles veillent aussi à l'hygiène de leur salle, essentielle pour la santé et le bien-être des enfants.

- **Cuisinière** : Chaque jour s'engage à préparer le repas annoncé par la directrice, et de conserver le plat témoin pendant 72 heures. Cela signifie qu'une petite quantité du repas préparé pendant la journée est conservée pour être analysée dans un laboratoire en cas de réclamation d'un parent concernant la nourriture de son enfant à la crèche. En l'absence de celle-ci, la gérante ou la directrice est chargée de la cuisine.
- **Chauffeur** : Son rôle est d'assurer le transport des enfants en toute sécurité. Il effectue généralement deux trajets : un premier pour les enfants dont les parents commencent le travail très tôt, et un second pour le reste du groupe. Grâce à lui, les enfants arrivent et repartent à la crèche dans de bonnes conditions.

2.2.3 Activités de Kenza Crèche

La crèche Kenza propose diverses activités pour favoriser l'éveil et le développement des enfants de manière ludique et adaptée à leur âge :

- **Exploration sensorielle** : Les enfants découvrent le monde avec leurs cinq sens. En touchant, regardant et écoutant, ils s'amuse avec des objets aux textures, couleurs et sons variés, ce qui stimule leur curiosité.
- **Mouvement et motricité** : Des parcours d'obstacles et de petits jeux comme enfiler des perles aident les enfants à mieux bouger et à développer leur coordination.
- **Découverte des matières** : En manipulant des éléments comme de l'eau, de la terre ou de la pâte, les enfants apprennent comment les choses changent et réagissent, en expérimentant les volumes et textures.
- **Jeux d'imitation** : En jouant à faire comme les grands (nourrir une poupée, conduire une voiture), les enfants comprennent mieux leur environnement et s'expriment naturellement.
- **Construction et tri** : Avec des jeux de construction, ils apprennent à empiler, assembler et trier des objets, renforçant ainsi leur logique de façon concrète.
- **Expression et créativité** : Les enfants participent à des activités comme les chansons, les marionnettes et la peinture qui leur permettent de s'exprimer et d'explorer leur créativité.

2.2.4 Organigramme de Kenza Crèche

L'organigramme de Kenza Crèche, présenté dans la figure suivante 2.1, montre comment l'équipe est organisée. La directrice est à la tête de la crèche. Elle est aidée par une gérante qui s'occupe de la gestion quotidienne. Trois autres rôles sont sous la responsabilité de la gérante : les éducateurs, qui s'occupent des enfants, le chauffeur, responsable du transport, et la cuisinière, chargée de préparer les repas. Cette organisation permet à chacun de savoir ce qu'il a à faire pour que tout fonctionne correctement.

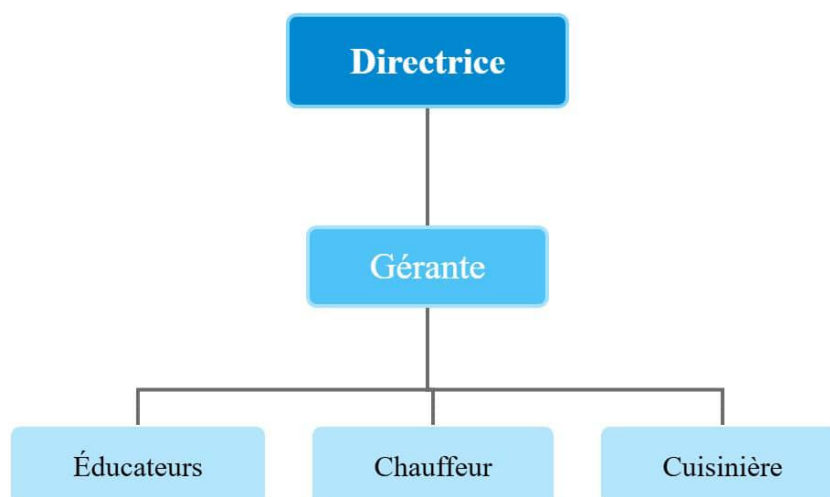


FIGURE 2.1 – Organigramme de KENZA Crèche.

2.3 Description du métier

Une crèche est un lieu d'accueil collectif pour les enfants âgés de quelques mois à six ans. Elle est généralement ouverte 5 jours sur 7, en semaine, et correspond aux horaires de travail des parents, tout en assurant une éducation et un environnement favorable à leurs enfants.

Les crèches sont gérées par des professionnels de la petite enfance qualifiés, afin de garantir la sécurité et le bien-être des enfants [16].

2.3.1 Principales responsabilités

La crèche joue un rôle primordial dans la société. Sa mission est d'accueillir les enfants, de veiller à leur bien-être et à leur confort, de les accompagner dans leurs découvertes et dans leurs apprentissages au quotidien, tout en respectant le rythme de chaque enfant. Elle offre des opportunités d'apprentissage, d'interaction sociale et de développement cognitif aux jeunes enfants.

La crèche assure également la communication avec les parents pour les tenir informés des progrès de leurs enfants, veille sur l'hygiène et la propreté, et encourage la collaboration entre l'équipe éducative.

Elle reste attentive à la santé des enfants, en surveillant les signes de fatigue, d'inconfort ou de problèmes de santé, et en réagissant rapidement si un enfant montre des signes de malaise, garantissant ainsi un environnement sécurisant et adapté à chaque enfant.

2.3.2 Compétences et qualités requises

Accéder à un emploi dans le secteur de la petite enfance nécessite des compétences techniques ainsi que des qualités personnelles. Elles sont indispensables pour créer un environnement d'apprentissage positif, sécurisé et motivant pour les enfants :

- **Connaissance du développement de l'enfant** : Une compréhension solide

des différentes étapes de croissance de l'enfant est nécessaire pour lui fournir un soutien éducatif et émotionnel approprié [17].

- **Patience et bienveillance** : Importants dans ces métiers. Être patient permet de créer un espace sécurisant où les enfants peuvent explorer et exprimer leurs émotions librement sans peur [17].
- **Capacité d'écoute et de communication** : Indispensables pour comprendre les besoins des enfants et interagir avec eux de manière constructive [17].
- **Créativité et flexibilité** : Travailler avec des enfants nécessite d'imaginer des activités captivantes pour stimuler leur curiosité. La flexibilité est également essentielle pour s'adapter aux changements imprévus et répondre de manière adéquate aux besoins individuels [17].
- **Sens des responsabilités et éthique professionnelle** : Les professionnels doivent faire preuve d'un haut degré de responsabilité et d'éthique, car ils jouent un rôle crucial dans le bien-être et la sécurité des enfants qui leur sont confiés [17].
- **Avoir un esprit d'équipe** : Travailler en collaboration avec l'équipe éducative pour garantir une prise en charge cohérente et de qualité [17].

2.3.3 Formation et qualifications nécessaires

Accéder aux métiers de la petite enfance exige une certaine passion, mais aussi une formation. Parmi les parcours de formation les plus reconnus et valorisés dans ce domaine, le CAP (Certificat d'Aptitude Professionnelle) de la petite enfance (Diplôme d'État d'auxiliaire de puériculture) est souvent exigé en Algérie et même à l'étranger pour accéder à ce poste.

De plus, d'autres formations peuvent être ajoutées à son expérience professionnelle, telles que les premiers secours ou l'accompagnement des enfants en situation de handicap.

2.3.4 Type de crèche existante

Il existe diverses structures destinées à l'accueil de la petite enfance :

- **Crèche collective** : Les crèches collectives, généralement gérées par une collectivité ou un organisme privé, accueillent un groupe d'enfants à partir de l'âge de 2 mois et demi, sous la supervision de plusieurs professionnels de la petite enfance [18].
- **Crèche parentale** : Les crèches parentales ou familiales fonctionnent sur le modèle participatif où un groupe de parents collabore pour s'occuper des tout-petits [18].
- **Crèche d'entreprise** : Certaines entreprises proposent des crèches sur leur lieu de travail ou à proximité. Cette démarche permet de faciliter la vie professionnelle et familiale de leurs collaborateurs [18].
- **Micro-crèche** : Les micro-crèches sont des structures de garde de petite taille pouvant accueillir jusqu'à 12 enfants. Les tout-petits, âgés de 10 semaines à 3 ans, sont regroupés dans le même groupe quel que soit leur âge, ce qui multiplie les occasions d'apprentissage [18].
- **Crèche multi-accueil** : Elle propose aux parents plusieurs types de garde (occasionnel, régulier, d'urgence) pour s'adapter à leurs besoins [18].

2.4 Étude des solutions existantes

L'analyse de l'existant est régulièrement utilisée lors de la conception d'une nouvelle solution informatique, elle vise à analyser la solution existante pour extraire les aspects positifs et négatifs. Afin de réaliser une solution meilleure.

Selon notre étude, nous avons trouvé plusieurs sites de gestion de crèche en Algérie et à l'étranger :

2.4.1 Clever Nursery (Algérie)

Clever Nursery est une solution conçue pour faciliter la gestion des crèches et des garderies. Elle permet aux responsables de crèches, aux éducateurs et aux administrateurs d'automatiser et de centraliser divers aspects de la gestion quotidienne.



FIGURE 2.2 – Clever Nursery[19].

2.4.1.1 Avantages

- Gestion administrative simplifiée.
- Communication améliorée.
- Suivi du développement.
- Sécurité des données.

2.4.1.2 Limites

- Manque d'informations publiques : Peu d'informations sont disponibles concernant les fonctionnalités détaillées, l'expérience utilisateur et les tarifs.
- Parfois surchargée d'informations sur une même page, ce qui peut nuire à la lisibilité.

2.4.2 Créacreche (Algérie)

Créacreche est une solution numérique dédiée à la gestion des crèches en Algérie. Elle a été spécialement conçue pour simplifier les tâches administratives, pédagogiques et organisationnelles des établissements de petite enfance. Elle permet une gestion plus fluide et une meilleure qualité de service.



FIGURE 2.3 – Créacreche[20].

2.4.2.1 Avantages

- Simplification des tâches administratives.
- Communication améliorée.
- Accessibilité et centralisation des données.
- Adaptation aux besoins locaux.

2.4.2.2 Limites

- Données limitées : des informations détaillées sur les fonctionnalités spécifiques, l'interface utilisateur et les tarifs sont peu accessibles publiquement.
- Graphisme peu moderne, ce qui donne une impression vieillotte à certains utilisateurs.

2.4.3 LutinRouge (International)

LutinRouge est une solution de gestion pour les crèches et micro-crèches, visant à simplifier les tâches administratives, pédagogiques et organisationnelles. Elle permet aux gestionnaires de centraliser les informations des enfants, de suivre les présences, de gérer le personnel, de communiquer avec les parents et d'automatiser la facturation. Cela assure une gestion plus fluide et une meilleure qualité de service pour les enfants et leurs familles.



FIGURE 2.4 – LutinRouge[21].

2.4.3.1 Avantages

- Simplicité d'utilisation.
- Communication améliorée avec les parents.
- Conformité avec les réglementations.
- Adaptabilité.

2.4.3.2 Limites

- Dépendance technologique : nécessite une connexion internet stable et une infrastructure informatique adéquate.

2.4.4 Procare (International)

Procare est un logiciel utilisé dans plusieurs pays pour gérer les crèches. Il permet de suivre les inscriptions, les présences, la facturation et de communiquer facilement avec les parents. Il aide le personnel à organiser son travail au quotidien.

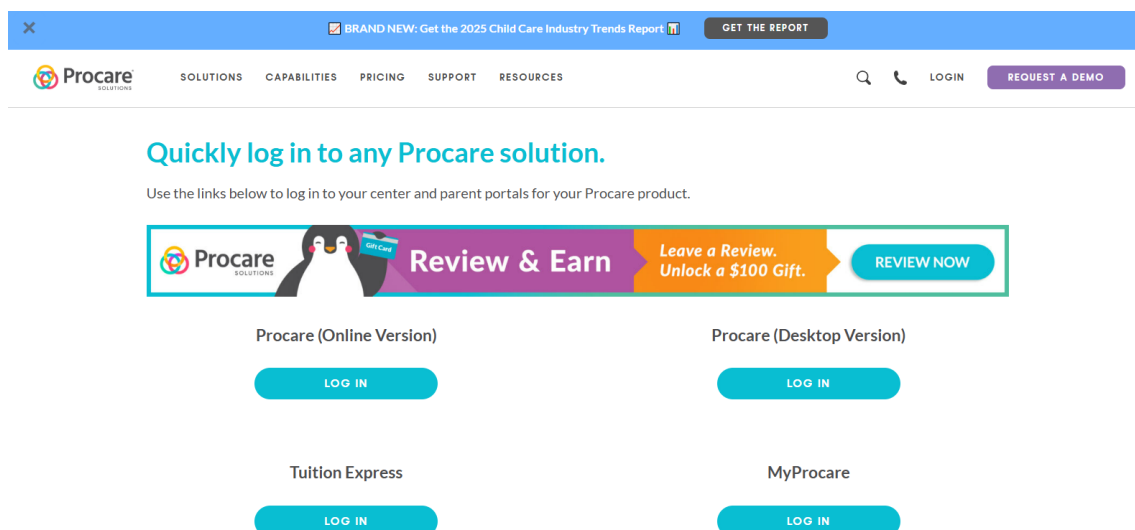


FIGURE 2.5 – Procare[22].

2.4.4.1 Avantages

- Communication avec les parents.
- Suivi des progrès et des activités de l'enfant.
- Sécurité et conformité.

2.4.4.2 Limites

- Complexité : nécessite un temps d'adaptation important au début.
- Dépendance à Internet : une connexion stable est requise pour un fonctionnement optimal.

2.4.5 Tableau comparatif des solutions existantes

Le tableau 2.1 présente une synthèse des différentes solutions de gestion de crèche étudiées. Il met en évidence leurs principaux avantages et inconvénients, ce qui nous aide à identifier les fonctionnalités pertinentes à intégrer dans notre propre solution.

Solution	Avantages	Inconvénients
Clever Nursery	- Gestion simplifiée - Bonne communication	- Peu d'infos publiques - Interface parfois chargée
Créacreche	- Adaptée au contexte local - Données centralisées	- Design peu moderne - Infos limitées
LutinRouge	- Facile à utiliser - Communication efficace	- Dépend d'une bonne connexion Internet
Procare	- Suivi complet - Bonne sécurité	- Complexe au début - Nécessite Internet stable

TABLE 2.1 – Tableau comparatif des solutions existantes de gestion de crèche

Après avoir étudié les différentes solutions existantes, nous avons relevé plusieurs constats. Tout d'abord, aucune ne propose un ensemble de fonctionnalités réellement complet pour la gestion d'une crèche. L'utilisateur qu'il s'agisse du directeur, des éducateurs ou des parents cherche aujourd'hui une solution simple qui lui permette de gérer les inscriptions, le suivi des enfants, la communication et l'organisation quotidienne à partir d'un seul outil. Par ailleurs, la plupart des solutions analysées restent assez complexes à utiliser et nécessitent un temps d'adaptation, ce qui peut rendre leur prise en main difficile pour certains utilisateurs.

2.5 Problématique

Dans un contexte où de nombreux parents jonglent entre la vie professionnelle et personnelle, trouver une crèche adaptée à leurs besoins devient un véritable défi. Ils recherchent un établissement sûr et flexible, garantissant un encadrement de qualité pour leurs enfants.

D'un autre côté, les crèches font face à des problématiques administratives et organisationnelles, cherchant à répondre efficacement aux besoins des familles tout en gérant les contraintes liées aux ressources, à la sécurité et à la qualité des services.

Après une observation continue de la crèche, nous avons pu observer les insuffisances suivantes :

- **Recherche difficile sur les registres qui engendre une perte de temps :** Cette difficulté de recherche peut ralentir les processus de gestion, accroître le risque d'erreurs et impacter la réactivité dans la prise de décision.
- **Possibilité d'erreur dans le remplissage des différents documents et registres :** La saisie manuelle des documents et registres augmente le risque d'erreurs, pouvant affecter la fiabilité des données et compliquer leur gestion.
- **Insécurité des informations :** Le manque de protection des données peut entraîner des fuites, des pertes ou des accès non autorisés, compromettant ainsi la confidentialité et l'intégrité des informations.
- **Un grand nombre de documents dans la crèche rend leur stockage et leur gestion plus complexes :** L'accumulation des documents en crèche complique leur gestion. Une solution numérique permet de les stocker et d'y accéder en ligne, évitant les déplacements et réduisant les risques de perte.
- **Manque de transparence sur les prestations offertes :** Il y a fréquemment peu d'informations disponibles sur les horaires et les types d'activités.
- **Défis de communication :** Maintenir une communication fluide avec la crèche, notamment pour les informations sur le suivi quotidien de l'enfant qui n'est pas toujours facile.
- **Gestion administrative et organisationnelle complexe :** La crèche doit gérer une grande quantité de données (informations sur les enfants, autorisations parentales, etc.), souvent via des processus manuels ou des logiciels non spécialisés, ce qui peut ralentir et complexifier les opérations quotidiennes.
- **Difficulté à gérer les inscriptions et les listes d'attente :** Pour la crèche, la gestion des demandes d'inscription devient complexe, car elle doit gérer des listes d'attente tout en répondant aux questions des parents. Cela nécessite une organisation rigoureuse et peut engendrer des erreurs ou des retards dans les réponses aux parents.

Pour remédier à ces limitations, nous proposons une solution optimisée pour la gestion de la crèche, répondant à l'ensemble des besoins et intégrant diverses améliorations.

2.6 Solution proposée

Nous visons à développer et mettre en place une plate-forme numérique, conçue pour la gestion d'une crèche. Cette solution permettra aux parents d'inscrire leurs enfants, de suivre leur évolution au quotidien et d'accéder aux informations proposées par l'établissement (planning, repas, activités, etc.).

Côté personnel de la crèche, la plateforme facilitera la gestion pédagogique et organisationnelle à travers diverses fonctionnalités :

- Gestion des enfants.
- Gestion des employés.
- Gestion des actualités.
- Gestion des factures.
- Gestion des inscriptions et réinscriptions.

- Gestion des dossiers administratifs des enfants.
- Communication directe avec les familles (messagerie).

L'objectif est de digitaliser l'environnement de la crèche pour améliorer la qualité des services, fluidifier la communication avec les parents et alléger les charges de travail du personnel.

La plateforme sera accessible aussi aux éducateurs, aux parents, selon des droits d'accès définis.

2.7 Conclusion

Dans ce chapitre, nous avons découvert le fonctionnement de la crèche Kenza ainsi que le contexte général du projet. Cette étape nous a permis de mieux comprendre les besoins réels du terrain et de réfléchir à une solution adaptée. Le chapitre suivant présentera la méthode choisie pour mener à bien le développement de cette application.

Chapitre 3

Langages de modélisation et méthodologies de développement

3.1 Introduction

Pour réussir un projet web, il ne suffit pas de savoir coder. Il faut aussi suivre une méthode claire et bien organisée. Ce chapitre explique les étapes importantes à suivre pour planifier et structurer un projet informatique. On y découvre différentes méthodes de développement, dont Scrum choisie pour notre projet. Il présente aussi des outils comme UML (Unified Modeling Language) pour la modélisation, ainsi que l'architecture MVC (Modèle-Vue-Contrôleur), très utile pour construire des applications bien structurées.

3.2 Méthodologie de développement

Le développement d'un système suit généralement une méthodologie bien définie pour garantir sa qualité, sa cohérence et sa maintenabilité. Il existe plusieurs types de méthodologies, chacune avec ses propres principes et domaines d'application. Les méthodes classiques, comme la méthode systémique, proposent une approche structurée et séquentielle, adaptée aux projets bien définis.

À côté de cela, les méthodes orientées objet ont introduit une nouvelle manière d'aborder le développement logiciel.

Enfin, les méthodologies agiles ont émergé pour répondre aux besoins de flexibilité et de collaboration dans les projets, avec des approches itératives, adaptatives et centrées sur l'utilisateur. Chaque famille de méthodes présente des avantages selon le contexte du projet, la taille de l'équipe, les délais et les attentes des utilisateurs.

3.2.1 Méthodes classiques

Les méthodes classiques de développement logiciel suivent une approche linéaire et structurée. Chaque étape, comme l'analyse, la conception, le codage, puis les tests, est réalisée l'une après l'autre, sans retour en arrière. La méthode systémique, par exemple, prend en compte l'ensemble du système dans son environnement global pour bien comprendre tous les éléments avant de commencer le développement. Ces

méthodes sont souvent utilisées pour des projets stables et bien définis dès le départ.

3.2.1.1 Méthode systémique

Les méthodes systémiques sont un ensemble d'approches utilisées pour analyser, concevoir et résoudre des problèmes complexes, en tenant compte des interactions et des relations entre les différentes parties d'un système [23].

3.2.1.2 Merise

C'est une méthode de gestion de projet informatique qui permet de décomposer un système en sous-systèmes [24].

Niveaux d'abstraction de Merise

- **Niveau conceptuel** : Il représente ce que fait le système, indépendamment de toute technologie, répond aux questions : que fait l'organisme ?
- **Niveau organisationnel (ou logique)** : Adapter les modèles conceptuels aux contraintes et règles de l'organisation. Répond à qui fait quoi, quand et comment ?
- **Niveau physique** : Traduire les modèles logiques en une mise en œuvre technique concrète.

3.2.2 Méthodes orientées objet

Les méthodes orientées objet se basent sur le concept d'objets, qui regroupent des données et des fonctions. Elles permettent de modéliser un système de façon plus naturelle, en représentant les éléments du monde réel comme des objets (ex. : un utilisateur, une voiture).

Ces méthodes facilitent la réutilisation du code, la maintenance et l'évolution du logiciel. Il existe plusieurs approches comme RUP (Rational Unified Process), OMT (Object Modeling Technique), OOSE (Object-Oriented Software Engineering) ou la méthode de Booch, qui aident à bien organiser le travail autour des objets dès la phase d'analyse.

3.2.2.1 OMT

La méthode OMT est une approche orientée objet utilisée pour modéliser et concevoir un système. Elle repose sur trois types de modèles :

- **Modèle d'objet (structure statique → classes, relations)** : Représente les différents objets, leurs relations et la structure statique.
- **Modèle dynamique (comportement → états, transitions)** : Décrit les comportements et interactions au fil du temps, ainsi que les changements d'état.
- **Modèle fonctionnel (traitement de l'information)** : Représente les différentes étapes ou processus du système, ainsi que les changements de données.

3.2.2.2 OOSE

OOSE est une méthode orientée objet développée par Ivar Jacobson. Elle repose sur l'utilisation des cas d'usage pour mieux cerner les besoins des utilisateurs vis-à-vis du système. Parmi ses points clés, on retrouve [25] :

- Met l'accent sur le besoin de l'utilisateur dès le début.
- Utilise des cas d'utilisation pour décrire les scénarios fonctionnels.
- Idéal pour l'analyse et la spécification des exigences.

3.2.2.3 Booch Method

La méthode de Booch est une méthode orientée objet utilisée pour analyser et concevoir des systèmes logiciels complexes. Elle repose sur des objets, des classes, et

des interactions entre eux. Parmi ses points clés, on retrouve :

- Utilise plusieurs types de diagrammes pour modéliser un système.
- Met l'accent sur la conception du système (architecture).
- Très visuelle, elle propose des symboles graphiques pour représenter les objets et leurs relations.

3.2.2.4 RUP

RUP est une méthode de développement orientée objet, proposée par IBM (International Business Machines). Elle organise le projet en phases (inception (la première phase du cycle de développement), élaboration, construction, transition) et se base sur des itérations [26]. Ses points clés sont :

- Repose sur 3 piliers : modélisation visuelle (UML), cas d'utilisation, et itérations.
- Fournit un cadre complet pour gérer tout le cycle de vie d'un logiciel.
- Peut être adaptée selon la taille du projet.

3.2.2.5 2TUP (Two Track Unified Process)

Variante de l'UP (Unified Process), conçue pour être plus légère, plus souple et mieux adaptée aux projets évolutifs. Elle repose sur deux pistes de travail complémentaires :

- **Branche fonctionnelle** : axée sur les besoins des utilisateurs et les fonctionnalités attendues.
- **Branche technique** : s'occupe de l'architecture, des contraintes techniques et des choix technologiques.

Cette séparation permet une meilleure répartition des responsabilités et facilite l'adaptation aux changements en cours de projet.

3.2.3 Méthodologie Agile

La méthodologie agile est une approche moderne de gestion de projet, basée sur un développement itératif et incrémental, permettant une adaptation rapide aux besoins changeants des utilisateurs. Contrairement aux méthodes classiques, elle favorise la flexibilité, la collaboration constante avec le client et des livraisons fréquentes de versions fonctionnelles du produit [27].

Cette méthodologie repose sur quatre principes fondamentaux :

- **Équipe** : Dans un environnement agile, les interactions entre les membres de l'équipe sont primordiales pour favoriser la créativité et la résolution des problèmes.
- **Application** : Elle met l'accent sur l'importance de livrer rapidement des résultats concrets afin que les équipes puissent obtenir des retours dès les premières étapes et améliorer le produit en continu.
- **Collaboration** : En collaborant étroitement avec le client, l'équipe peut mieux comprendre ses attentes, obtenir des retours réguliers et ajuster le produit en conséquence.
- **Acceptation du changement** : C'est la capacité des équipes agiles à s'ajuster rapidement aux changements de contexte. Cela permet d'accueillir les nouvelles idées et de réagir aux retours des utilisateurs, garantissant ainsi que le projet reste pertinent et efficace face à un environnement en constante évolution.

Plusieurs méthodologies ont été développées autour de ces principes, parmi lesquelles :

3.2.3.1 Scrum

- Méthode agile très populaire.
- Organisation du travail en sprints (itérations courtes).
- Rôles clés : Product Owner, Scrum Master, équipe de développement.
- Réunions régulières : daily scrum, sprint planning, sprint review, etc.

3.2.3.2 XP (Extreme Programming)

- Méthode agile orientée développement logiciel.
- Se concentre sur la qualité du code et la collaboration rapprochée avec le client.
- Rôles clés : Product Owner, Scrum Master, équipe de développement.
- Développement itératif, tests automatisés, Programmation en binôme, Restructuration du code.

3.3 Méthode de développement choisie

Pour assurer une gestion efficace du projet et répondre aux besoins des utilisateurs, il est important d'adopter une approche structurée. Parmi les différentes méthodes existantes, nous avons choisi la méthodologie agile, et plus particulièrement Scrum, pour sa flexibilité et son adaptation aux évolutions du projet. Cela nous aidera à optimiser le développement et à assurer une mise en production progressive de notre solution.

3.3.1 Scrum

Scrum est un cadre de travail agile qui aide les équipes à structurer et à gérer efficacement leur travail grâce à un processus itératif basé sur des sprints. Il est particulièrement adapté aux projets complexes aux exigences évolutives, permettant une livraison rapide et flexible. Scrum s'appuie sur trois piliers : transparence, inspection et adaptation, ainsi que sur des rôles, artefacts et événements clés pour assurer une amélioration continue et maximiser la valeur du produit final [28].

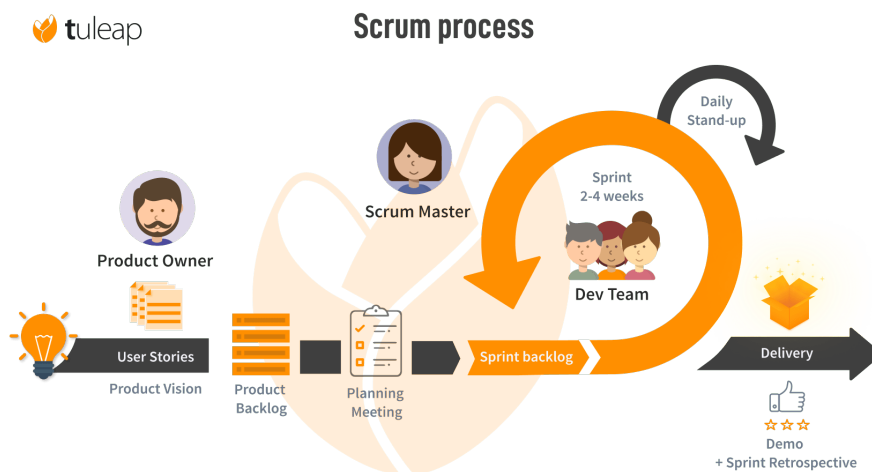


FIGURE 3.1 – Méthodologie Scrum [29].

3.3.1.1 Principes fondamentaux de Scrum

- **Transparence** : L'équipe et les parties prenantes doivent avoir accès à tous les aspects importants du processus. Cela favorise une compréhension partagée du travail et de ses objectifs [30].
- **Inspection** : L'inspection implique une évaluation régulière du travail en cours, des processus utilisés et des résultats obtenus afin de repérer d'éventuels écarts ou problèmes par rapport aux objectifs. Grâce à cette pratique, l'équipe peut suivre l'avancement et la qualité du travail [30].
- **Adaptation** : L'adaptation consiste à modifier les actions, le plan ou les processus en fonction des observations effectuées lors de l'inspection. Lorsque l'équipe identifie des écarts, des difficultés ou des opportunités d'amélioration, elle ajuste sa méthode pour répondre aux nouvelles conditions et optimiser les résultats[30].

3.3.1.2 Rôles

- **Product owner** : Le Product Owner est responsable de la gestion efficace du Product Backlog, y compris le développement et la communication explicite de l'objectif du produit, la création et la communication claire des éléments du Product Backlog, l'ordonnancement des éléments du Product Backlog et la garantie que le Product Backlog est transparent, visible et compris [31].
- **Scrum master** : Le Scrum Master est responsable de l'établissement de Scrum en aidant tout le monde à comprendre la théorie et la pratique de Scrum, à la fois au sein de l'équipe Scrum et de l'organisation, tout en servant l'équipe Scrum ainsi que l'organisation au sens large [32].
- **Scrum team** : La Scrum Team est une petite équipe (généralement de 10 personnes ou moins) auto-organisée et pluridisciplinaire qui travaille ensemble, en utilisant le cadre Scrum, pour créer quelque chose de valeur. L'équipe se compose d'un Product Owner qui maximise la valeur du produit résultant du travail de l'équipe Scrum, de développeurs qui créent le produit et d'un Scrum Master qui aide l'équipe à améliorer ses pratiques et son efficacité [33].

3.3.1.3 Événements de Scrum

Scrum s'appuie sur des événements clés pour organiser le travail et garantir la transparence :

- **Sprint** : Un sprint désigne une brève période limitée dans le temps généralement entre 1 et 4 semaines, dont une équipe Scrum a besoin pour accomplir une certaine quantité de travail. Il inclut tous les événements Scrum et se répète en continu, offrant un rythme de travail stable tout en permettant des ajustements réguliers [34].
- **Sprint planning** : Le Sprint Planning est la réunion de lancement du Sprint où l'équipe Scrum définit le travail à réaliser. Le Product Owner présente les éléments prioritaires du Product Backlog, et l'équipe répond à trois questions[35] :
 - **Pourquoi ?** (valeur via l'objectif de sprint).
 - **Quoi ?** (éléments sélectionnés du backlog).
 - **Comment ?**(plan de réalisation).Le résultat (Sprint Backlog) inclut les éléments choisis, l'objectif et le plan. Durée limitée à 8 h max pour un Sprint de 4 semaines (adaptée pour les Sprints plus courts).

- **Daily Scrum** : Réunion quotidienne de 15 minutes où les développeurs synchronisent leur travail, vérifient leur progression vers l'objectif de sprint et adaptent le Sprint Backlog. Elle favorise la communication rapide et l'auto-organisation de l'équipe [35].
- **Sprint Review** : Sprint Review est une réunion collaborative où l'équipe Scrum présente le travail accompli aux parties prenantes pour évaluer les progrès et adapter le Product Backlog. D'une durée limitée (max 4 h pour un Sprint mensuel), elle permet des décisions éclairées pour les prochains Sprints[35].
- **Sprint retrospective** : Sprint Retrospective est une réunion où l'équipe Scrum analyse le dernier Sprint pour identifier des améliorations. Elle examine ce qui a bien fonctionné, les difficultés rencontrées et définit des actions concrètes pour progresser. Limitée à 3 h maximum pour un Sprint d'un mois, elle clôture chaque cycle en favorisant l'amélioration continue [35].

3.3.1.4 Les artefacts :

Les artefacts Scrum (Product Backlog, Sprint Backlog et Incrément) fournissent une vision transparente du produit, du travail planifié et des réalisations. Ils guident l'équipe et les parties prenantes tout au long du développement [35].

- **Product backlog** : Le Product Backlog est une liste structurée, évolutive et priorisée de toutes les exigences liées au produit. Il contient l'objectif produit et évolue continuellement grâce à l'affinage. Les développeurs estiment l'effort nécessaire pour chaque élément, tandis que le Product Owner est chargé de les clarifier et de les prioriser [35].
- **Sprint backlog** : Plan de travail dynamique créé par les développeurs, comprenant l'objectif du sprint, les éléments du Product Backlog sélectionnés et la stratégie de réalisation. Il évolue durant le Sprint pour s'adapter aux nouvelles informations tout en maintenant l'objectif fixé [35].
- **Incrément** : L'incrément est le résultat concret et utilisable d'un Sprint, cumulant toutes les fonctionnalités terminées selon les critères de finition définis par l'équipe. Potentiellement livrable, il représente une avancée claire vers l'objectif du produit et doit répondre aux critères de qualité établis [35].

3.4 Langage UML

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue [36].

Il propose plusieurs types de diagrammes, répartis en deux catégories :

3.4.1 Diagrammes de structure (ou statiques)

Les diagrammes UML structurels, comme leur nom l'indique, ils décrivent la structure d'un système, notamment les classes, les objets, les paquetages, les composants, et les relations entre ces éléments. On trouve :

3.4.1.1 Diagramme de classes

Ce diagramme représente la structure statique du système en montrant les classes, leurs attributs, leurs méthodes et les relations entre elles [36].

Les éléments essentiels comprennent :

- **classes** : Description abstraite d'un ensemble d'objets qui partagent les mêmes propriétés et comportements [36].
- **Attributs** : Caractéristiques ou propriétés d'une classe. Un attribut peut posséder une multiplicité et une valeur initiale [36].
- **Opération** : Une opération représente une action définie globalement pour tous les objets d'une même classe. Elle peut prendre des paramètres et retourner une valeur d'un certain type [36].
- **Association** : Une association désigne une relation durable entre deux classes, reliant leurs instances. Elle est généralement bidirectionnelle, sauf si une flèche est ajoutée pour restreindre sa navigabilité [36].
- **Rôle** : Le nom attribué à une extrémité d'une association, par extension, il décrit la manière dont les instances d'une classe interagissent avec celles d'une autre classe à travers cette association [36].
- **Multiplicité** : Le nombre d'objets (min..max) qui peuvent participer à une relation avec un autre objet dans le cadre d'une association [36].

Multiplicités fréquentes :

- 0..1 = optionnel (mais pas multiple).
- 1 = exactement 1.
- 0..* = * = quelconque.
- 1..* = au moins 1.

3.4.1.2 Diagramme de contexte

Bien que non standardisé dans UML, ce diagramme est souvent utilisé pour montrer les interactions entre le système étudié et son environnement externe. Les éléments essentiels sont :

- **Système** : Le sujet principal du diagramme.
- **Acteurs externes** : Entités extérieures interagissant avec le système (utilisateurs, autres systèmes).
- **Interfaces** : Points de contact entre le système et les acteurs externes.

3.4.2 Diagrammes de comportement

Ce diagramme UML représente la manière dont le système se comporte et interagit avec lui-même et avec les utilisateurs, les autres systèmes et les autres entités. On trouve :

3.4.2.1 Diagramme de séquence

Les diagrammes de séquence UML montrent comment les objets interagissent dans le temps, en visualisant l'ordre des messages échangés entre les objets. Ils sont particulièrement efficaces pour illustrer les interactions dynamiques entre les différents composants d'un système.

Les diagrammes de séquence en UML s'illustrent principalement en deux types, chacun ayant un niveau d'abstraction et une focalisation distincte :

Diagramme de séquence système :

Illustrer la séquence verticale des messages passés entre objets au sein d'une interaction [36], les éléments essentiels sont :

- **Acteurs** : Les entités qui interagissent avec le système.

- **Système** : Le système est représenté par un rectangle intitulé "Système", qui encapsule l'ensemble des fonctionnalités internes. Il est traité comme une "boîte noire", ce qui signifie que ses processus internes ne sont pas détaillés dans ce diagramme.
- **Message** : Interactions ou événements échangés entre les acteurs et le système.
- **Ligne de vie** : Correspondant à une représentation dans un diagramme de séquence, montrant la durée de participation d'un élément (acteur, système ou objet) lors de la modélisation ou de la conception.

Diagramme de séquence détaillé

Fournir une vue approfondie des interactions internes du système, en mettant en évidence la collaboration entre les objets ou classes pour réaliser une fonctionnalité spécifique. Il suit le modèle MVC.

3.4.2.2 Diagramme de cas d'utilisation

Ce diagramme décrit les fonctionnalités offertes par le système du point de vue de l'utilisateur, en montrant les interactions entre les acteurs et les cas d'utilisation [36]. Les éléments essentiels comprennent :

- **Acteurs** : Un acteur représente un utilisateur ou un système externe qui interagit avec le système étudié à travers au moins un cas d'utilisation [36].
- **Cas d'utilisation** : Fonctionnalité ou service que le système fournit à un ou plusieurs acteurs pour répondre à un besoin spécifique[36].
- **Relations** : Liens entre les acteurs et les cas d'utilisation, tels que :
 - **Association** : L'association est utilisée dans ce type de diagramme pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement « participe à »[36].
 - **Inclusion** : Le cas d'utilisation principal fait obligatoirement appel à un autre cas d'utilisation, à un moment précis de son déroulement [36].
 - **Extension** : Un cas d'utilisation peut en inclure un autre de manière optionnelle et implicite à un moment non précisé, ce qui est généralement déconseillé [36].
 - **Généralisation** : Les cas d'utilisation hérités partagent les descriptions du parent, mais peuvent ajouter des relations spécifiques, ce qui est déconseillé. En revanche, généraliser les acteurs peut parfois être pertinent [36].

3.4.2.3 Diagramme d'activités

Le diagramme d'activités est une représentation graphique utilisée en UML pour modéliser les flux de travail ou les processus métier. Il illustre la séquence d'activités, les décisions associées et les interactions entre différentes actions au sein d'un système, similaire à un organigramme ou à un diagramme de flux de données.

Typiquement, un diagramme d'activité possède un point de départ, appelé « état initial » et un point de fin désigné comme « état final »[37].

- **Activités** : Tâches ou actions réalisées dans le processus.
- **Transitions** : Flux reliant les activités entre elles.
- **Décisions** : Points où des choix influencent le chemin suivi dans le processus.
- **Début/Fin** : Indicateurs des points de départ et de terminaison du processus.

Ces diagrammes offrent une vue d'ensemble complète des aspects structurels et

comportementaux d'un système, facilitant ainsi sa conception, sa documentation et sa communication entre les différentes parties prenantes.

3.4.3 Choix des diagrammes à utiliser

Comme mentionné précédemment, UML définit plusieurs types de diagrammes, chacun servant à modéliser différents aspects d'un système. Dans notre travail, nous avons utilisé :

- **Diagrammes de structure** : Diagramme de classes, diagramme de contexte.
- **Diagrammes de comportement** : Diagramme de cas d'utilisation, diagramme de séquence et diagramme d'activité.

Ces diagrammes nous ont permis d'avoir une vision claire et structurée du système, facilitant ainsi sa conception et son développement.

3.5 Architecture MVC

L'architecture MVC est un modèle de conception logicielle qui divise une application en trois composants principaux, chacun ayant des responsabilités distinctes, comme montré dans la figure suivante 3.2.

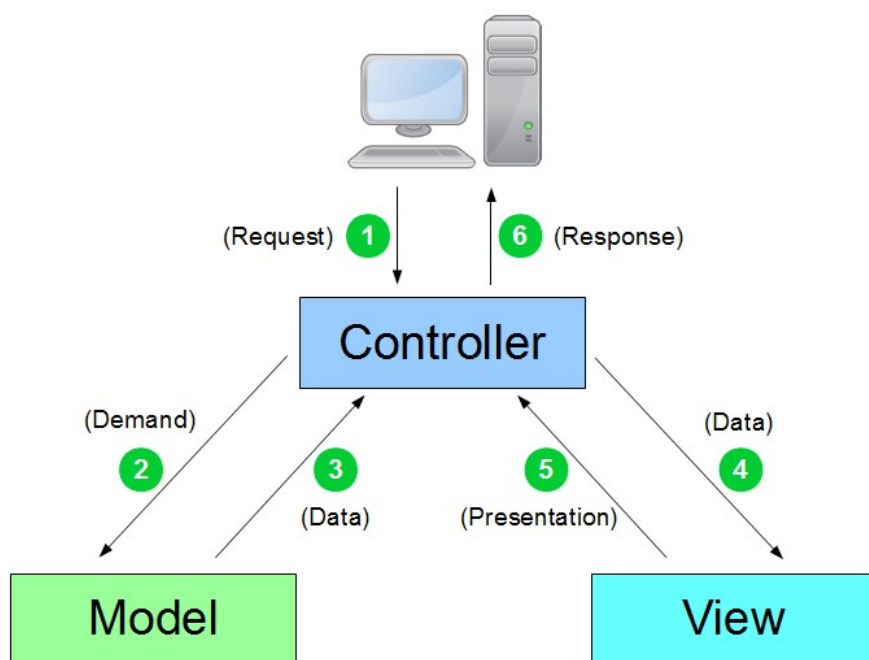


FIGURE 3.2 – L'architecture MVC [38].

- **Modèle (Model)** : Il gère les données et la logique métier de l'application. Le modèle est responsable de la récupération, de la manipulation et de la gestion des données, souvent en interaction avec une base de données ou d'autres sources de données [39].
- **Vue (View)** : Elle est chargée de la présentation de l'interface utilisateur. La vue affiche les données fournies par le modèle sous une forme adaptée à l'utilisateur, sans contenir de logique métier [39].

- **Contrôleur (Controller) :** Il traite les entrées de l'utilisateur et les traduit en actions à effectuer par le modèle ou la vue. Le contrôleur agit comme un intermédiaire, recevant les requêtes de l'utilisateur, les traitant et mettant à jour le modèle ou la vue en conséquence [39].

Grâce à ce type d'architecture, le développeur sait où intervenir et comment le faire, tout en minimisant les erreurs et les impacts sur les autres composants techniques et métiers. De plus, le modèle MVC est devenu un standard dans le développement d'applications web et mobiles en raison de sa capacité à structurer le code de manière claire et modulable.

3.6 Conclusion

A travers ce chapitre, nous avons présenté la méthodologie de développement, basée sur l'approche Agile, Scrum et UML afin de structurer et optimiser la gestion du projet. Les chapitres suivants détailleront la mise en œuvre des sprints.

Chapitre 4

Conception et développement

4.1 Introduction

Pour mener à bien le développement de notre application, nous avons choisi une méthode de travail qui nous permettait d'avancer de façon organisée et efficace. Nous avons donc opté pour Scrum, une méthode inspirée des approches agiles, basée sur des cycles courts appelés sprints. Grâce à cette organisation, nous avons pu progresser étape par étape, en nous concentrant à chaque fois sur des objectifs précis.

Ce chapitre présente l'ensemble des sprints réalisés durant le projet. À travers chacun d'eux, nous expliquerons ce que nous avons cherché à accomplir, les tâches que nous avons menées, les éventuelles difficultés rencontrées, ainsi que les résultats obtenus. Cette manière de travailler nous a permis de mieux gérer notre temps, d'identifier rapidement les problèmes et de faire évoluer le projet de façon régulière. Elle a aussi permis à l'équipe de mieux se coordonner, car chacun savait ce qu'il devait faire et pouvait suivre l'avancement du projet au fur et à mesure.

4.2 Sprint Zéro

Le Sprint Zéro marque le point de départ du projet. Il joue un rôle essentiel car il pose les bases nécessaires avant de commencer le développement proprement dit. C'est à ce moment que nous avons pris le temps de bien comprendre ce que l'application devait offrir, en identifiant les utilisateurs concernés et les besoins à satisfaire. Cette étape a aussi permis de mettre en place l'équipe chargée du projet et de préciser les grandes lignes du fonctionnement attendu. En définissant clairement les besoins fonctionnels et non fonctionnels dès le début, nous avons pu démarrer le travail sur des bases solides et bien organisées.

4.2.1 Spécification des besoins

C'est une étape essentielle dans le processus de développement logiciel. Son objectif est de comprendre et de clarifier d'une manière approfondie les exigences et les attentes des parties prenantes concernant le développement du système logiciel. Ainsi, nous procéderons de la manière suivante :

4.2.1.1 Identification des acteurs

Les acteurs d'un système sont des entités externes à celui-ci qui interagissent avec lui afin d'atteindre un but précis. Les personnes, les organisations, les systèmes informatiques et les dispositifs matériels peuvent être des acteurs. C'est en identifiant les acteurs de notre système et en collaborant avec les parties prenantes qu'il sera possible de collecter les besoins fonctionnels, non fonctionnels, ainsi que les cas d'utilisation.

Notre système comprendra les acteurs suivants :

1. **Directeur.**
2. **Éducateur.**
3. **Parent.**

4.2.1.2 Diagramme de contexte

La figure 4.1, comprend le diagramme de contexte, met en évidence les principales interactions entre le système et ses acteurs.

NB

E : représente les messages émis par le système en réponse aux actions des acteurs.
 R : représente les messages reçus par le système, c'est-à-dire les actions effectuées par les acteurs sur le système.

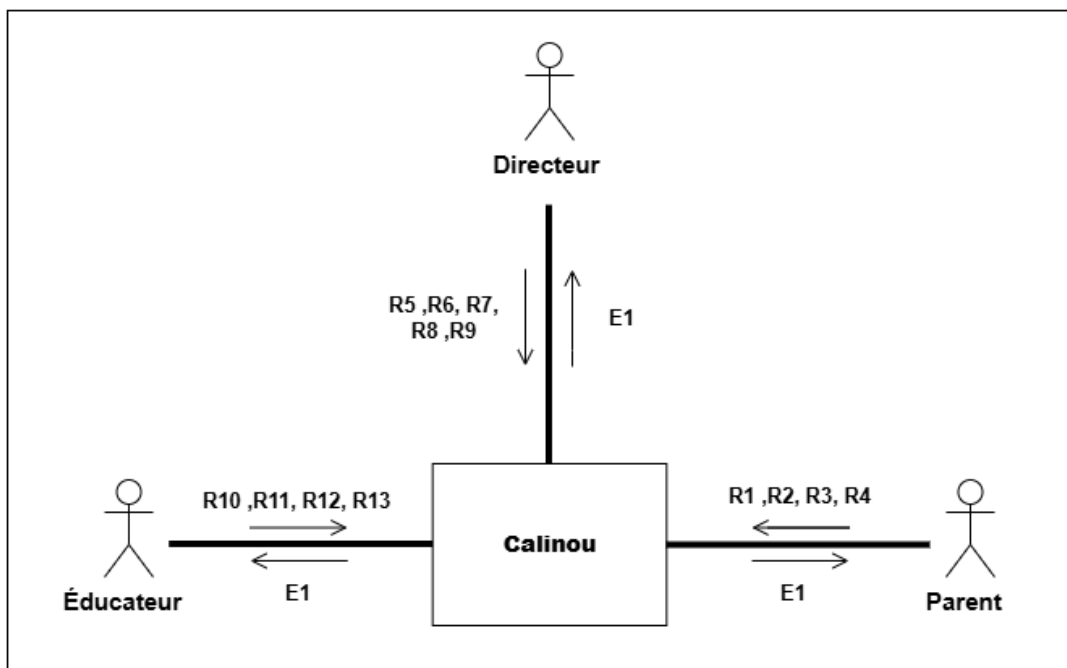


FIGURE 4.1 – Diagramme de Contexte".

4.2.1.3 Description des messages échangés

Le diagramme de contexte précédent décrit les trois acteurs qui utilisent le système. Pour chaque acteur, des flux d'échanges avec celui-ci sont observés. Les flèches sortantes du système représentent les messages émis aux acteurs, tandis que les flèches entrantes représentent les messages reçus par le système.

Le tableau suivant 4.1 illustre ces échanges :

ID	Description
Messages émis par le système	
E1	Afficher toutes les données du système (Directeur, Éducateur, Parent).
Messages reçus par le système	
R1	Envoyer une demande d'inscription ou de réinscription.
R2	Compléter le dossier d'inscription.
R3	Abandonner l'inscription.
R4	Suivre l'actualité.
R5	Traiter les inscriptions.
R6	Gérer les dossiers d'inscriptions.
R7	Gérer les factures.
R8	Gérer les plannings pédagogiques et les menus de repas.
R9	Gérer les personnels.
R10	Consulter le planning pédagogique de son groupe.
R11	Établir la fiche de suivi de l'enfant.
R12	Gérer les évaluations.
R13	Suivi les présences des enfants.

TABLE 4.1 – Messages échangés entre le système et les utilisateurs

4.2.1.4 Présentation de l'équipe scrum

Le Product Owner, le Scrum Master et une équipe de développement font partie de l'équipe Scrum.

M. Foudil MIR, le Scrum Master, simplifie le processus de développement en supprimant les difficultés et en veillant à ce que la méthodologie Scrum soit suivie. Mme Djouhra (la directrice de la crèche où nous avons effectué le stage) est responsable de la définition de la vision du produit et de la gestion du Product Backlog en accordant une priorité aux fonctionnalités en fonction des besoins des utilisateurs. Finalement, Malika LOULOU et Fadila MAIOU, membres de l'équipe de développement, travaillent ensemble afin de concevoir, développer et tester les fonctionnalités du projet.

4.2.2 Capture des besoins fonctionnels

Dans cette partie, nous allons capturer les besoins fonctionnels de la plate-forme pour une meilleure conception de la solution.

4.2.2.1 Les cas d'utilisation

Notre plate-forme possède un directeur ayant des privilèges et une responsabilité globale sur le système. Il est chargé de la gestion de la crèche et des fonctionnalités du système.

Bien que chaque acteur possède ses propres fonctionnalités, celles-ci ont été subdivisées et représentées comme suit dans le tableau 4.2 :

Acteur	Cas d'utilisation
Directeur	<ul style="list-style-type: none"> • Traitement des demandes d'inscriptions • Gestion des employés • Gestion des sections • Gestion des groupes • Gestion des saisons • Gestion des matières • Gestion des séances • Gestion des plannings pédagogiques et les menus des repas • Gestion des factures • Suivi des présences des employés • Gestion de la galerie de la crèche • Actualité • Gérer la messagerie
Éducateur	<ul style="list-style-type: none"> • Établir la fiche de suivi des enfants • Suivi des présences des enfants • Gestion des évaluations • Consulter le planning pédagogique de son groupe • Gérer la messagerie
Parent	<ul style="list-style-type: none"> • Envoyer une demande d'inscription • Suivre l'état d'avancement de sa demande • Communication avec la crèche via la messagerie • Consulter l'actualité de son enfant • Consulter le planning pédagogique • Consulter le menu des repas • Gérer la messagerie

TABLE 4.2 – Cas d'utilisation par acteur.

4.2.3 Capture des besoins non fonctionnels

Ce genre d'exigence met l'accent sur le fonctionnement du système et sur les critères de qualité à respecter, c'est-à-dire :

1. Les interfaces doivent être compatibles avec toutes les tailles d'écran (responsive).

2. Une interface claire et agréable visuellement, offrant une expérience utilisateur fluide et plaisante.
3. La possibilité de changer de langue.
4. Les informations personnelles des utilisateurs étant des données sensibles, elles doivent être sécurisées par des mécanismes de cryptage et protégées contre tout accès non autorisé.

4.2.4 Backlog produit du sprint zéro

La gestion du **Product Backlog** est une responsabilité clé du **Product Owner**. Après des échanges avec lui, nous avons recueilli les attentes des utilisateurs sous forme de **User Stories**, qui décrivent les besoins fonctionnels de manière claire :

"En tant que [acteur], je peux [action] afin de [objectif]."

Chaque User Story est évaluée selon sa **priorité** et son **niveau de risque** pour orienter efficacement le développement. Le tableau suivant 4.3 présente un aperçu global du **Backlog produit du sprint zéro** avec les fonctionnalités identifiées.

ID	User Story	Priorité	Risque	Ordre
US-01	En tant qu'utilisateur, je souhaite posséder un compte, et y accéder en m'authentifiant.	Élevée	Moyen	1
US-11	En tant que parent, je souhaite envoyer une demande d'inscription.	Élevée	Important	2
US-12	En tant que parent, je dois compléter le dossier d'inscription.	Élevée	Important	3
US-13	En tant que parent, je peux abandonner une inscription.	Élevée	Important	4
US-02	En tant que directeur, je peux gérer les catégories d'abonnements et les suppléments.	Élevée	Faible	5
US-03	En tant que directeur, je peux gérer les saisons.	Élevée	Faible	6
US-04	En tant que directeur, je peux gérer les groupes et sections.	Élevée	Faible	7
US-05	En tant que directeur, je peux gérer les employés.	Élevée	Moyen	8
US-06	En tant que directeur, je peux gérer les factures.	Élevée	Important	9
US-07	En tant que directeur, je peux gérer les plannings de cours et le menu des repas.	Élevée	Important	10
US-09	En tant qu'éducateur, je peux établir la fiche de suivi de chaque enfant et gérer ses évaluations.	Élevée	Important	11
US-10	En tant qu'éducateur, je peux signaler l'absence des enfants.	Élevée	Important	12
US-14	En tant que parent, je peux communiquer avec la crèche via une messagerie.	Élevée	Important	13

Suite du tableau 4.3				
ID	User Story	Priorité	Risque	Ordre
US-08	En tant que directeur, je peux gérer la galerie de la crèche ainsi que l'actualité.	Basse	Faible	14

TABLE 4.3 – Le Backlog produit du sprint zéro avec ordre de priorité.

4.2.5 Planification des sprints

La planification de notre projet s'est faite en trois sprints, chacun durant entre deux et quatre semaines.

Le premier sprint a permis de développer les principales fonctionnalités pour le directeur. Le deuxième sprint a ajouté celles destinées aux parents et aux éducateurs, tout en poursuivant celles du directeur. Enfin, le dernier sprint a intégré la messagerie interne, la gestion des actualités, de la galerie, des événements, des factures, ainsi que d'autres fonctionnalités pour les parents et les éducateurs.

N°du Sprint	Objectif principale	Durée
Sprint Un	Développement des fonctionnalités de base pour le directeur de la crèche.	4 semaines
Sprint Deux	Compléter le processus d'inscription de l'enfant ainsi que les tâches relatives aux utilisateurs du système.	4 semaines
Sprint Trois	Mise en place de la messagerie, gestion des actualités, de la galerie, des événements, des factures, ainsi que quelques fonctionnalités pour les parents et les éducateurs.	4 semaines

TABLE 4.4 – Règles de gestion : S'authentifier.

4.2.6 Présentation du logo de la plate-forme

Le logo de notre plateforme s'appelle Calinou. Ce nom est formé à partir des mots "câlin" et "nous". Il reflète un moment de tendresse et de proximité, comme un câlin partagé entre l'enfant et l'éducateur, ou entre l'enfant et ses parents. À travers ce nom, nous voulons transmettre une atmosphère chaleureuse, bienveillante et rassurante, où chaque enfant se sent en sécurité et entouré. Le logo s'inspire de cet univers doux, avec des formes simples et des couleurs douces qui évoquent la tendresse et l'enfance.



FIGURE 4.2 – Logo de la plate-forme.

4.3 Sprint Un

Le Sprint Un marque le début du développement concret de notre application. Pendant cette première itération, nous avons commencé à réaliser plusieurs fonctionnalités prévues dans le Backlog.

L'objectif était de traduire les besoins exprimés en éléments fonctionnels, prêts à être intégrés dans le système. Cette étape a donc permis de passer de la phase de préparation à celle de la mise en œuvre, en passant par les étapes de spécification, de développement et de validation.

4.3.1 Backlog produit du sprint Un

Lors du Sprint un, on a sélectionné des User Stories essentielles pour le projet, notamment celles liées à l'authentification des utilisateurs, telles que la gestion des sessions et des mots de passe. On a également travaillé sur d'autres fonctionnalités comme la gestion des employés, des saisons, des sections, des groupes, des catégories d'abonnement, des suppléments, notamment l'affectation des éducateurs à des groupes.

Ces tâches, sélectionnées à partir du Product Backlog, ont été intégrées au Sprint Backlog du sprint un, pour être complétées durant cette période.

Le tableau suivant 4.5 illustre le **Backlog produit du sprint un** :

ID US	Tâche	Durée
US-01	Élaborer le CU et la description de la fonctionnalité "S'authentifier". Ajouter les diagrammes de séquence nécessaires	4h
	Développer et tester les fonctionnalités d'authentification.	5h
US-02	Élaborer le CU et description de la fonctionnalité "gestion des catégories d'abonnement". Ajouter les diagrammes de séquences nécessaires	4h
	Développer et tester la gestion des catégories d'abonnement.	18h
US-03	Élaborer le CU et la description de la fonctionnalité "gestion des suppléments". Ajouter les diagrammes de séquences nécessaires	4h
	Développer et tester la gestion des suppléments.	18h
US-04	Élaborer le CU et la description de la fonctionnalité "gestion des sections". Ajouter les diagrammes de séquences nécessaires	4h
	Développer et tester la gestion des sections.	18h
US-05	Élaborer le CU et la description de la fonctionnalité "gestion des groupes". Ajouter les diagrammes de séquences nécessaires	4h
	Développer et tester la gestion des groupes.	18h
US-06	Élaborer le CU et la description de la fonctionnalité "gestion saisons". Ajouter les diagrammes de séquences nécessaires	5h
	Développer et tester la gestion des saisons.	20h
US-07	Élaborer le CU et la description de la fonctionnalité "gestion des employés". Ajouter les diagrammes de séquences nécessaires	4h
	Développer et tester la gestion des employés.	20h

Continuation du tableau 4.5		
ID US	Tâche	Durée
US-08	Élaborer le CU de la fonctionnalité "affecter des éducateurs aux groupes". Ajouter les diagrammes de séquences nécessaires	4h
	Développer et tester la fonctionnalité d'affecter des éducateurs aux groupes. Ajouter les diagrammes de séquences nécessaires	10h

TABLE 4.5 – Backlog produit du sprint un

4.3.2 Spécifications des besoins

Cette étape est importante car elle permet de bien cadrer le projet dès le départ. On y précise ce que le système devra faire, les objectifs visés, et les contraintes à prendre en compte, qu'elles soient techniques ou liées au contexte de la crèche. Cela aide à éviter les imprévus plus tard.

Une fois que tout est validé, ces éléments servent de base pour la suite : ils orientent la conception, facilitent le développement et permettent de mieux préparer les tests.

4.3.3 Diagramme de cas d'utilisation du sprint un

Le diagramme présenté dans la Figure 4.3 représente les cas d'utilisation identifiés lors du premier sprint. Il décrit les différentes interactions possibles entre le directeur et le système, en mettant en évidence les fonctionnalités principales liées à la gestion de la crèche.

Parmi ces fonctionnalités, on retrouve la gestion des employés, des éducateurs, des saisons, des sections, des groupes, des catégories d'abonnement, des services et des suppléments. Pour chacune de ces entités, le directeur a la possibilité d'ajouter, modifier, supprimer, rechercher ou encore activer ou désactiver certains éléments. Le diagramme montre également des opérations spécifiques comme l'affectation d'un éducateur à un groupe ou l'export des données en format PDF ou Excel.

Ce schéma permet d'avoir une vue d'ensemble claire sur les besoins fonctionnels exprimés durant cette phase du projet et servira de base pour les étapes de conception détaillée.

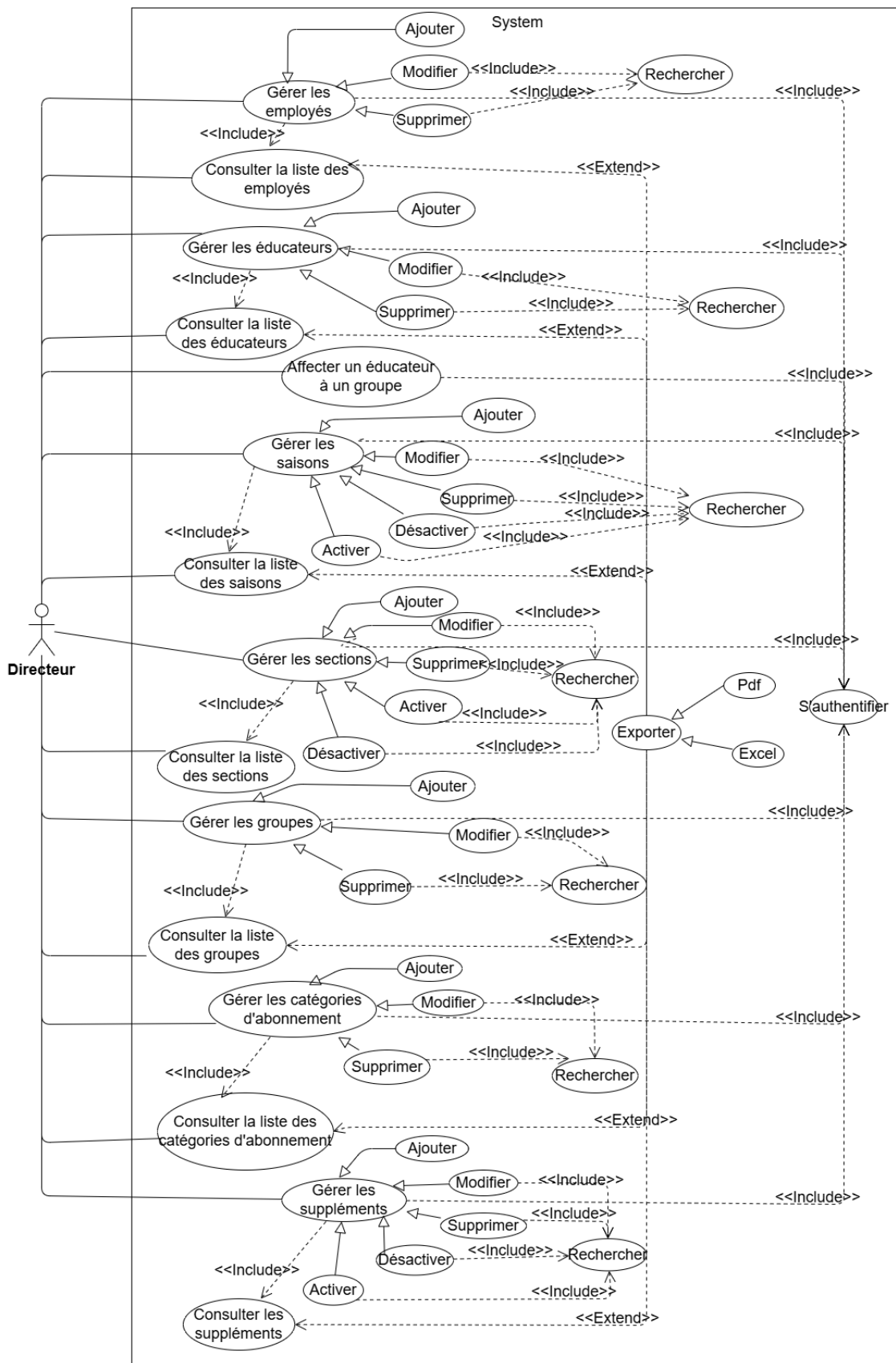


FIGURE 4.3 – Diagramme du cas d'utilisation du sprint un

4.3.4 Description textuelle

Dans cette section, nous décrivons les cas d'utilisation les plus importants réalisés lors du sprint un. Chaque description présente le déroulement de l'action du point de vue de l'utilisateur, tout en prenant en compte les vérifications nécessaires au bon fonctionnement de la fonctionnalité. Cela permet de mieux comprendre comment se déroule chaque interaction avec le système et sous quelles conditions elle peut être validée.

4.3.4.1 Cas d'utilisation "S'authentifier"

TABLE 4.6 – Règles de gestion : S'authentifier.

ID	Règles de gestion
RG-001	L'email doit exister et être valide.
RG-002	Le mot de passe doit être identique à celui de la base de données.

TABLE 4.7 – Description textuelle du CU : S'authentifier.

CU : S'authentifier
ID : 1
Description brève : L'étape primaire pour accéder au système.
Acteurs primaires : Tous
Pré-condition : L'utilisateur doit posséder un compte.
Enchaînement principal : Le cas d'utilisation démarre lorsque l'utilisateur accède au système. 1. L'utilisateur saisit son e-mail et son mot de passe. 2. L'utilisateur clique sur « se connecter » et attend la fin du chargement. 3. Le système valide l'authentification de l'utilisateur.
Post-condition : L'utilisateur accède à sa session.
Enchaînement alternatif : 3.1. Si le mot de passe est incorrect, l'application indique à l'utilisateur que ses informations d'identification sont erronées et l'invite à les ressaisir de nouveau. Le scénario reprend au point 1. 3.2.1. En cas d'oubli de mot de passe, l'utilisateur clique sur le bouton "Mot de passe oublié". 3.2.2. Le système lui affiche un formulaire en lui demandant l'e-mail associé à son compte. 3.2.3. L'utilisateur saisit son e-mail, et envoie les informations. 3.2.4. Le système envoie un e-mail à l'utilisateur contenant un lien de réinitialisation. 3.2.5. L'utilisateur reçoit l'e-mail et clique sur le lien qui le redirige vers une page de réinitialisation du mot de passe. 3.2.6. L'utilisateur saisit son nouveau mot de passe, le confirme et valide l'opération. 3.2.7. Le système enregistre les nouvelles informations. Le scénario reprend au point 1. 3.3. Si l'utilisateur se connecte avec un e-mail inexistant, le système affiche une erreur indiquant que l'adresse e-mail n'est pas enregistrée dans la base de données.

4.3.4.2 Cas d'utilisation "Rechercher"

TABLE 4.8 – Description textuelle du CU : Recherche côté client.

CU : Recherche côté client
ID : 2
Description brève : Permet à l'utilisateur de rechercher une information (ex : catégorie d'abonnement, saison, groupe, enfant, etc.) de manière rapide et efficace via le champ de recherche.
Acteurs primaires : Tous les utilisateurs connectés
Pré-condition : L'utilisateur doit être connecté au système.
Enchaînement principal : Le cas d'utilisation commence lorsque l'utilisateur souhaite retrouver une information précise. <ol style="list-style-type: none"> 1. L'utilisateur accède à la page contenant la liste recherchée (par exemple : liste des saisons). 2. Il saisit un mot-clé dans la barre de recherche (par exemple : "2024"). 3. Le système envoie la requête au serveur (backend) pour filtrer les données. 4. Le serveur retourne les résultats correspondants, qui sont affichés à l'utilisateur.
Post-condition : Les résultats de la recherche sont affichés à l'écran.
Enchaînements alternatifs : <ol style="list-style-type: none"> 4.1 Aucun résultat trouvé : Le système affiche un message : "Aucun résultat trouvé". 4.2 Saisie modifiée : L'utilisateur peut corriger ou modifier le mot-clé pour relancer la recherche.

4.3.4.3 Cas d'utilisation "Gérer les catégories d'abonnement"

TABLE 4.9 – Description textuelle du CU : Gérer les catégories d'abonnement.

CU : Gérer les catégories d'abonnement
ID : 3
Description brève : Permettre au directeur de la crèche de gérer les informations relatives aux catégories d'abonnement proposées par la crèche, ainsi que l'ajout, la modification et la suppression.
Acteurs primaires : Directeur
Acteurs secondaire : Aucun
Pré-condition : <ol style="list-style-type: none"> 1. Le directeur doit être authentifié. 2. Une saison active doit exister dans le système. 3. La catégorie d'abonnement qu'on souhaite modifier ou supprimer doit exister dans la base de données.
Enchaînement principal : A. Le directeur souhaite ajouter une nouvelle catégorie d'abonnement. <ol style="list-style-type: none"> 1. Le directeur accède à l'interface de gestion des catégories d'abonnement. 2. Le directeur clique sur le bouton "Ajouter une catégorie d'abonnement". 3. Le système affiche un formulaire pour saisir les informations de la catégorie d'abonnement telles que : le type de la catégorie d'abonnement (journée ou demi-journée). L'option "avec repas" (oui /non), et une description optionnelle pour décrire la catégorie choisie. 3. Le directeur saisit et valide les informations.

Continuation du tableau 4.9
<p>4. Le système enregistre la nouvelle catégorie d'abonnement dans la base de données si elle n'existe pas déjà, et affiche un message de succès confirmant l'ajout.</p> <p>B. Le directeur souhaite modifier une catégorie d'abonnement existante.</p> <ol style="list-style-type: none"> 1. Le directeur accède à l'interface de gestion des catégories d'abonnement. 2. Il saisit un critère de recherche (par exemple : type de la catégorie) dans le champ de recherche. 3. Le système effectue une recherche dans la base de données et affiche la ou les catégories d'abonnement correspondantes. 4. Le directeur clique sur "Éditer" pour la catégorie d'abonnement souhaitée dans la liste. 5. Le système affiche un formulaire déjà rempli avec les informations actuelles de la catégorie d'abonnement. 6. Le directeur modifie les informations de la catégorie d'abonnement souhaitée. 7. Le directeur valide les modifications. 8. Le système met à jour les informations de la catégorie d'abonnement dans la base de données et affiche un message de succès. <p>C. Le directeur souhaite supprimer une catégorie d'abonnement :</p> <ol style="list-style-type: none"> 1. Le directeur accède à l'interface de gestion des catégories d'abonnement. 2. Il saisit un critère de recherche (par exemple, type de la catégorie) dans le champ de recherche. 3. Le système effectue une recherche dans la base de données et affiche les résultats correspondants. 4. Le directeur clique sur "Supprimer" pour la catégorie d'abonnement souhaitée dans la liste. 5. Le système affiche un message de confirmation. 6. Après confirmation, le système supprime la catégorie d'abonnement de la base de données et affiche un message de succès.
<p>Post-condition : Les modifications (ajout, mise à jour ou suppression) des catégories d'abonnement sont correctement enregistrées dans le système et visibles dans la liste des catégories d'abonnement.</p>
<p>Enchaînements alternatifs :</p> <ol style="list-style-type: none"> 1. Aucune saison active : Le système affiche un message d'erreur et empêche l'ajout, la modification ou la suppression. 2. Ajout d'une catégorie d'abonnement existante : Le système affiche un message d'erreur signalant que cette catégorie existe déjà. 3. Modification avec données invalides : Le système affiche un message d'erreur indiquant les champs invalides. 4. Suppression d'une catégorie liée à des abonnements ou sections : Le système empêche la suppression et affiche un message expliquant que la catégorie est utilisée.

4.3.4.4 Cas d'utilisation "Gérer les suppléments"

TABLE 4.10 – Description textuelle du CU : Gérer les suppléments.

CU : Gérer les suppléments
ID : 4

Continuation du tableau 4.10
Description brève : Permettre au directeur de la crèche de gérer les informations relatives aux suppléments proposés par la crèche (comme le transport ou des services additionnels), y compris l'ajout, la modification, l'activation et la suppression.
Acteurs primaires : Directeur
Acteurs secondaires : Aucun
Pré-condition : <ol style="list-style-type: none"> 1. Le directeur doit être authentifié. 2. Une saison active doit exister dans le système. 3. Le supplément à modifier, supprimer ou activer doit exister dans la base de données.
Enchaînement principal : A. Le directeur souhaite ajouter un nouveau supplément. <ol style="list-style-type: none"> 1. Le directeur accède à l'interface de gestion des suppléments. 2. Il clique sur le bouton "Ajouter un supplément". 3. Le système affiche un formulaire de saisie comprenant : le nom du supplément, sa description (facultative) et son prix. 4. Le directeur saisit et valide les informations. 5. Le système vérifie l'unicité du supplément (même nom et prix). 6. Si les données sont valides et non dupliquées, le supplément est ajouté à la base de données avec un lien actif à la saison actuelle, et le système affiche un message de succès. 7. Le système lie le supplément aux autres saisons futures avec un état inactif. B. Le directeur souhaite modifier un supplément existant. <ol style="list-style-type: none"> 1. Le directeur accède à l'interface de gestion des suppléments. 2. Il recherche le supplément à modifier. 3. Le système affiche les résultats de la recherche. 4. Le directeur clique sur "Éditer". 5. Le système affiche un formulaire pré-rempli. 6. Le directeur modifie les informations puis valide. 7. Le système met à jour les données dans la base et affiche un message de succès. C. Le directeur souhaite supprimer un supplément. <ol style="list-style-type: none"> 1. Le directeur accède à l'interface de gestion des suppléments. 2. Il recherche le supplément à supprimer. 3. Le système affiche les résultats. 4. Le directeur clique sur "Supprimer". 5. Le système affiche un message de confirmation. 6. Après confirmation, le système détache le supplément de toutes les saisons, puis le supprime définitivement. D. Le directeur souhaite activer ou désactiver un supplément. <ol style="list-style-type: none"> 1. Le directeur accède à l'interface de gestion des suppléments. 2. Il recherche et sélectionne le supplément souhaité. 3. Le système vérifie l'attachement du supplément à la saison actuelle. 4. Pour l'activation : <ul style="list-style-type: none"> - Si le supplément est lié mais inactif, le système l'active et affiche un message de succès. - Sinon (déjà actif ou non lié à la saison actuelle), un message explicatif est affiché. 5. Pour la désactivation : <ul style="list-style-type: none"> - Si le supplément est lié et actif, le système le désactive et affiche un message de succès. - Sinon (déjà inactif ou non lié à la saison actuelle), un message explicatif est affiché.

Continuation du tableau 4.10
Post-condition : Les actions d'ajout, de mise à jour, d'activation, de désactivation ou de suppression des suppléments sont prises en compte dans la base de données et visibles dans l'interface.
Enchaînements alternatifs :
1. Aucune saison active : Le système empêche toute opération et affiche un message d'erreur.
2. Ajout d'un supplément existant : Le système rejette l'opération et affiche une erreur.
3. Modification avec données invalides : Le système affiche les champs incorrects et demande correction.
4. Activation d'un supplément déjà actif : Le système affiche un message indiquant que le supplément est déjà actif.
5. Désactivation d'un supplément déjà inactif : Le système affiche un message indiquant que le supplément est déjà inactif.
6. Suppression impossible : Si une erreur survient lors de la suppression (ex. : dépendance technique), le système affiche un message d'erreur.
7. Supplément non trouvé : Si un ID incorrect est fourni, un message d'erreur est retourné.

4.3.4.5 Cas d'utilisation "Gérer les sections"

TABLE 4.11 – Description textuelle du CU : Gérer les sections.

CU : Gérer les sections
ID : 5
Description brève : Permettre au directeur de la crèche de gérer les informations relatives aux sections : ajout, modification, suppression, activation ou désactivation.
Acteurs primaires : Directeur
Acteurs secondaires : Aucun
Pré-conditions :
1. Le directeur est authentifié.
2. La section à modifier, supprimer, activer ou désactiver doit exister dans la base de données.
Enchaînement principal :
A. Le directeur souhaite ajouter une nouvelle section.
1. Le directeur accède à l'interface de gestion des sections.
2. Le directeur clique sur "Ajouter une section".
3. Le système affiche un formulaire pour saisir les informations : nom, âge minimum, âge maximum, description.
4. Le directeur remplit et valide le formulaire.
5. Le système vérifie que la section n'existe pas déjà (nom ou tranche d'âge).
6. Si tout est valide, la section est enregistrée et rattachée à la saison actuelle.
7. Le système affiche un message de succès.
B. Le directeur souhaite modifier une section existante.
1. Le directeur accède à l'interface de gestion des sections.
2. Il recherche la section à modifier.
3. Le système affiche les résultats de la recherche.
4. Le directeur clique sur "Éditer".
5. Le système affiche un formulaire pré-rempli avec les informations actuelles.

Continuation du tableau 4.11
<p>6. Le directeur modifie les champs souhaités et valide. 7. Le système vérifie la validité des données et l'unicité du nom ou de la tranche d'âge. 8. Le système met à jour la base de données et affiche un message de succès.</p> <p>C. Le directeur souhaite supprimer une section.</p> <p>1. Le directeur accède à l'interface de gestion des sections. 2. Il recherche la section à supprimer. 3. Le système affiche les résultats de la recherche. 4. Le directeur clique sur "Supprimer". 5. Le système affiche un message de confirmation. 6. Après confirmation, le système vérifie si la section est liée à un groupe actif. 7. Si non, le système supprime la section et affiche un message de succès.</p> <p>D. Le directeur souhaite activer une section.</p> <p>1. Le directeur accède à l'interface de gestion des sections. 2. Il recherche la section à activer. 3. Le système affiche les résultats. 4. Le directeur clique sur "Activer". 5. Le système vérifie si la section est rattachée à la saison actuelle. 6. Si elle est inactive, elle est activée et un message de succès est affiché. Sinon, un message d'information est affiché.</p> <p>E. Le directeur souhaite désactiver une section.</p> <p>1. Le directeur accède à l'interface de gestion des sections. 2. Il recherche la section à désactiver. 3. Le système affiche les résultats. 4. Le directeur clique sur "Désactiver". 5. Le système vérifie si la section est active et attachée à la saison actuelle. 6. Si oui, la section est désactivée et un message de succès est affiché. Sinon, un message d'information est affiché.</p>
<p>Post-condition : Toutes les modifications sont enregistrées et reflétées dans la liste des sections.</p>
<p>Enchaînements alternatifs :</p> <p>1. Ajout d'une section existante : Le système affiche un message d'erreur. 2. Informations invalides (âge min > âge max) : Message d'erreur sur les champs invalides. 3. Suppression d'une section liée à un groupe actif : Le système bloque l'action et affiche un message d'information. 4. Activation/désactivation d'une section non liée à la saison : Le système affiche un message d'erreur.</p>

4.3.4.6 Cas d'utilisation "Gérer les groupes"

TABLE 4.12 – Description textuelle du CU : Gérer les groupes.

CU : Gérer les groupes
ID : 6
Description brève : Permet au directeur de la crèche de gérer les groupes : ajout, modification, suppression, tri et recherche.
Acteurs primaires : Directeur
Acteurs secondaires : Aucun
Pré-conditions :

Continuation du tableau 4.12

1. Le directeur est authentifié.
2. Une saison scolaire active existe.
3. Des sections actives sont disponibles.
4. Au moins une salle est enregistrée.

Enchaînement principal :

A. Le directeur souhaite ajouter un groupe :

1. Le directeur accède à l'interface de gestion des groupes.
2. Le directeur clique sur "Ajouter un groupe".
3. Le système affiche un formulaire de saisie.
4. Le directeur renseigne le nom, la section, la salle, la description et les mois min/max.
5. Le système vérifie l'unicité du groupe, la validité des données et la disponibilité de la salle.
6. Le système enregistre le groupe et affiche un message de succès.

C. Le directeur souhaite modifier un groupe existant :

1. Le directeur accède à l'interface de gestion des groupes.
2. Il recherche le groupe à modifier en filtrant par nom, section ou état.
3. Le système affiche les résultats de la recherche.
4. Le directeur clique sur "Éditer" à côté du groupe souhaité.
5. Le système affiche un formulaire pré-rempli avec les données actuelles du groupe.
6. Le directeur modifie les champs nécessaires (nom, salle, section, description, etc.).
7. Le système vérifie la validité des données saisies et la disponibilité de la salle.
8. Si les données sont valides, le système enregistre les modifications.
9. Le système affiche un message de succès.

D. Le directeur souhaite supprimer un groupe :

1. Le directeur accède à l'interface de gestion des groupes.
2. Il recherche le groupe à supprimer.
3. Le système affiche les résultats de la recherche.
4. Le directeur clique sur "Supprimer" à côté du groupe concerné.
5. Le système affiche un message de confirmation.
6. Après confirmation, le système vérifie si des enfants sont affectés à ce groupe.
7. Si aucun enfant n'est affecté, le système supprime le groupe.
8. Le système affiche un message de succès.

Post-conditions : Les modifications sont enregistrées et visibles dans la liste des groupes.

Enchaînements alternatifs :

1. Ajout d'un groupe avec un nom déjà existant dans la même section :
Le système affiche une erreur : "Ce groupe existe déjà".
2. Période invalide ($\text{mois_min} > \text{mois_max}$) : Le système refuse la soumission et affiche un message d'erreur.
3. Salle déjà affectée à un groupe pour la saison en cours :
Le système empêche l'ajout ou la modification du groupe et signale une collision de salle.
4. Suppression impossible (enfants encore affectés au groupe) :
Le système affiche un message bloquant la suppression : " Ce groupe contient encore des enfants ".
5. Aucune saison active ou section disponible :
Le système redirige l'utilisateur vers la gestion des saisons ou des sections et affiche un message explicatif.

4.3.4.7 Cas d'utilisation "Gérer les employés"

TABLE 4.13 – Description textuelle du CU : Gérer les employés.

CU : Gérer les employés
ID : 7
Description brève : Permettre au directeur de la crèche la possibilité d'ajouter, de modifier, de visualiser, et de supprimer les employés selon son poste. Un compte utilisateur peut être créé lui permettant d'accéder à certaines fonctionnalités.
Acteurs primaires : Directeur
Acteurs secondaire : Aucun
Pré-condition : <ol style="list-style-type: none"> 1. Le directeur doit être authentifié et il a les droits d'accès à la gestion des employés. 2. L'employé à gérer doit exister dans la base de données.
Enchaînement principal : <ol style="list-style-type: none"> 1. Le directeur accède à l'interface de gestion des employés : A. Le directeur souhaite ajouter un nouvel employé : <ol style="list-style-type: none"> A.a. Le poste d'emploi ne nécessite pas un compte utilisateur. 2. Le système affiche un formulaire avec des informations importantes à remplir telles que : nom, prénom, date de naissance, adresse, etc. 3. Le directeur saisit et valide les informations personnelles. 4. Le système enregistre le nouvel employé dans la base de données, et affiche un message de succès confirmant l'ajout. <ol style="list-style-type: none"> A.b. Le poste d'emploi nécessite un compte utilisateur (chauffeur, gérant, éducateur). 2. Le système affiche un formulaire des informations importantes à remplir. 3. Le directeur saisit et valide les informations personnelles ainsi que l'adresse e-mail et mot de passe pour que l'utilisateur se connecte au système. 4. Le système enregistre le nouvel employé et un compte utilisateur est créé, envoie un e-mail de confirmation avec les identifiants, et affiche un message de succès confirmant l'ajout. B. Le directeur souhaite modifier un employé existant : <ol style="list-style-type: none"> 1.2. Le directeur peut rechercher l'employé spécifique qu'il souhaite gérer, en utilisant des critères de recherche. 1.3. Le système affiche une liste des employés correspondant aux critères de recherche. 1.4. Le directeur clique sur "Éditer" pour l'employé souhaité dans la liste. 1.5. Le système affiche un formulaire déjà rempli avec les informations actuelles de l'employé. 1.6. Le directeur modifie les informations de l'employé souhaité et valide les modifications. 1.7. Le système met à jour les informations du groupe dans la base de données et affiche un message de succès. C. Le directeur souhaite supprimer un employé : <ol style="list-style-type: none"> 1.2. Le directeur peut rechercher l'employé spécifique qu'il souhaite gérer, en utilisant des critères de recherche. 1.3. Le système affiche une liste des employés correspondant aux critères de recherche. 1.4. Le directeur clique sur "Supprimer" pour l'employé souhaité dans la liste. 1.5. Le système affiche un message de confirmation. 1.6. Après confirmation, le système supprime l'employé de la base de données et affiche un message de succès.
Post-condition :

Continuation du tableau 4.13
1. Les informations de l'employé sont enregistrées et mises à jour dans la base de données. 2. Un compte utilisateur est créé c'est nécessaire.
Enchaînement alternatif : A. Lors de l'ajout : A.3. Le directeur saisit les informations invalides, le système affiche un message d'erreur indiquant que certaines informations sont invalides. A.3.1. Le directeur insère un employé avec une adresse e-mail ou un numéro de carte selon le type de carte existant dans la base de données, le système affiche un message l'informant que l'employé existe déjà. B. Lors de la modification : 4.1. Le directeur modifie le poste d'un employé, le système affiche un message l'informant que certaines informations doivent être renseignées selon le nouveau poste. C. Lors de la suppression : 4.2. Le directeur ne peut pas supprimer un employé possédant un compte actif.

4.3.4.8 Cas d'utilisation "Exportation en PDF ou en Excel"

TABLE 4.14 – Description textuelle du CU : Exporter en PDF ou Excel.

CU : Exporter en PDF ou Excel
ID : 8
Description brève : Permet à l'utilisateur d'exporter des données affichées dans le système au format PDF ou Excel.
Acteurs primaires : Tous
Pré-condition : 1. L'utilisateur doit être authentifié et dispose des droits d'accès aux données à exporter. 2. La donnée souhaitée à exporter doit exister dans la base de données.
Enchaînement principal : Le cas d'utilisation commence lorsque l'utilisateur souhaite exporter une liste. 1. L'utilisateur accède à la page contenant la liste à exporter. 2. L'utilisateur clique sur le bouton "Exporter". 3. Le système affiche les options de format : DF ou Excel. 4. L'utilisateur sélectionne le format souhaité. 5. Le système génère le fichier dans le format choisi. 6. Le fichier est automatiquement téléchargé sur l'ordinateur de l'utilisateur.
Post-condition : Le fichier est généré et téléchargé sur la machine de l'utilisateur.
Enchaînement alternatif : 4.1. Si l'utilisateur ferme la fenêtre sans sélectionner un format, l'exportation est annulée. 5.1. Si le système rencontre une erreur lors de la génération du fichier (par exemple : erreur serveur), un message d'erreur est affiché. 5.2. L'utilisateur a la possibilité de relancer l'exportation.

4.3.4.9 Cas d'utilisation "Activer ou Désactiver une saison"

TABLE 4.15 – Description textuelle du CU : Activer et désactiver une saison.

CU : Activer et désactiver une saison
ID : 9
Description brève : Permet à l'utilisateur de gérer l'état d'une saison en l'activant ou en la désactivant pour l'utilisation dans le système.
Acteurs primaires : Directeur
Pré-condition : Le directeur doit être authentifié.
Enchaînement principal : Le cas d'utilisation commence lorsque le directeur souhaite activer ou désactiver une saison. 1. Le directeur accède à la liste des saisons disponibles. 2. Il sélectionne la saison à gérer. 3. Le système affiche une option pour activer ou désactiver la saison. 4. Le directeur clique sur "Activer" ou "Désactiver". 5. Le système met à jour l'état de la saison dans la base de données. 6. Le système affiche un message confirmant le succès de l'opération.
Post-condition : L'état de la saison est modifié (active ou désactivée) selon l'action du directeur.
Enchaînement alternatif : 5.1. En cas d'erreur lors de la mise à jour, le système conserve l'état actuel de la saison.

4.3.5 Diagrammes de séquences détaillés

Dans cette étape du projet, nous avons cherché à établir une base claire pour le développement de l'application. L'objectif était de bien définir l'architecture du système, ses composants principaux et les éléments nécessaires à sa mise en œuvre. Une conception bien pensée facilite non seulement la compréhension du fonctionnement général, mais aussi l'évolution et la maintenance future du logiciel.

Dans ce cadre, les diagrammes de séquence détaillés jouent un rôle important. Ils illustrent comment les différents acteurs ou objets du système interagissent entre eux, en précisant l'ordre des échanges. Cela permet de visualiser le déroulement des actions et de mieux comprendre le fonctionnement interne du système.

Pour représenter ces échanges, nous avons utilisé l'approche MVC, où :

- M désigne le modèle : c'est-à-dire la gestion des données.
- V représente la vue : qui correspond à l'interface utilisateur.
- C correspond au contrôleur : chargé de gérer les interactions et les traitements.

4.3.5.1 Diagramme de séquence détaillé "S'authentifier"

Le cas d'utilisation « S'authentifier » est essentiel pour toute plate-forme ou système informatique.

L'utilisateur renseigne ses identifiants, lesquels sont ensuite vérifiés par le système à travers une comparaison avec les données enregistrées dans la base de données. En fonction du résultat de la vérification, l'utilisateur est soit authentifié, soit invité à saisir de nouveau ses identifiants. Ce cas d'utilisation peut s'étendre à d'autres situations selon les conditions prédéfinies.

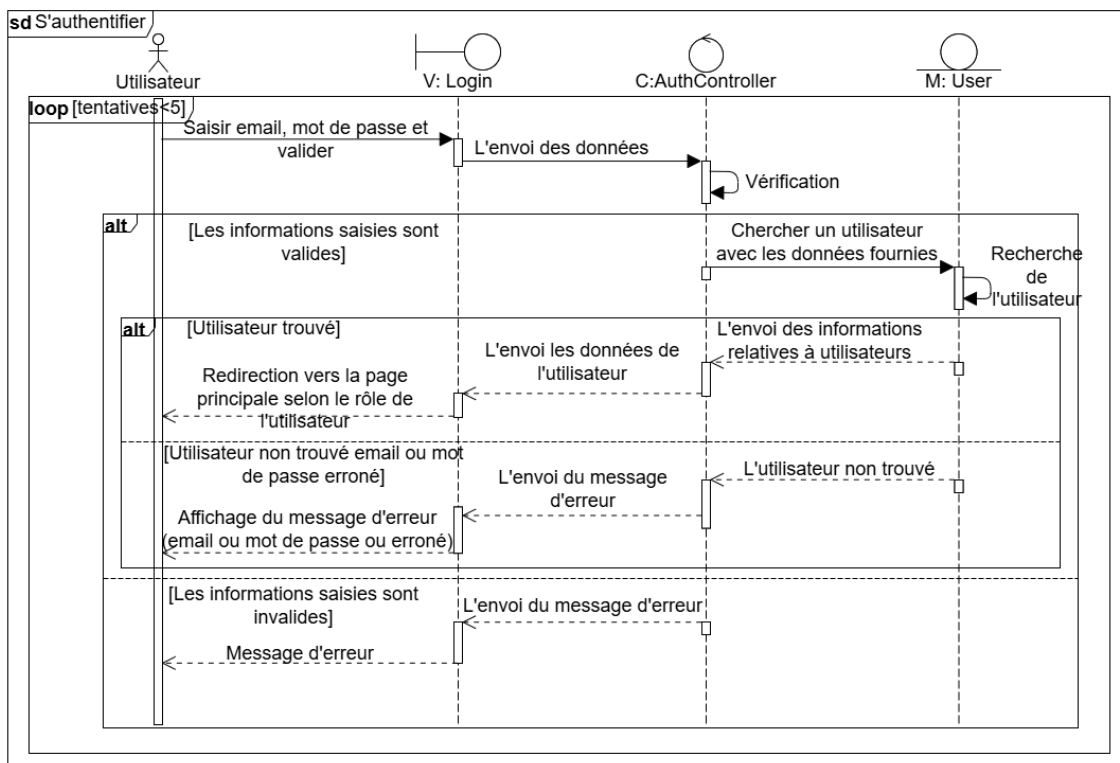


FIGURE 4.4 – Diagramme de séquence détaillé "S'authentifier".

4.3.5.2 Diagramme de séquence détaillé "Rechercher un objet"

Le cas d'utilisation « Rechercher un objet » permet à l'utilisateur de retrouver rapidement un objet précis enregistré dans le système. Pour cela, il lui suffit de saisir un ou plusieurs critères dans le champ de recherche. Le système traite la demande et affiche une liste de résultats correspondant aux informations fournies.

Ce diagramme de séquence, illustré dans la Figure 4.5, décrit les échanges entre l'utilisateur et le système lors de cette opération.

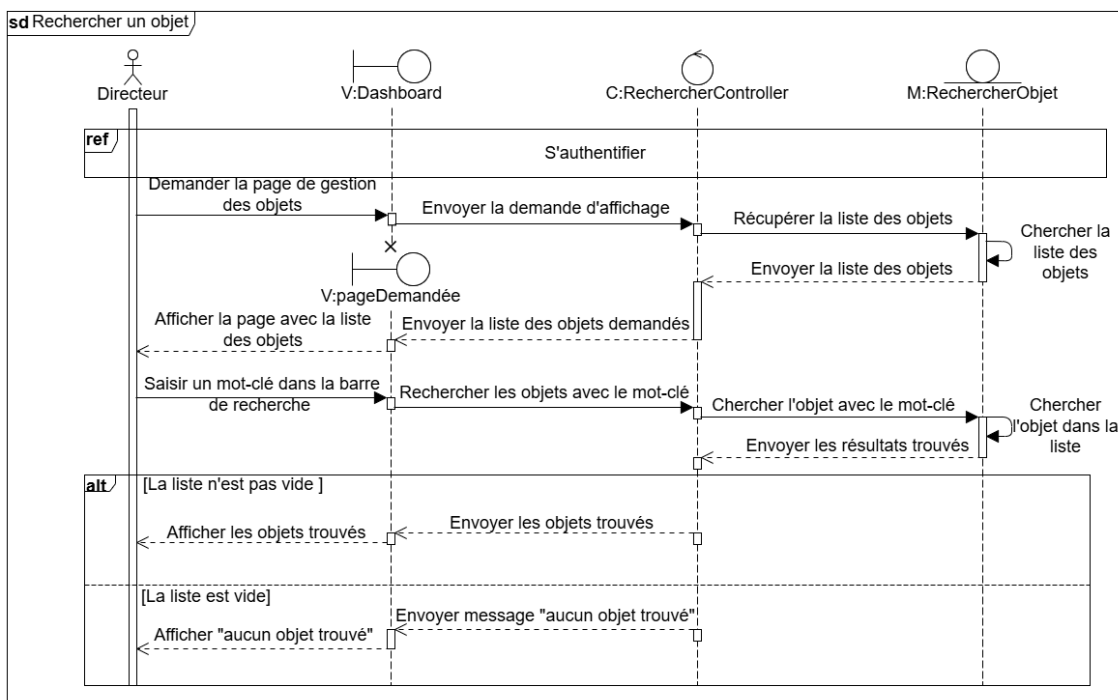


FIGURE 4.5 – Rechercher un objet.

4.3.5.3 Diagramme de séquence détaillé "Rechercher une saison"

Le cas d'utilisation « Rechercher une saison » permet à l'utilisateur d'accéder facilement aux informations d'une saison précise enregistrée dans le système. Pour cela, il saisit des critères de recherche, comme l'année ou le nom de la saison. Le système analyse la requête, interroge la base de données, puis affiche les résultats correspondants.

Le diagramme de séquence présenté dans la Figure 4.6 illustre les différentes interactions entre l'utilisateur et le système lors de cette recherche.

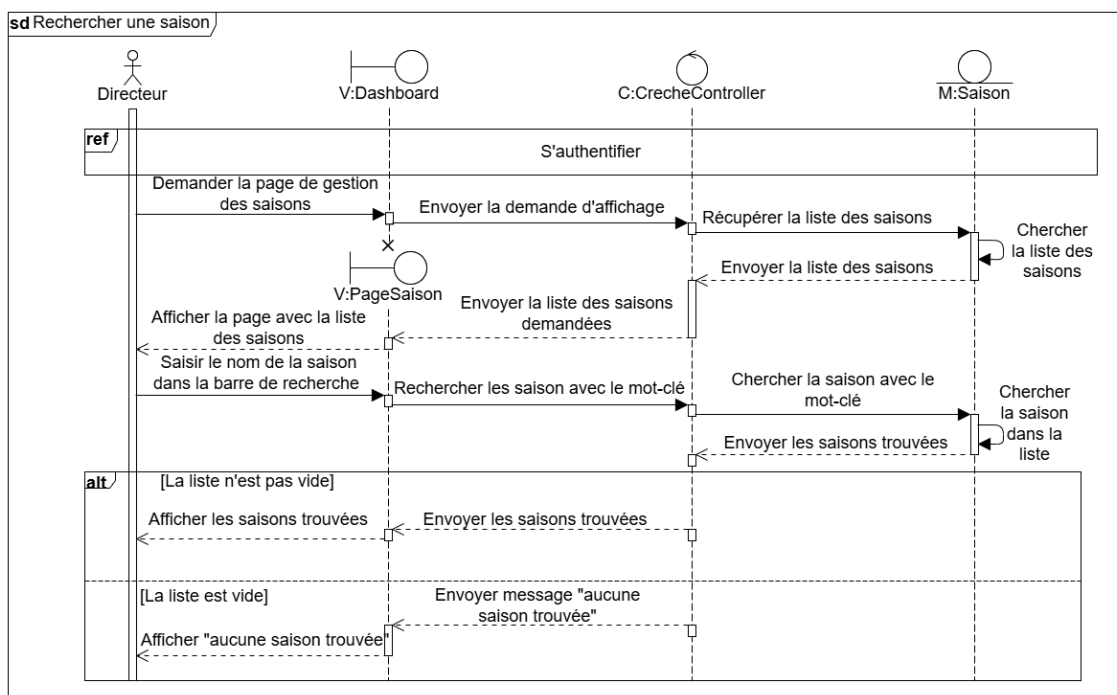


FIGURE 4.6 – Diagramme de séquence détaillé "Rechercher une saison".

4.3.5.4 Diagramme de séquence détaillé "Ajouter un employé"

Le cas d'utilisation « Gérer un employé » permet d'ajouter, modifier ou supprimer un employé.

Après l'authentification, le directeur peut effectuer l'ajout d'un employé, le système vérifie la validité et l'unicité des informations et qu'aucun employé n'existe déjà avec les mêmes identifiants.

Si toutes les données sont valides, l'employé est ajouté avec succès. En cas d'erreur, le système retourne un message d'échec de l'ajout.

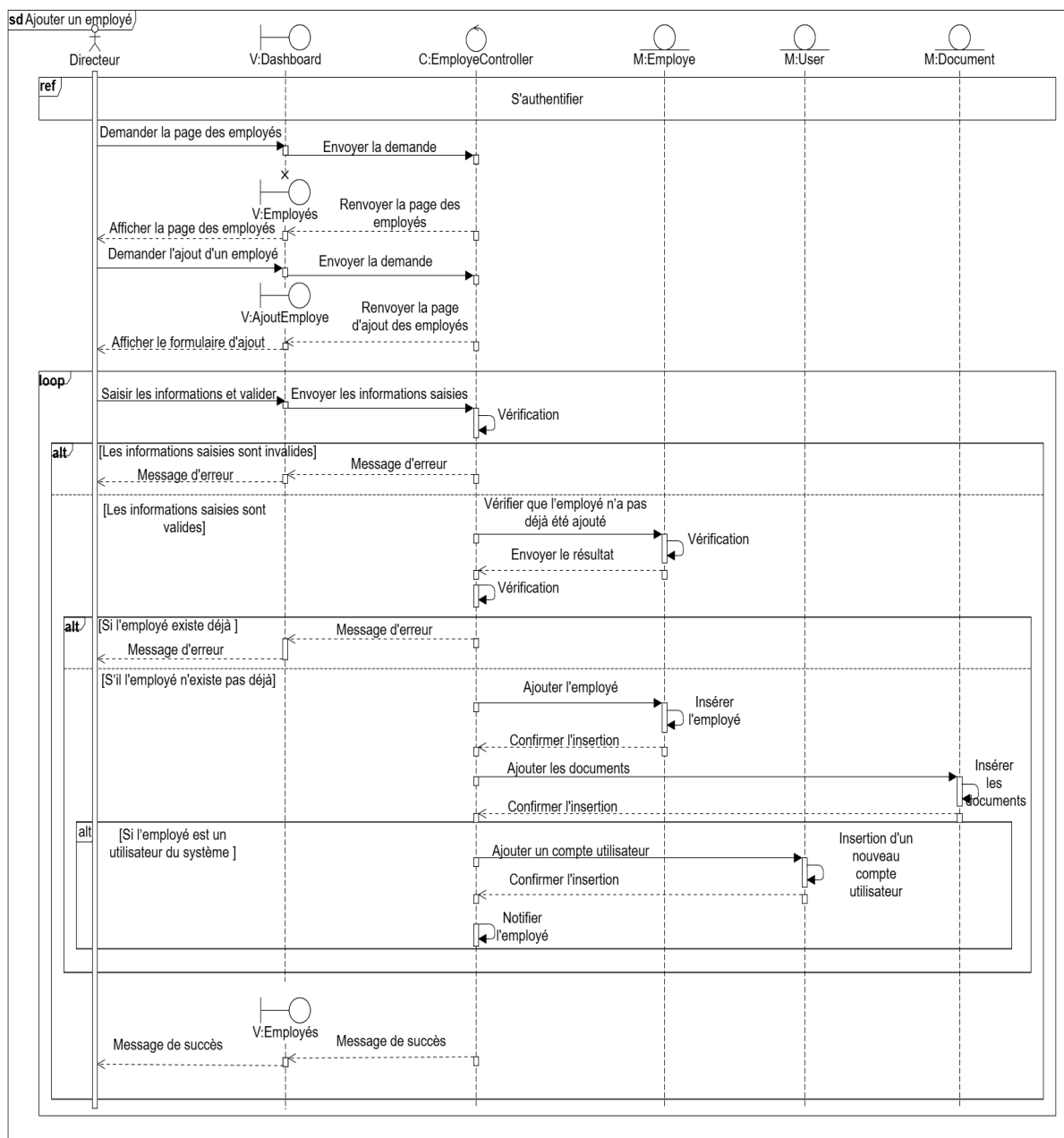


FIGURE 4.7 – Diagramme de séquence détaillé "Ajouter un employé".

4.3.5.5 Diagramme de séquence détaillé "Modifier une catégorie d'abonnement"

Le cas d'utilisation « Gérer une catégorie d'abonnement » inclut la fonctionnalité permettant l'ajout, la modification et la suppression d'une catégorie d'abonnement. L'option "Modifier une catégorie d'abonnement" permet au directeur de modifier les informations d'une catégorie d'abonnement existante.

Le système vérifie les données saisies avant de faire la mise à jour. Dès que l'opération est effectuée, il enregistre les modifications et envoie un message du succès au directeur. En cas d'erreur, un message d'erreur est retourné.

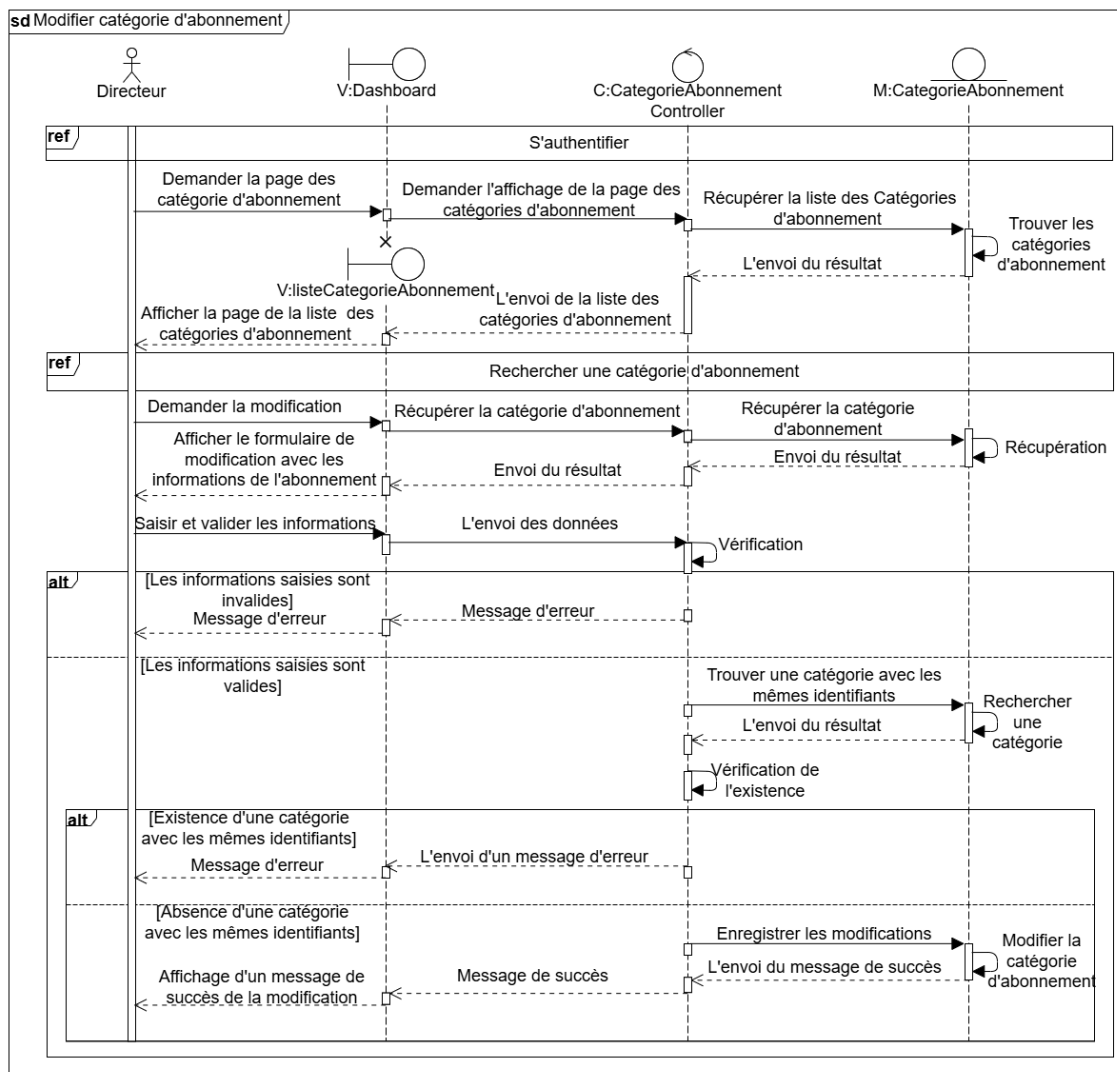


FIGURE 4.8 – Diagramme de séquence détaillé "Modifier une catégorie d'abonnement".

4.3.5.6 Diagramme de séquence détaillé "Supprimer une section"

Le cas d'utilisation « Gérer une section » permet au directeur d'ajouter de nouvelles sections si nécessaire, ainsi que la modification et la suppression.

L'option "Supprimer une section" permet au directeur de supprimer les informations d'une section existante. Une fois la suppression confirmée, le système supprime la section de manière définitive.

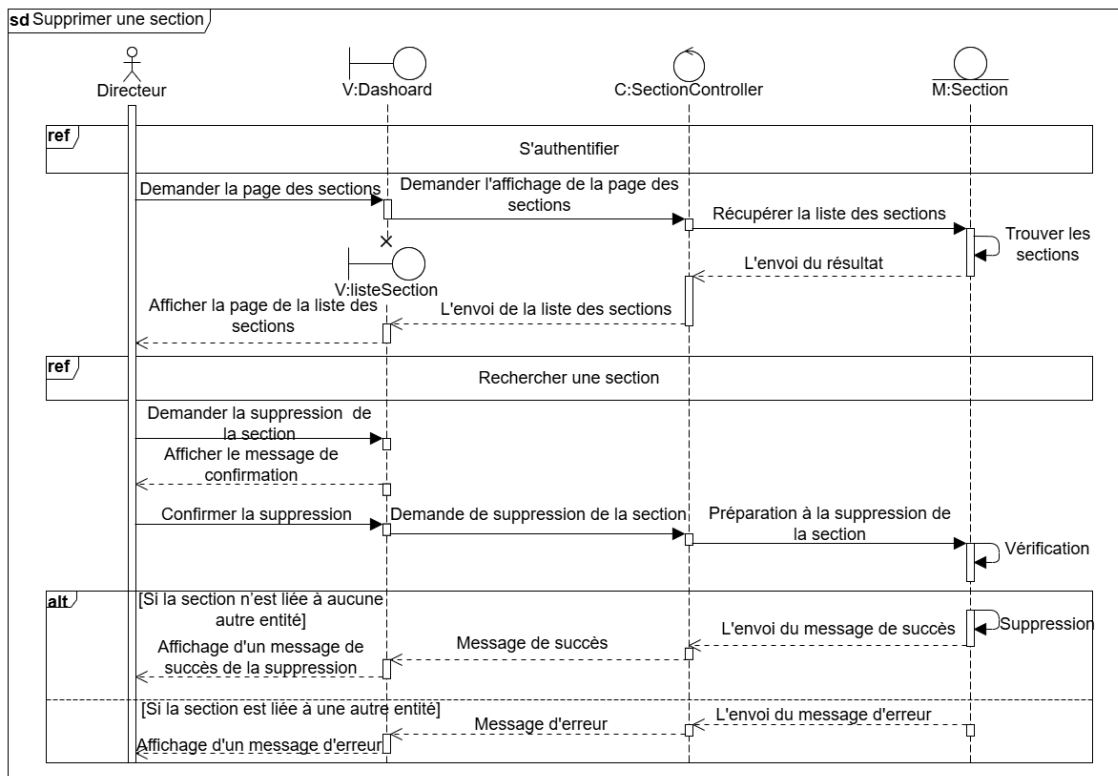


FIGURE 4.9 – Diagramme de séquence détaillé "Supprimer une section".

4.3.5.7 Diagramme de séquence détaillé "Activer/Désactiver un supplément"

Le cas d'utilisation « Activer/Désactiver un supplément » permet de gérer la disponibilité d'un supplément dans le système. Lorsqu'un supplément est désactivé, il devient temporairement indisponible pour les utilisateurs, empêchant toute nouvelle interaction avec celui-ci. À l'inverse, l'activation rend à nouveau le supplément accessible.

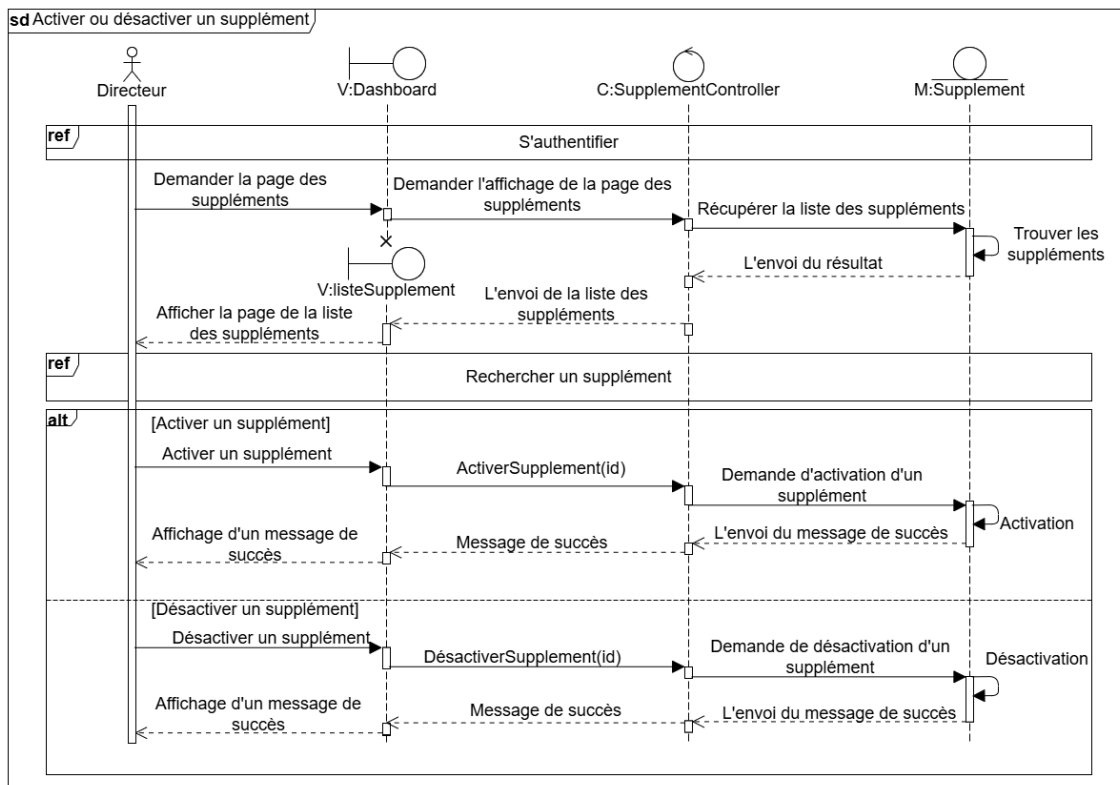


FIGURE 4.10 – Diagramme de séquence détaillé "Activer/Désactiver un supplément".

4.3.5.8 Diagramme de séquence détaillé "Exporter la liste des groupes"

Le cas d'utilisation « Exporter la liste des groupes » permet à l'utilisateur de récupérer les informations liées aux différents groupes enregistrés dans le système, sous forme de fichier téléchargeable. L'utilisateur peut choisir le format de sortie, soit en PDF, soit en Excel, selon ses besoins.

Le diagramme de séquence présenté dans la Figure 4.11 illustre les différentes étapes du processus, depuis la demande d'export jusqu'à la génération du fichier.

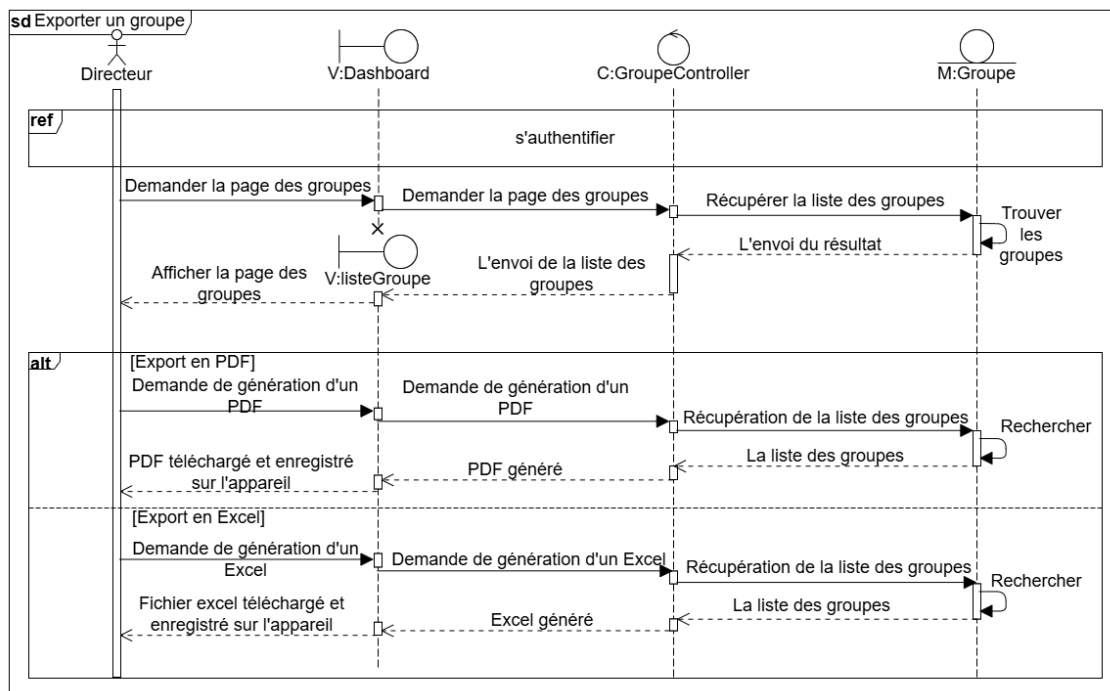


FIGURE 4.11 – Diagramme de séquence détaillé "Exporter la liste d'un groupe".

4.3.5.9 Diagramme d'activité "S'authentifier"

Le diagramme d'activité du cas d'utilisation S'authentifier décrit les différentes étapes suivies par l'utilisateur lors de la connexion au système.

Il commence par la saisie de l'e-mail et du mot de passe, puis se poursuit par la vérification des informations. Selon le résultat, l'utilisateur est soit redirigé vers sa page d'accueil, soit informé d'une erreur et invité à réessayer.

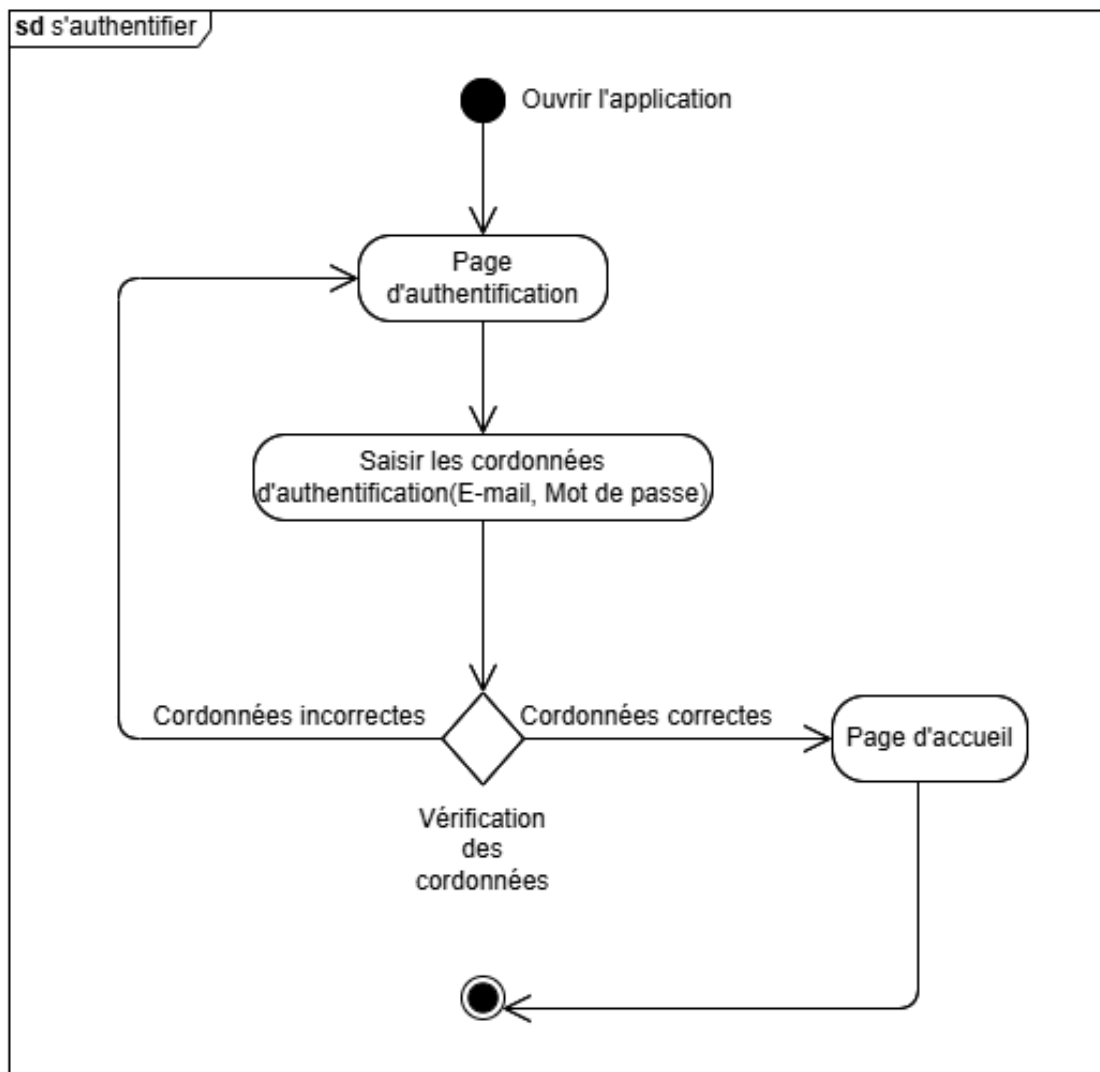


FIGURE 4.12 – Diagramme d'activité "S'authentifier".

4.4 Sprint deux

Le Sprint deux poursuit le travail engagé lors de la première phase du projet. Il nous a permis d'approfondir le développement en ajoutant de nouvelles fonctionnalités et en apportant des améliorations aux éléments déjà réalisés. Cette étape nous a permis de progresser dans le projet, tout en s'appuyant sur ce qui avait été fait auparavant.

4.4.1 Backlog produit du sprint deux

Lors du Sprint deux, nous avons travaillé sur plusieurs fonctionnalités présentées dans la section ci-dessous. Dans lequel on a intégré des nouveaux acteurs tels que l'éducateur et le parent, chacun avec ses propres fonctionnalités.

Ces ajouts permettent d'améliorer le système et de mieux répondre aux besoins réels de la crèche.

Le tableau suivant 4.16 représente un aperçu du **Backlog produit du sprint deux** avec les fonctionnalités identifiées.

ID US	Tâche	Durée
US-09	Élaborer le CU et la description de la fonctionnalité "Traitement des inscriptions" par le directeur. Ajouter les diagrammes de séquence nécessaires.	6h
	Développer et tester la fonctionnalité.	36h
US-10	Élaborer le CU et la description de la fonctionnalité "Gestion des dossiers" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	20h
US-11	Élaborer le CU et la description de la fonctionnalité "Gestion des règlements intérieurs de la crèche" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la gestion des règlements intérieurs de la crèche.	18h
US-12	Élaborer le CU et la description de la fonctionnalité "Gestion des plannings pédagogiques" par le directeur. Ajouter les diagrammes de séquence nécessaires.	2h
	Développer et tester la gestion des plannings pédagogiques.	5h
US-13	Élaborer le CU et la description de la fonctionnalité "Gérer des menus des repas" par le directeur. Ajouter les diagrammes de séquence nécessaires.	2h
	Développer et tester la gestion de menu des repas.	8h
US-14	Élaborer le CU et la description de la fonctionnalité "Suivi des présences des employés" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	18h
US-15	Élaborer le CU et la description de la fonctionnalité "Gestion des matières" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la gestion des matières.	8h

Continuation du tableau 4.16		
ID US	Tâche	Durée
US-16	Élaborer le CU et la description de la fonctionnalité "Gestion des séances" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la gestion des séances.	8h
US-17	Élaborer le CU et la description de la fonctionnalité "Gestion des salles" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la gestion des salles.	6h
US-18	Élaborer le CU et la description de la fonctionnalité "Affecter les enfants aux groupes" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	20h
US-19	Élaborer le CU et la description de la fonctionnalité "Établir la fiche de suivi enfant" par l'éducateur. Ajouter les diagrammes de séquence nécessaires.	5h
	Développer et tester la fonctionnalité.	20h
US-20	Élaborer le CU et la description de la fonctionnalité "Gérer les évaluations pédagogiques par matière" par l'éducateur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	16h
US-21	Élaborer le CU et la description de la fonctionnalité "Gestion de profil éducateur" par l'éducateur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la gestion de profil éducateur.	18h
US-22	Élaborer le CU de la fonctionnalité "Consulter le planning pédagogique" par l'éducateur. Ajouter les diagrammes de séquence nécessaires.	2h
	Développer et tester la consultation du planning pédagogique.	6h
US-23	Élaborer le CU et la description de la fonctionnalité "Consulter la liste des enfants" par l'éducateur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la Consulter la liste des enfants.	6h
US-24	Élaborer le CU de la fonctionnalité "Suivi de la présence des enfants" par l'éducateur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	10h
US-25	Élaborer le CU de la fonctionnalité "Envoyer une demande d'inscription" par le parent. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	19h
US-26	Élaborer le CU de la fonctionnalité "Suivre l'état d'avancement de la demande d'inscription" par le parent. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	19h

Continuation du tableau 4.16		
ID US	Tâche	Durée
US-27	Élaborer le CU de la fonctionnalité "Compléter le dossier d'inscription" par le parent. Ajouter les diagrammes de séquence nécessaires.	5h
	Développer et tester la fonctionnalité.	19h
US-28	Élaborer le CU de la fonctionnalité "Abandonner l'inscription" par le parent. Ajouter les diagrammes de séquence nécessaires	2h
	Développer et tester la fonctionnalité.	5h
US-29	Élaborer le CU de la fonctionnalité "Consulter le planning des cours" par le parent. Ajouter les diagrammes de séquence nécessaires	2h
	Développer et tester la fonctionnalité.	5h
US-30	Élaborer le CU de la fonctionnalité "Consulter le menu des repas" par le parent. Ajouter les diagrammes de séquence nécessaires.	2h
	Développer et tester la fonctionnalité.	5h
US-31	Élaborer le CU de la fonctionnalité "Gérer le profil du parent" par le parent. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	19h
US-32	Élaborer le CU de la fonctionnalité "Gérer les repas" par le directeur. Ajouter les diagrammes de séquence nécessaires.	2h
	Développer et tester la fonctionnalité.	7h

TABLE 4.16 – Backlog du produit du sprint deux

4.4.2 Spécification des besoins

Comme dans le premier sprint, cette phase débute par la spécification des besoins. Cette étape permet de structurer le travail à venir, notamment en vue de la conception, du développement et des tests.

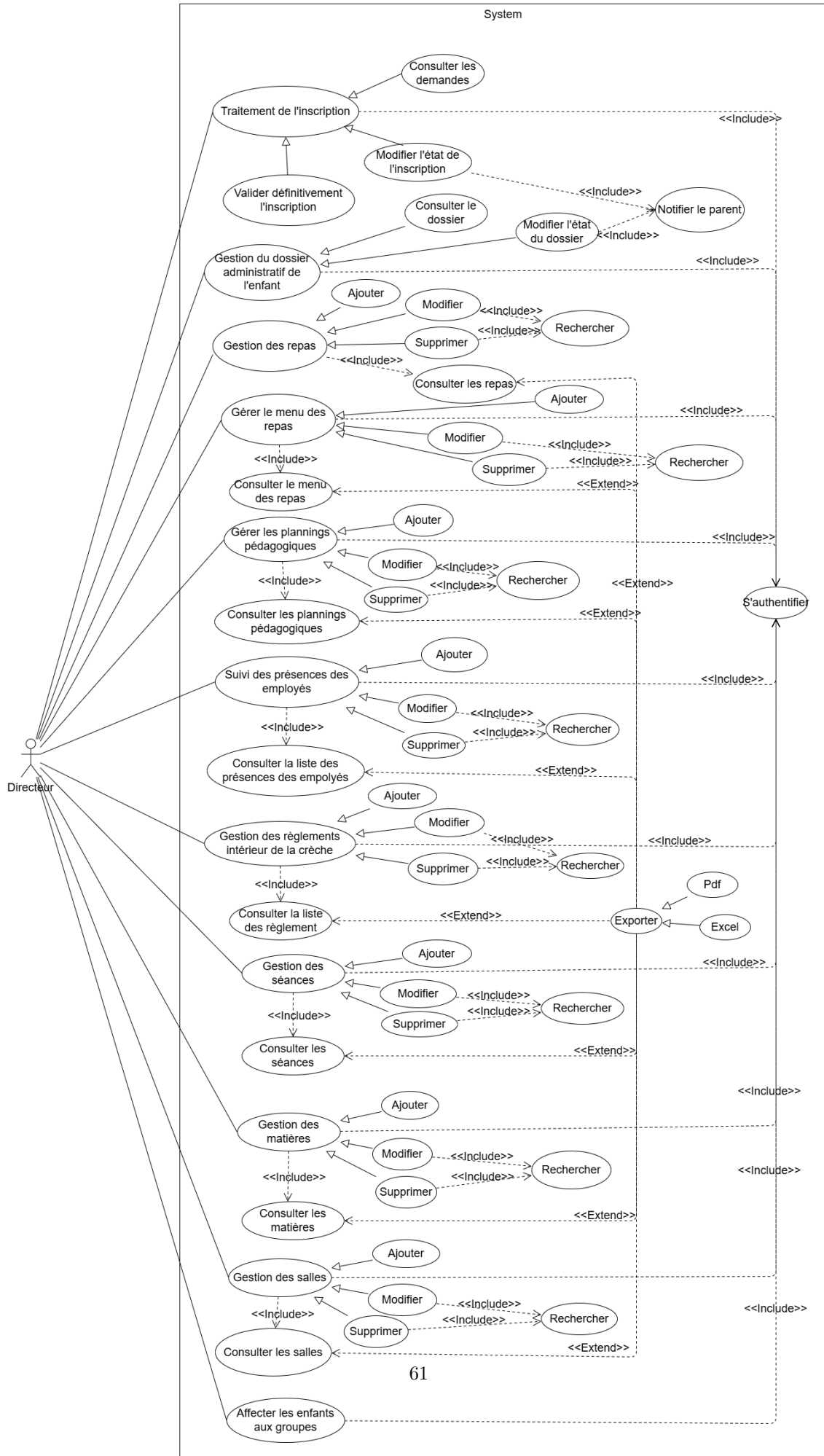
4.4.3 Diagramme de cas d'utilisation du sprint deux

Les diagrammes suivants englobent la modélisation des cas d'utilisation déduits à partir du deuxième Sprint :

4.4.3.1 Diagramme de cas d'utilisation "directeur"

Le directeur joue un rôle important dans le fonctionnement de la crèche. Il s'occupe de plusieurs aspects, comme la gestion des inscriptions, le suivi des employés, l'organisation du planning ou encore les repas.

Le diagramme ci-dessous Figure 4.13 montre l'ensemble des actions qu'il peut réaliser dans le système. On y retrouve par exemple la validation des dossiers, la consultation des demandes, la gestion des matières, des salles, ou encore l'affectation des enfants aux groupes. Ce diagramme donne une bonne vision des fonctionnalités qui lui sont réservées.



4.4.3.2 Diagramme de cas d'utilisation "Éducateur"

L'éducateur intervient directement auprès des enfants et joue un rôle essentiel dans leur accompagnement quotidien. Il s'occupe de l'animation des activités, veille au bon déroulement de la journée et assure un suivi régulier de chaque enfant.

Pour cela, il utilise le système pour consulter le planning pédagogique de son groupe, enregistrer les présences, évaluer les enfants selon des critères précis, et échanger des informations avec le directeur ou les parents. Cela lui permet de gagner du temps, d'être mieux organisé et de garder une trace des différentes actions menées.

Le diagramme présenté dans la Figure 4.14 illustre l'ensemble des fonctionnalités qui lui sont accessibles.

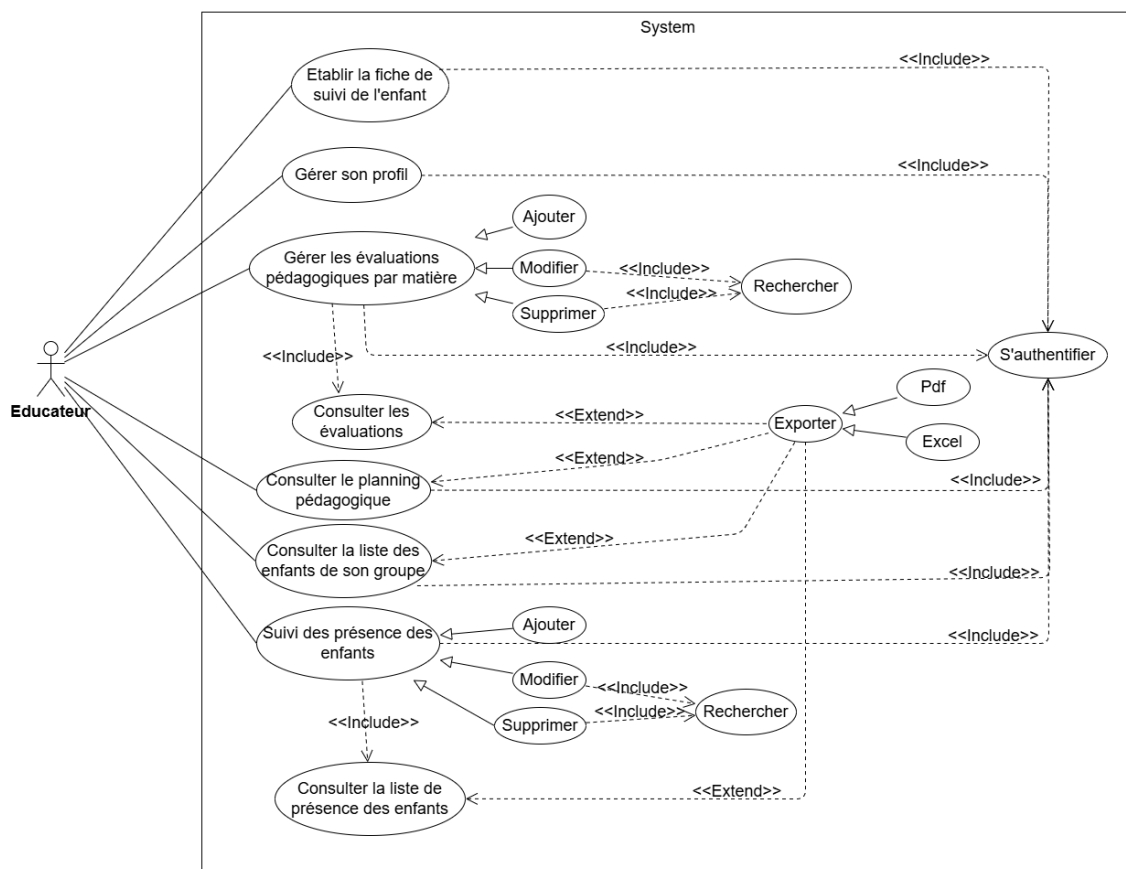


FIGURE 4.14 – Diagramme de cas d'utilisation "Éducateur".

4.4.3.3 Diagramme de cas d'utilisation "Parent"

Le parent utilise la plateforme pour rester informé du déroulement de la journée de son enfant à la crèche. À travers son espace personnel, il peut suivre les activités, consulter le planning, vérifier les menus proposés, ou encore lire les actualités publiées par l'équipe. Il a également la possibilité de soumettre une demande d'inscription et d'entrer en contact avec les éducateurs ou le directeur en cas de besoin.

L'objectif est de lui offrir un espace simple et pratique pour suivre la scolarité de son enfant et échanger avec les responsables.

Le diagramme ci-dessous Figure 4.15 montre l'ensemble des fonctionnalités qui lui sont accessibles dans le système.

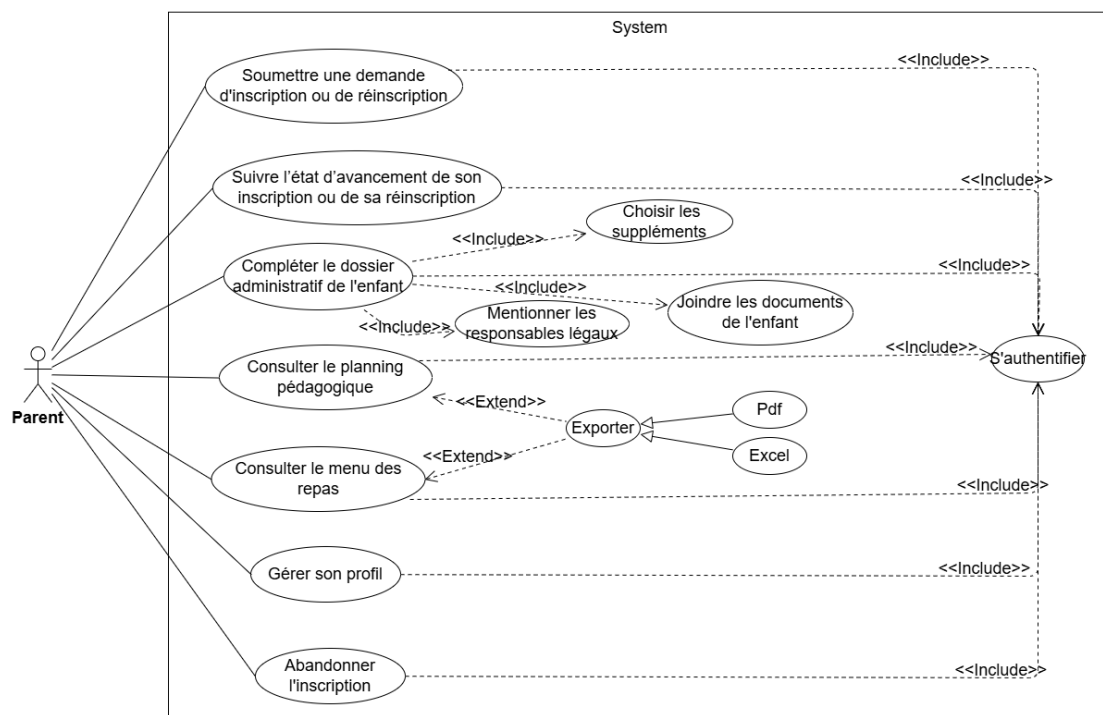


FIGURE 4.15 – Diagramme de cas d'utilisation "Parent".

4.4.4 Description textuelle

Dans cette section, nous présentons une description écrite des cas d'utilisation les plus importants réalisés lors du sprint deux. Ces descriptions permettent de suivre, étape par étape, le déroulement de certaines fonctionnalités du système du point de vue de l'utilisateur.

4.4.4.1 Cas d'utilisation "Gérer les matières"

TABLE 4.17 – Description textuelle du CU : Gérer les matières.

CU : Gérer les matières
ID : 1
Description brève : Ce cas d'utilisation permet au directeur d'ajouter, modifier ou supprimer une matière dans la crèche en saisissant son nom.
Acteurs primaires : Directeur
Acteurs secondaires : Aucun
Pré-condition : Le directeur est authentifié.
<p>Enchaînement principal :</p> <p>A. Ajouter une matière</p> <ol style="list-style-type: none"> 1. Le directeur accède à la page d'ajout des matières. 2. Il saisit le nom de la matière. 3. Le système vérifie que le nom n'existe pas déjà. 4. Si le nom est unique, la matière est enregistrée. 5. Le système affiche un message de succès. <p>B. Modifier une matière</p> <ol style="list-style-type: none"> 1. Le directeur consulte la liste des matières. 2. Il sélectionne une matière à modifier. 3. Il modifie le nom. 4. Le système vérifie le nouveau nom. 5. Le système enregistre les modifications et affiche un message de succès. <p>C. Supprimer une matière</p> <ol style="list-style-type: none"> 1. Le directeur consulte la liste des matières. 2. Il sélectionne une matière à supprimer. 3. Le système affiche un message de confirmation. 4. Le directeur confirme la suppression. 5. Le système supprime la matière et affiche un message de succès.
Post-condition : La matière est ajoutée, modifiée ou supprimée dans la base de données selon l'action réalisée.
<p>Enchaînements alternatifs :</p> <ol style="list-style-type: none"> 1. Le nom de la matière existe déjà : Le système affiche un message d'erreur et bloque l'enregistrement. 2. La matière sélectionnée n'existe plus : Le système affiche un message d'erreur. 3. Erreur technique lors de l'enregistrement : Le système affiche un message d'erreur et aucune modification n'est enregistrée.

4.4.4.2 Cas d'utilisation "Gérer les séances"

TABLE 4.18 – Description textuelle du CU : Gérer les séances.

CU : Gérer les séances
ID : 2
Description brève : Permettre au directeur d'ajouter, modifier ou supprimer des séances associées à une matière.
Acteurs primaires : Directeur
Acteurs secondaires : Aucun
Pré-condition : 1. Le directeur est authentifié. 2. Des matières sont déjà enregistrées.
Enchaînement principal : A. Le directeur crée une séance. 1. Le directeur accède à la page du planning voulu afin de pouvoir gérer les séances 2. Le système affiche la liste des séances existantes. 3. Le directeur clique sur « Ajouter une séance ». 4. Le système affiche un formulaire avec les champs : matière, heure début, heure fin, jour. 5. Le directeur remplit les champs et valide le formulaire. 6. Le système enregistre la séance dans la base de données. 7. Le système affiche un message de confirmation et met à jour la liste des séances. B. Le directeur modifie une séance. 1. Le directeur sélectionne une séance. 2. Le système affiche les détails de la séance. 3. Le directeur modifie les champs souhaités. 4. Le directeur valide. 5. Le système enregistre les modifications. 6. Le système affiche un message de succès. C. Le directeur supprime une séance. 1. Le directeur sélectionne une séance à supprimer. 2. Le système demande une confirmation. 3. Le directeur confirme. 4. Le système supprime la séance. 5. Le système affiche un message de succès.
Post-condition : La séance est ajoutée, modifiée ou supprimée selon l'action effectuée.
Enchaînements alternatifs : 1. Aucune matière enregistrée : Le système affiche le message : « Aucune matière disponible. Veuillez d'abord créer une matière ». 2. Validation du formulaire échouée : Le système affiche les erreurs et bloque l'enregistrement. 3. La séance à modifier n'existe pas : Le système affiche le message : « Séance introuvable ». 4. Échec de la suppression : Le système affiche un message d'erreur et la séance n'est pas supprimée. 5. Chevauchement détecté avec une autre séance le même jour : Le système bloque l'enregistrement et affiche un message d'erreur indiquant un conflit d'horaires avec une séance existante.

4.4.4.3 Cas d'utilisation "Gérer les plannings pédagogiques"

TABLE 4.19 – Description textuelle du CU : Gérer les plannings de cours.

CU : Gérer les plannings de cours
ID : 3
Description brève : Permet au directeur d'ajouter, modifier ou supprimer un planning de cours associé à une section d'une saison.
Acteurs primaires : Directeur
Acteurs secondaires : Aucun
Pré-conditions : Le directeur est authentifié. La saison actuelle est créée. Au moins une section active est liée à la saison actuelle. Des matières sont enregistrées dans le système.
<p>Enchaînement principal :</p> <p>A. Liste des plannings :</p> <ol style="list-style-type: none"> 1. Le directeur accède à la page de gestion des plannings. 2. Le système vérifie qu'une saison actuelle existe. Le système vérifie que des sections actives existent pour la saison. 3. Le système récupère la liste des plannings existants et l'affiche. <p>A. Ajout d'un nouveau planning :</p> <ol style="list-style-type: none"> 1. Le directeur accède à la page de gestion des plannings. 2. Le directeur sélectionne une section active. 3. Le système vérifie qu'aucun planning n'existe pour cette section dans la saison actuelle. 4. Le système enregistre le planning. 5. Le système renvoie un message de succès. <p>B. Modification d'un planning :</p> <ol style="list-style-type: none"> 1. Le directeur choisit un planning existant à modifier. 2. Le système vérifie l'existence du planning. 3. Le directeur modifie les informations souhaitées (par exemple, les matières ou les jours). 4. Le système met à jour les données du planning. 5. Le système affiche un message de confirmation. <p>C. Suppression d'un planning :</p> <ol style="list-style-type: none"> 1. Le directeur sélectionne un planning à supprimer. 2. Le système vérifie l'existence du planning. 3. Le système supprime le planning de la base de données. 4. Le système affiche un message de succès.
Post-condition : Le planning est ajouté, modifié ou supprimé selon l'action effectuée.
<p>Enchaînements alternatifs :</p> <ol style="list-style-type: none"> 1. Aucune saison actuelle : Le système redirige vers la liste des saisons avec un message d'erreur. 2. Aucune section active : Le système redirige vers la liste des sections avec un message d'erreur. 3. Aucune matière disponible : Le système retourne une erreur. 4. Section invalide ou inexistante : Un message d'erreur est retourné. 5. Un planning existe déjà pour la section : Un message d'erreur est retourné. 6. Le planning n'existe pas : Redirection avec message d'erreur. 7. Échec de la modification ou suppression : Un message d'erreur est affiché.

4.4.4.4 Cas d'utilisation "Gérer les évaluations"

TABLE 4.20 – Description textuelle du CU : Gérer les évaluations.

CU : Gérer les évaluations
ID : 4
Description brève : Ce cas d'utilisation permet à l'éducateur d'ajouter, consulter, modifier ou supprimer une évaluation attribuée à un enfant pour une matière donnée, dans la saison actuelle.
Acteurs primaires : Educateur
Acteurs secondaires : Aucun
Pré-condition : <ol style="list-style-type: none"> 1. L'éducateur est authentifié et une saison actuelle est définie. 2. Les matières doivent être disponibles.
Enchaînement principal : A. Ajouter une évaluation <ol style="list-style-type: none"> 1. L'éducateur accède à la page des évaluations. 2. Le système vérifie qu'une saison actuelle est bien définie. 3. Le système affiche la liste des matières et des enfants liés à cette saison. 4. L'éducateur sélectionne une matière et un enfant, puis saisit une note et une remarque facultative. 5. Le système vérifie que cette évaluation n'existe pas déjà. 6. Le système enregistre l'évaluation et affiche un message de succès. B. Modifier une évaluation <ol style="list-style-type: none"> 1. L'éducateur consulte la liste des évaluations. 2. Il sélectionne une évaluation à modifier. 3. Il modifie la note et/ou la remarque. 4. Le système enregistre les modifications et affiche un message de succès. C. Supprimer une évaluation <ol style="list-style-type: none"> 1. L'éducateur consulte la liste des évaluations. 2. Il sélectionne une évaluation à supprimer. 3. Le système affiche un message de confirmation. 4. L'éducateur confirme la suppression. 5. Le système supprime l'évaluation et affiche un message de succès.
Post-condition : L'évaluation est ajoutée, modifiée ou supprimée dans la base de données.
Enchaînements alternatifs : <ol style="list-style-type: none"> 1. Aucune saison actuelle n'est définie : Le système affiche un message d'erreur et bloque l'accès à la fonctionnalité. 2. L'évaluation existe déjà pour cet enfant et cette matière : Le système affiche un message d'erreur et empêche l'ajout. 3. L'évaluation sélectionnée n'existe plus : Le système affiche un message d'erreur. 4. Erreur technique lors de l'enregistrement ou de la suppression : Le système affiche un message d'erreur et aucune opération n'est effectuée.

4.4.4.5 Cas d'utilisation "Demander une inscription ou une réinscription"

TABLE 4.21 – Description textuelle du CU : Gestion des inscriptions.

CU : Demander une inscription ou une réinscription
ID : 5
Description brève : Permet au parent d'inscrire ou réinscrire un enfant pour la saison en cours.
Acteurs primaires : Parent
Acteurs secondaires : Aucun
Pré-condition : <ol style="list-style-type: none"> 1. Le parent est authentifié. 2. Une saison active existe. 3. Au moins une section active est disponible.
Enchaînement principal : <ol style="list-style-type: none"> 1. Le parent accède à la page d'inscription. 2. Le système vérifie l'existence d'une saison active et de sections disponibles. 3. Le parent remplit le formulaire avec : <ul style="list-style-type: none"> - Informations personnelles de l'enfant - Photo d'identité - Jours de fréquentation - Services demandés (matin/après-midi avec ou sans repas) 4. Le parent soumet le formulaire. 5. Le système valide les données et vérifie : <ol style="list-style-type: none"> a. L'âge de l'enfant correspond à une section active b. L'enfant n'est pas déjà inscrit pour cette saison c. Les horaires sont compatibles avec les règles de gestion 6. Le système enregistre l'inscription et génère un matricule. 7. Le système affiche un message de succès.
Post-condition : L'enfant est inscrit avec les services demandés pour la saison en cours.
Enchaînements alternatifs : <ol style="list-style-type: none"> 2.a. Aucune saison active : Le système affiche "Aucune année scolaire active". 2.b. Aucune section active : Le système affiche "Aucune section disponible". 5.a. Âge inapproprié : Le système refuse avec message "L'âge ne correspond à aucune section". 5.b. Enfant déjà inscrit : Le système refuse avec message "Inscription existante pour cette saison". 5.c. Horaires invalides : Le système explique les créneaux autorisés. 5.d. Données manquantes : Le système liste les champs incomplets.
Exceptions : <ol style="list-style-type: none"> 1. Échec d'enregistrement : Message d'erreur technique. 2. Problème de téléchargement photo : L'utilisateur est invité à réessayer.
Remarques : La réinscription suit le même processus avec vérification préalable de l'existence de l'enfant.

4.4.4.6 Cas d'utilisation "Compléter le dossier d'inscription"

TABLE 4.22 – Description textuelle du CU : Compléter le dossier d'inscription.

CU : Compléter le dossier d'inscription
ID : 6
Description brève : Permettre au parent de compléter le dossier administratif de son enfant après l'acceptation de la demande d'inscription.
Acteurs primaires : Parent
Acteurs secondaires : Aucun
Pré-condition : <ol style="list-style-type: none"> 1. Le parent est authentifié. 2. La demande d'inscription a été acceptée. 3. Une saison académique active est disponible.
Enchaînement principal : <ol style="list-style-type: none"> 1. Le parent accède à la page de complétion du dossier. 2. Le système vérifie l'existence d'une saison active. 3. Le système affiche les champs du formulaire à remplir : <ul style="list-style-type: none"> – Cartes d'identité des deux représentants légaux. – Liste de 3 à 5 tuteurs (nom, prénom, genre, photo, carte d'identité, lien avec l'enfant, téléphone). – Documents obligatoires (acte de naissance, carnet de santé, certificats médicaux). – Suppléments à choisir (optionnels). – Signatures électroniques du premier représentant. 4. Le parent complète le formulaire et soumet. 5. Le système valide les données saisies. 6. Le système sauvegarde les fichiers transmis dans son espace de stockage. 7. Le système génère les fichiers PDF (fiche d'inscription, autorisation parentale, engagement, règlement signé). 8. Le dossier est marqué comme complété. 9. Un message de confirmation est affiché.
Post-condition : Le dossier est complété et prêt à être examiné par l'administration. Les documents sont stockés dans le système.
Enchaînements alternatifs : <ol style="list-style-type: none"> 1. Aucune saison active : Le système redirige le parent avec un message : « Aucune saison active disponible ». 2. Erreur de validation : Le système affiche les erreurs détectées (champs manquants, formats invalides, nombre insuffisant de tuteurs, etc). 3. Erreur lors de l'enregistrement des fichiers : Le système affiche un message d'échec de l'opération et invite à réessayer.

4.4.4.7 Cas d'utilisation "Traitement des inscriptions"

TABLE 4.23 – Description textuelle du CU : Traitement des inscriptions.

CU : Traitement des inscriptions
ID : 7
Description brève : Permet au directeur de consulter les demandes d'inscription et de modifier leur état (acceptée, refusée, en attente, etc.) en fonction de la disponibilité et des règles de gestion.
Acteurs primaires : Directeur
Acteurs secondaires : Parent (notifié uniquement)
Pré-condition : <ol style="list-style-type: none"> 1. Le directeur est authentifié dans le système. 2. Une saison active est définie.
Enchaînement principal : A. Consulter les demandes d'inscription : <ol style="list-style-type: none"> 1. Le directeur s'authentifie et accède à la page des inscriptions. 2. Le système récupère et affiche la liste des inscriptions de l'année en cours. B. Modifier l'état d'une inscription : <ol style="list-style-type: none"> 3. Le directeur sélectionne une demande d'inscription à traiter. 4. Le système affiche un formulaire de modification de l'état. 5. Le directeur modifie l'état (ex. accepté, refusé, en attente) et soumet les données. 6. Le système valide les données saisies. 7. Si l'état sélectionné est « accepté », le système vérifie la capacité maximale. 8. Si la capacité le permet, l'état est enregistré comme « accepté » et le parent est notifié. 9. Le système affiche un message de succès.
Post-condition : L'état de l'inscription est mis à jour dans le système et le parent est notifié du changement.
Enchaînements alternatifs : <ol style="list-style-type: none"> 1. Les informations saisies sont invalides : Le système affiche un message d'erreur et invite à corriger les champs erronés. 2. Capacité maximale atteinte : Le système rejette la demande, marque l'inscription comme non validée, et notifie le parent. 3. Autres états que "accepté" : Le système applique directement le changement d'état et notifie le parent.

4.4.4.8 Cas d'utilisation "Valider définitivement l'inscription"

TABLE 4.24 – Description textuelle du CU : Valider définitivement l'inscription.

Continuation du tableau 4.24
CU : Valider définitivement l'inscription
ID : 8
Description brève : Permet au directeur de valider définitivement une inscription d'un enfant, en s'assurant que les données sont valides, que l'état est bien « accepté » et que le dossier est complet. Une fois validée, une prestation de type « frais d'admission » est générée et le parent est notifié.
Acteur primaire : Directeur

TABLE 4.24 – Description textuelle du CU : Valider définitivement l'inscription.

Continuation du tableau 4.24
Acteurs secondaires : Parent (notifié uniquement)
Pré-conditions : 1. Le directeur est authentifié dans le système. 2. Une inscription existe avec un état modifiable.
Enchaînement principal : 1. Le directeur accède à la page des inscriptions. 2. Le système affiche la liste des inscriptions de l'année en cours. 3. Le directeur sélectionne une inscription à valider. 4. Le système affiche un formulaire de modification. 5. Le directeur choisit l'état « accepté » comme validation définitive et soumet le formulaire. 6. Le système vérifie les données saisies : a. La validité des données saisies. b. Si les données sont valides, il vérifie que l'état sélectionné est bien « accepté ». c. Si c'est le cas, il vérifie ensuite la complétude du dossier. 7. Si toutes les conditions sont remplies, le système : a. Rend l'inscription définitive. b. Crée une prestation de type « frais d'admission ». c. Notifie le parent de l'acceptation définitive.. 8. Le système affiche un message de succès.
Post-condition : L'inscription est définitivement validée, la prestation est enregistrée, et le parent est notifié.
Enchaînements alternatifs : 1. Les informations saisies sont invalides : Le système affiche un message d'erreur. 2. L'état sélectionné n'est pas « accepté » : Le système empêche la validation définitive et affiche un message d'erreur. 3. Le dossier de l'enfant est incomplet : Le système affiche un message d'erreur indiquant que la validation définitive est impossible tant que le dossier n'est pas complet.

4.4.5 Diagrammes de séquences détaillés

Cette phase consiste à élaborer des diagrammes de séquences détaillés relatifs au sprint deux :

4.4.5.1 Diagramme de séquence détaillé "Demander une inscription ou une réinscription"

Le cas d'utilisation « Demander une inscription ou une réinscription » permet à un parent de faire une demande pour inscrire un nouvel enfant ou réinscrire un enfant déjà inscrit à la crèche.

Le parent remplit un formulaire avec les informations nécessaires, puis l'envoie. Le système vérifie les données et enregistre la demande. Ensuite, le directeur peut consulter et décider de l'accepter ou non.

Le diagramme présenté dans la Figure 4.16 montre les différentes étapes de cette interaction.

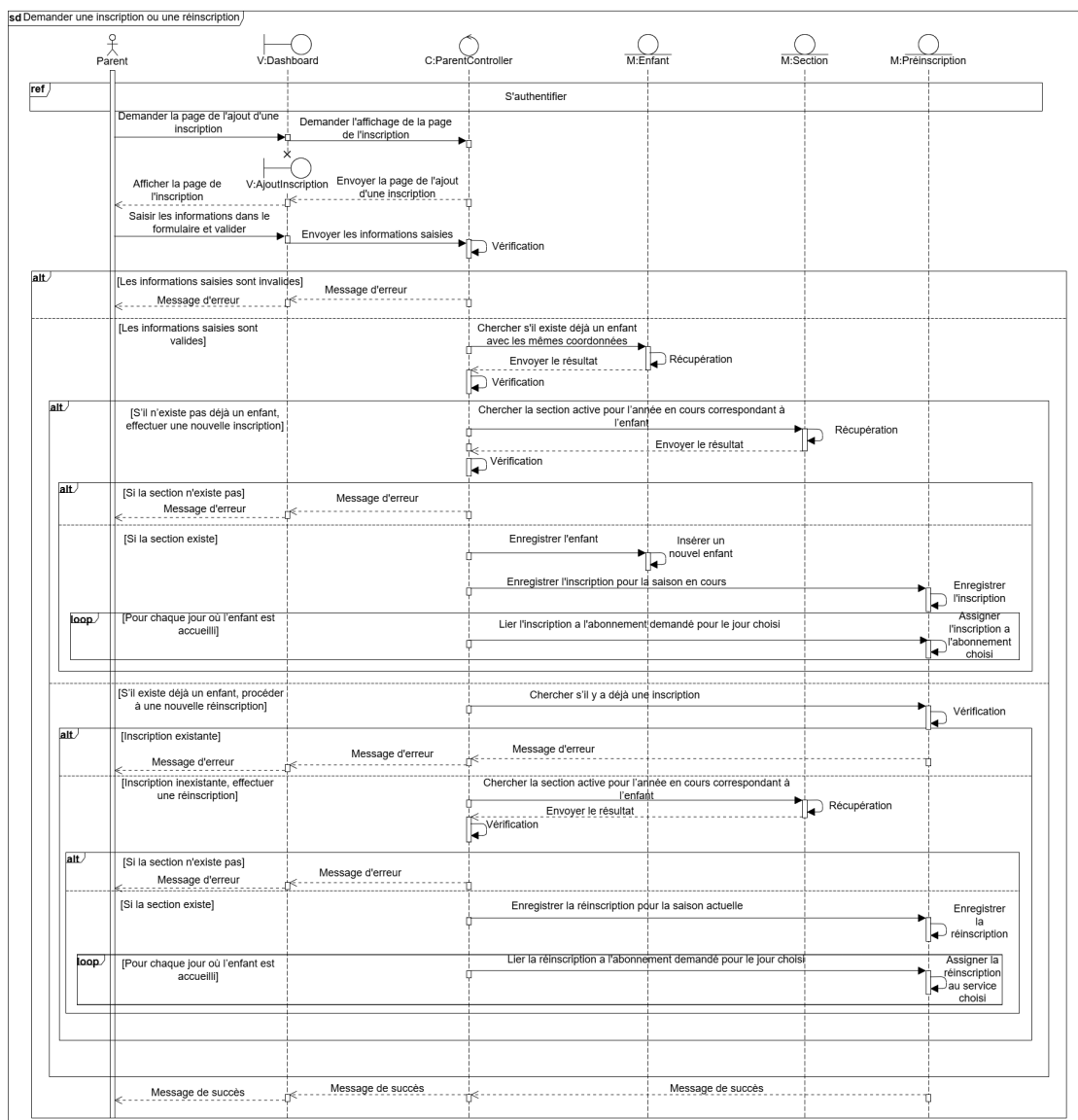


FIGURE 4.16 – Diagramme de séquence détaillé "Demander une inscription ou une réinscription".

4.4.5.2 Diagramme de séquence détaillé "Ajouter un planning pédagogique"

Le cas d'utilisation « Gérer les plannings de cours » permet au directeur de créer, modifier ou supprimer les plannings liés aux différentes sections de la crèche.

Pour ajouter un planning, le directeur sélectionne simplement la section concernée, puis lance l'enregistrement du nouveau planning. Le système se charge alors de sauvegarder l'information et affiche un message de confirmation pour indiquer que l'opération s'est bien déroulée. Ce processus, bien que simple, permet de garder une bonne organisation au sein de la structure.

Le diagramme de séquence ci-dessous 4.17 présente les différentes étapes de cette action, depuis le choix de la section jusqu'à la validation finale

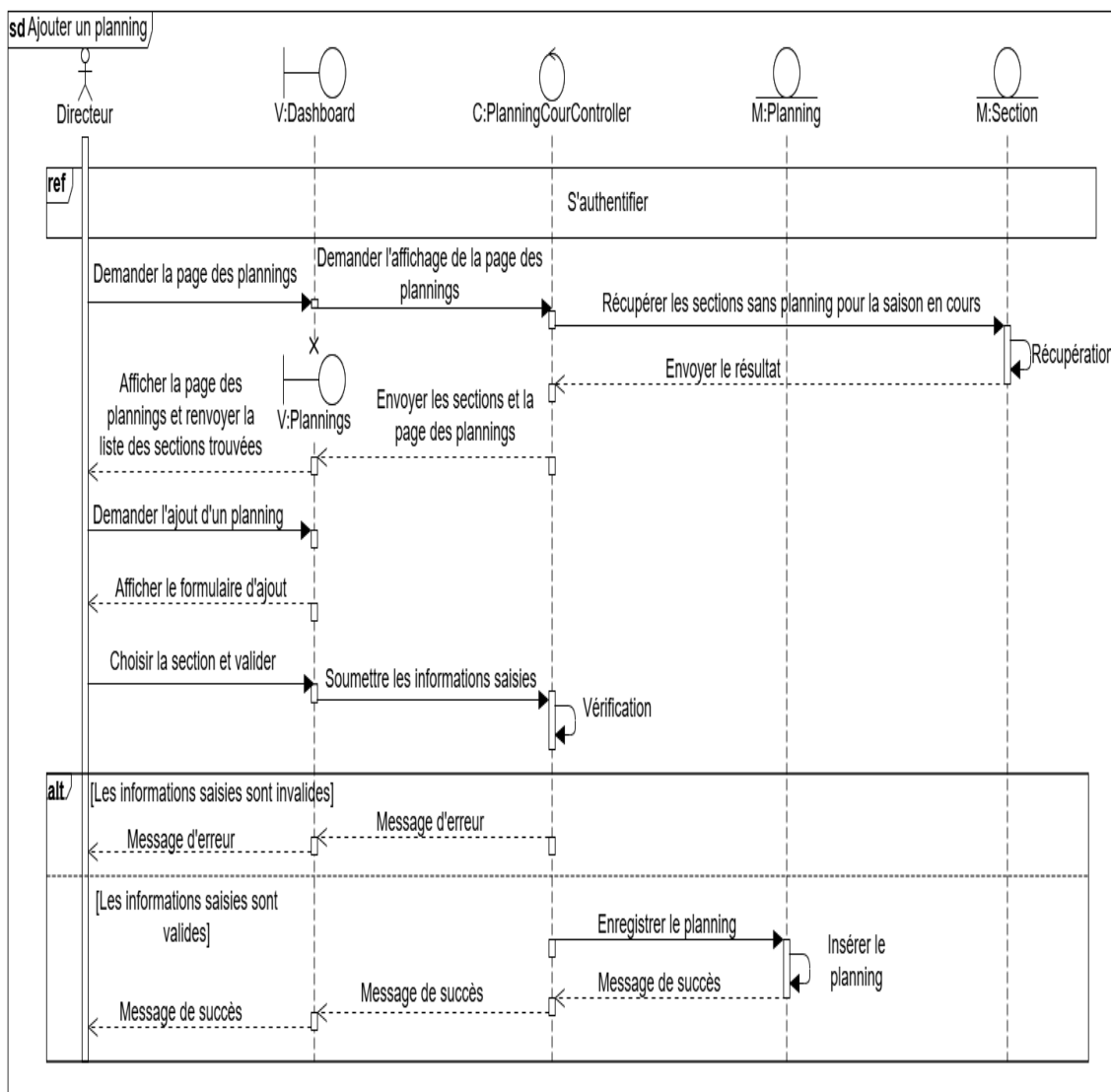


FIGURE 4.17 – Diagramme de séquence détaillé " Ajouter un planning pédagogique".

4.4.5.3 Diagramme de séquence détaillé "Modifier une séance"

Le cas d'utilisation « Gérer les séances » permet au directeur d'effectuer différentes actions sur les séances liées à une matière, telles que l'ajout, la modification ou la suppression.

Lorsqu'il souhaite modifier une séance, le directeur accède d'abord à la page des plannings via le tableau de bord. Il sélectionne le planning concerné, puis choisit la séance à modifier. Le système affiche alors un formulaire prérempli avec les informations existantes de la séance.

Après avoir saisi les modifications souhaitées (comme l'horaire, la matière, etc.), le directeur valide le formulaire. Le système effectue d'abord une vérification de la validité des données saisies. Si celles-ci sont incorrectes ou incomplètes, un message d'erreur est affiché.

Dans le cas contraire, le système vérifie s'il y a un chevauchement avec une autre séance identique dans le même planning et le même jour.

- S'il y a un chevauchement, un message d'erreur est affiché, et la modification est rejetée.
- S'il n'y a pas de conflit, la modification est enregistrée, et un message de confirmation s'affiche pour indiquer le succès de l'opération.

Le diagramme présenté dans la Figure 4.18 illustre les différentes étapes de ce processus, y compris les cas alternatifs liés à la validation des données et aux chevauchements éventuels.

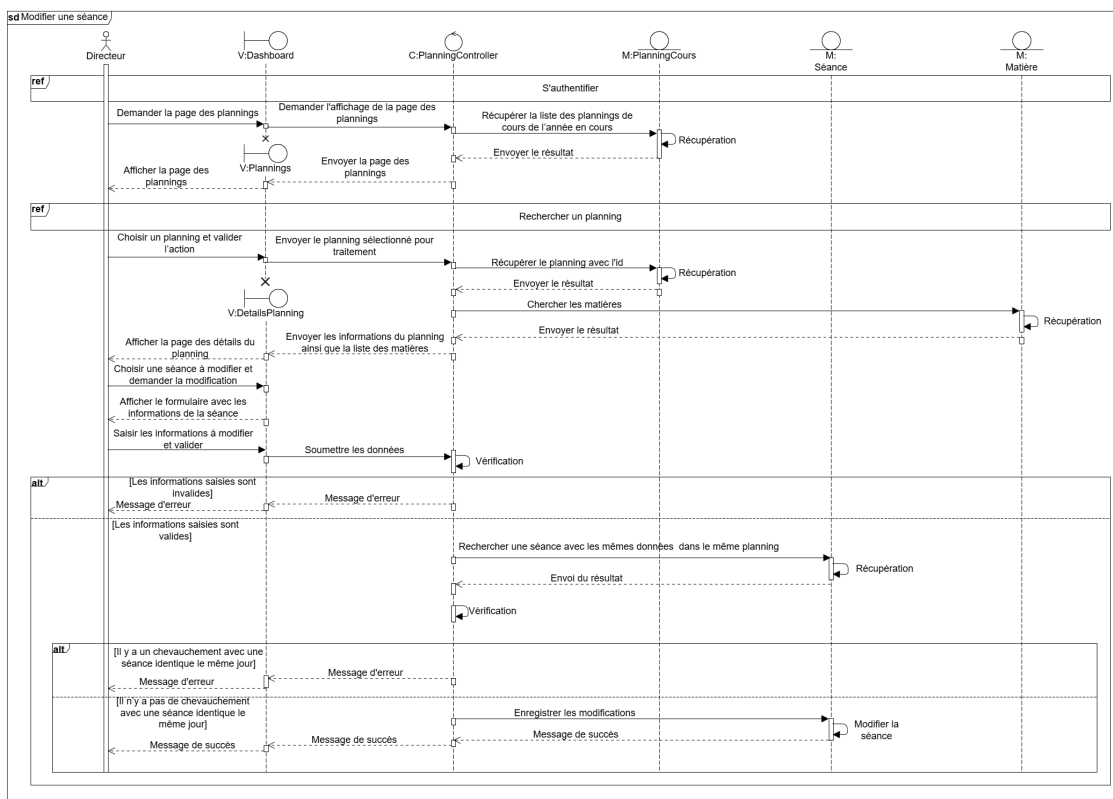


FIGURE 4.18 – Diagramme de séquence détaillé "Modifier une séance".

4.4.5.4 Diagramme de séquence détaillé "Modifier l'état d'une inscription"

Ce cas d'utilisation permet au directeur de changer l'état d'une demande d'inscription. Il sélectionne une demande, il modifie son état (acceptée, refusée ou en attente).

Si l'état choisi est « acceptée », le système vérifie s'il reste des places disponibles. Si tout est correct, le système enregistre le changement et informe le parent.

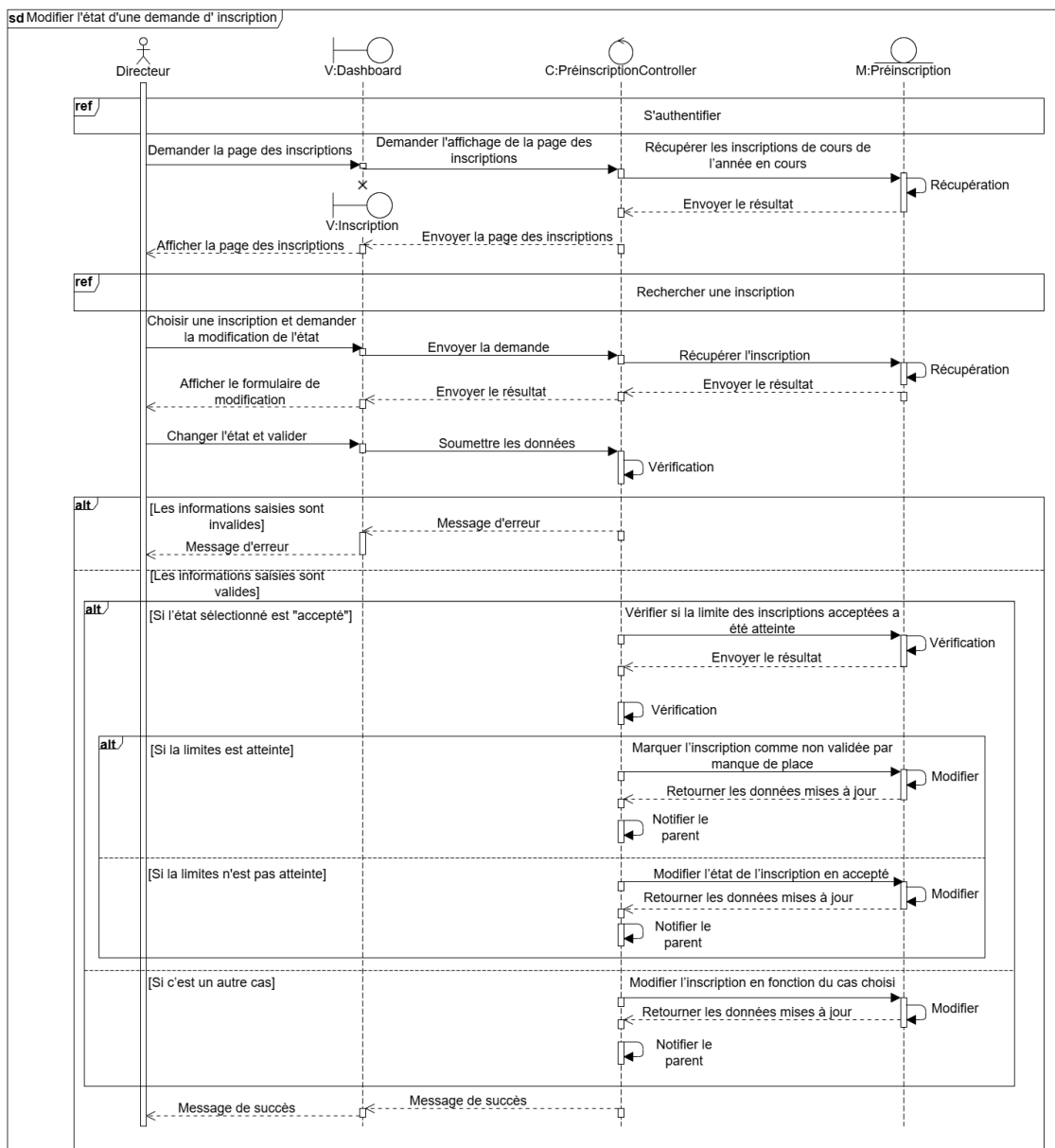


FIGURE 4.19 – Diagramme de séquence détaillé "Modifier l'état d'une inscription".

4.4.5.5 Diagramme de séquence détaillé "Accepter définitivement l'inscription"

Ce cas d'utilisation permet au directeur de finaliser le processus d'inscription d'un enfant. Une fois la demande d'inscription traitée et acceptée, le directeur accède au dossier concerné pour vérifier que toutes les informations sont complètes et conformes : documents requis, informations sur l'enfant, et pièces justificatives. Si le dossier est bien rempli et que l'état de la demande est marqué comme « accepté », il peut valider définitivement l'inscription.

Une fois cette action confirmée, le système effectue automatiquement plusieurs opérations : il enregistre l'état final de l'inscription, génère les frais d'inscription à régler, et envoie une notification au parent pour l'informer que l'inscription de son enfant est désormais définitive.

Le diagramme présenté dans la Figure 4.20 détaille les échanges entre le directeur et le système à travers cette procédure.

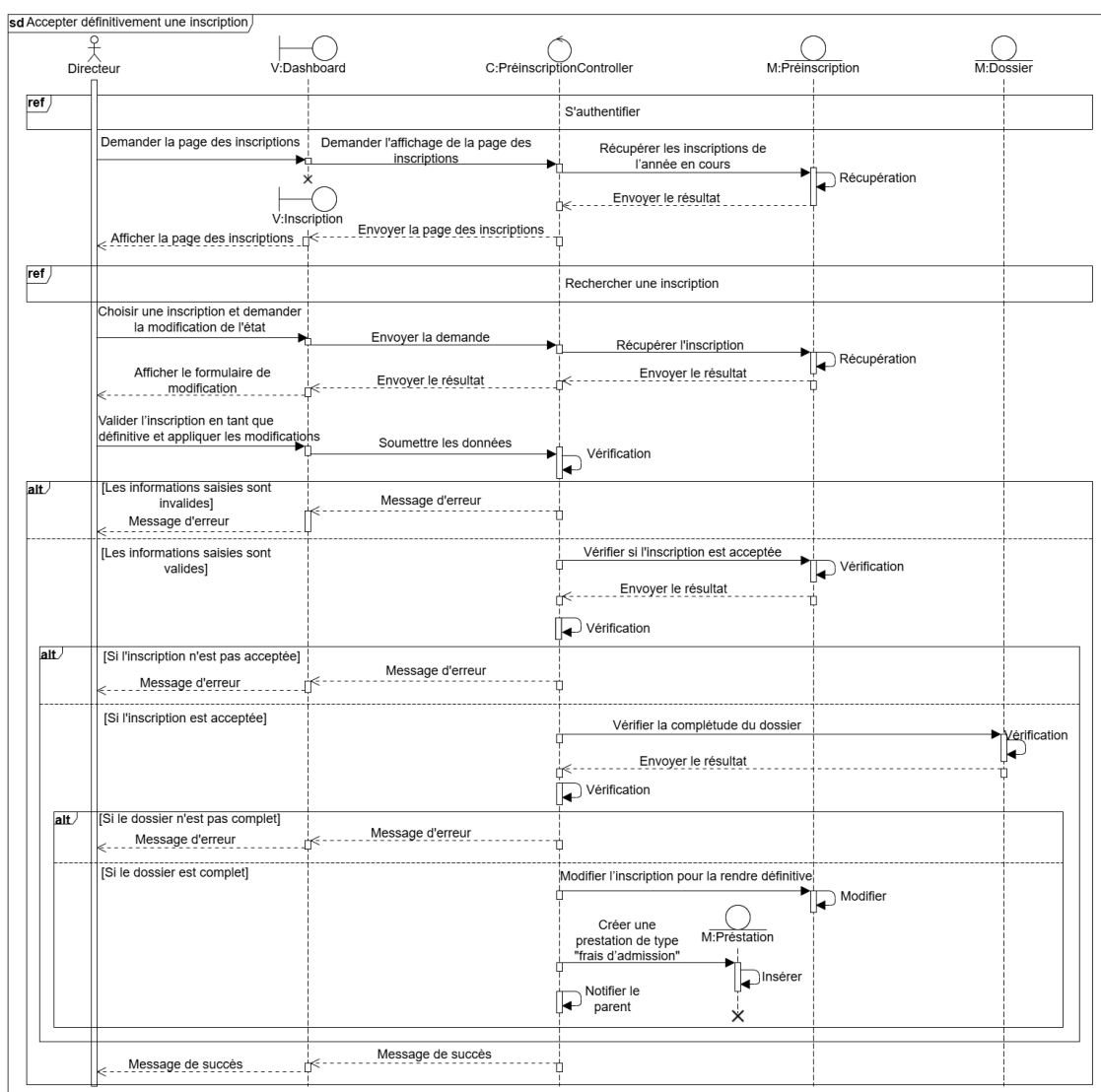


FIGURE 4.20 – Diagramme de séquence détaillé " Accepter définitivement l'inscription ".

4.5 Sprint trois

Dans cette dernière étape du projet, nous avons poursuivi le développement de l'application web de gestion de crèche afin de la finaliser. Ce sprint a permis d'ajouter les dernières fonctionnalités prévues, comme la messagerie, tout en corrigeant les erreurs repérées et en améliorant certains éléments existants.

L'objectif est de livrer une version complète, stable et facile à utiliser, qui réponde pleinement aux besoins des utilisateurs de la crèche.

4.5.1 Backlog produit du sprint trois

Au cours du Sprint trois, nous avons poursuivi le développement de l'application en ajoutant de nouvelles fonctionnalités importantes. Ce sprint a permis de compléter le système avec des modules utiles comme la messagerie, la gestion des factures, des événements et des actualités.

Ces ajouts viennent renforcer les fonctionnalités déjà mises en place dans les sprints précédents, tout en améliorant l'organisation, la communication et la gestion quotidienne au sein de la crèche.

Le tableau suivant 4.25 représente un aperçu du **Backlog produit du sprint trois** avec les fonctionnalités identifiées.

TABLE 4.25 – Le Backlog du produit du sprint trois

ID US	Tâche	Durée
US-33	Élaborer le CU et la description de la fonctionnalité "Gestion des factures". Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	25h
US-34	Élaborer le CU et la description de la fonctionnalité "Gestion des évènements" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	20h
US-35	Élaborer le CU et la description de la fonctionnalité "Valider le paiement d'une facture" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	18h
US-36	Élaborer le CU et la description de la fonctionnalité "Gérer la galerie" par le directeur et les éducateurs. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la gestion de la galerie.	18h
US-37	Élaborer le CU et la description de la fonctionnalité "Gérer les actualités" par le directeur et les éducateurs. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la gestion des actualités.	18h
US-38	Élaborer le CU et la description de la fonctionnalité "Gérer la messagerie" par le directeur, les éducateurs et le parent. Ajouter les diagrammes de séquence nécessaires.	5h
	Développer et tester la fonctionnalité.	25h

Continuation du tableau 4.25		
ID US	Tâche	Durée
US-39	Élaborer le CU et la description de la fonctionnalité "Signaler l'absence de l'enfant au parent" par l'éducateur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la gestion des matières.	18h
US-40	Élaborer le CU et la description de la fonctionnalité "Consulter la galerie" par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	18h
US-41	Élaborer le CU et la description de la fonctionnalité "Consulter l'actualité " par le directeur. Ajouter les diagrammes de séquence nécessaires.	4h
	Développer et tester la fonctionnalité.	18h

4.5.2 Spécifications des besoins

Comme dans la section précédente dédiée au Sprint deux, nous entamons ici l'étape de spécification des besoins pour le troisième sprint.

Cette phase est essentielle car elle permet de poser clairement ce que le système doit offrir en termes de fonctionnalités. Elle nous aide à structurer le travail à venir, en préparant la conception technique, le développement de chaque fonctionnalité, ainsi que les tests qui permettront de vérifier que tout fonctionne correctement.

En définissant précisément les attentes, nous pouvons avancer de manière plus organisée et éviter les imprévus lors des étapes suivantes.

4.5.3 Diagramme de cas d'utilisation du sprint trois

Le diagramme présenté dans la Figure 4.21 regroupe les cas d'utilisation définis lors du troisième sprint. Il met en évidence les fonctionnalités ajoutées et les interactions entre les différents utilisateurs du système.

Le directeur peut gérer les factures (consultation, validation), suivre les événements associés, publier ou modifier les actualités et la galerie, et utiliser la messagerie pour échanger avec les parents ou les éducateurs si besoin.

Le parent, de son côté, peut déclarer l'absence de son enfant, consulter la galerie et les actualités, et utiliser la messagerie pour communiquer directement avec le directeur ou l'éducateur en charge de son enfant.

Enfin, l'éducateur peut accéder aux actualités et à la galerie, et utiliser la messagerie pour informer les parents, notamment en cas d'absence.

Ce diagramme donne une vue d'ensemble des fonctionnalités développées pendant ce sprint et des rôles de chaque utilisateur.

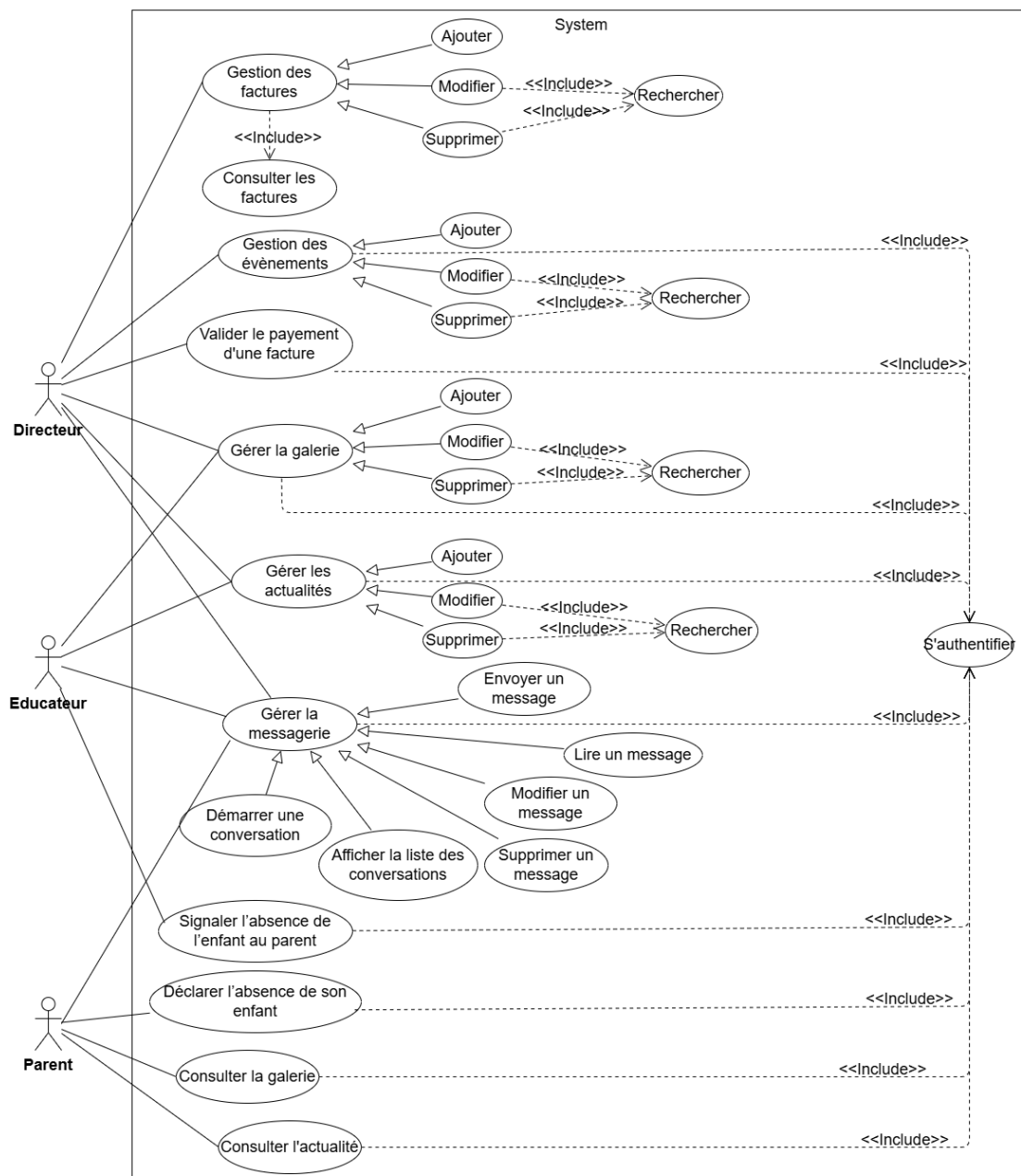


FIGURE 4.21 – Diagramme de cas d'utilisation du sprint trois.

4.5.4 Description textuelle

Dans cette section, nous décrivons par écrit les cas d'utilisation les plus importants développés pendant le sprint trois. Chaque description présente de façon claire ce que fait l'utilisateur, étape par étape, ainsi que les réactions attendues du système. Cela permet de mieux comprendre le fonctionnement des principales fonctionnalités ajoutées au cours de cette phase.

4.5.4.1 Cas d'utilisation "Gérer les actualités"

TABLE 4.26 – Description textuelle du CU : Gérer les actualités.

CU : Gérer les actualités
ID : 1
Description brève : Permettre au directeur de la crèche d'ajouter, modifier ou supprimer des actualités liées à la crèche.
Acteurs primaires : Directeur
Acteurs secondaires : Educateur
Pré-condition : <ol style="list-style-type: none"> 1. Le directeur doit être authentifié. 2. Une saison active doit être définie dans le système.
Enchaînement principal : A. Le directeur souhaite ajouter une actualité. <ol style="list-style-type: none"> 1. Le directeur accède à la page de gestion des actualités via le menu d'accueil. 2. Le système affiche la page de gestion des actualités. 3. Le directeur clique sur "Ajouter une actualité". 4. Le système affiche le formulaire d'ajout. 5. Le directeur saisit les informations (titre, contenu, image facultative) et valide. 6. Le système vérifie les données saisies. 7. Si les informations sont valides, l'actualité est enregistrée et un message de succès est affiché. B. Le directeur souhaite modifier une actualité. <ol style="list-style-type: none"> 1. Le directeur clique sur "Modifier" pour une actualité existante. 2. Le système affiche un formulaire prérempli. 3. Le directeur modifie les champs souhaités (titre, contenu) et valide. 4. Le système met à jour les données et affiche un message de succès.
C. Le directeur souhaite supprimer une actualité : <ol style="list-style-type: none"> 1. Le directeur clique sur "Supprimer" pour une actualité donnée. 2. Le système supprime l'actualité (ainsi que l'image associée si elle existe) et affiche un message de succès. Post-condition : Les actualités sont correctement mises à jour dans le système et reflétées dans la liste des actualités disponibles.
Enchaînements alternatifs : <ol style="list-style-type: none"> 1. Informations invalides lors de l'ajout ou la modification : Le système affiche un message d'erreur précisant les champs invalides. 2. Absence de saison active : Le système empêche l'enregistrement d'une actualité et affiche un message demandant de créer ou activer une saison.

4.5.4.2 Cas d'utilisation "Gérer la galerie"

TABLE 4.27 – Description textuelle du CU : Gérer la galerie.

CU : Gérer la galerie
ID : 2
Description brève : Permettre au directeur d'ajouter, de modifier la visibilité ou de supprimer des photos dans la galerie de la crèche.
Acteurs primaires : Directeur
Acteurs secondaires : Educateur
Pré-condition : <ol style="list-style-type: none"> 1. Le directeur doit être authentifié. 2. Une saison active doit être définie dans le système.
Enchaînement principal : A. Le directeur souhaite ajouter une photo à la galerie. <ol style="list-style-type: none"> 1. Le directeur accède à la page de gestion de la galerie via le menu. 2. Le système affiche la liste des photos de la saison actuelle. 3. Le directeur clique sur "Ajouter une photo". 4. Le système affiche le formulaire d'ajout. 5. Le directeur sélectionne une photo (format jpg, jpeg, png) et peut renseigner un titre (facultatif), puis valide. 6. Le système vérifie la validité du fichier et enregistre la photo. 7. Un message de confirmation s'affiche. B. Le directeur souhaite modifier la visibilité d'une photo. <ol style="list-style-type: none"> 1. Le directeur sélectionne une photo existante. 2. Il coche ou décoche la case "visible". 3. Le système enregistre le changement de visibilité et affiche un message de succès. C. Le directeur souhaite supprimer une photo : <ol style="list-style-type: none"> 1. Le directeur clique sur "Supprimer" pour une photo donnée. 2. Le système supprime le fichier de stockage et l'enregistrement en base de données. 3. Un message de confirmation s'affiche.
Post-condition : La galerie est correctement mise à jour et affiche les photos visibles liées à la saison en cours.
Enchaînements alternatifs : <ol style="list-style-type: none"> 1. Informations invalides lors de l'ajout : Le système affiche un message d'erreur précisant le format ou la taille du fichier invalide. 2. Absence de saison active : Le système empêche l'ajout d'une photo et redirige vers la gestion des saisons avec un message d'erreur.

4.5.4.3 Cas d'utilisation "Gérer la messagerie"

TABLE 4.28 – Description textuelle du CU : Gérer la messagerie.

CU : Gérer la messagerie
ID : 3
Description brève : Permettre à un utilisateur authentifié d'échanger des messages avec d'autres utilisateurs via une interface de messagerie intégrée.
Acteurs primaires : Utilisateur (parent, éducateur, directeur)
Acteurs secondaires : Aucun
Pré-condition : <ol style="list-style-type: none"> 1. L'utilisateur est authentifié. 2. L'utilisateur a accès à la messagerie via son tableau de bord.
Enchaînement principal : <ol style="list-style-type: none"> 1. L'utilisateur accède à la page de messagerie via le menu du tableau de bord. 2. Le système récupère et affiche la liste de ses conversations existantes. 3. L'utilisateur peut cliquer sur une conversation pour consulter et envoyer de nouveaux messages. 4. L'utilisateur peut aussi initier une nouvelle conversation. 5. Le système propose un champ de recherche permettant de trouver un autre utilisateur (par nom, prénom ou e-mail). 6. Si une conversation existe déjà avec cet utilisateur, le système l'affiche. 7. Sinon, une nouvelle conversation est créée et un message de bienvenue est automatiquement ajouté. 8. L'utilisateur saisit un message et l'envoie. 9. Le système enregistre le message et met à jour la conversation. 10. Le nouveau message apparaît dans la boîte de discussion.
Post-condition : La conversation est enregistrée avec tous les messages envoyés et peut être consultée ou poursuivie ultérieurement.
Enchaînements alternatifs : <ol style="list-style-type: none"> 1. Aucun utilisateur trouvé lors de la recherche : Le système affiche un message indiquant qu'aucun résultat ne correspond aux critères. 2. Problème lors de l'enregistrement du message : Le système affiche une erreur et le message n'est pas envoyé.

4.5.4.4 Cas d'utilisation "Gérer les évènements"

TABLE 4.29 – Description textuelle du CU : Gérer les évènements.

CU : Gérer les évènements
ID : 4
Description brève : Permettre au directeur de la crèche d'ajouter, modifier ou supprimer des évènements liés à la crèche (fêtes, sorties, réunions, etc).
Acteurs primaires : Directeur
Acteurs secondaires : Aucun
Pré-condition : <ol style="list-style-type: none"> 1. Le directeur doit être authentifié. 2. Une saison active doit être définie dans le système.

Continuation du tableau 4.29

Enchaînement principal :**A. Le directeur souhaite ajouter un évènement.**

1. Le directeur accède à la page de gestion des évènements via le menu d'accueil.
2. Le système affiche la liste des évènements existants.
3. Le directeur clique sur "Ajouter un évènement".
4. Le système affiche un formulaire vide.
5. Le directeur saisit les informations (description, date, heure, etc.) et valide.
6. Le système vérifie les données saisies.
7. Si les données sont valides, l'évènement est enregistré et un message de succès est affiché.

B. Le directeur souhaite modifier un évènement.

1. Le directeur accède à la page des évènements.
2. Le système affiche la liste des évènements.
3. Le directeur sélectionne un évènement et clique sur "Modifier".
4. Le système affiche un formulaire prérempli avec les informations existantes de l'évènement.
5. Le directeur modifie les champs souhaités et valide.
6. Le système vérifie les nouvelles données.
7. Si elles sont valides, les modifications sont enregistrées et un message de succès est affiché.

C. Le directeur souhaite supprimer un évènement.

1. Le directeur clique sur "Supprimer" à côté de l'évènement concerné.
2. Le système affiche une demande de confirmation.
3. Le directeur confirme la suppression.
4. Le système supprime l'évènement et affiche un message de succès.

Post-condition : Les évènements sont correctement enregistrés, modifiés ou supprimés dans le système et mis à jour dans la liste affichée.

Enchaînements alternatifs :

1. Informations invalides lors de l'ajout ou la modification : Le système affiche un message d'erreur précisant les champs invalides à corriger.
2. Absence de saison active : Le système empêche toute opération sur les évènements et affiche un message demandant de créer ou d'activer une saison.

4.5.5 Diagrammes de séquences détaillés

Cette phase vise à représenter de manière précise le déroulement des nouvelles fonctionnalités mises en place lors du Sprint trois. Pour cela, des diagrammes de séquence ont été élaborés afin de visualiser les interactions entre les différents acteurs et le système.

Chaque diagramme décrit étape par étape le comportement attendu de l'application dans un scénario spécifique, ce qui permet de mieux comprendre le fonctionnement interne des cas d'utilisation développés.

4.5.5.1 Diagramme de séquence détaillé "Ajouter une actualité"

Le cas d'utilisation « Gérer une actualité » permet au directeur de publier de nouvelles informations destinées aux parents ou aux éducateurs. Cela inclut l'ajout, la modification ou la suppression d'actualités visibles sur la plateforme.

Dans le cas particulier de l'ajout d'une actualité, le directeur ou l'éducateur accède à l'interface prévue à cet effet, puis saisit les informations nécessaires : titre, description, contenu et éventuellement une image. Une fois tous les champs remplis, il valide l'opération.

Le système enregistre alors l'actualité dans la base de données et l'affiche dans l'espace dédié, consultable par les utilisateurs concernés. Un message de confirmation est affiché pour indiquer que la publication a bien été effectuée.

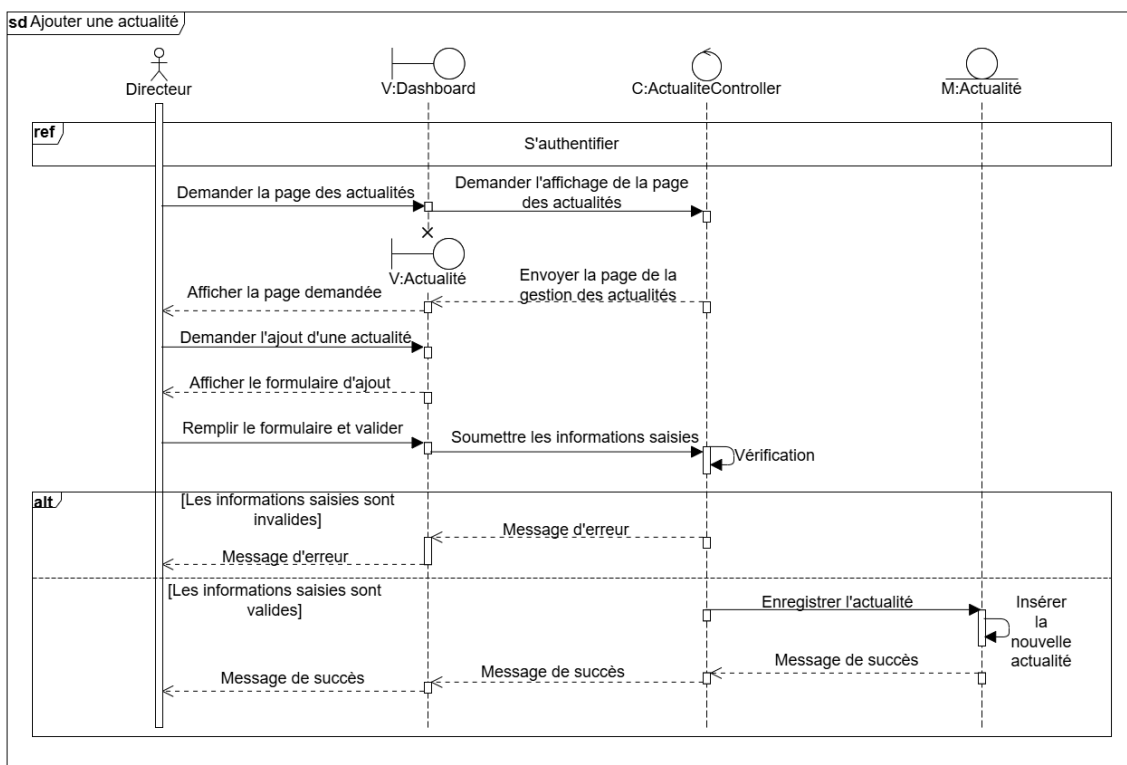


FIGURE 4.22 – Diagramme de séquence détaillé "Ajouter une actualité".

4.5.5.2 Diagramme de séquence détaillé "Gérer la messagerie"

Le cas d'utilisation « Gérer la messagerie » permet à un utilisateur authentifié (parent, éducateur, directeur.) de consulter ses conversations, d'envoyer des messages ou d'initier de nouveaux échanges avec d'autres utilisateurs via une interface dédiée.

Lorsqu'un utilisateur envoie un message, le système vérifie l'existence d'une conversation ou en crée une nouvelle, enregistre le message, puis met à jour l'affichage en conséquence.

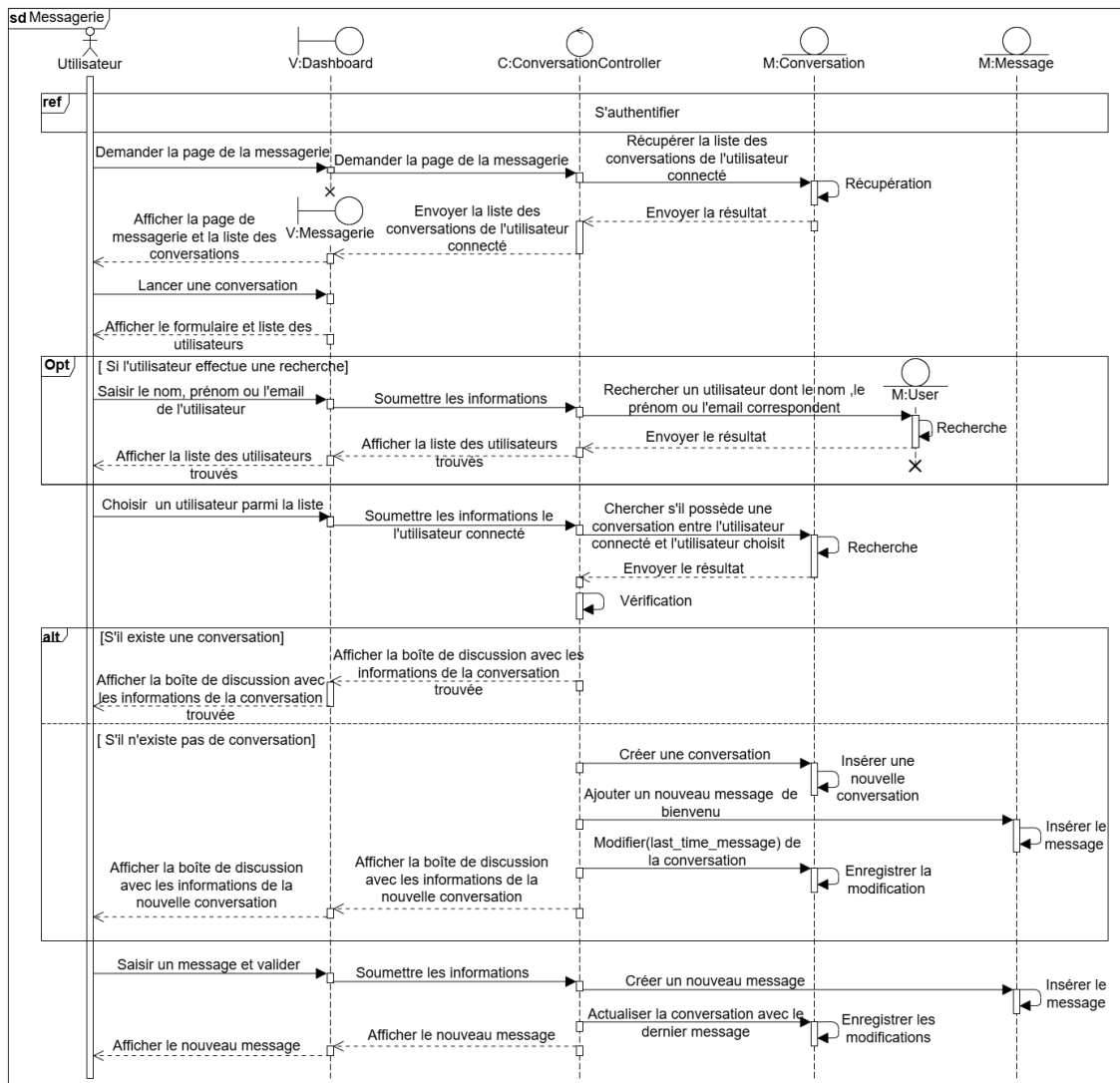


FIGURE 4.23 – Diagramme de séquence détaillé "Gérer la messagerie".

4.5.5.3 Diagramme de séquence détaillé "Modifier un évènement"

Le cas d'utilisation « Gérer les évènement » permet au directeur d'ajouter, modifier ou supprimer des évènement dans la crèche.

Pour l'option "Modifier un évènement", le directeur choisit un évènement dans la liste. Le système montre les détails de cet évènement. Le directeur peut alors changer les informations comme le titre, la date, l'heure, la description.

Après avoir fait les modifications, le directeur clique sur un bouton pour valider. Le système vérifie que les informations sont correctes, enregistre les changements, puis affiche un message pour dire que tout s'est bien passé.

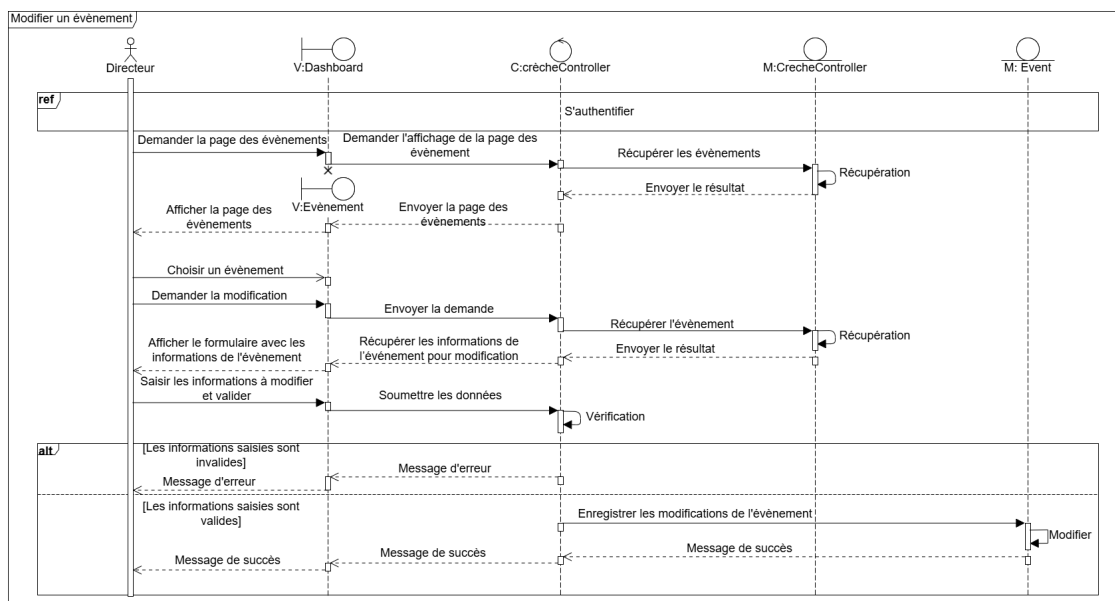


FIGURE 4.24 – Diagramme de séquence détaillé "Modifier un évènement".

4.5.6 Implémentation

Dans cette phase, nous avons programmé les fonctionnalités du système en utilisant les langages et outils choisis. Nous avons suivi les spécifications et la conception faites auparavant pour transformer les idées en code. Chaque partie a été développée pour fonctionner comme prévu, selon les besoins du système.

4.5.6.1 Diagramme de classe

Le diagramme de classes global donne une vue d'ensemble de la structure de notre application. Il met en évidence les principales classes, leurs attributs, ainsi que les relations qui les lient, tout en tenant compte des besoins fonctionnels et des évolutions prévues du système. La figure 4.25 illustre ce diagramme.

4.5.6.2 Modèle relationnel

Le modèle relationnel est une façon d'organiser les données dans une base de données sous forme de tables reliées entre elles. ce qui facilite le stockage et l'accès aux informations. Pour passer d'un modèle conceptuel à un modèle relationnel, il est nécessaire d'appliquer des règles de transformation précises [40].

Règles de passage

Ces règles permettent de transformer les classes et les associations en tables et relations dans la base de données. Elles assurent une bonne correspondance entre le modèle conceptuel et le modèle logique relationnel utilisé pour la création de la base de données.

1. Transformation des classes

- Chaque classe devient une table dans le modèle relationnel.
- Les attributs de la classe deviennent les colonnes de la table.
- Il faut choisir un attribut comme clé primaire (généralement un attribut ou un identifiant unique)

2. Transformation des associations

- **Association 1..1 (un à un) ou 0..1 :**
On ajoute une clé étrangère dans l'une des deux tables.
- **Association 1..N (un à plusieurs) :**
La table du côté "plusieurs" reçoit une clé étrangère vers la table du côté "un".
- **Association N..N (plusieurs à plusieurs) :**
On crée une table associative avec deux clés étrangères.
- **Présence d'héritage :**
 - **Push-Up :** Une seule table pour toutes les classes, avec des clés étrangères pour gérer les associations.
 - **Push-Down :** Chaque sous-classe a sa propre table, avec des clés étrangères vers la super-classe.
 - **La distinction :** Chaque sous-classe a une table contenant tous les attributs, avec des clés étrangères de la super-classe.

Le schéma de la base de donnée équivalent à notre diagramme de classe est le suivant :

Parent (id_parent, est_profilCompleet, lien_parental, est_firstRepresentant, profession, relation, #id_user, #id_deuxiemeRepresentant)

Tuteur (id_tuteur, nom, prenom, tel, adresse, photo_path, nin)

Utilisateur (id_user, nom, prenom, date_naissance, adresse, telephonel, email, etat_civil, role, photo_path, type_carte, numero_carte, mot_de_passe, email, nin, is_actif, insta_url, fcb_url, sexe, date_expiration)

Employe (id_employe, poste, date_embauche, type_contrat, annee_experience, date_sortie, #id_user)

Educateur (id_educateur, #id_employe)

Directeur (id_directeur, #id_user)

Creche (id_creche, nom_creche, adresse_creche, tel1_creche, tel2_creche, email_creche, heure_ouverture, heure_fermeture, horaires_debut_sortie, num_agrement, capacité_accueil, fcb_url, insta_url, logo_path)

Enfant (id_enfant, nom, prenom, matricule, photo_path, sexe, date_naissance, lieu_naissance, groupe_sanguin, adresse, langues_parlees, date_sortie, active, est_handicape, details_handicap, a_regime_alimentaire, details_regime_alimentaire, a_allergies, details_allergies, a_maladie_chronique, details_maladies, a_maladies_passées, details_maladies_passees, numero_act_naissance, #id_parent)

Presence_enfant (id_presence_enfant, date_presence, heure_presence, status, note, #id_enfant)

Presence_educateur (id_presence_educateur, date_presence, heure_presence, status, note, #id_educateur)

Evaluation (id_evaluation, remarque, note, #id_preinscription, #id_matiere, #id_educateur)

Message (id_message, body, read, is_visible_sender, is_visible_receiver, #id_sender, #id_receiver, #id_conversation)

Conversation (id_conversation, last_time_message, sender_is_pinned, receiver_is_pinned, sender_deleted, receiver_deleted, #id_sender, #id_receiver)

Actualite (id_actualite, contenu, titre, image, date_publication, #id_user)

Galerie (id_galerie, path, est_visible, #id_user)

Planning_Cour (id_planning, #id_saison, #id_section)

Seance (id_seance, heure_debut, heure_fin, jour, #id_matiere, #id_planning)

Matiere (id_matiere, nom, description)

Groupe (id_groupe, nom_groupe, description, mois_min, mois_max, #id_section, #id_salle, #id_saison)

Affectation (#id_educateur, #id_groupe, date_affectation, date_fin_affectation, status, #id_saison)

Salle (id_salle, numero_salle, capacite, numero_etage)

Repas (id_repas, nom_repas, description, type_repas)

Jour (id_jour, nom_jour)

Jour_Repas (#id_jour, #id_repas, #id_saison)

Document (id_document, titre, type_document, fichier_path, documentable_type, documentable_id, #id_employe)

Preinscription (id_preinscription, etat, commentaire, est_inscrit_definitivement, date_entree, date_creation, #id_enfant, #id_saison, #id_section, #id_parant, #id_groupe)

Preinscription_Abonnement (#id_preinscription, #id_cat_abon, jour, matin, apres_midi, avec_repas, heure_arrivee, heure_depart)

Dossier (id_dossier, etat, date_depot, date_validation, numero_dossier, commentaire, remarque, autorisation_publicationPhotos, date_creation_dossier, #id_preinscription)

Dossier_Tuteur (#id_dossier, #id_tuteur, relation)

Saison (id_saison, nom_saison, date_debut, date_fin, active, #id_reglement)

Supplement (id_supplement, nom_supplement, description)

Supplement_Saison (#id_supplement, #id_saison, montant, active)

Supplement_Dossier (#id_supplement, #id_dossier)

Section (id_section, nom_section, description, age_min, age_max)

Categorie Abonnement (id_cat_abon, description, type, avec_repas)

Section_Abonnement (#id_section, #id_cat_abon, #id_saison, active, montant)

Saison_Section (#id_saison, #id_section, frais_inscription)

Prestation (id_prestation, montant_total, montant_restant, montant_paye, date_debut_periode, date_fin_periode, description, #id_facture, #id_preinscription)

Facture (id_facture, numero_facture, date_emission, date_echeance, statut, montant_total, commentaire, montant_paye, date_paiement)

Evenement (id_event, titre_event, type_event, date_debut, date_fin, lieu, description, fichier_path, status, #id_preinscription)

Participant (id_participant, participantable_id, participantable_type)

Evenement_Participant (#id_event, #id_participant, est_confirme)

4.6 Conclusion

Ce chapitre nous a permis de retracer toutes les étapes du développement de notre application, depuis les premières fonctionnalités jusqu'aux dernières améliorations. En suivant la méthode Scrum, nous avons pu avancer de façon structurée, en découpant le travail en sprints clairs et progressifs. Chaque étape a contribué à faire évoluer le projet, à corriger les imperfections et à répondre au mieux aux besoins des utilisateurs.

Cette démarche nous a non seulement aidés à mieux organiser nos tâches, mais aussi à travailler de manière plus collaborative et efficace. Au final, elle a abouti à une application plus complète, plus stable et mieux adaptée aux attentes de la crèche.

Chapitre 5

Réalisation

5.1 Introduction

Dans ce chapitre, nous abordons la phase de réalisation de notre application. Il s'agit de montrer concrètement comment le projet a été mis en œuvre, en expliquant les choix techniques que nous avons faits.

Nous présentons d'abord les outils et langages utilisés, ainsi que l'environnement de développement mis en place. Ensuite, nous décrivons les principales interfaces de l'application et les fonctionnalités développées, afin d'illustrer comment celles-ci répondent aux besoins identifiés lors des étapes précédentes.

5.2 Outils et environnement de développement

Plusieurs outils sont mis à notre disposition pour faciliter le processus de réalisation d'une application informatique. Pour mener ce projet jusqu'à son terme. On a procédé comme suit :

5.2.1 Environnement de développement

Laravel

Laravel est un framework PHP open-source qui facilite le développement web grâce à une syntaxe claire et des outils puissants comme le routage, Eloquent pour la gestion des bases de données, et une architecture MVC. Il prend en charge l'injection de dépendances, la gestion des tâches planifiées, les files d'attente, et les tests, offrant une solution adaptée aux développeurs de tous niveaux [41].



FIGURE 5.1 – Logo de Laravel [42].

Bootstrap

Bootstrap est un framework CSS open-source conçu pour faciliter et accélérer le développement de sites et d'applications web responsives et mobiles-first. Il fournit des outils préconçus en HTML, CSS et JavaScript, incluant des composants tels que des grilles flexibles, des boutons, des formulaires et des menus de navigation, permettant aux développeurs de créer des interfaces cohérentes et esthétiques sans avoir à coder chaque élément à partir de zéro [43].



FIGURE 5.2 – Logo de Bootstrap [44].

Visual Studio Code (VS Code)

Visual Studio Code est un éditeur de code léger et extensible développé par Microsoft. Il supporte plusieurs langages et dispose d'une grande communauté pour les extensions [45].



FIGURE 5.3 – Logo de VS Code [46].

WAMP Server

WAMP (Windows, Apache, MySQL, PHP) est un environnement de développement local sous Windows qui permet de créer et de tester des sites ou applications web. WampServer est une version de WAMP utilisée comme environnement de développement web sous Windows. Il constitue une plateforme dynamique intégrant le serveur Apache 2, le langage de script PHP et le système de gestion de base de données MySQL. Il inclut également phpMyAdmin, un outil qui facilite la gestion des bases de données via une interface graphique[47].



FIGURE 5.4 – Logo de WAMP Server [48].

GitHub

GitHub est une plateforme d'hébergement de code basée sur Git. Elle permet de collaborer sur des projets open-source ou privés, tout en offrant des fonctionnalités comme les pull requests, les issues, et les actions CI (Continuous Integration)/CD (Continuous Delivery) [49].



FIGURE 5.5 – Logo de GitHub [50].

5.2.2 Technologies web utilisées

HTML

HTML est un langage de balisage utilisé pour structurer le contenu des pages web. Il définit la structure des documents en utilisant des balises comme `<h1>`, `<p>`, ``, etc [51].



FIGURE 5.6 – Logo de HTML [52]

CSS

CSS est un langage de style utilisé pour décrire l'apparence des documents HTML. Il permet de contrôler les couleurs, polices, marges, et la disposition des éléments[53].



FIGURE 5.7 – Logo de CSS [54]

JavaScript

JavaScript est un langage de programmation interprété qui permet d'ajouter des fonctionnalités dynamiques aux pages web, comme des animations, des interactions avec l'utilisateur, ou la manipulation du DOM (Document Object Model) [55].



FIGURE 5.8 – Logo de JavaScript [56]

PHP

PHP est un langage de programmation backend utilisé principalement pour le développement d'applications web dynamiques. Il s'intègre facilement avec HTML et gère des bases de données comme MySQL [57].



FIGURE 5.9 – Logo de PHP [58].

SQL

Le langage SQL est un langage informatique utilisé pour exploiter des bases de données. Il permet d'exécuter des tâches telles que la création, la gestion et la manipulation des données dans une base [59].



FIGURE 5.10 – Logo de SQL [60].

5.2.3 Autres Outils

Visual Paradigm

Un logiciel est un outil en ligne puissant, multiplateforme et pourtant facile à utiliser qui permet la création des diagrammes UML, concevoir des bases de données, et gérer des projets logiciels [61].



FIGURE 5.11 – Logo de Visual Paradigm [62].

MySQL

MySQL est un système de gestion de bases de données relationnelles open- source. Il est utilisé pour stocker et gérer les données d'applications web et de logiciels[63].



FIGURE 5.12 – Logo de MySQL [64].

Git

Git est un système de gestion de versions distribué utilisé pour suivre les modifications du code source. Il permet aux développeurs de collaborer efficacement en gérant les différentes versions d'un projet [65].



FIGURE 5.13 – Logo de Git [66].

Composer

Composer dans le contexte de Laravel est un gestionnaire de dépendances pour PHP. Il permet d'installer, de mettre à jour et de gérer les bibliothèques et outils nécessaires pour développer une application Laravel ou toute autre application PHP [67].

Exemple d'utilisation dans Laravel :

1. **Installer laravel :**
"composer create-project laravel/laravel nom-du-projet"
2. **Ajouter un package (laravel/ui) :**
"composer require laravel/ui"
3. **composer update :**
"composer update"



FIGURE 5.14 – Logo de Composer [68].

5.3 Principales interfaces

5.3.1 Accueil

La page d'accueil, illustrée dans la Figure 5.15, joue un rôle central dans la navigation. Elle permet aux visiteurs d'accéder facilement aux différentes rubriques disponibles : programmes proposés, informations pour les parents, galerie de photos, actualités de la crèche, et formulaire de contact. Deux boutons situés dans l'en-tête

permettent également aux utilisateurs de se connecter à leur compte ou d'en créer un. Cette interface claire a été conçue pour faciliter l'accès aux informations essentielles, tout en garantissant une navigation intuitive pour tous les profils d'utilisateurs.

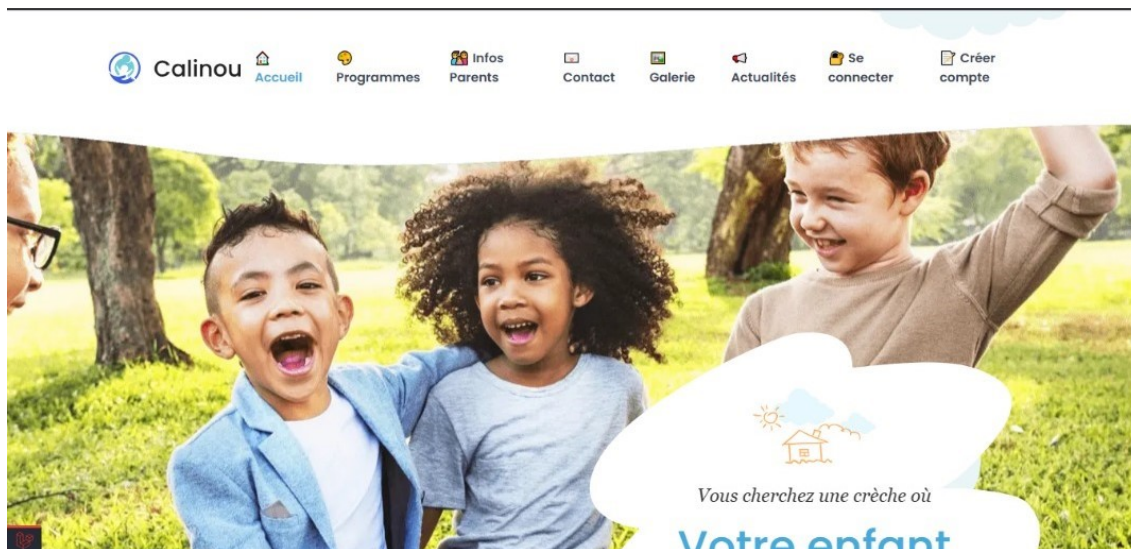


FIGURE 5.15 – Interface d'accueil

5.3.2 Authentification

Comme illustré dans la Figure 5.16, cette interface permet au directeur, aux éducateurs et aux parents de s'authentifier afin d'accéder à leur espace personnel. L'utilisateur saisit son adresse e-mail et son mot de passe, puis clique sur le bouton « Se connecter ». En cas d'erreur (mot de passe incorrect ou compte inexistant), des messages d'alerte sont affichés pour guider l'utilisateur. Un lien « Mot de passe oublié » est également disponible pour initier une procédure de réinitialisation.

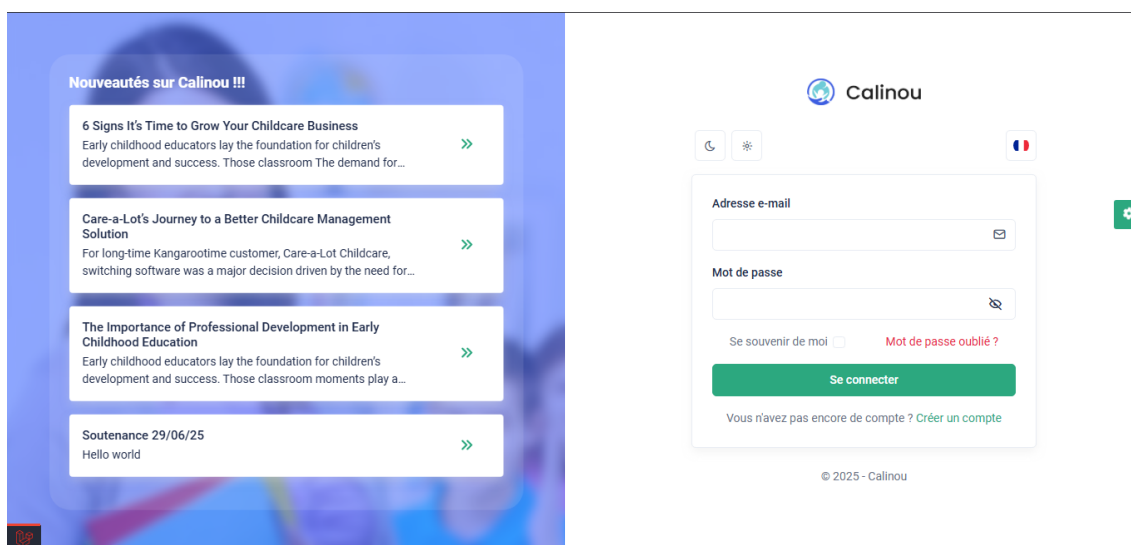


FIGURE 5.16 – Interface d'authentification

5.3.3 S'inscrire

La Figure 5.17 présente l'interface d'inscription, destinée aux parents n'ayant pas encore de compte. Elle leur permet de créer un compte personnel en remplissant les champs requis (nom, prénom, adresse e-mail, mot de passe). Une fois le formulaire complété et validé, les données sont enregistrées.

Après la création du compte, un e-mail de validation est automatiquement envoyé à l'adresse saisie. Le parent doit cliquer sur le lien reçu afin d'activer son compte. Une fois cette étape est effectuée, il peut accéder à la plateforme et compléter son profil utilisateur.

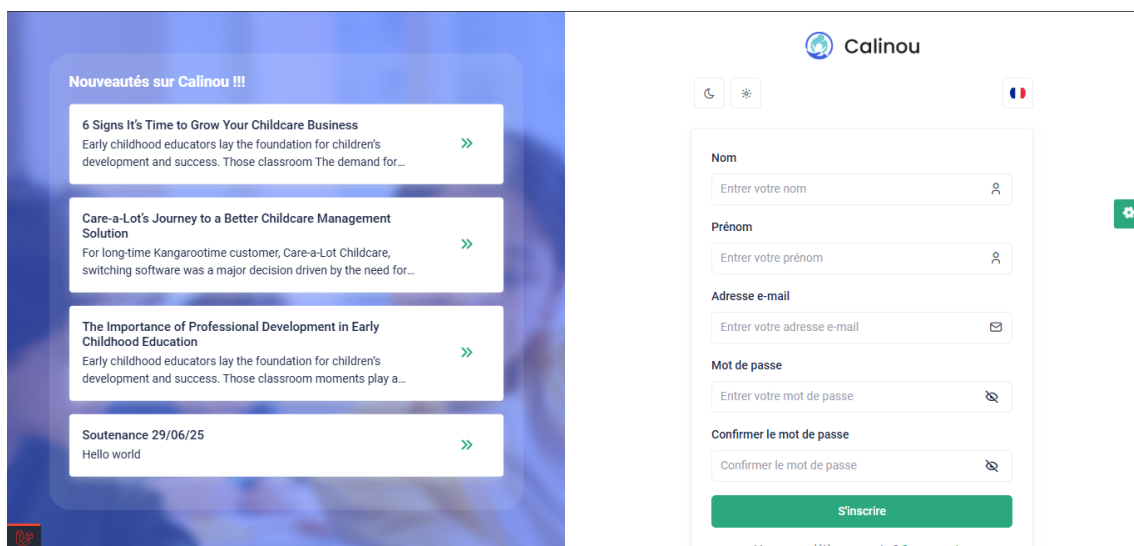
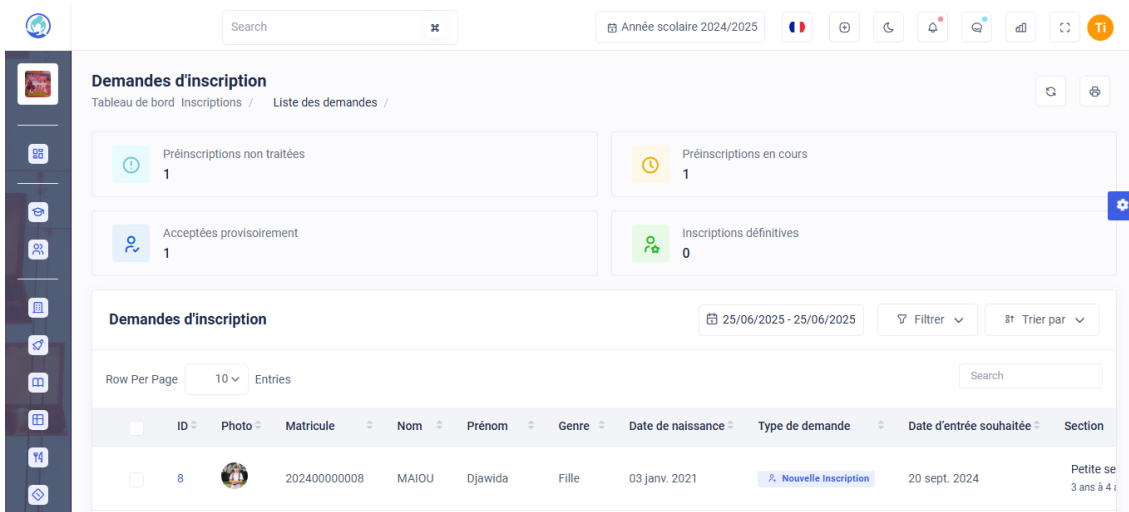


FIGURE 5.17 – Interface d'inscription

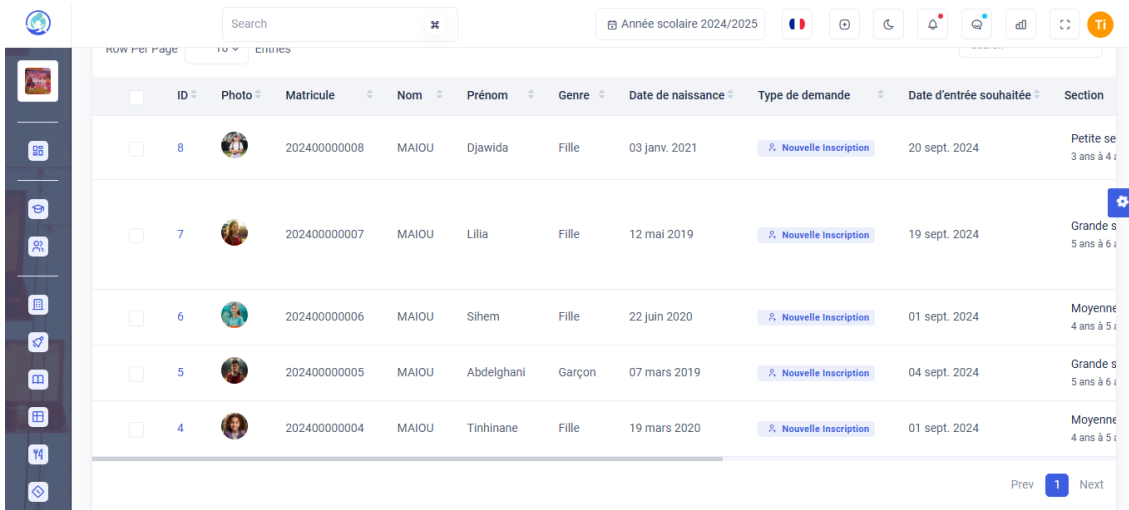
5.3.4 Demandes d'inscription reçues

L'interface présentée à la Figure 5.18 permet au directeur de consulter l'ensemble des demandes d'inscription envoyées par les parents, qu'il s'agisse de nouvelles inscriptions ou de réinscriptions. Chaque demande est accompagnée d'informations détaillées telles que la photo, la matricule, le nom, le prénom, le sexe, la date de naissance de l'enfant, le type de la demande (nouvelle, réinscription), la date d'entrée souhaitée, la section demandée, ainsi que l'état d'avancement de la demande.

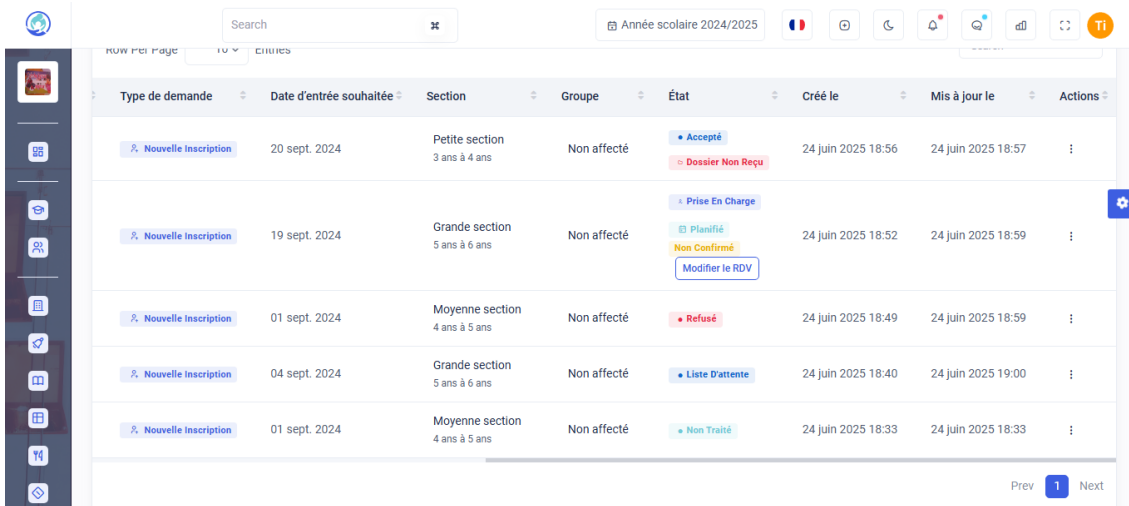
Cet espace permet au directeur de suivre, traiter et organiser efficacement les demandes d'inscription.



(a) Première partie de l'interface



(b) Deuxième partie de l'interface

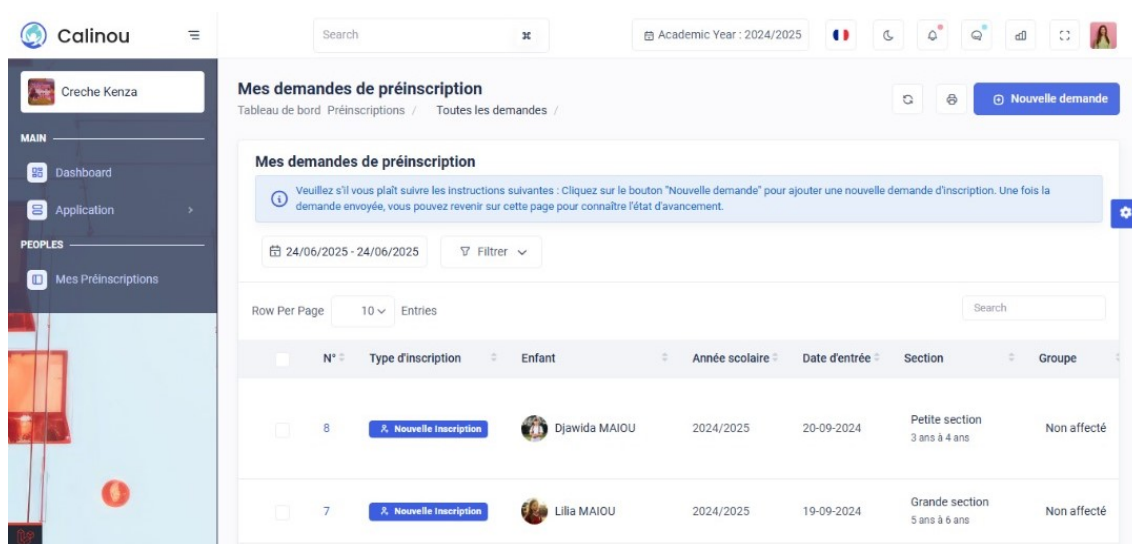


(c) Troisième partie de l'interface

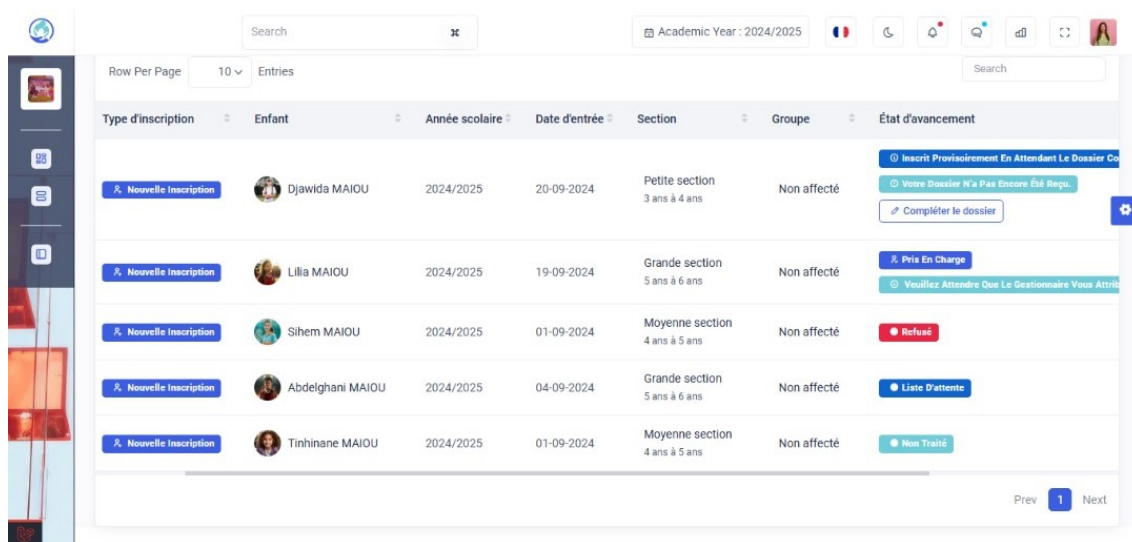
FIGURE 5.18 – Interface complète des demandes d'inscription envoyées au directeur

5.3.5 Liste des inscriptions effectuées par le parent

L'interface présentée à la Figure 5.19 affiche l'ensemble des inscriptions réalisées par le parent, qu'il s'agisse d'une nouvelle inscription ou d'une réinscription. Chaque inscription est accompagnée d'informations détaillées telles que le nom et le prénom de l'enfant, la saison concernée, la date d'entrée souhaitée, la section demandée, ainsi que l'état d'avancement de sa demande d'inscription. Ce module permet au parent de suivre l'évolution de ses demandes en temps réel, garantissant ainsi un suivi administratif clair et structuré.



(a) Première partie de l'interface



(b) Deuxième partie de l'interface

FIGURE 5.19 – Interface des inscriptions effectuées par le parent

5.3.6 Soumission d'une demande d'inscription par le parent

L'interface présentée à la Figure 5.20 permet au parent de remplir une demande d'inscription pour son enfant. Il peut y indiquer les informations personnelles, choisir les horaires d'accueil, la date d'entrée souhaitée, et signaler d'éventuels problèmes de santé. Ces éléments aident la crèche à organiser un accueil adapté.

Ajouter une demande d'inscription
Tableau de bord Préinscriptions / Ajouter une demande d'inscription /

Type d'inscription : Nouvelle inscription :

Informations personnelles

Photo d'identité
Taille max 3Mo, formats autorisés : JPG, PNG, SVG

Nom: Maïou Prénom: Tinhinane Genre: Fille

Date de naissance: 10-02-2021 Lieu de naissance: Béjala Numéro d'acte de naissance: 2392

Groupe sanguin: A+ Adresse: El hamma Amizour Bejala Langues parlées: Arabe Kabyle Français Anglais

(a) Étape 1 : Informations personnelles de l'enfant

Horaires d'accueil

Jours et horaires Date d'entrée: 01-09-2024

Plage	Dimanche	Lundi	Mardi	Mercredi	Jeudi
Matin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Repas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Après-midi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Heure d'arrivée	08:50	09:30	08:50	09:10	09:00
Heure de départ	09:10	09:10	09:10	09:10	09:00

Veillez répondre à ces quelques questions

(b) Étape 2 : Sélection des horaires et de la date d'entrée

Veillez répondre à ces quelques questions

L'enfant a-t-il un régime alimentaire particulier ?
Oui Non

L'enfant souffre-t-il d'une maladie chronique ?
Oui Non

L'enfant a-t-il eu de : oreillons, conjonctivite, varicelle, rougeole, ou des poux ?
Oui Non
elle a eu oreillons, poux

L'enfant a-t-il des allergies ?
Oui Non

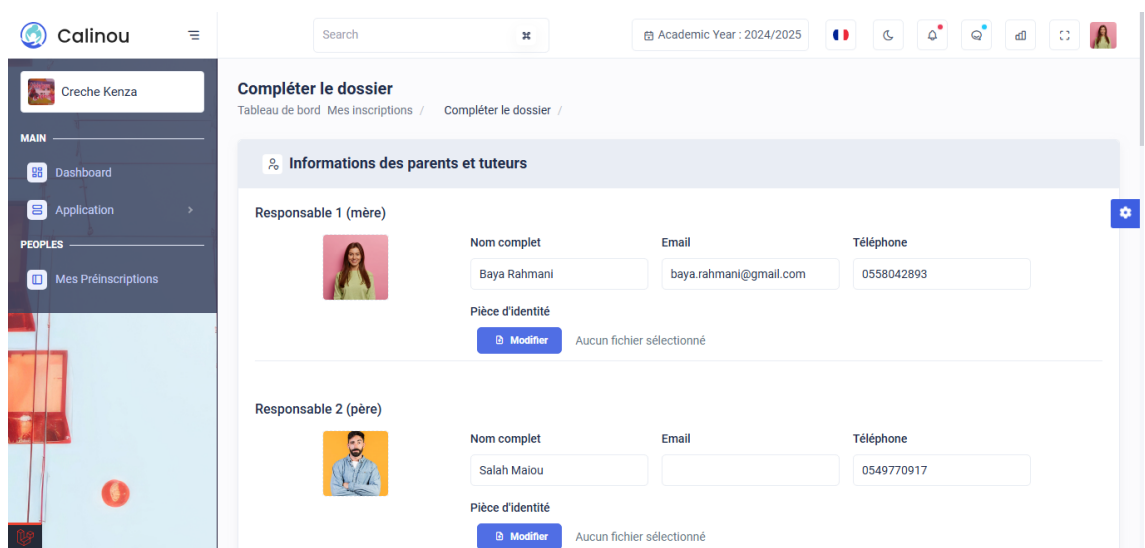
L'enfant est-il en situation de handicap ?
Oui Non

(c) Étape 3 : Questions de santé et validation

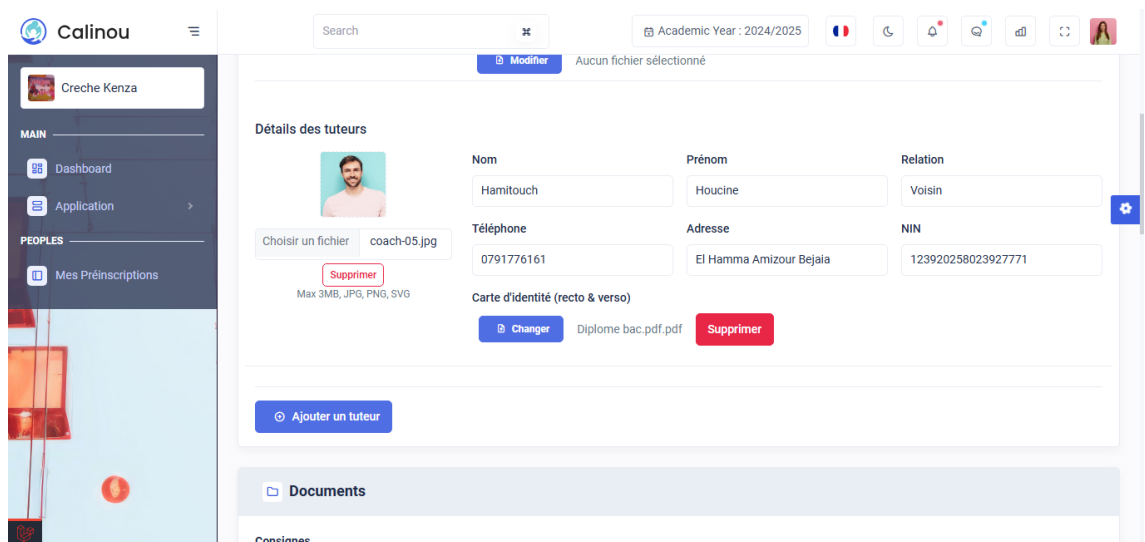
FIGURE 5.20 – Interface de soumission d'une demande d'inscription par le parent

5.3.7 Compléter le dossier d'inscription

L'interface représentée dans la Figure 5.21 permet aux parents de compléter le dossier d'inscription de leurs enfants. Elle propose des champs à renseigner concernant les informations personnelles des responsables de l'enfant et les documents à fournir, ainsi qu'un champ réservé à la signature des représentants.

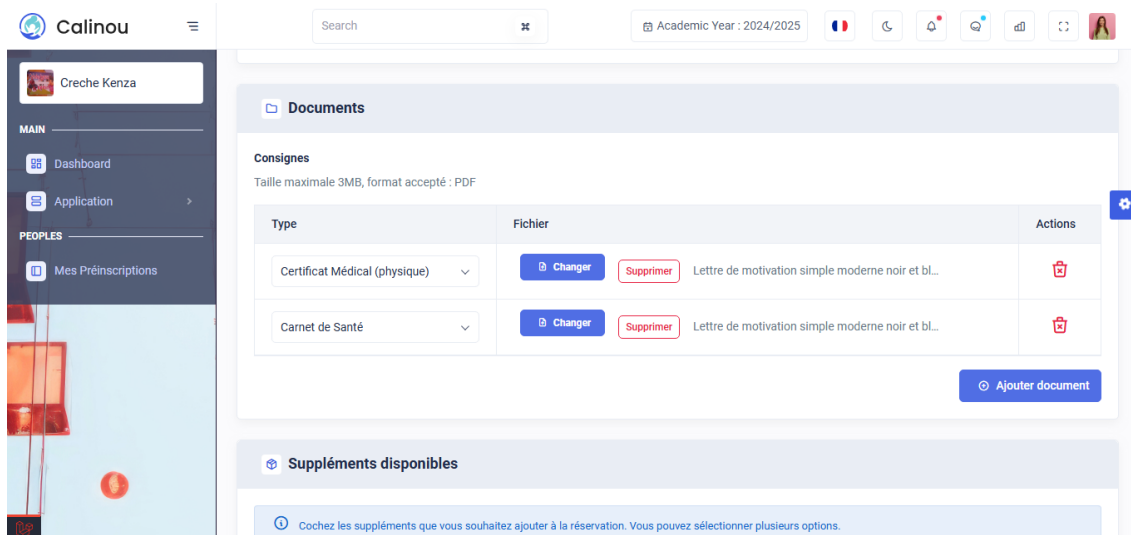


(a) Informations des deux responsables

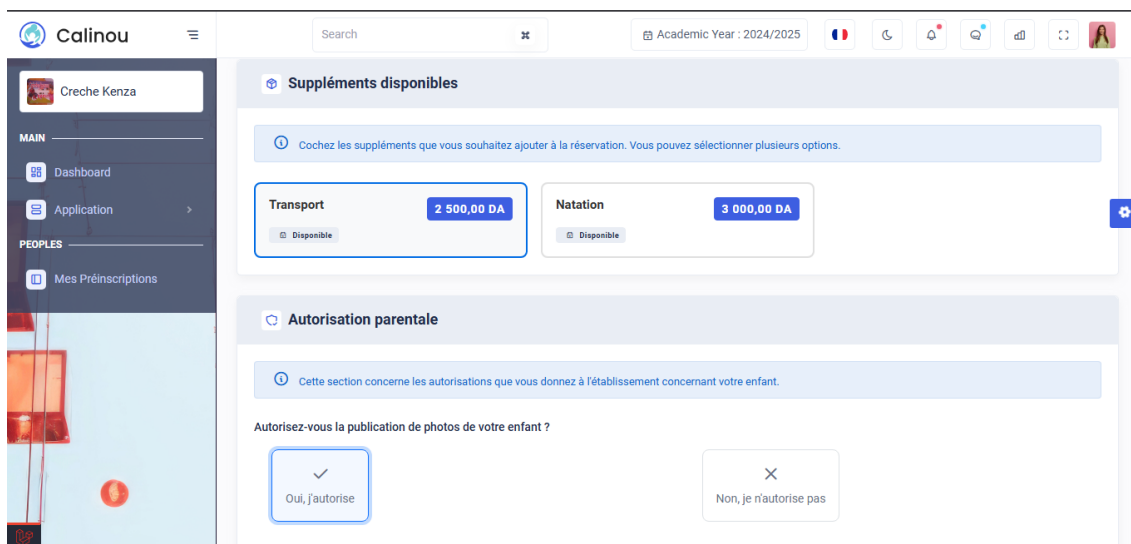


(b) Informations des tuteurs

FIGURE 5.21 – Interface de saisie des responsables légaux



(a) Ajout des documents



(b) Choix du supplément et autorisation parentale

FIGURE 5.22 – Interface de complétion du dossier d'inscription

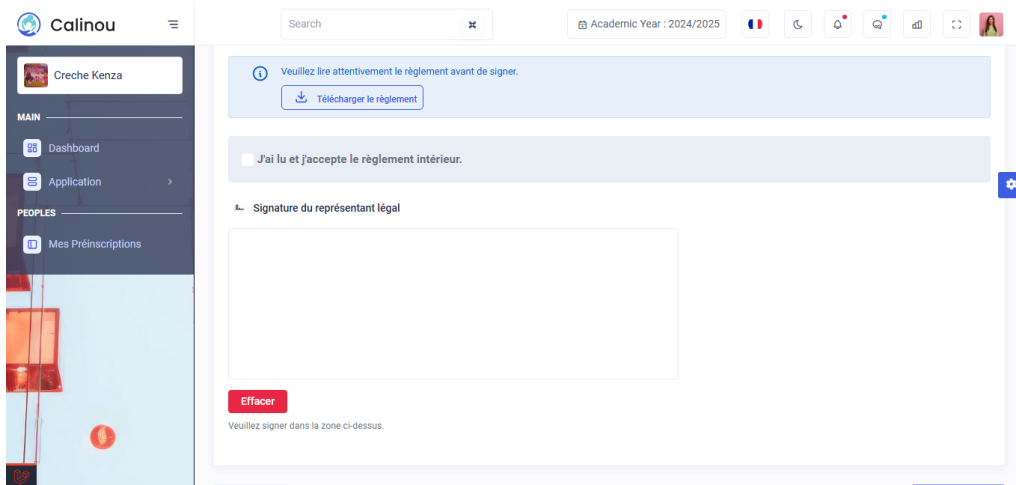


FIGURE 5.23 – Espace de signature électronique du représentant légal

5.3.8 Prestation

L'interface illustrée dans la Figure 5.24 permet d'afficher les prestations liées à un enfant inscrit à la crèche. Elle présente différentes informations comme le type de prestation (par exemple transport ou natation), la période concernée (date de début et date de fin), le montant total ainsi que le montant payé. Cette interface permet au directeur de suivre les prestations attribuées à chaque enfant et d'ajouter des factures si nécessaire.

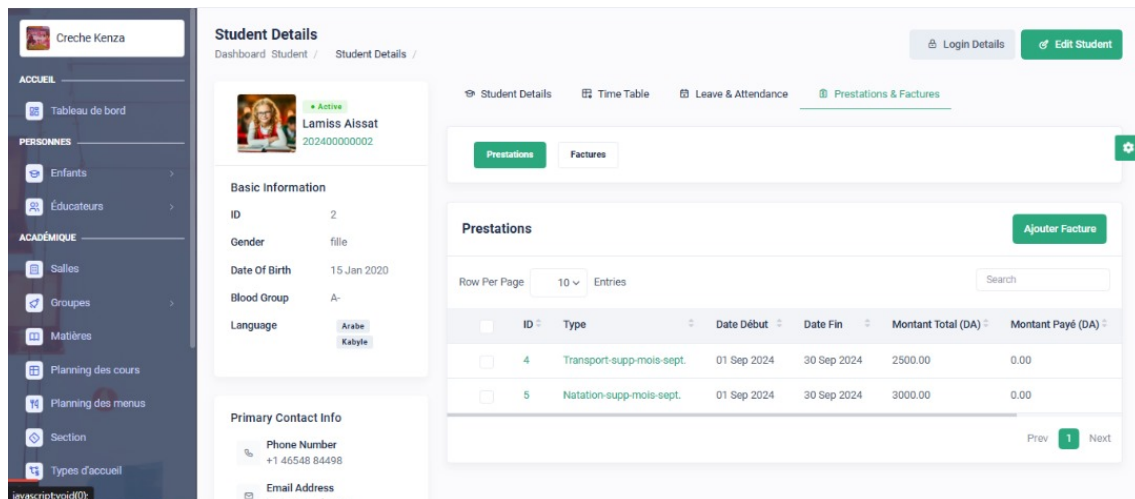


FIGURE 5.24 – Interface de prestation

5.3.9 Facture

L'interface présentée dans la Figure 5.25 permet de consulter les factures associées à un enfant. Elle affiche pour chaque facture un numéro, la date d'émission, la date d'échéance, le montant total, ainsi que le montant déjà payé. Cela permet de suivre les paiements effectués et de savoir si une facture est réglée ou non. L'interface aide aussi le directeur de la crèche à gérer la facturation de manière claire et organisée.

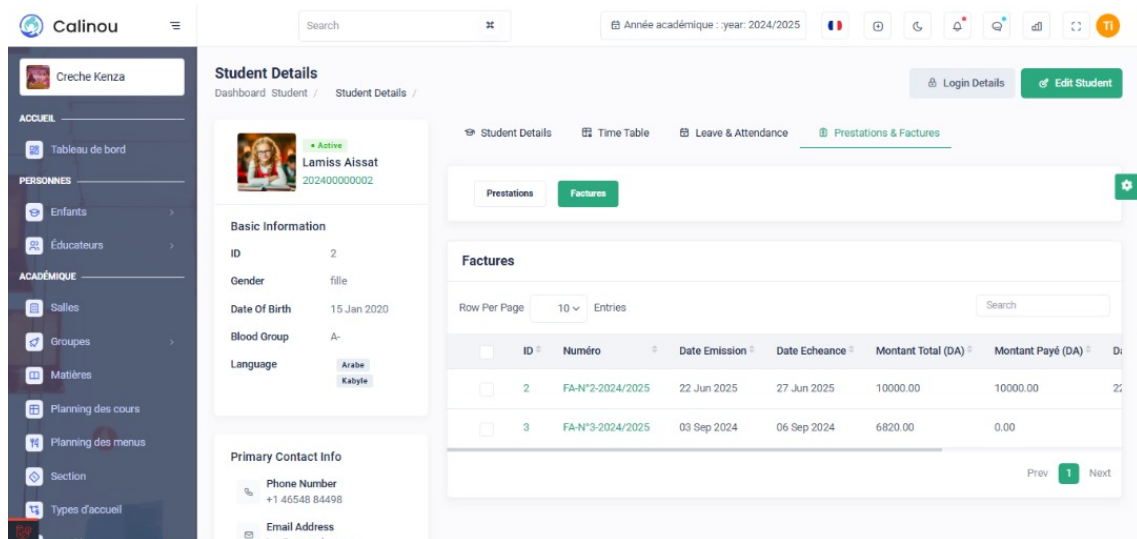


FIGURE 5.25 – Interface de facture

5.3.10 Facture générer

L’interface affichée dans la Figure 5.26 s’ouvre après avoir cliqué sur une facture dans la liste. Elle présente tous les détails liés à la facture sélectionnée : les informations de l’enfant (nom, matricule, date de naissance), le numéro de la facture, la date d’émission, la date d’échéance, ainsi que les coordonnées du parent concerné. On y trouve aussi un tableau avec la description de la prestation facturée, les dates de début et de fin, le montant total, le montant payé et le montant restant. Un résumé à droite indique clairement ce qu’il reste à payer, s’il y a lieu. Cette interface permet d’avoir une vue complète et précise sur chaque facture.

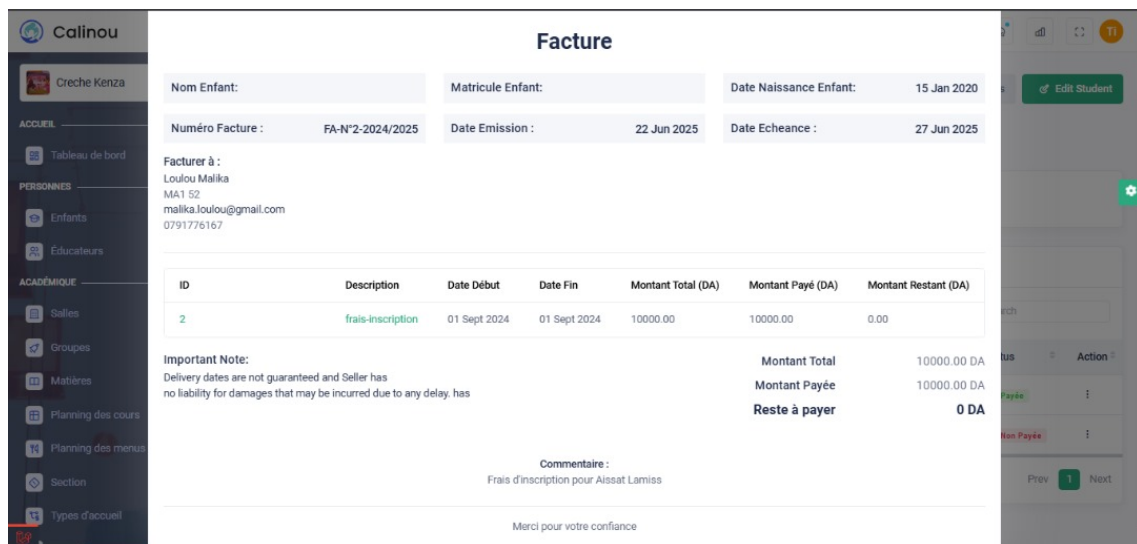


FIGURE 5.26 – Interface de facture générer

5.3.11 Engagement

Ce document d'engagement 5.27 a été automatiquement généré depuis notre plateforme, à partir des informations fournies. Il comprend le nom du premier représentant légal, sa date de naissance, sa relation avec l'enfant, ainsi que le type et le numéro de sa pièce d'identité. Il contient aussi une déclaration indiquant les tuteurs autorisés à récupérer l'enfant en cas d'absence des parents. Un champ est également prévu pour la signature du premier représentant légal.



CRECHE KENZA

Lotissement Akal Aberkane, El-Kseur (w) Béjaïa

Tél: 0560053470 / 0553545222

ENGAGEMENT

Je soussigné(e) Mme Lilia Maiou

Né(e) le 12/05/2000

N.P. Passeport n°123456781

Père ou Mère mère

m'engage à ce que celui-ci soit confié à sa sortie de votre établissement uniquement aux personnes indiquées ci-dessous (inscrire nom et prénom et lien de parenté) :

1. tuteur 1 tuteur 1 (oncle)
2. tuteur 2 tuteur 2 (Tante)
3. tuteur 3 tuteur 3 (Oncle)
- 4.
- 5.

Et m'engage à respecter le règlement intérieur intégralement.

Fait à _____, le 22/05/2025

FIGURE 5.27 – Interface d'engagement

5.3.12 Fiche d'inscription

La figure 5.28 présente un exemple de fiche d'inscription générée automatiquement à partir de notre plateforme, en fonction des données renseignées. Elle regroupe l'engagement des parents, les informations sur l'enfant, les horaires de garde choisis, les éventuels suppléments, ainsi que les renseignements sur les deux responsables légaux. Elle inclut aussi les informations médicales importantes concernant l'enfant. Un champ est également prévu pour la signature du premier représentant légal.



CRECHE KENZA

Lotissement Akal Aberkane, El-Kseur (w) Béjaia
Tél: 0560053470 / 0553545222

FICHE D'INSCRIPTION 2024/2025

ENGAGEMENT DES PARENTS

Je soussigné(e) Mme Maiou Lilia

N.P. Passeport n°123456781

Père/Mère de l'enfant mère

Inscris mon enfant à la crèche Creche Kenza, du 01/09/2024 au 30/06/2025



INFORMATIONS SUR L'ENFANT

Nom de l'enfant Maiou

Prénom de l'enfant Fadila

Date de naissance 05/02/2020

Section Moyenne section (4 ans - 5 ans)

Date d'entrée 01/09/2024

HORAIRES DE GARDE

Jour	Période	Heure d'arrivée	Heure de départ	Tarif (DZ)
Lundi	Journée complète	-	-	410,00

FIGURE 5.28 – Première partie du document de fiche d'inscription générer par la plateforme

Jour	Période	Heure d'arrivée	Heure de départ	Tarif (DZ)
Mardi	Journée complète	-	-	410,00
Mercredi	Journée complète	-	-	410,00
Judi	Journée complète	-	-	410,00
Dimanche	Journée complète	-	-	410,00

SUPPLÉMENTS

Aucun supplément enregistré

RESPONSABLE LÉGAL 1 (mère)

Nom et prénom Maïou Lilia

Profession Educatrice de crèche

Téléphone 0558042893

Adresse tayroudja El hamma amizour bejaia

Email lilia.maïou6@gmail.com

RESPONSABLE LÉGAL 2 (père)

Nom et prénom Zidan Zidine

Profession Footballeur

Téléphone 0549770917

Adresse Tayroudja el hamma amizour bejaia

Email zidane.zidan@gmail.com

FIGURE 5.29 – Deuxième partie du document de fiche d'inscription généré par la plateforme



FIGURE 5.30 – Troisième partie du document de fiche d’inscription généré par la plateforme

5.3.13 Autorisation parentale

La Figure 5.31 montre un exemple de document d’autorisation parentale généré automatiquement à partir de notre plateforme, selon les informations fournies. Il concerne l’autorisation de publier des photos ou vidéos de l’enfant sur les supports de communication, notamment les réseaux sociaux. Ce document contient le nom complet de l’enfant, le nom du responsable légal 1, une zone pour les remarques, les comptes de réseaux sociaux des deux responsables légaux, ainsi que la signature du responsable concerné.



CRECHE KENZA

Lotissement Akal Aberkane, El-Kseur (w) Béjaïa

Tél: 0560053470 / 0553545222

ARTICLE 35 : AUTORISATION PARENTALE

J'autorise la publication de photos/vidéos de mon enfant sur les supports de communication (site web, réseaux sociaux, supports imprimés) de Creche Kenza

Nom complet de l'enfant : Maiou Fadila

Responsable légal 1 (mère) : Maiou Lilia

Remarques particulières : Aucune remarque

Réseaux sociaux de responsable légal 1 :

Facebook: <https://www.facebook.com/seen.solo>

Instagram: <https://www.instagram.com/seen.solo>

Réseaux sociaux de responsable légal 2 :

Facebook: <https://www.facebook.com/zindine.zidan>

Instagram: <https://www.instagram.com/zindine.zidan>

Fait à _____, le 2025-05-22 12:15:26

Signature du représentant légal

FIGURE 5.31 – Document de l'autorisation parentale généré par la plateforme

5.3.14 Messagerie

La messagerie intégrée à la plateforme permet aux différents utilisateurs de communiquer facilement entre eux. Le directeur peut échanger avec l'ensemble des parents ainsi qu'avec tous les éducateurs. De leur côté, les parents peuvent uniquement discuter avec les éducateurs qui encadrent leur propre enfant et le directeur.

Ce système garantit une communication ciblée, tout en respectant la structure de l'établissement. La Figure 5.32 illustre l'interface de cette messagerie.

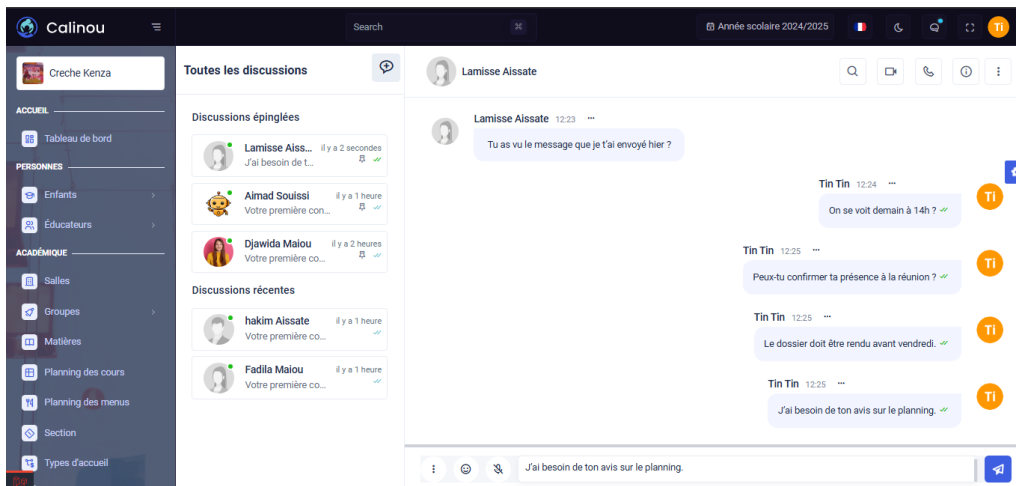


FIGURE 5.32 – Interface de la messagerie

5.4 Conclusion

La phase de réalisation nous a permis de transformer nos idées en une application réelle et fonctionnelle, grâce à l'utilisation d'outils modernes et efficaces. Les interfaces créées répondent aux besoins du projet et offrent une utilisation simple et agréable. Cette étape a aussi permis de confirmer que les choix techniques faits au départ étaient adaptés à la mise en œuvre de l'application.

Conclusion et perspectives

Dans ce mémoire, nous avons exploré le domaine de la garde d'enfants en nous concentrant sur la gestion interne d'une crèche. Pour cela, nous avons conçu et développé une application web dédiée à une seule structure, Kenza Crèche, en utilisant notre système appelé **Calinou**.

L'objectif principal de ce projet était de proposer une solution simple, claire et efficace pour faciliter le travail quotidien au sein de la crèche, notamment à travers la gestion des enfants, du personnel, des groupes et des inscriptions. Ce système permet de centraliser les informations et d'améliorer l'organisation interne.

Nous avons suivi une démarche de développement agile en utilisant la méthode Scrum, ce qui nous a permis de planifier nos tâches de manière structurée et de faire évoluer progressivement notre solution en fonction des besoins identifiés.

Ce projet nous a permis de mettre en pratique nos connaissances en développement logiciel et d'acquérir de nouvelles compétences utiles pour notre parcours professionnel.

Cette expérience a été enrichissante à la fois sur le plan technique et humain. Elle nous a permis de répondre à un besoin concret et d'apporter une première solution numérique pour améliorer la gestion d'une crèche dans un contexte réel.

Au vu des retours positifs et des possibilités d'évolution qu'offre le système Calinou, il apparaît pertinent d'envisager des pistes d'amélioration et d'extension, dans le but de répondre à des besoins plus larges et d'accompagner d'autres structures similaires.

Afin de prolonger ce travail et d'enrichir davantage le système Calinou, plusieurs évolutions peuvent être envisagées. Il serait tout d'abord pertinent de développer un **portail multi-crèches**, permettant à plusieurs établissements de fonctionner sur la même plateforme tout en conservant une gestion autonome. Ensuite, le **développement d'une application mobile** offrirait une meilleure accessibilité et une utilisation plus pratique pour les utilisateurs au quotidien. Enfin, l'intégration d'un **chatbot** pourrait faciliter la communication avec les utilisateurs et automatiser certaines tâches récurrentes.

Bibliographie

- [1] AQUILAPP. *Définition d'une application mobile*. consulté le 06 novembre 2024. URL : <https://www.aquilapp.fr/ressources/projet-mobile/quest-ce-quune-application-mobile#definition-dune-application-mobile>.
- [2] AQUILAPP. *L'application native*. consulté le 06 novembre 2024. URL : <https://www.aquilapp.fr/ressources/projet-mobile/quest-ce-quune-application-mobile#lapplication-native>.
- [3] AQUILAPP. *Les applications hybrides*. consulté le 06 novembre 2024. URL : <https://www.aquilapp.fr/ressources/projet-mobile/quest-ce-quune-application-mobile#les-applications-hybrides>.
- [4] AQUILAPP. *Les Progressive Web Apps*. consulté le 06 novembre 2024. URL : <https://www.aquilapp.fr/ressources/projet-mobile/quest-ce-quune-application-mobile#les-progressive-web-apps>.
- [5] TESTLIFY. *Desktop application*. consulté le 06 novembre 2024. URL : <https://testlify.com/tech-glossary/desktop-application/>.
- [6] Jainli CONSULTING. *Application web statique*. consulté le 06 novembre 2024. URL : <https://www.jainliconsulting.com/les-5-types-dapplications-web/#application-web-statique>.
- [7] Jainli CONSULTING. *Application web dynamique*. consulté le 06 novembre 2024. URL : <https://www.jainliconsulting.com/les-5-types-dapplications-web/#application-web-dynamique>.
- [8] Jainli CONSULTING. *Application web e-commerce*. consulté le 06 novembre 2024. URL : <https://www.jainliconsulting.com/les-5-types-dapplications-web/#application-web-e-commerce>.
- [9] Jainli CONSULTING. *Application web portail*. consulté le 06 novembre 2024. URL : <https://www.jainliconsulting.com/les-5-types-dapplications-web/#application-web-portail>.
- [10] Alain WISNER. *Réflexions sur l'ergonomie (1962-1995)*. Consulté le 24 novembre 2024. Toulouse : Octares Éditions, 1995. ISBN : 2-906769-24-X.
- [11] Léa RABATEL. *Les règles de base pour un design de site web harmonieux*. consulté le 24 novembre 2024. URL : <https://learabatel.fr/blog/ergonomie-web-10-regles-de-base-pour-un-design-de-site-web-harmonieux/>.
- [12] YUMENS. *Quels sont les objectifs de l'ergonomie web ?* consulté le 24 novembre 2024. URL : <https://www.yumens.fr/expertise/ux-design/ergonomie/>.
- [13] CRÉA. *Les grandes règles d'ergonomie de votre site web*. Consulté le 24 novembre 2024. Nov. 2016. URL : <https://www.nobilito.fr/club/les-grandes-regles-dergonomie-de-votre-site-web/>.
- [14] ROCKET. *What are the types of UI/UX designers ?* Consulté le 24 novembre 2024. Mars 2024. URL : <https://www.getrocket.com/post/ui-ux-designers>.
- [15] ESOKIA. *UX / UI : qu'est-ce que ça signifie ?* Consulté le 24 novembre 2024. URL : <https://esokia.com/fr/blog/ux-ui-quest-ce-que-ca-signifie>.

- [16] L'équipe Les Parents ZENS. *Qu'est-ce qu'une crèche ?* Consulté le 10 novembre 2024. Sept. 2022. URL : <https://www.lesparentszens.com/blog-famille/creche-definition>.
- [17] Centre Européen de FORMATION. *Les compétences et qualités pour un travail avec les jeunes enfants.* Consulté le 10 novembre 2024. URL : <https://www.centre-europeen-formation.fr/metiers/petite-enfance/>.
- [18] L'équipe Les Parents ZENS. *Les différents types de crèches.* Consulté le 10 novembre 2024. Sept. 2022. URL : <https://www.lesparentszens.com/blog-famille/creche-definition>.
- [19] *CleverNursery*. Consulté le 15 novembre 2024. 2024. URL : <https://blog.clevernursery.com/un-aperçu-de-clever-nursery-revolutionnez-la-gestion-des-creches>.
- [20] *creacreche*. Consulté le 15 novembre 2024. 2024. URL : <https://creacreche.createdz.com/>.
- [21] *lutinrouge*. Consulté le 15 novembre 2024. 2024. URL : <https://www.lutinrouge.eu/>.
- [22] *procare*. Consulté le 15 novembre 2024. 2024. URL : <https://www.procaresoftware.com/>.
- [23] NPOLIVE. *Approche systémique des systèmes informatiques.* Consulté le 15 novembre 2024. Oct. 2016. URL : <https://npolive.net/2016/10/31/approche-systemique-des-systemes-informatiques/>.
- [24] TECHNO-SCIENCE.NET. *Merise (informatique) : méthode d'analyse, de conception et de gestion de projet.* Consulté le 26 juin 2025. URL : <https://www.techno-science.net/glossaire-definition/Merise-informatique.html>.
- [25] Ivar JACOBSON et al. *Object-Oriented Software Engineering : A Use Case Driven Approach.* Consulté le 28 novembre 2024. Addison-Wesley, 1992. ISBN : 978-0-201-54435-0.
- [26] TECHLIB. *Définition RUP (Rational Unified Process).* Consulté le 28 novembre 2024. URL : <https://techlib.fr/definition/rup.html>.
- [27] Thibault BAHEUX. *Méthode agile : Principes de fonctionnement.* Mise à jour le 4 novembre 2022, consulté le 01 décembre 2024. URL : <https://www.reussirsesprojets.com/methode-agile/>.
- [28] Par Max REHKOPF. *Agile et Scrum. Comment choisir la méthodologie qui répond le mieux à vos besoins.* Consulté le 01 décembre 2024. URL : <https://www.atlassian.com/fr/agile/scrum/agile-vs-scrum>.
- [29] TULEAP. *What is scrum agile method ?* Consulté le 03 décembre 2024. URL : <https://www.tuleap.org/agile/agile-scrum-in-10-minutes>.
- [30] AUDREY DEGUARA. *Piliers Scrum : Comment ils contribuent à l'amélioration continue dans un projet agile.* Consulté le 01 décembre 2024. URL : <https://blog-gestion-de-projet.com/piliers-scrum/>.
- [31] SCRUM.ORG. *What is a Product Owner ?* Consulté le 01 décembre 2024. URL : <https://www.scrum.org/resources/what-is-a-product-owner?>.
- [32] SCRUM.ORG. *What is a Scrum Master ?* Consulté le 01 décembre 2024. URL : <https://www.scrum.org/resources/what-is-a-scrum-master?>.
- [33] SCRUM.ORG. *What is a Scrum Team ?* Consulté le 01 décembre 2024. URL : <https://www.scrum.org/resources/what-scrum-team?>.
- [34] Max REHKOPF. *Que sont les sprints dans la gestion de projet ?* Consulté le 01 décembre 2024. URL : <https://www.atlassian.com/fr/agile/scrum/sprints>.
- [35] Ken SCHWABER et Jeff SUTHERLAND. *The Scrum Guide™.* Consulté le 01 décembre 2024. 2020. URL : <https://www.scrumguides.org>.
- [36] Pascal ROQUES. *UML 2 : Modéliser une application web.* 4^e éd. Consulté le 20 novembre 2024. Eyrolles, oct. 2008. ISBN : 978-2-212-12389-0.
- [37] Volatine GREEN. *Présentation des 14 types et exemples de diagrammes UML.* Consulté le 20 novembre 2024. Mai 2021. URL : https://gitmind.com/fr/types-diagrammes-uml.html#_11.
- [38] ADVENTY. *MVC - Adventy.* Consulté le 15 novembre 2024. URL : <https://adventy.org/fr/mvc>.

- [39] GEEKSFORGEEKS. *MVC Framework - Introduction*. Consulté le 15 novembre 2024. Sept. 2022. URL : <https://www.geeksforgeeks.org/mvc-framework-introduction>.
- [40] ORACLE. *What Is a Relational Database ?* Consulté le 03 décembre 2024. June 18, 2021. URL : <https://www.oracle.com/database/what-is-a-relational-database/>.
- [41] Ian CALLAHAN. *Laravel*. Consulté le 30 novembre 2024. URL : <https://laravel.com/>.
- [42] LARAVEL. *Logo laravel*. Consulté le 30 novembre 2024. URL : <https://download.logo.wine/logo/Laravel/Laravel-Logo.wine.png>.
- [43] V-LABS. *C'est quoi Bootstrap ?* Consulté le 30 novembre 2024. URL : <https://www.v-labs.fr/glossaire/bootstrap/>.
- [44] CLEANPNG. *Logo Bootstrap*. Consulté le 30 novembre 2024. URL : <https://download.logo.wine/logo/Laravel/Laravel-Logo.wine.png>.
- [45] Visual Studio CODE. *VS Code*. Consulté le 30 novembre 2024. URL : <https://code.visualstudio.com/>.
- [46] Visual Studio CODE. *VS Code*. Consulté le 30 novembre 2024. URL : <https://code.visualstudio.com/brand>.
- [47] Romain BOURDON. *WampServer,plate-forme de développement Web sous Windows*. Consulté le 30 novembre 2024. URL : <https://www.wampserver.com/>.
- [48] LOGOTYPES. *WampServer Logo*. Consulté le 30 novembre 2024. URL : <https://logosandtypes.com/alphabet/letter-w/wampserver/>.
- [49] Romain BOURDON. *WampServer,plate-forme de développement Web sous Windows*. Consulté le 30 novembre 2024. URL : <https://github.com>.
- [50] Dave GANDY. *GitHub Logo*. Consulté le 30 novembre 2024. URL : https://www.flaticon.com/free-icon/github-logo_25231.
- [51] Romain BOURDON. *HTML : HyperText Markup Language*. Consulté le 02 décembre 2024. URL : <https://developer.mozilla.org/en-US/docs/Web/HTML>.
- [52] CLEANPNG. *HTML5 Logo for Web Development*. Consulté le 02 décembre 2024. URL : <https://www.cleanpng.com/png-html5-logo-8265088/>.
- [53] Romain BOURDON. *CSS : Cascading Style Sheets*. Consulté le 02 décembre 2024. URL : <https://developer.mozilla.org/en-US/docs/Web/CSS>.
- [54] FREEPIK. *CSS Logo*. Consulté le 02 décembre 2024. URL : https://www.flaticon.com/fr/icone-gratuite/css-3_5968242?term=logo+css+3&page=1&position=2&origin=search&related_id=5968242.
- [55] Romain BOURDON. *JavaScript*. Consulté le 02 décembre 2024. URL : <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- [56] Pixel PERFECT. *JavaScript Logo*. Consulté le 02 décembre 2024. URL : https://www.flaticon.com/fr/icone-gratuite/script-java_1199124?term=logo+javascript&page=1&position=4&origin=search&related_id=1199124.
- [57] PHP. *Qu'est-ce que PHP et que peut-il faire ? Qu'est ce que PHP ?* Consulté le 02 décembre 2024. URL : <https://www.php.net/manual/fr/introduction.php>.
- [58] PHP. *Php Logo*. Consulté le 02 décembre 2024. URL : <https://www.php.net/images/logos/new-php-logo.png>.
- [59] La Rédaction JDN. *SQL (Structured Query Language) : définition, traduction et acteurs*. Consulté le 02 décembre 2024. Jan. 2019. URL : <https://www.journaldunet.fr/web-tech/guide-du-big-data/1203603-sql-structured-query-language-definition-traduction-et-acteurs/>.
- [60] SQL. *SQL Logo*. Consulté le 02 décembre 2024. URL : <https://www.sqlfr.com/>.
- [61] Visual PARADIGM. *Finding a Free online drawing software ?* Consulté le 03 décembre 2024. URL : <https://online.visual-paradigm.com/fr/diagrams/solutions/free-visual-paradigm-online?>

- [62] SQL. *Visual Paradigm Logo*. Consulté le 03 décembre 2024. URL : <https://www.pngegg.com/en/search?q=visual+Paradigm>.
- [63] MYSQL. *Chapter 1 General Information*. Consulté le 03 décembre 2024. URL : <https://dev.mysql.com/doc/refman/8.0/en/tutorial.html>.
- [64] FREEPIK. *MySQL Logo*. Consulté le 03 décembre 2024. URL : https://www.flaticon.com/kr/free-icon/mysql_919836.
- [65] GITLAB. *Qu'est-ce que Git ? – Système de gestion de versions distribué*. Consulté le 03 décembre 2024. URL : <https://about.gitlab.com/fr-fr/topics/version-control/what-is-git-version-control/>.
- [66] ICONS. *Git Logo*. Consulté le 03 décembre 2024. URL : <https://icones8.fr/icons/set/git-logo>.
- [67] COMPOSER. *Composer - Dependency Manager for PHP*. Consulté le 03 décembre 2024. URL : <https://getcomposer.org/doc/00-intro.md>.
- [68] VECTOR ICONS. *Composer Logo*. Consulté le 03 décembre 2024. URL : <https://iamvector.com/all-icons?search=composer>.

Annexes

Annexe A : Dictionnaire de Données

Le dictionnaire de données présente de manière détaillée les entités de la base de données, leurs attributs, la signification de chaque attribut, ainsi que leur type. Ce tableau 1 permet d'avoir une vision claire de la structure des données manipulées par le système et sert de référence tout au long du développement.

TABLE 1 – Dictionnaire de données des entités du système

Entité	Attribut	Définition Attribut	Type
Parent	id_parent	Identifiant du parent	Numérique
	est_profilComple	Statut du profil	Booléen
	lien_parental	Type de lien parental (mère, père, tuteur)	Alphabétique
	est_firstRepresentant	Est le représentant principal	Booléen
	profession	Profession du parent	Alphabétique
	relation	Relation avec l'enfant	Alphabétique
Tuteur	id_tuteur	Identifiant du tuteur	Numérique
	nom	Nom du tuteur	Alphabétique
	prenom	Prénom du tuteur	Alphabétique
	num_tel	Numéro de téléphone du tuteur	Numérique
	adresse	Adresse du tuteur	Alphabétique
	photo_path	Chemin de la photo du tuteur	Alphanumérique
	nin	Numéro d'identification nationale du tuteur	Alphanumérique
Utilisateur	id_user	Identifiant de l'utilisateur	Numérique
	nom	Nom de l'utilisateur	Alphabétique
	prenom	Prénom de l'utilisateur	Alphabétique
	date_naissance	Date de naissance	Date
	adresse	Adresse de l'utilisateur	Alphabétique
	telephone	Numéro de téléphone de l'utilisateur	Numérique
	email	Adresse email	Alphanumérique
	etat_civil	État civil de l'utilisateur(marie, célibataire, divorce, veuf)	Alphabétique
	role	Rôle dans le système	Alphabétique

Suite page suivante

Continuation du tableau 1			
Entité	Attribut	Définition Attribut	Type
	photo_path	Chemin de la photo de l'utilisateur	Alphanumérique
	type_carte	Type de carte (carte d'identité, passeport, permis de conduite)	Alphabétique
	numero_carte	Numéro de la carte d'identité	Alphanumérique
	mot_de_passe	Mot de passe de l'utilisateur	Alphanumérique
	nin	Numéro d'identification nationale de l'utilisateur	Alphanumérique
	is_actif	Utilisateur actif ou non	Booléen
	insta_url	Lien Instagram	Alphanumérique
	fcb_url	Lien Facebook	Alphanumérique
	sexe	Sexe de l'utilisateur	Alphabétique
	date_expiration	Date d'expiration de la carte	Date
Employe	id_employe	Identifiant de l'employé	Numérique
	poste	Poste occupé	Alphabétique
	date_embauche	Date d'embauche	Date
	type_contrat	Type de contrat	Alphabétique
	nbr_annee_experience	Années d'expérience	Numérique
	date_sortie	Date de sortie	Date
Educateur	id_educateur	Identifiant de l'éducateur	Numérique
Directeur	id_directeur	Identifiant du directeur	Numérique
Crèche	id_creche	Identifiant de la crèche	Numérique
	nom_creche	Nom de la crèche	Alphabétique
	adresse_creche	Adresse de la crèche	Alphabétique
	tel1_creche	Téléphone principal	Numérique
	tel2_creche	Téléphone secondaire	Numérique
	email_creche	Email de la crèche	Alphanumérique
	heure_ouverture	Heure d'ouverture de la crèche	Heure
	heure_fermeture	Heure de fermeture de la crèche	Heure
	horaires_debut_sortie	Heure début sortie des enfants	Heure
	joursFermeture	Jours de fermeture de ma crèche	Alphabétique
	num_agrement	Numéro d'agrément de la crèche	Alphanumérique
	capacité_accueil	Capacité d'accueil de la crèche en nombre d'enfants	Numérique
	fcb_url	URL Facebook de la crèche	Alphanumérique
	insta_url	URL Instagram de la crèche	Alphanumérique
logo_path	Chemin vers le logo	Alphanumérique	
Enfant	id_enfant	Identifiant de l'enfant	Numérique
	nom	Nom de l'enfant	Alphabétique
	prenom	Prénom de l'enfant	Alphabétique
	matricule	Matricule unique	Alphanumérique
	photo_path	Chemin de la photo de l'enfant	Alphanumérique
	sexe	Sexe de l'enfant	Alphabétique
	date_naissance	Date de naissance de l'enfant	Date
	lieu_naissance	Lieu de naissance de l'enfant	Alphabétique

Suite page suivante

Continuation du tableau 1			
Entité	Attribut	Définition Attribut	Type
	groupe_sanguin	Groupe sanguin e l'enfant(A+, A-, B+, B-, AB-, AB+, O-, O+)	Alphabétique
	adresse	Adresse de l'enfant	Alphabétique
	langues_parlees	Langues parlées	Alphabétique
	date_sortie	Date de sortie de la crèche	Date
	active	l'enfant est toujours actif ou non	Booléen
	est_handicape	Présence d'un handicap	Booléen
	details_handicap	Détails du handicap	Alphabétique
	a_regime_alimentaire	Régime alimentaire spécifique	Booléen
	details_regime_alimentaire	Détails régime alimentaire	Alphabétique
	a_allergies	Allergies connues	Booléen
	details_allergies	Détails allergies	Alphabétique
	a_maladie_chronique	Maladie chronique	Booléen
	details_maladies	Détails maladies chroniques	Alphabétique
	a_maladies_passées	Maladies passées	Booléen
	details_maladies_passées	Détails maladies passées	Alphabétique
numero_act_naissance	Numéro acte de naissance	Alphanumérique	
Presence_enfant	id_presence_enfant	Identifiant pointage enfant	Numérique
	date_presence	date de la présence	Date
	heure_presence	heure de la présence	Heure
	status	Indique la présence de l'enfant (présent, en retard, absent)	Alphabétique
	note	Commentaire ou observation liée au statut de présence	Alphabétique
Presence_educateur	id_presence_educateur	Identifiant pointage de l'éducateur	Numérique
	date_presence	date de la présence	Date
	heure_presence	heure de la présence	Heure
	status	Indique la présence de l'éducateur (présent, en retard, absent)	Alphabétique
	note	Commentaire ou observation liée au statut de présence	Alphabétique
Evaluation	id_evaluation	Identifiant de l'évaluation	Numérique
	note	note de l'évaluation	Alphabétique
	remarque	Commentaire sur l'évaluation	Alphabétique
Message	id_message	Identifiant du message	Numérique
	body	Contenu du message	Alphabétique
	read	Statut lu/non lu	Booléen
	is_visible_sender	Visibilité du message pour l'expéditeur	Booléen
	is_visible_receiver	Visibilité du message pour le destinataire	Booléen
Conversation	id_conversation	Identifiant de la conversation	Numérique

Suite page suivante

Continuation du tableau 1			
Entité	Attribut	Définition Attribut	Type
	last_time_message	Date et heure du dernier message	DateTime
	sender_is_pinned	Conversation épinglée par l'expéditeur	Booléen
	receiver_is_pinned	Conversation épinglée par le destinataire	Booléen
	sender_deleted	Suppression de la conversation par l'expéditeur	Booléen
	receiver_deleted	Suppression de la conversation par le destinataire	Booléen
Actualité	id_actualite	Identifiant de l'actualité	Numérique
	contenu	Contenu textuel de l'actualité	Alphabétique
	titre	Titre de l'actualité	Alphabétique
	image	Chemin de l'image associée	Alphanumérique
	date_publication	Date de publication de l'actualité	Date
Galerie	id_galerie	Identifiant de la galerie	Numérique
	path	Chemin des images	Alphanumérique
	est_visible	Visibilité publique (oui/non)	Booléen
Planning_Cour	id_planning	Identifiant du planning	Numérique
Séance	id_seance	Identifiant de la séance	Numérique
	heure_debut	Heure de début de la séance	Heure
	heure_fin	Heure de fin de la séance	Heure
	jour	Jour de la séance	Alphabétique
Matière	id_matiere	Identifiant de la matière	Numérique
	nom	Nom de la matière	Alphabétique
	description	Description de la matière	Alphabétique
Groupe	id_groupe	Identifiant du groupe	Numérique
	nom_groupe	Nom du groupe	Alphabétique
	description	Description du groupe	Alphabétique
	mois_min	Âge minimum en mois	Numérique
	mois_max	Âge maximum en mois	Numérique
Affectation	id_affectation	Identifiant de l'affectation	Numérique
	date_affectation	Date de début d'affectation des éducateurs	Date
	date_fin_affectation	Date de fin d'affectation des éducateurs	Date
	status	Statut de l'affectation des éducateurs(actif, en congé, remplaçant, terminé)	Alphabétique
Salle	id_salle	Identifiant de la salle	Numérique
	numero_salle	Numéro de la salle	Numérique
	capacite	Capacité de la salle	Numérique
	numero_etage	Numéro de l'étage	Numérique
Repas	id_repas	Identifiant du repas	Numérique
	nom_repas	Nom du repas	Alphabétique
	description	Description du repas	Alphabétique

Suite page suivante

Continuation du tableau 1			
Entité	Attribut	Définition Attribut	Type
	type_repas	Type de repas (entrée, plat principale, dessert)	Alphabétique
Jour	id_jour	Identifiant du menu	Numérique
	nom_jour	Indique le jour de la semaine pour le repas	Alphabétique
Jour_Repas	/	/	/
Document	id_document	Identifiant du document	Numérique
	titre	titre du document	Alphabétique
	type_document	Type du document (ex : cv, diplôme, certificat médical physique, certificat médical psychiatrique, photo d'identité, pièce d'identité, règlement, règlement signé, autorisation parentale, engagement, fiche d'inscription, carnet de santé, acte de naissance et la fiche familiale)	Alphabétique
	fichier_path	Chemin du fichier	Alphanumérique
	documentable_type	Entité associée au document (employé ou enfant)	Alphabétique
	documentable_id	Identifiant de l'entité associée au document	Numérique
Préinscription	id_preinscription	Identifiant de la préinscription	Numérique
	etat	État de la préinscription	Alphabétique
	commentaire	Commentaire éventuel	Alphabétique
	est_inscrit_definitivement	Indique si l'inscription est définitive (oui/non)	Booléen
	date_entree	Date d'entrée de l'enfant	Date
	date_creation	Date de création de la préinscription	Date
Préinscription- _Abonnement	jour	Jour concerné par l'abonnement	Alphabétique
	matin	Présence matin (oui/non)	Booléen
	apres_midi	Présence après-midi (oui/non)	Booléen
	avec_repas	Présence repas (oui/non)	Booléen
	heure_arrivee	Heure d'arrivée	Heure
	heure_depart	Heure de départ	Heure
Dossier	id_dossier	Identifiant du dossier	Numérique
	etat	État du dossier (complet, incomplet, reçu, non reçu)	Alphabétique
	date_depot	Date de dépôt	Date
	date_validation	Date de validation	Date
	numero_dossier	Numéro du dossier	Alphanumérique
	commentaire	Commentaire associé	Alphabétique

Suite page suivante

Continuation du tableau 1			
Entité	Attribut	Définition Attribut	Type
	remarque	Remarque éventuelle	Alphabétique
	autorisation_publicationPhotos	Autorisation publication des photos des enfants (oui/non)	Booléen
	date_creation_dossier	Date de création du dossier d'inscription	Date
Dossier_Tuteur	relation	Relation du tuteur avec l'enfant	Alphabétique
Saison	id_saison	Identifiant de la saison	Numérique
	nom_saison	Nom de la saison (ex : 2024-2025)	Alphabétique
	date_debut	Date de début de saison	Date
	date_fin	Date de fin de saison	Date
	active	Indique si saison active (oui/non)	Booléen
Supplement	id_supplement	Identifiant du supplément	Numérique
	nom_supplement	Nom du supplément	Alphabétique
	description	Description du supplément	Alphabétique
Supplement- _Saison	montant	Montant du supplément à payer par mois	Numérique
	active	Statut actif (oui/non)	Booléen
Supplement- _Dossier	/	/	/
Section	id_section	Identifiant de la section	Numérique
	nom_section	Nom de la section	Alphabétique
	description	Description de la section	Alphabétique
	age_min	Âge minimum requis	Numérique
	age_max	Âge maximum autorisé	Numérique
Catégorie Abonnement	id_cat_abon	Identifiant de la catégorie d'abonnement	Numérique
	description	Description de la catégorie	Alphabétique
	type	Type de l'abonnement (journal, demi journée)	Alphabétique
	avec_repas	Si repas inclus (oui/non)	Booléen
Section_Abon- nement	active	Statut actif de l'abonnement	Booléen
	montant	Montant de l'abonnement à payer	Numérique
Saison_Section	frais_inscription	Frais d'inscription pour de la section	Numérique
Prestation	id_prestation	Identifiant de la prestation	Numérique
	montant_total	Montant total de la prestation	Numérique
	montant_restant	Montant restant à payer	Numérique
	montant_paye	Montant déjà payé	Numérique
	date_debut_periode	Début de la période	Date
	date_fin_periode	Fin de la période	Date
	description	description de la prestation	Alphabétique
Facture	id_facture	Identifiant de la facture	Numérique
	numero_facture	Numéro de la facture	Numérique
	date_emission	Date d'émission de la facture	Date

Suite page suivante

Continuation du tableau 1			
Entité	Attribut	Définition Attribut	Type
	date_echeance	Date d'échéance de paiement	Date
	statut	Statut de la facture (payée, non payé , partiellement payé ,annulé)	Alphabétique
	montant_total	Montant total dû	Numérique
	commentaire	Commentaire lié à la facture	Alphabétique
	montant_paye	Montant payé	Numérique
	date_paiement	Date de paiement	Date
Evenement	id_event	Identifiant de l'événement	Numérique
	titre_event	Titre de l'événement	Alphabétique
	type_event	Type d'événement (ex : fête, rendez-vous, vacance, sortie, réunion, autre)	Alphabétique
	date_debut	Date de début	Date
	date_fin	Date de fin	Date
	lieu	Lieu de l'événement	Alphabétique
	description	Description de l'évènement	Alphabétique
	fichier_path	chemin de fichier	Alphanumérique
status	Statut de l'événement (terminé, annulé, en cours, planifié)	Alphabétique	
Participant	id_participant	Identifiant du participant	Numérique
	participantable_id	Identifiant de l'entité liée au participant	Numérique
	participantable_type	Type d'entité (Parent, Éducateur ou Enfant)	Alphabétique
Évènement- _Participant	est_confirme	Indique si le participant a confirmé (oui/non)	Booléen
Fin du tableau 1			

Résumé

La gestion quotidienne d'une crèche implique de nombreuses tâches administratives et organisationnelles qui peuvent rapidement devenir complexes sans un outil adapté. C'est dans ce contexte que nous avons entrepris la réalisation d'une application web pour la crèche Kenza Crèche, afin de faciliter son fonctionnement interne et de moderniser sa gestion.

Cette application centralise les informations relatives aux enfants, aux employés, aux inscriptions et à d'autres aspects essentiels du suivi administratif. Elle permet également d'améliorer la communication entre le personnel et les parents, en offrant un accès simple et rapide aux données pertinentes. Le développement a été mené selon la méthode agile Scrum, qui nous a permis d'adopter une démarche itérative et collaborative, en intégrant progressivement les besoins de l'utilisateur final. Le résultat est une solution numérique complète, pensée pour alléger la charge de travail des responsables et offrir un meilleur service aux familles.

Ce projet constitue une réponse concrète et moderne aux besoins de gestion d'un établissement préscolaire, en s'adaptant aux réalités du terrain et aux attentes des utilisateurs.

Mots-clés : application web, crèche, gestion, Scrum, communication.

Abstract

The daily management of a daycare center involves numerous administrative and organizational tasks that can quickly become complex without a suitable tool. In this context, we developed a web application for Kenza Crèche, with the goal of facilitating its internal operations and modernizing its overall management.

This application centralizes essential information related to children, staff, registrations, and other key administrative activities. It also enhances communication between staff and parents by providing quick and easy access to relevant data. The development was carried out using the agile Scrum methodology, which enabled us to adopt an iterative and collaborative approach, gradually integrating user needs. The result is a complete digital solution designed to reduce the workload of staff and offer better service to families.

This project provides a practical and modern response to the management needs of a preschool institution, adapted to real-world conditions and user expectations.

Keywords : web application, daycare, management, Scrum, communication.