

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



MÉMOIRE DE FIN DE CYCLE

En vue de l'obtention du diplôme de Master Professionnel en Informatique
Option : Génie Logiciel

THÈME

**Conception et Réalisation d'une Application Web
pour la Gestion des Formations**
Cas d'étude : Cevital Agro-Industrie

Réalisé par :

ADJTOUTAH Sara
BELAID Siham

Devant le jury composé de :

Président : Dr. Y.BENNAI
Encadrant : Pr. K.AMROUN
Co-encadrante : Dr. Z.TAHAKOURT
Examineur : Dr. R.OUZEGGANE
Examineur : Dr. H.KHALED

Année Universitaire 2024 – 2025

Remerciements

*Nous tenons à la fin de ce travail à remercier **ALLAH** le Tout-Puissant de nous avoir donné la santé et de nous avoir permis de mener à terme ce projet.*

*En premier lieu, nous souhaitons exprimer notre profonde gratitude à nos encadrants, **M. Amroun Kamel** et **Mme Tahakourt Zineb**, pour leur disponibilité, leur patience, et la qualité de leur encadrement. Leur accompagnement constant, leurs conseils avisés et leur bienveillance ont grandement contribué à la réalisation de ce travail.*

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

*Nous remercions aussi chaleureusement **Mlle Z.Belkacemi** et toutes l'équipe CEVITAL, pour leur soutien et leur précieuse collaboration durant notre stage.*

Nous exprimons notre plus grande gratitude à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

*À ma chère famille,
A la femme qui a souffert sans me laisser souffrir, A toi ma très chère
mère quoi que je fasse ou que je dise, je ne saurai point te remercier
comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta
présence à mes côtés a toujours été ma source de force pour affronter les
différents obstacles.*

*A mes grandes sœurs bien-aimées **Hanane**, **Ouarda** et **Lydia**, vous
avez toujours été là, présentes, protectrices, attentives.
En tant que cadette, j'ai grandi sous vos regards bienveillants et vos
conseils précieux, chacune à sa manière.
Merci d'avoir cru en moi .*

*À ma précieuse binôme **Siham**, ta rigueur, ton engagement et ta
bienveillance ont fait de cette collaboration une belle aventure humaine.
Merci pour ton écoute, ton sérieux, et ta constance.*

*À mes chers amis **Achour**, **Louiza** et **Haoua**, Votre amitié m'a été
précieuse tout au long de ce parcours. Merci pour vos encouragements,
votre soutien inconditionnel, vos éclats de rire au bon moment et votre
présence toujours réconfortante.
Vous avez contribué à alléger les moments de stress et à rendre ce voyage
plus lumineux.*

Dédicaces

Je dédie ce travail à tous ceux qui ont cru en moi et m'ont soutenu tout au long de ce parcours.

À mes chers parents,

Aucune dédicace ne saurait exprimer pleinement mon respect, mon amour éternel et ma profonde reconnaissance pour les sacrifices que vous avez consentis pour mon éducation et mon bien-être.

Je vous remercie infiniment pour tout le soutien et l'amour constants que vous m'avez prodigués tout au long de ma vie.

Puisse Dieu Tout-Puissant vous accorder santé, bonheur et longue vie.

À mon frère, qui a toujours été un exemple de sagesse et de soutien, merci pour ta présence rassurante et tes encouragements tout au long de mon parcours.

À mes deux sœurs, pour leur affection, leur complicité et leur soutien réconfortant tout au long de ce parcours.

À ma chère binôme et amie Sara avec qui j'ai passé des moments agréables et inoubliables et avec qui j'ai partagé ce travail.

À mon cher ami Khalil, Merci pour ton encouragement constant et ton soutien précieux. Tes mots et ta bienveillance ont été d'un grand réconfort tout au long de ce parcours

Table des matières

Liste des Figures	i
Liste des Tableaux	iii
Liste des Abréviations	iv
Introduction Générale	1
1 Étude Préalable	1
1.1 Introduction	1
1.2 Présentation de l'organisme d'accueil	1
1.2.1 Activités et filiales	2
1.2.2 Présence nationale et internationale	2
1.2.3 Objectifs de CEVITAL	3
1.2.4 Organigramme du groupe CEVITAL	3
1.2.5 Direction des systèmes d'informations DSI	4
1.3 Problématique et Objectifs	5
1.3.1 Problématique	5
1.3.2 Objectifs	6
1.4 Les besoins fonctionnels et non fonctionnels	6
1.4.1 Les besoins fonctionnels	6
1.4.2 Les besoins non fonctionnels	7
1.5 Langage et processus de développement	7
1.6 Conclusion	8
2 Analyse Des Besoins	9
2.1 Introduction	9
2.2 Identification des acteurs	9
2.3 Diagramme de contexte	10
2.4 Identification des messages	11
2.5 Identification des cas d'utilisation	11
2.6 Diagramme de cas d'utilisation	13
2.7 Description textuelle des principaux cas d'utilisation	14
2.7.1 Authentification	14
2.7.2 Gestion et suivi des formations	15
2.7.3 Evaluer les employés	16
2.7.4 Consulter le suivi des formations	17
2.7.5 Etablir un plan de formations	18
2.7.6 Valider le plan annuel des formations	19
2.8 Diagrammes de séquence	19

2.8.1	Diagramme de séquence « Authentification »	20
2.8.2	Diagramme de séquence « Gestion et suivi des formations »	21
2.8.3	Diagramme de séquence « Évaluer les employés »	22
2.8.4	Diagramme de séquence « Consulter le suivi des formations »	23
2.8.5	Diagramme de séquence « Établir un plan de formations »	24
2.8.6	Diagramme de séquence « Valider le plan annuel des formations »	25
2.9	Conclusion	26
3	Conception et Analyse du Domaine	27
3.1	Introduction	27
3.2	Diagrammes d'interactions	27
3.2.1	Diagramme d'interaction de cas d'utilisation «S'authentifier »	28
3.2.2	Diagramme d'interaction du cas d'utilisation «Gestion et suivi des formations »	29
3.2.3	Diagramme d'interaction de cas d'utilisation « Valider le plan annuel des formations »	30
3.3	Diagramme de classe	31
3.4	Dictionnaire de données	32
3.5	Schéma relationnel	36
3.6	Passage au modèle relationnel	36
3.7	Conclusion	37
4	Réalisation	38
4.1	Introduction	38
4.2	Technologies et Outils Utilisés	38
4.2.1	Outils et plateformes utilisés	38
4.2.2	Langages et frameworks	40
4.2.3	Bibliothèques utilisées	41
4.3	Structure du code source	42
4.4	Présentation des interfaces de notre plateforme	42
4.4.1	Interface d'Authentification	42
4.4.2	Interface des statistiques de formation	43
4.4.3	Interface des besoins en formation	44
4.4.4	Interface de planification des formations	45
4.4.5	Interface du plan annuel	46
4.4.6	Interface de validation du plan annuel	46
4.4.7	Interface d'évaluation à froid	47
4.4.8	Interface d'évaluation à chaud	47
4.5	Conclusion	48
	Conclusion générale	49
	Annexe A	53
	Annexe B	56
	Annexe C	58

Liste des Figures

1.1	Logo de CEVITAL	2
1.2	Organigramme de CEVITAL	4
1.3	Organigramme DSI	5
2.1	Diagramme de contexte dynamique	10
2.2	Diagramme de cas d'utilisation «Globale»	13
2.3	Diagramme de séquence du cas d'utilisation « S'authentifier ».	20
2.4	Diagramme de séquence du cas d'utilisation « Gestion des formations ». . .	21
2.5	Diagramme de séquence du cas d'utilisation « Évaluer les employés ». . . .	22
2.6	Diagramme de séquence du cas d'utilisation « Consulter le suivi des formations ».	23
2.7	Diagramme de séquence du cas d'utilisation « Établir un plan de formations ». .	24
2.8	Diagramme de séquence du cas d'utilisation « Valider le plan annuel des formations ».	25
3.1	Classe d'interface	27
3.2	Classe de contrôle	28
3.3	Classe entité	28
3.4	Diagramme d'interaction du cas d'utilisation «S'authentifier »	29
3.5	Diagramme d'interaction de cas d'utilisation «Gestion et suivi des formations»	30
3.6	Diagramme d'interaction du cas d'utilisation «Valider le plan annuel des formations »	31
3.7	Diagramme de classe	32
4.1	Logo Lucidchart.	38
4.2	Logo Visual Studio Code.	39
4.3	Logo XAMPP.	39
4.4	Logo Github.	39
4.5	Logo React.	40
4.6	LogoNode.js.	40
4.7	Interface d'authentification	43
4.8	Tableau de bord analytique du système de formation	44
4.9	Interface Besoin en formations	45

4.10	Formulaire d'ajout d'un besoin en formation.	45
4.11	Interface Suivi et planification des formations	46
4.12	Interface du plan annuel de formation	46
4.13	Interface de validation du plan annuel	47
4.14	Interface de l'évaluation à froid	47
4.15	Interface de l'évaluation à chaud	48
4.16	Diagramme de cas d'utilistion de l'administrateur	53
4.17	Diagramme de cas d'utilistion de Responsable direction	53
4.18	Diagramme de cas d'utilisation de Responsable de formation	54
4.19	Diagramme de cas d'utilistion du DRH	54
4.20	Diagramme de cas d'utilistion du Directeur General	55
4.21	Diagramme de cas d'utilistion de l'employe	55
4.22	Logo Postman	56
4.23	La requête GET dans le logiciel Postman	57
4.24	La requête POST dans le logiciel Postman	57
4.25	Email envoyé suite à la création d'un compte	58
4.26	Inscription confirmée à une session de formation	59
4.27	Fonctionnalité d'exportation des données	60

Liste des Tableaux

1.1	Activités des sites de production de CEVITAL	2
2.1	Identification des messages échangés	11
2.2	Identification des cas d'utilisation	12
2.3	Description du cas d'utilisation « Authentification »	14
2.4	Description du cas d'utilisation « Gestion et suivi des formations »	15
2.5	Description du cas d'utilisation « Evaluer les employés »	16
2.6	Description du cas d'utilisation « Consulter le suivi des formations »	17
2.7	Description du cas d'utilisation « Etablir un plan de formations »	18
2.8	Description du cas d'utilisation « Valider le plan annuel des formations »	19
3.1	Description des attributs du diagramme de classe	33

Liste des Abréviations

API	Application Programming Interface
CAT	Confitures
CSS	Cascading Style Sheets
DG	Directeur Général
DOM	Document Object Model
DRH	Directeur des Ressources Humaines
DSI	Direction du Système d'Information
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
JS	JavaScript
JSX	JavaScript XML
PET	Polyéthylène Téréphtalate
SDK	Software Development Kit
SGBD	Système de Gestion de Base de Données
SMTP	Simple Mail Transfer Protocol
SPA	Société Par Actions
SQL	Structured Query Language
UI	User Interface
UML	Unified Modeling Language
UP	Unified Process
URL	Uniform Resource Locator
XML	eXtensible Markup Language

Introduction Générale

Dans un environnement économique mondial marqué par une forte compétitivité et une évolution technologique constante, les entreprises sont plus que jamais appelées à innover, s'adapter et se réinventer. Qu'elles soient de petite, moyenne ou grande taille, opérant dans l'industrie, les services ou le numérique, elles doivent faire face à des défis multiples : digitalisation, gestion des talents, optimisation des processus, satisfaction client ou encore transformation durable.

Ainsi, la réussite d'une entreprise ne repose plus uniquement sur ses ressources matérielles ou financières, mais aussi sur ses ressources humaines ; sa capacité à mobiliser, former et faire évoluer son capital humain.

Parmi les facteurs déterminants de cette réussite, la gestion des compétences, et notamment la formation continue des employés, occupe une place stratégique. En effet, la formation professionnelle en entreprise peut être définie comme l'ensemble des dispositifs visant à permettre aux salariés d'acquérir, de maintenir ou de développer leurs compétences dans le cadre de leur activité professionnelle. Elle couvre un large éventail d'actions : formations techniques, développement personnel, certifications, perfectionnements... Pour être pleinement efficace, elle doit s'intégrer dans un cycle structuré allant de l'analyse des besoins à l'évaluation des résultats, en passant par la planification, l'organisation et le suivi administratif et pédagogique. À cette complexité s'ajoute la nécessité, pour les entreprises modernes, de disposer d'outils numériques performants pour gérer l'ensemble de ce processus de manière fluide, centralisée et traçable.

C'est dans ce contexte que s'inscrit notre étude, menée au sein de l'entreprise CEVITAL, spécialisée dans l'agroalimentaire, avec un effectif de 18 000 employés et une volonté affirmée de valoriser le capital humain. L'objectif principal de cette application est de simplifier la gestion des formations au sein de l'entreprise en offrant une interface intuitive pour la planification, le suivi et l'évaluation des actions de formation. Elle vise également à améliorer la traçabilité des données et la génération des indicateurs statistiques via un tableau de bord.

Le présent mémoire est structuré en quatre chapitres principaux.

Le premier chapitre est consacré à l'étude préalable du contexte général et des enjeux liés à la gestion des formations. Il vise à introduire l'organisme d'accueil, à exposer la problématique rencontrée, ainsi qu'à définir les objectifs du projet. Le deuxième chapitre porte sur l'analyse des besoins fonctionnels et techniques, en s'appuyant sur les attentes des utilisateurs. Il a pour but de modéliser le système à travers l'identification des acteurs, les cas d'utilisation et les scénarios d'interaction. Le troisième chapitre aborde la phase de conception de l'application, incluant la modélisation du domaine, la structuration des données ainsi que le passage vers un modèle relationnel, en vue de préparer la phase d'implémentation. Enfin, le quatrième chapitre est dédié à la réalisation du projet, à la présentation des outils utilisés, au développement des interfaces principales et à l'illustration des fonctionnalités implémentées dans l'application.

Chapitre 1

Étude Préalable

1.1 Introduction

Donner accès à la bonne formation, au bon moment et à la bonne personne est aujourd'hui un enjeu stratégique pour les entreprises. Celles-ci doivent être capables de cartographier et de développer les compétences de leurs collaborateurs afin de soutenir leur croissance et de répondre aux exigences du marché.

Ce chapitre a pour objectif de présenter l'environnement de réalisation du projet, à savoir l'entreprise CEVITAL. Nous y décrivons ses principales activités, sa politique en matière de gestion des ressources humaines, ainsi que les défis rencontrés dans le domaine de la formation continue. Cette analyse permettra de mieux comprendre les motivations ayant conduit à la mise en œuvre d'une solution numérique pour la gestion des formations.

1.2 Présentation de l'organisme d'accueil

CEVITAL est une entreprise spécialisée dans l'industrie agroalimentaire, créée par Issad Rebrab en mai 1998 et implantée à l'extrême est du port de Béjaïa, sous forme d'une société par actions (SPA).

Portée par 18 000 employés répartis sur 3 continents, CEVITAL est l'un des fleurons de l'industrie agroalimentaire en Algérie, qui est constituée de plusieurs unités de production ultramodernes, telles que : une raffinerie d'huile, deux raffineries de sucre, une margarinerie, une unité de conditionnement d'eau minérale, une unité de fabrication et de conditionnement de boissons rafraîchissantes, une conserverie, des silos portuaires ainsi qu'un terminal de déchargement portuaire [1].

D'ailleurs, elle couvre les besoins nationaux et a permis de faire passer l'Algérie du stade d'importateur à celui d'exportateur.

La figure 1.1 présente le logo officiel de l'entreprise CEVITAL.



FIGURE 1.1 – Logo de CEVITAL

1.2.1 Activités et filiales

L'ensemble des activités de CEVITAL est concentré sur la production et la commercialisation des huiles végétales, margarines, sucres, eau minérale et jus de fruits. La production est répartie sur les trois sites (Bejaia, El Kseur et Agueni Gueghane).

Le tableau 1.1 présente les principales activités de production de CEVITAL réparties sur ses trois sites industriels.

Site de Bejaia	Site d'El Kseur	Site d'Agueni Gueghane
<ul style="list-style-type: none"> — Sucre liquide 500 T/jour. — Raffinage du sucre 5000 T/jour. — Raffinage de l'huile 1800 T/jour. — Production de margarines. — Emballage en PET (différentes dimensions). — Silos portuaires. 	<ul style="list-style-type: none"> — Production du jus Fruits COJEK 1130 T/jour. — CAT (confitures) 66 T/jour. — Sauce 100 T/jour. 	<ul style="list-style-type: none"> — Production d'eau minérale Lalla Khedidja 2688 T/jour.

TABLE 1.1 – Activités des sites de production de CEVITAL

1.2.2 Présence nationale et internationale

CEVITAL est considérée comme un leader global dans le secteur de l'agro-industrie, que ce soit à l'échelle nationale ou bien internationale. D'ailleurs, elle a permis de faire passer l'Algérie du stade d'importateur à celui d'exportateur en commercialisant ses produits agroalimentaires (sucre, huiles, margarine) vers de nombreux pays en Afrique, au Moyen-Orient et même en Europe. Cela renforce sa position sur les marchés internationaux.

Tout cela est rendu possible grâce à son implication dans des partenariats et des acquisitions internationales, visant à diversifier ses domaines d'intervention et à renforcer sa compétitivité globale.

1.2.3 Objectifs de CEVITAL

Les objectifs que le groupe CEVITAL ambitionne de réaliser sont les suivants :

- ✓ L'élargissement de la distribution de ses produits à l'ensemble du territoire national.
- ✓ L'importation de graines oléagineuses en vue de l'extraction directe des huiles brutes.
- ✓ Le soutien aux agriculteurs à travers des aides financières pour favoriser la production locale de ces graines
- ✓ L'amélioration de la visibilité de ses offres d'emploi sur le marché du travail.
- ✓ La satisfaction des besoins des clients
- ✓ L'augmentation du chiffre d'affaires.
- ✓ Le renforcement de ses relations de partenariat à l'international.
- ✓ Le positionnement de ses produits sur les marchés étrangers grâce à leurs exportations.

Les objectifs de cette entreprise sont atteints par l'application de ses valeurs d'**intégrité**, de **respect**, d'**initiative**, de **persévérance** et de **solidarité**, qui orientent et motivent chaque action entreprise [2].

1.2.4 Organigramme du groupe CEVITAL

Le schéma organisationnel présenté dans la figure 1.2 reflète la structure hiérarchique de CEVITAL et met en lumière la répartition des différentes directions. Il offre une meilleure compréhension de l'organisation interne et de la coordination entre les services.

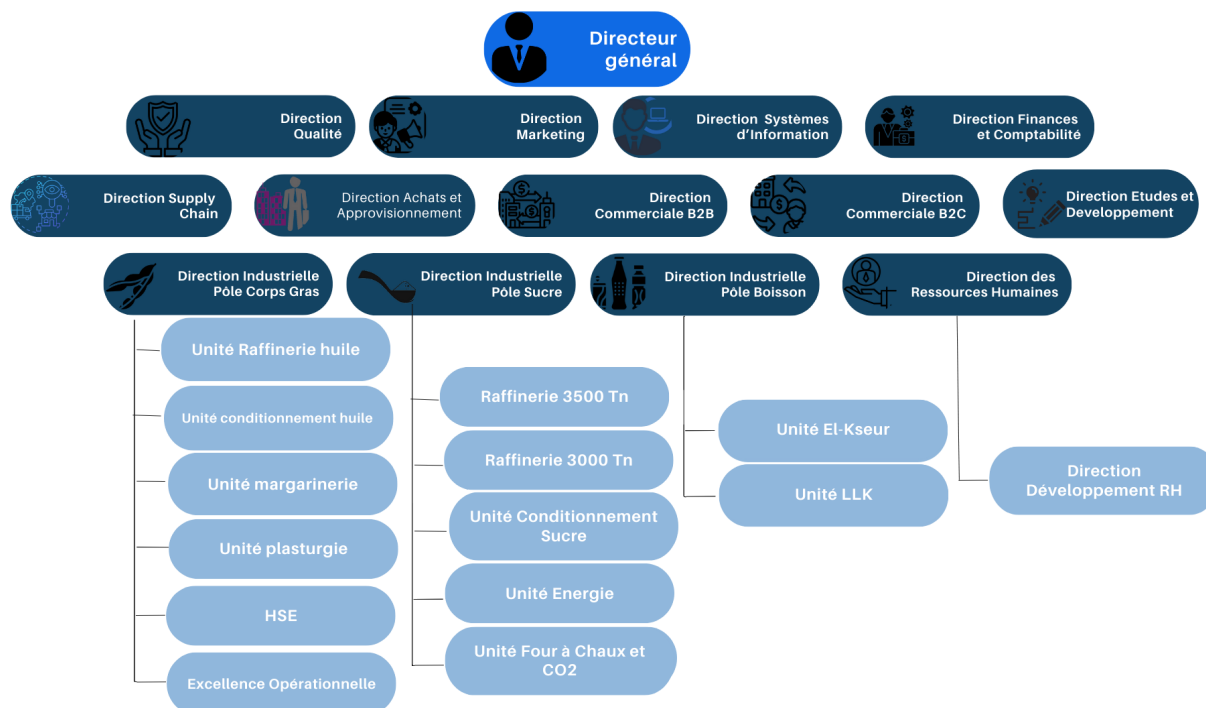


FIGURE 1.2 – Organigramme de CEVITAL

1.2.5 Direction des systèmes d'informations DSI

La direction des systèmes d'information veille sur la mise en place des moyens des technologies de l'information nécessaires pour supporter et améliorer l'activité, la stratégie et la performance de l'entreprise.

Comme illustré sur la figure 1.3, elle est constituée de plusieurs départements, dont le département technique, le département de sécurité, le département des transformations digitales et le département des applications métiers, où nous avons bénéficié de séances de stage pour collecter des informations relatives aux formations.

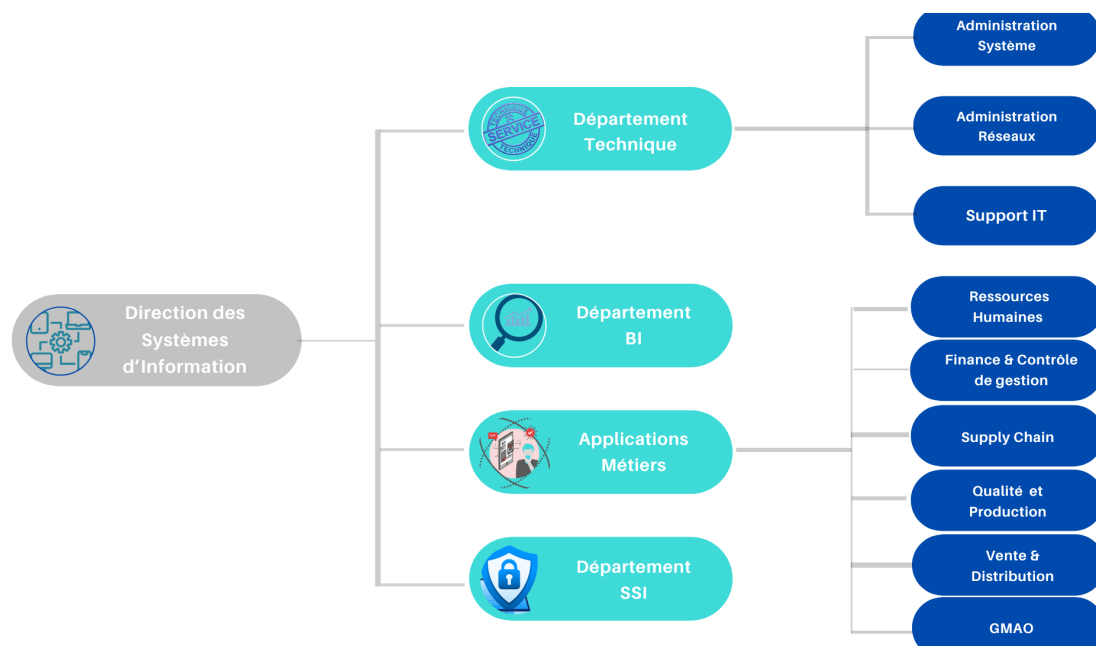


FIGURE 1.3 – Organigramme DSI

1.3 Problématique et Objectifs

1.3.1 Problématique

Les formations constituent un levier puissant pour renforcer les compétences, la créativité et l'engagement des collaborateurs, surtout au sein des grandes entreprises comme CEVITAL. De plus, elles permettent d'accompagner l'adaptation aux avancées technologiques et aux exigences du marché en dotant chaque personne contribuant à sa réussite, que ce soit salarié, stagiaire ou apprenti, des savoir-faire nécessaires pour exploiter les outils modernes et relever efficacement les défis.

Cependant, la gestion de ces formations entraîne de grandes difficultés, ce que nous avons remarqué durant notre stage. Les principaux problèmes identifiés sont les suivants :

- Absence d'un système dédié à la gestion des formations : Actuellement, la gestion des formations repose uniquement sur Excel, un outil qui, bien que flexible, présente plusieurs inconvénients majeurs. Il ne permet pas une gestion optimisée et centralisée des sessions de formation, entraîne des risques d'erreurs humaines, rend difficile le suivi des participants et des évaluations, et ne garantit pas une traçabilité fiable des données.
- La difficulté de mesurer l'efficacité des formations , ce qui peut compliquer les décisions concernant les investissements futurs.
- Le nombre important de données qui engendre d'une part une difficulté de stockage et d'autre la complexité de recherche ainsi la redondance des données.
- Vulnérabilité des données sensibles aux accès non autorisés.
- La gestion excessive de la paperasse, ce qui entraîne une lourdeur administrative.

1.3.2 Objectifs

Afin de remédier aux problèmes mentionnés précédemment, nous proposons une solution consistant à concevoir et à développer une application qui palliera les limites identifiées, et visera à atteindre les objectifs suivants :

- La gestion des éléments de base, tels que les employés et les utilisateurs, avec une interface ergonomique qui facilite la navigation entre les différentes fonctionnalités du système.
- Assurer un meilleur suivi des sessions de formation.
- Faciliter l'accès aux données et sécuriser leurs sauvegarde.
- Faciliter la tâche d'évaluation que ce soit pour les responsable ou bien les employés.
- Réduire le taux d'utilisation du papier et cela en automatisant toutes les fonctions nécessaires.

1.4 Les besoins fonctionnels et non fonctionnels

1.4.1 Les besoins fonctionnels

Les besoins fonctionnels représentent les fonctionnalités que l'application développée doit réaliser[3], et qui sont :

1. **Authentification** : Chaque utilisateur doit s'authentifier pour accéder à l'application.
2. **Gestion des comptes** : Gérer les comptes utilisateurs (ajouter, modifier, désactiver et attribuer des rôles).
3. **Saisie des besoins en formation** :
 - Créer, modifier et supprimer des besoins en formation identifiés pour les employés de chaque direction.
 - Regrouper les besoins et les filtrer par priorité(Élevé, Moyenne, faible).
4. **Gestion et suivie des formations** :
 - Créer, modifier et supprimer des formations, avec possibilité de téléchargement et impression de la liste des formations.
 - Regrouper les formations par catégories pour faciliter la recherche.
 - Définir et organiser les détails logistiques des formations, tels que la date, le groupe de participants, le mode de réalisation (en ligne ou en présentiel) ainsi que d'autres aspects pertinents.
 - Créer, modifier et supprimer les directions et les postes associés. Cela facilite l'organisation hiérarchique de l'entreprise et l'affectation précise des employés selon leur poste et leur direction.
 - Inscrire des employés aux formations de manière efficace, en sélectionnant les participants directement à partir de la liste des besoins.
 - Créer, modifier et supprimer des formateurs en leurs assignant des formations spécifiques.
 - Gérer également les lieux de formation qui collaborent avec l'entreprise pour la réalisation des formations.

5. **Plan de formations :**

- Gérer des plans de formations avec possibilité d'exportation et de téléchargement si nécessaire.

6. **Système de notification :**

- Envoi automatique d'une notification à chaque employé à la fin d'une formation, l'invitant à réaliser une évaluation à chaud.
- Envoi automatique d'une notification aux responsables, six mois après la formation, l'invitant à réaliser une évaluation à froid.

7. **Évaluation :**

- Remplir une fiche d'évaluation portant sur le contenu de la formation, la qualité du formateur et les conditions de déroulement (évaluation à chaud).
- Évaluer l'impact de la formation sur les performances des employés (évaluation à froid).

8. **Reporting et statistiques :** Afficher des statistiques sur les formations réalisées et le taux de satisfaction et participation.

1.4.2 Les besoins non fonctionnels

Les besoins non fonctionnels représentent les exigences techniques que le système doit vérifier [3] :

1. **Sécurité** : l'application doit garantir une sécurité optimale des données .
2. **Ergonomie** : fournir une interface simple et conviviale, accessible même pour des utilisateurs non techniques.
3. **Performance** : l'application doit assurer un temps de réponse minime, tout en répondant aux exigences de l'utilisateur.
4. **Maintenabilité** : fiabilité et possibilité de mise à jour des informations.
5. **Convivialité** : Offrir une bonne expérience utilisateur avec des interfaces responsives et intuitives, une navigation fluide et une facilité d'utilisation, permettant ainsi aux utilisateurs d'accomplir leurs tâches de manière efficace et agréable.

1.5 Langage et processus de développement

Pour les phases d'analyse et de conception de notre application, nous avons opté pour l'utilisation du langage UML (Unified Modeling Language) afin de modéliser les différents aspects du système à développer, en nous appuyant sur des diagrammes clés. Afin de structurer et d'organiser cette démarche, nous avons également adopté la méthode UP (Unified Process) adaptée à nos besoins spécifiques. Cette approche nous permettra de répondre efficacement aux exigences du client, tout en respectant les coûts et les délais, assurant ainsi la réussite du projet.

1.6 Conclusion

Dans ce premier chapitre, nous avons tout d'abord présenté l'organisme d'accueil CE-VITAL, ainsi que les services directement liés à notre travail. Cette étude préalable constitue ainsi une étape essentielle pour mieux comprendre le cadre du projet et définir des orientations claires pour la suite du développement. Les concepts clés liés à cette solution seront approfondis dans le chapitre suivant.

Chapitre 2

Analyse Des Besoins

2.1 Introduction

Après avoir défini les besoins fonctionnels et techniques de notre application dans le chapitre précédent, nous allons maintenant approfondir l'analyse des besoins en nous focalisant sur la modélisation des interactions entre les acteurs et le système, pour bien comprendre le fonctionnement global de l'application avant d'entamer la phase de conception détaillée.

Dans ce chapitre, nous allons identifier les différents acteurs interagissant avec l'application et établir un diagramme de contexte qui représentera les échanges entre eux et le système. Ensuite, élaborer des diagrammes de cas d'utilisation, et détailler les interactions dynamiques à travers les diagrammes de séquence, illustrant le déroulement des scénarios d'utilisation.

2.2 Identification des acteurs

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il peut interagir directement avec l'état du système en le consultant et/ou en le modifiant, à travers l'émission et/ou la réception de messages contenant éventuellement des données [5].

Pour notre système, nous avons identifié les acteurs suivants :

- **Administrateur** : Chargé de la gestion des comptes utilisateurs et de l'administration des accès à la plateforme de formation.
- **Employé** : A la possibilité de participer aux formations proposées et évaluer ces dernières afin de mesurer leurs impacts.
- **Responsable de direction** : Responsable de l'identification des besoins en formation de son équipe et de l'évaluation pour mesurer leurs efficacités.
- **Responsable de formation** : chargé de gérer le catalogue des formations en mettant à jour et en organisant les formations internes et externes par calibrage. Il est également responsable de la gestion des formateurs internes et externes ainsi que des lieux de formations, en veillant à leur sélection et leur suivi.
- **Directeur des Ressources Humaines (DRH)** : Responsable de la première validation du plan annuel de formation.
- **Directeur Général (DG)** : Intervient en dernier lieu pour valider le plan de formation ainsi que le budget alloué.

2.3 Diagramme de contexte

Un diagramme de contexte présente un système logiciel entier comme un processus unique et montre comment les entités externes interagissent avec le système [6].

À la suite de l'analyse des besoins fonctionnels et de l'identification des acteurs du système, le contexte dynamique a été modélisé à l'aide d'un diagramme, présenté dans la figure 2.1 :

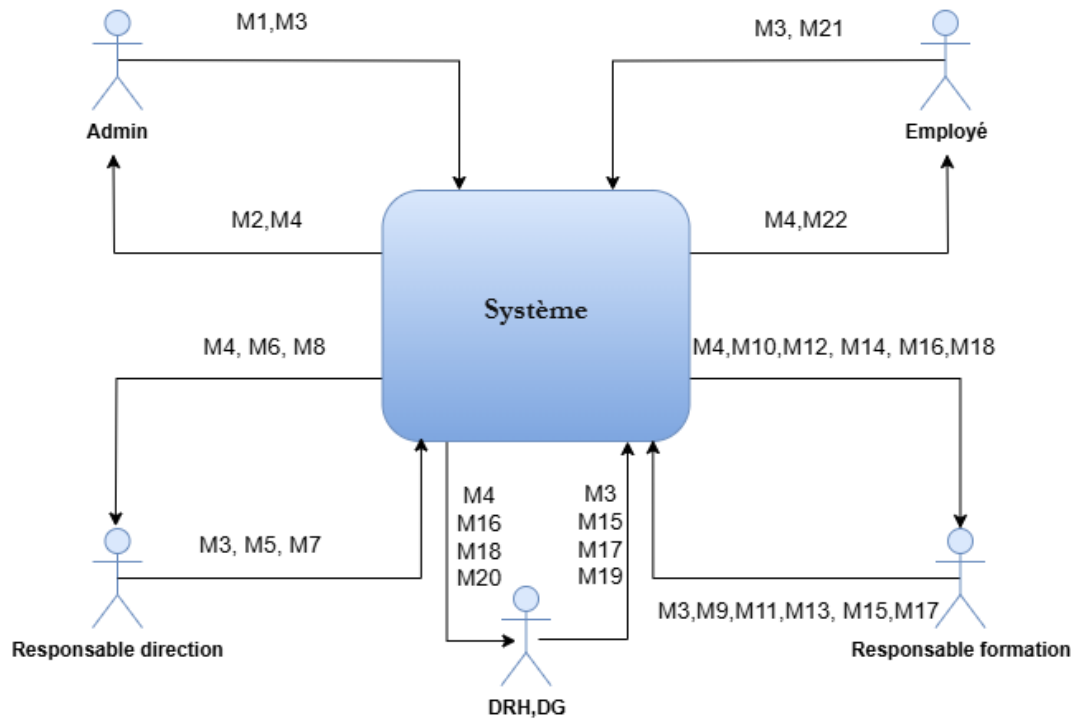


FIGURE 2.1 – Diagramme de contexte dynamique

2.4 Identification des messages

N°	Message Acteur vers Sys- tème	N°	Message Système vers Ac- teur
M1	Gestion des comptes utilisateurs	M2	Afficher la liste des utilisateurs
M3	Authentification	M4	Affichage de l'interface d'accueil correspondant à chaque acteur en cas de succès et l'interface d'authentification en cas d'échec
M5	Insérer des besoins en formations	M6	Afficher la page « Besoins en formations ».
M7	Évaluer les employés	M8	Afficher la fiche d'évaluation.
M9	Gestion et suivie des formations	M10	Afficher le catalogue des formations
M11	Accès à la gestion des formateurs et lieux de formations	M12	Affichage de la liste des formateurs et des lieux
M13	Gestion des employés	M14	Afficher la liste des employés
M15	Demande la page « plan de formation »	M16	Afficher la page « plan de formation »
M17	Consulter les statistiques	M18	Affichage des statistiques
M19	Validation du plan de formation	M20	Afficher une boîte de dialogue « Plan validé » ou bien « Plan rejeté »
M21	Evaluer les formations	M22	« Évaluation enregistrée avec succès »

TABLE 2.1 – Identification des messages échangés

2.5 Identification des cas d'utilisation

Un cas d'utilisation décrit une fonction qu'un système exécute pour atteindre l'objectif de l'utilisateur. Il doit renvoyer un résultat observable qui est utile pour l'utilisateur du système [7].

Dans ce qui suit, nous allons détailler les différents cas d'utilisation associés à chaque acteur du système afin de mieux les représenter.

N°	Cas d'utilisation		Acteur
1	S'authentifier		Utilisateurs
2	Gérer les comptes	Ajouter	Administrateur
		Modifier	
		Désactiver	
3	Gérer les besoins en formation	Ajouter	Responsable de direction
		Modifier	
		Supprimer	
	Évaluer les employés		
4	Gérer et suivre les formations	Ajouter	Responsable de formation
		Modifier	
		Supprimer	
	Gérer les employés	Ajouter	
		Modifier	
		Supprimer	
	Gérer les formateurs	Ajouter	
		Modifier	
		Supprimer	
	Gérer les lieux de formation	Ajouter	
		Modifier	
		Supprimer	
	Gérer les structures	Ajouter	
		Modifier	
		Supprimer	
	Informers les participants par e-mail		
	Établir un plan de formation		
	Consulter les évaluations		
	Consulter les statistiques		
5	Valider une première fois le plan		DRH
6	Valider une deuxième fois le plan		DG
	Consulter le suivi des formations		
	Consulter les statistiques		
7	Évaluer les formations		Employé

12
TABLE 2.2 – Identification des cas d'utilisation

2.6 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation en UML illustre les interactions possibles entre un utilisateur et un système. Il représente les scénarios d'utilisation pour lesquels le système est conçu [8].

La figure 2.2 présente le diagramme de cas d'utilisation de notre système.

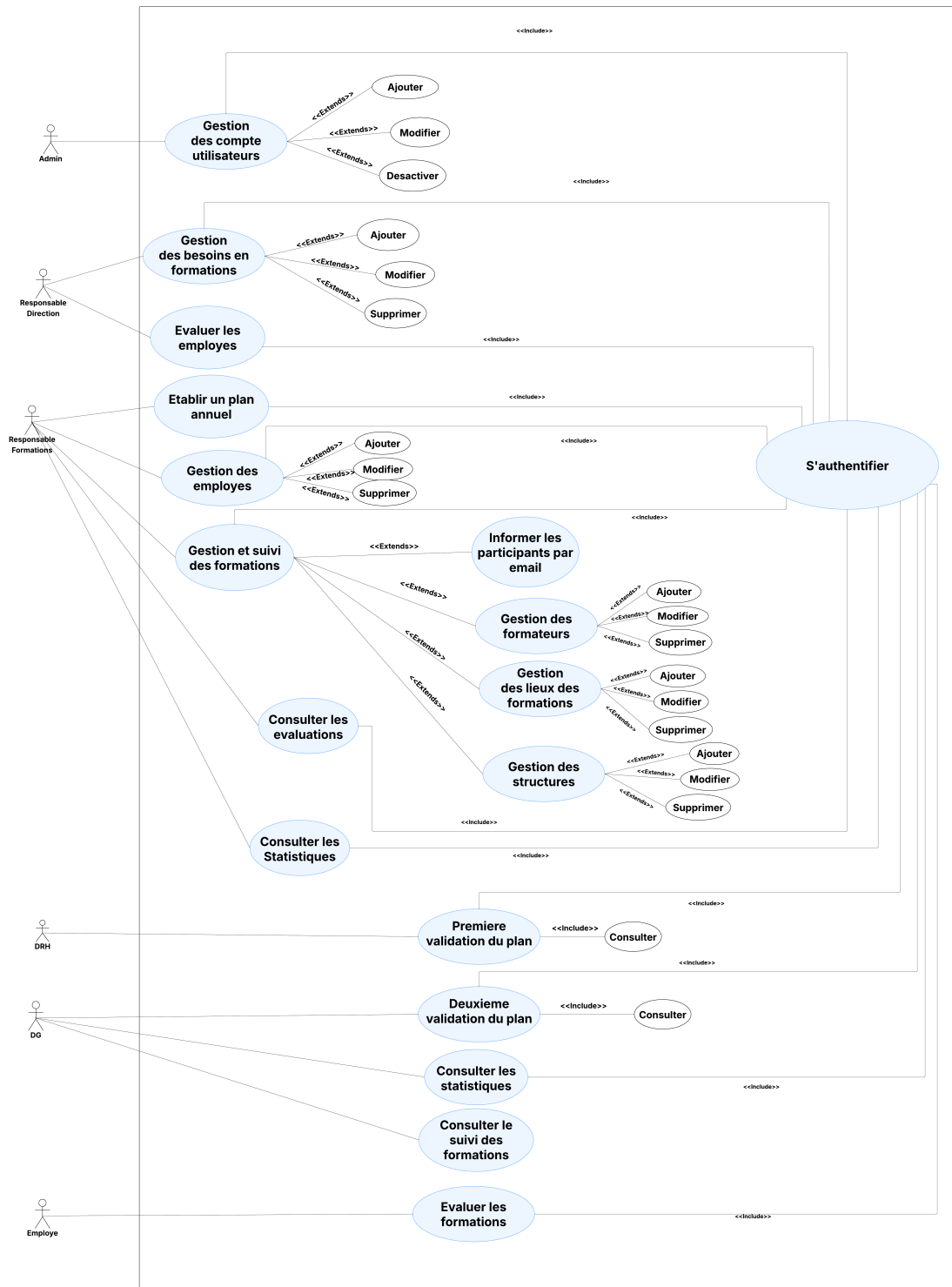


FIGURE 2.2 – Diagramme de cas d'utilisation «Globale»

2.7 Description textuelle des principaux cas d'utilisation

Cette étape consiste à décrire textuellement chaque cas d'utilisation selon une structure bien définie.

2.7.1 Authentification

Le tableau ci-dessous représente la description du cas d'utilisation « Authentification » :

Cas d'utilisation	Authentification
Acteurs	<ul style="list-style-type: none">- Administrateur- Responsable de direction- Responsable de formation- Employé- DRH- DG
Objectif	Permettre à chaque utilisateur de s'identifier de manière sécurisée afin d'accéder à l'application avec les droits correspondant à son rôle
Pré-condition	L'utilisateur doit avoir un compte valide (nom d'utilisateur et mot de passe enregistrés dans la base de données).
Scénario nominal	<ol style="list-style-type: none">1. L'utilisateur accède à la page de connexion.2. Le système affiche la page d'authentification.3. Il saisit son identifiant et son mot de passe.4. Le système vérifie l'existence des données.5. Le système donne l'accès à l'interface correspondante.
Scénario alternatif	<ol style="list-style-type: none">1. Erreur d'authentification : email ou mot de passe non valide ; le mot de passe doit contenir au moins 8 caractères et l'email doit respecter le format prenom.nom@cevital.com.2. Champs obligatoires vides.
Post-condition	<ul style="list-style-type: none">- Si l'authentification réussit, l'utilisateur accède à son espace personnel.

TABLE 2.3 – Description du cas d'utilisation « Authentification »

2.7.2 Gestion et suivi des formations

Le tableau ci-dessous représente la description du cas d'utilisation « Gestion et suivi des formations » :

Cas d'utilisation	Gestion et suivi des formations
Acteurs	Responsable de formation
Objectif	Assurer une bonne organisation des formations ,un suivi éfficace et à jour du plan de formation.
Pré-condition	Le responsable de formation doit être authentifié.
Scénario nominal	<ol style="list-style-type: none">1. Le responsable de formation accède à l'interface de gestion des formations.2. Il consulte la liste des formations disponibles.3. Il peut choisir l'une des trois actions suivantes : .Ajouter : Le responsable de formation choisit l'option pour créer une nouvelle formation. Un formulaire s'affiche où il peut renseigner les informations nécessaires. .Modifier : Le responsable de formation sélectionne une formation existante pour modifier ses informations .Rechercher : Le responsable de formation saisit un mot-clé ou applique des filtres (date, formateur, statut...) pour retrouver rapidement une formation spécifique dans la liste.4. Le système enregistre les actions effectuées et met à jour la base de données avec les nouvelles informations (ajout ou modification de formation).5. Le système affiche un message de confirmation pour lui indiquer que l'action a été effectuée avec succès.
Scénario alternatif	<ol style="list-style-type: none">1.Erreur lors de la création d'une formation (Si des champs obligatoires sont manquants, le système affiche un message d'erreur).2.Recherche infructueuse : Si aucune formation ne correspond aux critères de recherche saisis, le système affiche un message indiquant qu'aucun résultat n'a été trouvé.
Post-condition	<ol style="list-style-type: none">1.Une formation validée est enregistrée et affichée dans la liste des formations.2.Toute modification ou suppression est prise en compte.

TABLE 2.4 – Description du cas d'utilisation « Gestion et suivi des formations »

2.7.3 Evaluer les employés

Le tableau ci-dessous représente la description du cas d'utilisation « Evaluer les employés » :

Cas d'utilisation	Evaluer les employés
Acteurs	- Responsable de direction
Objectif	Permettre au responsable hiérarchique (direction) d'évaluer, après un certain délai, l'impact réel de la formation suivie par un employé sur ses performances et son comportement professionnel.
Pré-condition	-Le responsable de direction doit être authentifié. -La session de formation à évaluer est terminée.
Scénario nominal	1.Le responsable de direction accède à l'espace dédié aux évaluations. 2.Le responsable de direction sélectionne la formation concernée ainsi que l'employé à évaluer dans la liste des formations terminées. 3. Il remplit les champs du formulaire d'évaluation à froid en renseignant les informations requises, telles que les compétences acquises, l'impact de la formation et les suggestions d'amélioration. 4.Il valide et soumet la fiche d'évaluation. 5.Le système enregistre l'évaluation .
Scénario alternatif	Si des champs obligatoires ne sont pas remplis, le système affiche un message d'erreur et demande à l'utilisateur de les compléter avant de soumettre l'évaluation.
Post-condition	L'évaluation est enregistrée et accessible aux administrateurs pour analyse.

TABLE 2.5 – Description du cas d'utilisation « Evaluer les employés »

2.7.4 Consulter le suivi des formations

Le tableau ci-dessous représente la description du cas d'utilisation « Consulter le suivi des formations » :

Cas d'utilisation	Consulter le suivi des formations
Acteurs	Directeur Général
Objectif	Suivre le déroulement des formations en temps réel ou en différé, afin d'avoir une vision globale de l'état d'avancement du plan de formation
Pré-condition	L'authentification du Directeur Général a été effectuée avec succès.
Scénario nominal	<ol style="list-style-type: none">1.Le DG accède à l'espace de suivi des formations.2.Il consulte la liste des formations organisées.3.Il sélectionne une formation spécifique pour voir son état d'avancement4.Le système affiche les détails de la formation, y compris : La liste des employés à formés Les formateurs ayant encadré la formation Les lieux de formations Les dates de la formation (début et fin) Toute autre information pertinente sur le suivi de la formation.
Scénario alternatif	Problème d'accès aux informations : Un message d'erreur s'affiche en cas de problème technique.
Post-condition	Le DG a une vue complète sur le suivi des formations et peut utiliser ces informations pour les décisions futures.

TABLE 2.6 – Description du cas d'utilisation « Consulter le suivi des formations »

2.7.5 Etablir un plan de formations

Le tableau ci-dessous représente la description du cas d'utilisation « Établir un plan de formations » :

Cas d'utilisation	Etablir un plan de formations
Acteurs	Responsable de formation
Objectif	Élaborer un programme annuel de formation en recueillant les besoins exprimés par les différents services, tout en tenant compte des priorités de l'entreprise, des compétences à développer, et des contraintes budgétaires.
Pré-condition	Authentification et accès autorisé.
Scénario nominal	<ol style="list-style-type: none">1.Le responsable de formation accède à l'interface dédiée à l'établissement du plan de formation.2.Il sélectionne l'option "Établir un nouveau plan de formation".3.Le système propose une interface permettant à l'utilisateur d'établir le plan de formation en sélectionnant les formations à planifier, à partir de listes déroulantes ou d'options prédéfinies.4.Le responsable de formation renseigne le budget des formations à venir5.Une fois toutes les informations saisies.6.Le système enregistre le plan de formation et le rend accessible pour consultation et validation.
Scénario alternatif	Si des champs obligatoires sont manquants ou des erreurs dans la planification, le système affiche un message d'erreur demandant de corriger ces problèmes avant la soumission.
Post-condition	Le plan de formation est enregistré et visible pour les responsables concernés. Il peut être consulté et modifié à tout moment avant sa validation finale.

TABLE 2.7 – Description du cas d'utilisation «Etablir un plan de formations »

2.7.6 Valider le plan annuel des formations

Le tableau ci-dessous représente la description du cas d'utilisation « Valider le plan annuel des formations » :

Cas d'utilisation	Valider le plan annuel des formations
Acteurs	-Directeur Général (DG). -Directeur des ressources humains (DRH).
Objectif	Permettre au DRH et au DG de consulter et valider le plan annuel des formations proposé, en s'assurant qu'il est aligné avec les objectifs stratégiques de l'entreprise, tout en respectant les priorités, les compétences visées et le budget alloué.
Pré-condition	-Le plan annuel a été préalablement élaboré par le responsable formation. -Le plan est soumis pour validation. -L'utilisateur (DRH ou DG) est authentifié dans le système.
Scénario nominal	1.Il accède à la section "Plan annuel des formations". 2.Il consulte les détails du plan proposé (formations prévues, budgets, objectifs, services concernés...). 3.Une fois satisfait, il clique sur "Valider le plan". 4.Le système enregistre la validation.
Scénario alternatif	l'utilisateur rejette le plan ,refuse de le valider et formule des remarques lorsque une incohérence est identifiée(budget dépassé, formations non pertinentes...).
Post-condition	-Le plan annuel des formations est validé et devient exécutable.

TABLE 2.8 – Description du cas d'utilisation «Valider le plan annuel des formations »

2.8 Diagrammes de séquence

Un diagramme de séquence UML est une représentation graphique qui décrit l'interaction des objets d'un système dans le temps. Ces diagrammes capturent la séquence des messages échangés entre les objets et l'ordre dans lequel ces interactions se produisent. [9].

Dans ce qui suit, nous présenterons quelques diagrammes de séquence relatifs aux cas d'utilisation précédemment décrits.

2.8.1 Diagramme de séquence « Authentification »

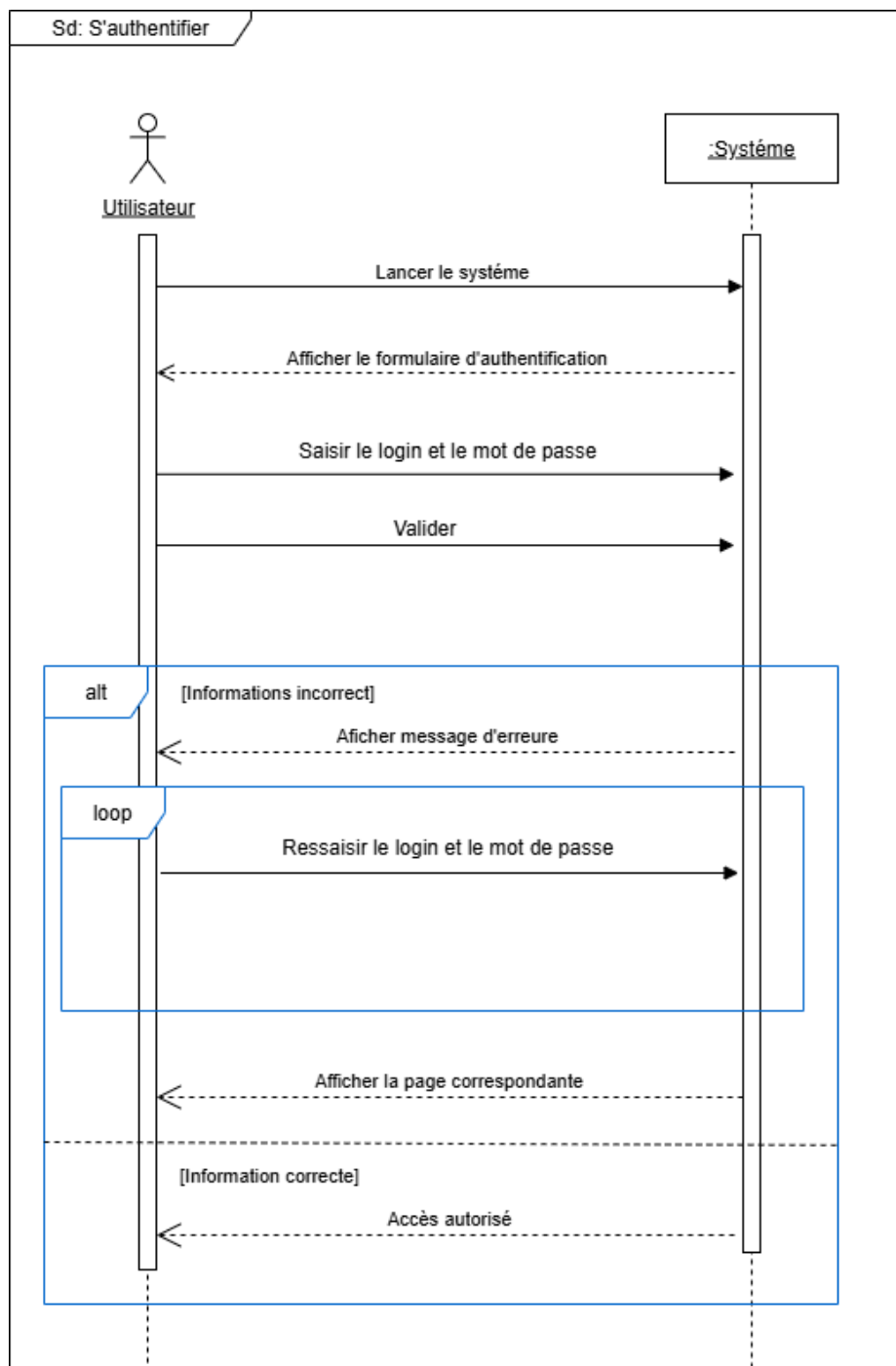


FIGURE 2.3 – Diagramme de séquence du cas d'utilisation « S'authentifier ».

2.8.2 Diagramme de séquence « Gestion et suivi des formations »

»

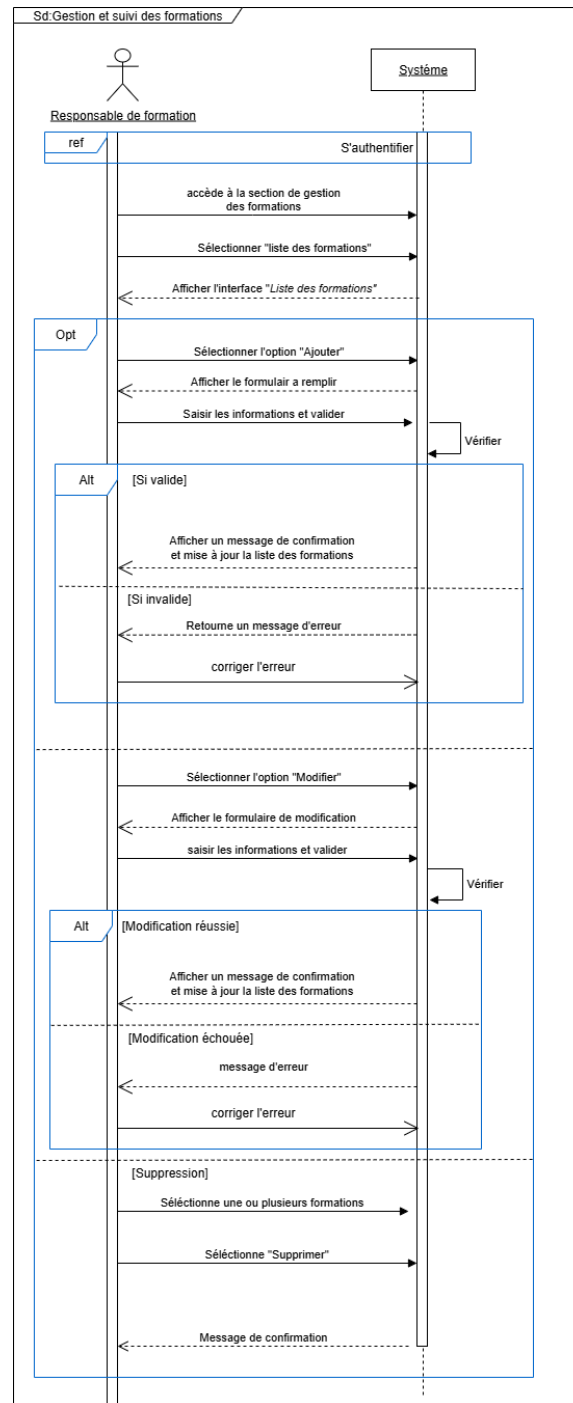


FIGURE 2.4 – Diagramme de séquence du cas d'utilisation « Gestion des formations ».

2.8.3 Diagramme de séquence « Évaluer les employés »

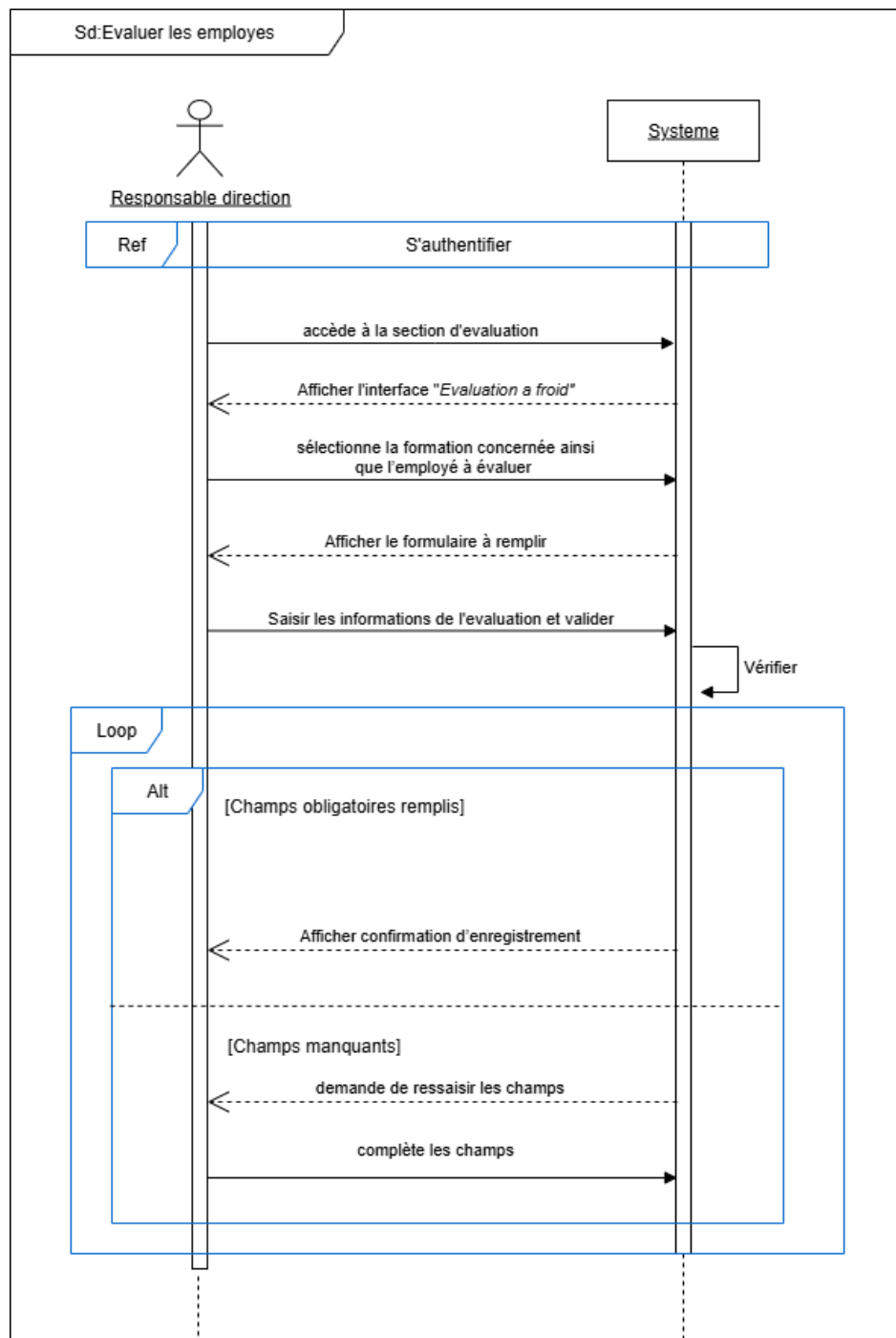


FIGURE 2.5 – Diagramme de séquence du cas d'utilisation « Évaluer les employés ».

2.8.4 Diagramme de séquence « Consulter le suivi des formations »

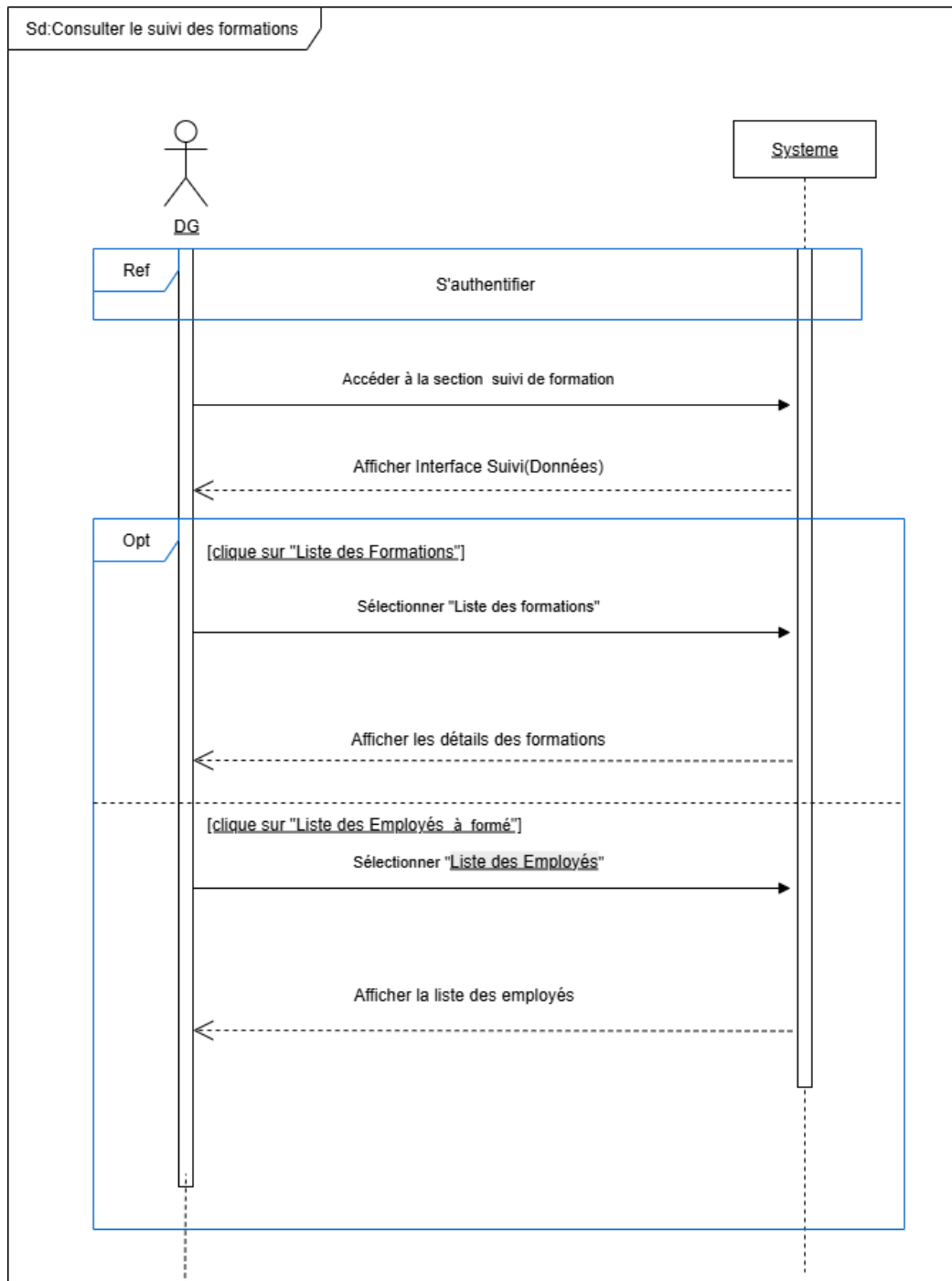


FIGURE 2.6 – Diagramme de séquence du cas d’utilisation « Consulter le suivi des formations ».

2.8.5 Diagramme de séquence « Établir un plan de formations »

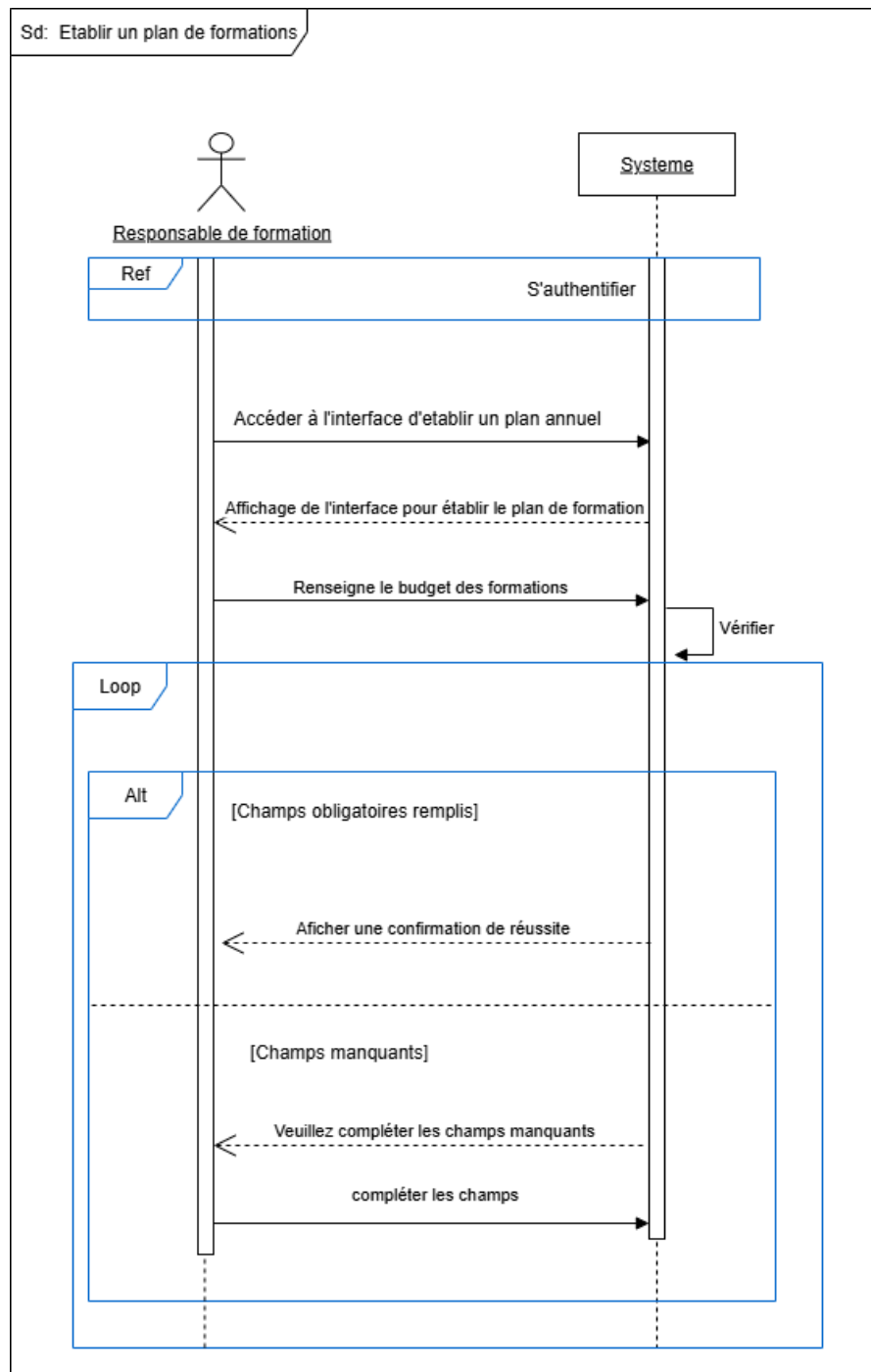


FIGURE 2.7 – Diagramme de séquence du cas d’utilisation « Établir un plan de formations ».

2.8.6 Diagramme de séquence « Valider le plan annuel des formations »

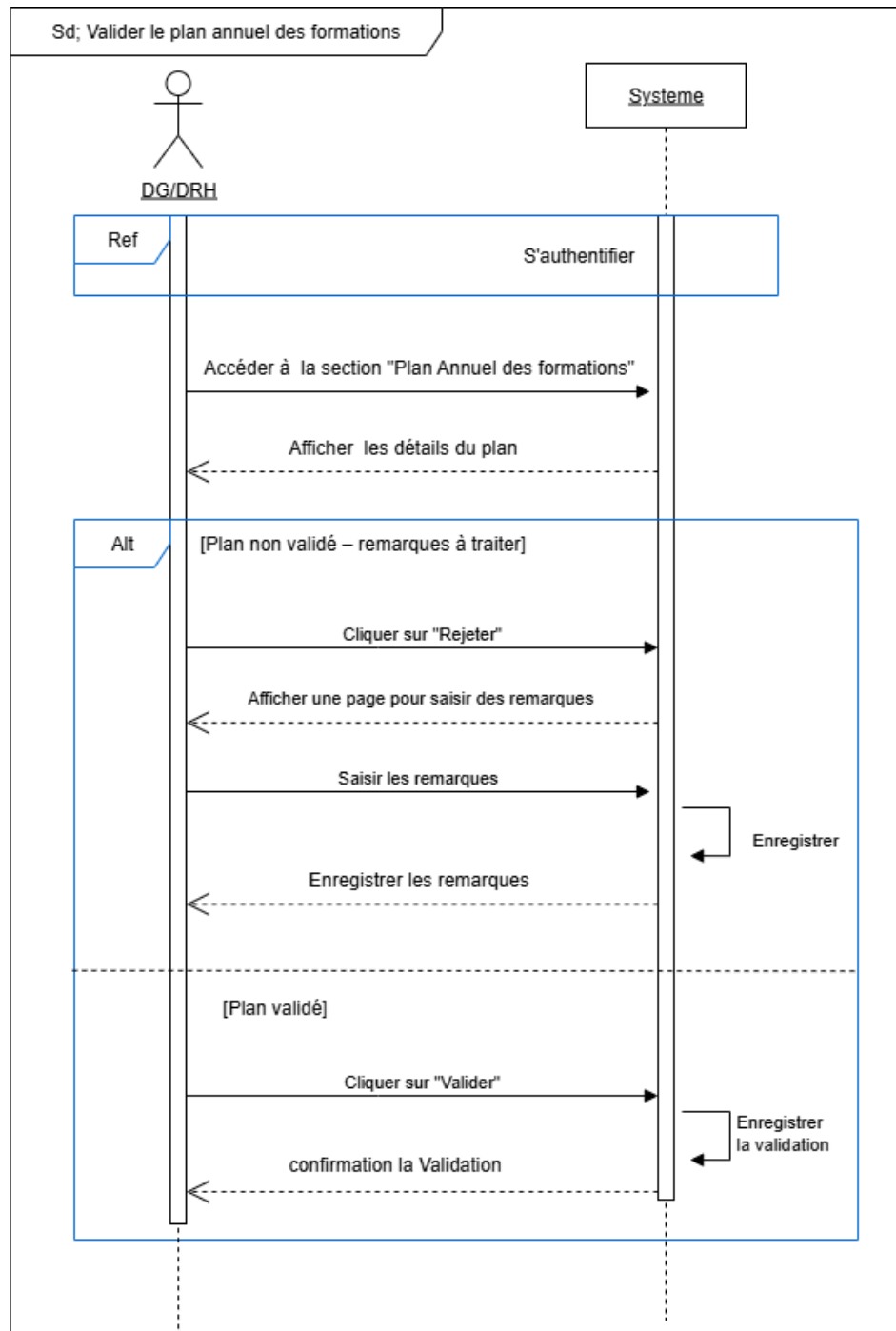


FIGURE 2.8 – Diagramme de séquence du cas d'utilisation « Valider le plan annuel des formations ».

2.9 Conclusion

Dans ce chapitre, nous avons identifié les principaux acteurs intervenant dans le processus de gestion des formations, ainsi que leurs interactions avec le système. Un diagramme de contexte a été élaboré afin de visualiser de manière claire les échanges entre le système et son environnement. Nous avons également présenté les exigences fonctionnelles à travers les diagrammes de cas d'utilisation et de séquence, ce qui nous a permis d'avoir une vision globale des comportements attendus. Cette analyse constitue une base solide pour aborder, dans le chapitre suivant, la phase de conception détaillée de l'application.

Chapitre 3

Conception et Analyse du Domaine

3.1 Introduction

Avant de passer à l'implémentation de notre application, il est essentiel d'effectuer une phase de conception approfondie. Cette étape permet de structurer et de clarifier les principales fonctionnalités du système afin d'assurer une mise en œuvre cohérente et efficace.

Dans ce chapitre, nous présentons les éléments clés de cette conception. Nous débutons par les diagrammes de séquence, qui illustrent les interactions entre les acteurs et le système dans différents scénarios. Ensuite, nous décrivons le diagramme de classes, qui représente les entités principales du système, leurs attributs et leurs relations. Nous poursuivons avec le dictionnaire des données, qui détaille les informations manipulées par l'application, et terminons par le modèle relationnel.

3.2 Diagrammes d'interactions

Un diagramme de séquence détaillé est une représentation graphique qui décrit l'interaction des objets dans un système au fil du temps. Ces diagrammes représentent la séquence des messages échangés entre les objets et l'ordre dans lequel ces interactions se produisent[12].

La modélisation proposée repose sur trois classes principales :

La classe d'interface : Cette classe représente l'interface entre le système et ses utilisateurs ou acteurs externes. Elle gère les interactions, comme la saisie des données ou l'affichage des résultats.

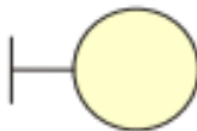


FIGURE 3.1 – Classe d'interface

La classe de contrôle : La classe de contrôle fait la jonction entre les classes d'interface et les classes entité. Elle contient la logique métier, les règles applicatives, et orchestre

le déroulement des cas d'utilisation.



FIGURE 3.2 – Classe de contrôle

La classe entité : Les classes entité représentent les objets métier ou données persistantes du système, issus du modèle du domaine.



FIGURE 3.3 – Classe entité

3.2.1 Diagramme d'interaction de cas d'utilisation «S'authentifier »

L'authentification est un pilier essentiel pour garantir la sécurité et la confidentialité des données au sein de notre application. Elle permet de vérifier l'identité de chaque utilisateur avant de lui donner accès aux fonctionnalités et ressources appropriées.

La figure 3.4 ci-dessous représente le diagramme d'interaction du cas d'utilisation « S'authentifier ».

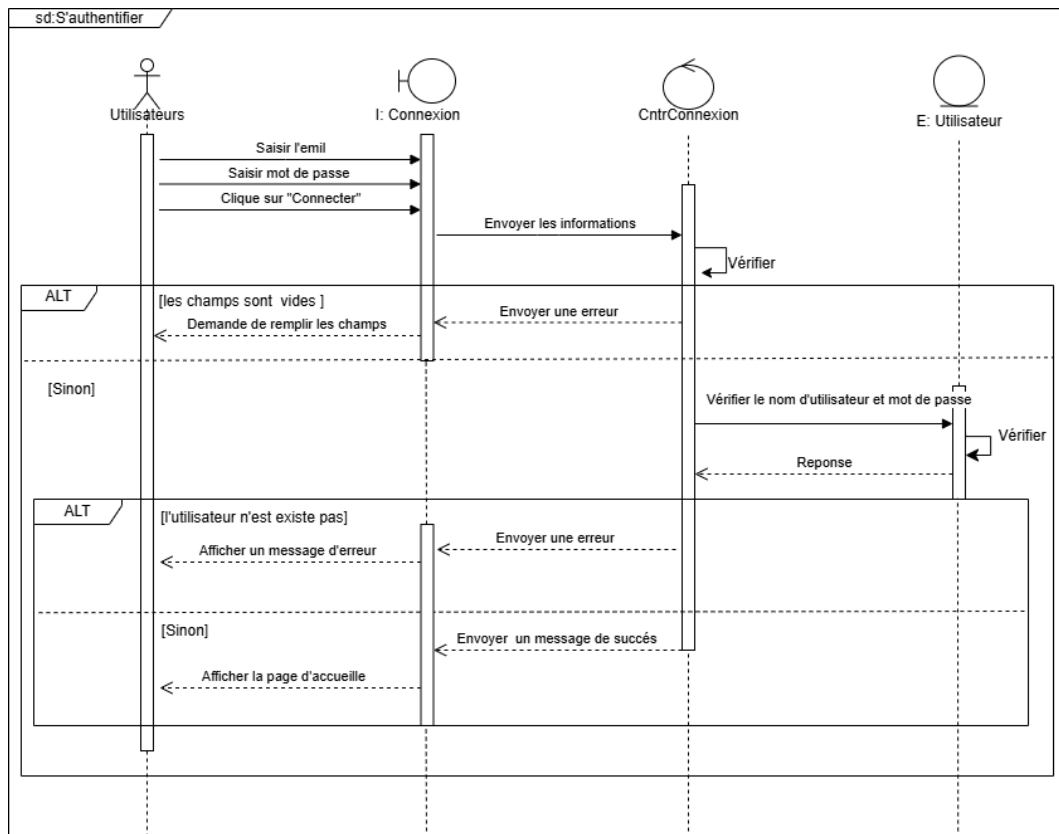


FIGURE 3.4 – Diagramme d'interaction du cas d'utilisation «S'authentifier »

3.2.2 Diagramme d'interaction du cas d'utilisation «Gestion et suivi des formations »

La gestion des formations permet de planifier, organiser et suivre les formations afin d'assurer leur bon déroulement et leur efficacité.

La figure 3.5 ci-dessous représente le diagramme d'interaction du cas d'utilisation « Gérer les formations ».

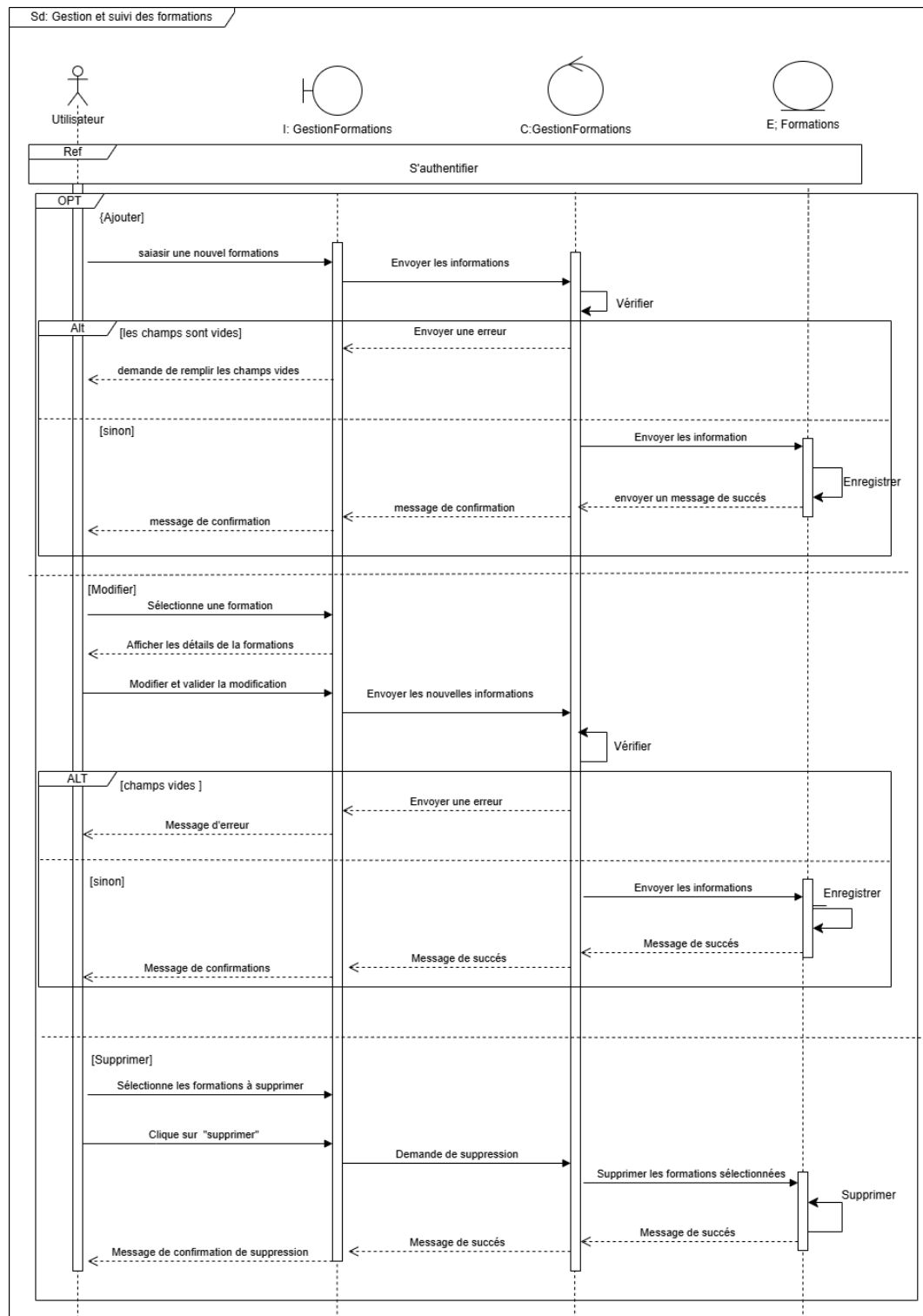


FIGURE 3.5 – Diagramme d’interaction de cas d’utilisation «Gestion et suivi des formations»

3.2.3 Diagramme d’interaction de cas d’utilisation « Valider le plan annuel des formations »

Valider le plan annuel permet d’accepter les formations prévues pour l’année afin de pouvoir les réaliser.

La figure 3.6 ci-dessous représente le diagramme d'interaction du cas d'utilisation « Valider le plan annuel des formations ».

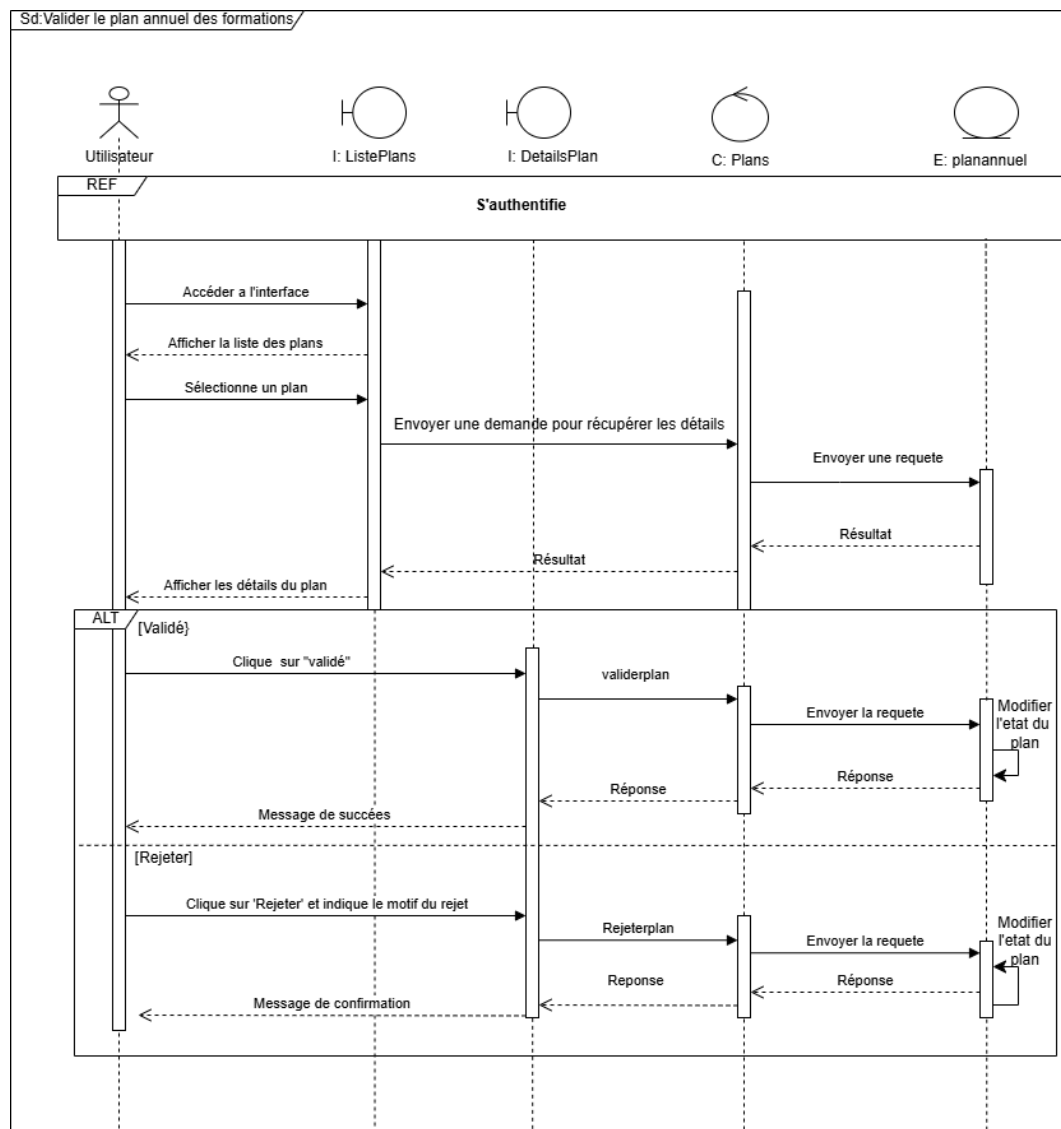


FIGURE 3.6 – Diagramme d'interaction du cas d'utilisation «Valider le plan annuel des formations »

3.3 Diagramme de classe

Le diagramme de classes est considéré comme le plus important dans la modélisation orientée objet, et constitue souvent un élément obligatoire dans ce contexte. Il fournit une vue globale et statique d'un système en présentant ses classes, interfaces, collaborations, ainsi que les relations entre elles. Contrairement aux diagrammes dynamiques, il montre ce qui interagit, mais pas comment ces interactions se déroulent dans le temps[10].

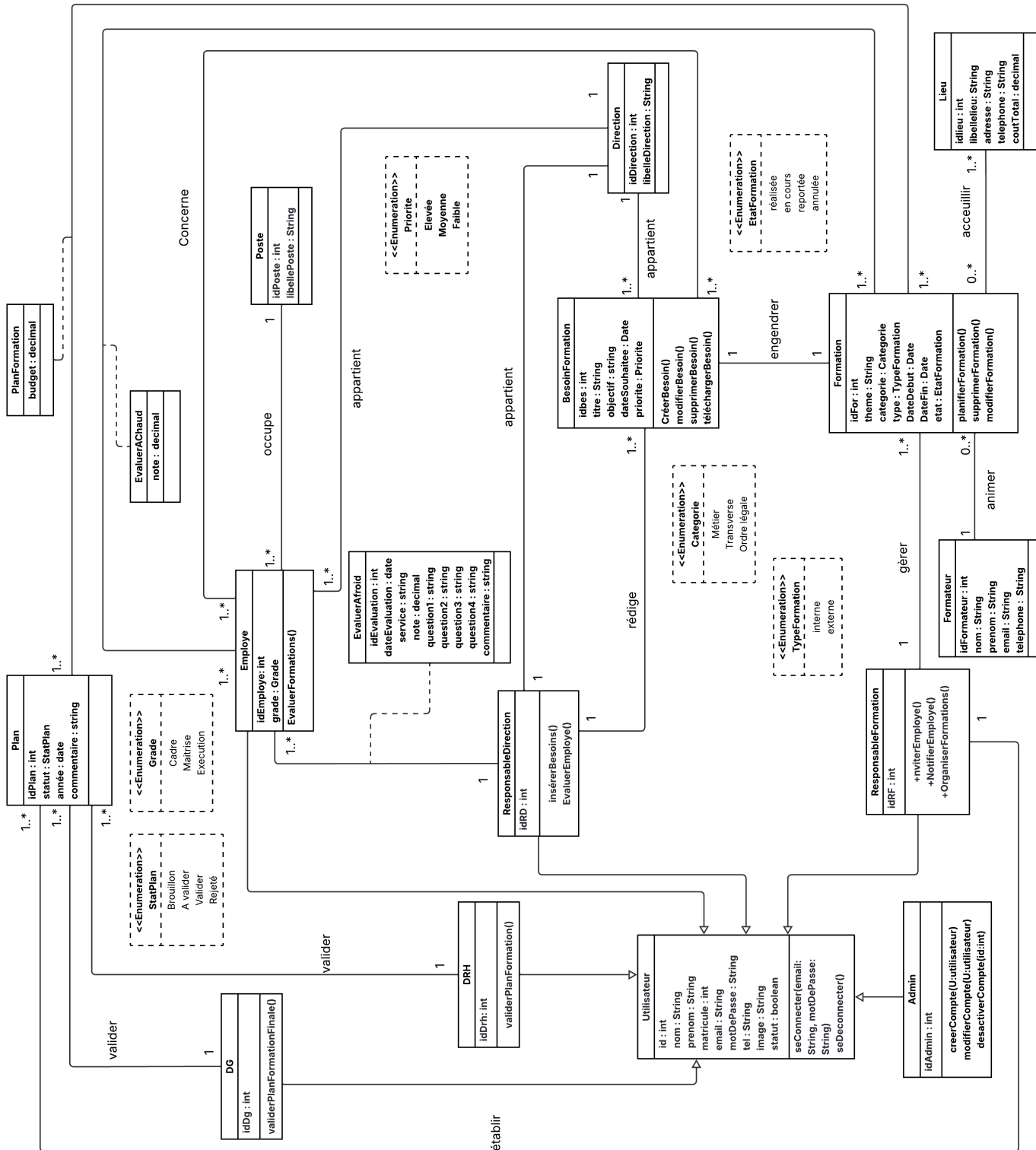


FIGURE 3.7 – Diagramme de classe

3.4 Dictionnaire de données

Le tableau ci-dessous présente une description détaillée des attributs de notre diagramme de classes pour en faciliter la compréhension.

TABLE 3.1 – Description des attributs du diagramme de classe

Classe	Responsabilité	Attributs		
		Désignation	Signification	Type
Utilisateurs	Gérer les comptes utilisateurs du système	id	Identifiant unique pour chaque utilisateur	Entier
		matricule	Matricule interne des utilisateurs (peut être utilisé comme ID RH)	Entier
		nom	Nom de l'utilisateur	Chaîne de caractères
		prenom	Prenom de l'utilisateur	Chaîne de caractères
		email	adresse email de l'utilisateur (utilisé pour la connexion)	Chaîne de caractères
		motDepasse	Mot de passe de l'utilisateur (doit être stocké d'une façon chiffrée)	Chaîne de caractères
		tel	Numéro de téléphone	Chaîne de caractères
		image	Chemin ou données de la photo utilisateur	Texte
formations	Gérer les formations planifiées et réalisées	idFor	Identifiant unique de la formation	Entier
		catégorie	Catégorie de la formation	Énumération
		type	Type de la formation	Énumération
		date_debut	Date début de la formation	Date
		date_fin	Date fin de la formation	Date
		etat	Etat ou statut actuel de la formation	Énumération
		theme	Theme principal de la formation	Chaîne de caractères

Classe	Responsabilité	Attributs		
		Désignation	Signification	Type
employe	Stocké les informations professionnells des employes	idemploye	Identifiant unique de l'employe	Entier
		grade	Grade hierarchique de l'employé (Maitrise ,Exécution , Cadre)	Énumération
Admin	Stocker les informations de l'administrateur	idAdmin	Identifiant unique de l'administrateur	Entier
Responsable formation	Stocker les informations des responsables formation	idRF	Identifiant unique du responsable formation	Entier
Responsable direction	Stocker les informations des responsables direction	idRD	Identifiant unique du responsable direction	Entier
DRH	Stocker les informations du DRH	idDRH	Identifiant unique du DRH	Entier
DG	Stocker les informations du DG	idDG	Identifiant unique du DG	Entier
besoininformation	Gérer les besoins en formations exprimés par les directions	idbes	Identifiant unique du besoin	Entier
		titre	Intitulé du besoin en formation exprimés	Chaîne de caractères
		objectif	Objectif de la formations	Texte
		datesouhaite	Date souhaitée pour la realisation de la formation	Date
		priorité	Degré de priorité du besoin exprimés	Énumération
direction	Lister les differentes directions de l'entreprise	iddirection	Identifiant unique de la direction	Entier
		libelledirection	nom de la direction	Chaîne de caractères
poste	Lister les differentes postes de l'entreprise	id_poste	Identifiant unique du poste	Entier
		libelleposte	nom du poste	Chaîne de caractères

Classe	Responsabilité	Attributs		
		Désignation	Signification	Type
lieu	Lister les lieux disponible pour les formations	idlieu	Identifiant unique du lieu	Entier
		libellelieu	nom du lieu	Chaîne de caractères
		telephone	Téléphone du lieu	Chaîne de caractères
		adresse	adresse du lieu	Chaîne de caractères
		coutTotal	Budget associé au lieu	Nombre décimal
formateur	Gérer les informations sur les formateurs	idformateur	Identifiant unique du formateur	Entier
		nom	Nom du formateur	Chaîne de caractères
		prenom	Prénom du formateur	Chaîne de caractères
		email	adresse email du formateur	Chaîne de caractères
		telephone	Numéro de téléphone du formateur	Chaîne de caractères
plan	Stocké les plan annuel des formations	idplan	Identifiant unique pour le plan	Entier
		commentaire	Informations ou commentaires sur le plan	Texte
		statut	Etat du plan annuelle (rejeté ou validé)	Énumération
		annee	Année à laquelle se rapporte le plan	Entier
Plan_Formation	Associer des budget aux formations	idpf	Identifiant unique	Entier
		budget	Montant budgétaire	Nombre décimal
evaluationchaud	Evaluation immédiates des formations par les participants	idevaluation	Identifiant unique	Entier
		note	Note attribuée	Nombre décimal
evaluation_froid	Evaluation a froid des formation par les responsable	id_evaluation	Identifiant unique	Entier
		note	Note attribuée	Nombre décimal
		question 1 a4	Reponses aux question d'évaluation	Texte
		date_evaluation	Date de l'évaluation	Date
		service	Service concerné	Chaîne de caractères
		commentaire	commentaire supplémentaire	Texte

3.5 Schéma relationnel

Le modèle relationnel est un modèle dans lequel les données sont organisées sous forme de tables qui sont associées entre elles. Une table correspond à une relation. Ce modèle permet d'indiquer toutes les caractéristiques d'une table de données, avec notamment les liens entre les différentes relations [11].

Dans ce qui suit, nous allons convertir notre diagramme de classes en schéma relationnel.

3.6 Passage au modèle relationnel

Après application des règles de convertissement, nous obtenons le schéma relationnel suivant :

Utilisateur(id, nom, prenom, matricule, email, motDePasse, tel, image, statut)

Admin(idAdmin, #idUtilisateur).

DG(idDg, #idUtilisateur).

DRH(idDrh, #idUtilisateur).

ResponsableFormation(idRF, #idUtilisateur).

ResponsableDirection(idRD, #idUtilisateur).

Employe(idEmploye, grade, #idUtilisateur, #idPoste, #idDirection).

Direction(idDirection, libelleDirection).

Poste(idPoste, libellePoste).

Lieu(idLieu, libelleLieu, adresse, telephone).

Formateur(idFormateur, nom, prenom, email, telephone).

BesoinFormation(idbes, titre, objectif, dateSouhaitee, priorite, #idDirection).

Concerne(#idbes, #idEmploye).

Formation(idFor, theme, categorie, type, dateDebut, dateFin, etat, #idbes, #idLieu, #idFormateur).

Plan(idPlan, statut, année, commentaire, #idFor).

PlanFormation(#idPlan, #idFor, budget).

EvaluerAChaud(#idEmploye, #idFor, note).

EvaluerAFroid(idEvaluation, dateEvaluation, service, note, question1, question2, question3, question4, commentaire, #idEmploye, #idRD).

3.7 Conclusion

Dans ce chapitre, nous avons exploré en détail la conception de notre projet, en nous appuyant sur des outils tels que des diagrammes de séquence détaillés et un diagramme de classes. Afin de clarifier davantage la structure de notre système, nous avons également mis en place un dictionnaire de données, explicitant les attributs de chaque classe. Cette approche a permis d'assurer une compréhension unifiée des composantes fondamentales de notre système.

Dans le chapitre suivant, nous allons passer à l'étape de réalisation de notre projet. Nous utiliserons les outils et les technologies appropriés pour mettre en œuvre notre plateforme.

Chapitre 4

Réalisation

4.1 Introduction

Dans ce chapitre, nous abordons la phase pratique de notre projet en présentant l'ensemble des outils, technologies et environnements de développement retenus pour la réalisation de notre plateforme. Nous détaillerons d'abord les logiciels et les frameworks sélectionnés. Ensuite, nous détaillerons les langages de programmation utilisés. Enfin, nous conclurons ce chapitre en exposant quelques interfaces de cette plateforme.

4.2 Technologies et Outils Utilisés

4.2.1 Outils et plateformes utilisés

Pour le développement de notre application, nous avons utilisé les outils et technologies suivants :

Lucidchart : Lucidchart est une plateforme cloud intuitive qui permet de créer facilement des diagrammes et des schémas, quel que soit l'appareil, le navigateur ou le système d'exploitation utilisé. Elle facilite la collaboration en temps réel grâce à des fonctionnalités comme la co-crédation simultanée, le chat intégré, les commentaires spécifiques à une forme et les curseurs collaboratifs. Lucidchart propose une vaste bibliothèque de formes pour divers types de diagrammes (UML, organigrammes, diagrammes EA, etc.)[\[13\]](#).



FIGURE 4.1 – Logo Lucidchart.

Visual Studio Code : Visual Studio Code (VSCode) est un éditeur de code source et un environnement de développement intégré (IDE) de Microsoft. Il est open-source et cross-platform, c'est-à-dire qu'il fonctionne sur Windows, Linux et Mac. Il a été conçu pour les développeurs web, mais il prend en charge de nombreux autres langages de programmation tels que C++, C, Python, Java, etc. Il offre de nombreuses fonctionnalités comme la

coloration syntaxique, l'auto-complétion, la mise en évidence des erreurs, la navigation de code, le débogage, la gestion de versions, l'intégration avec Git, et beaucoup d'autres [14].

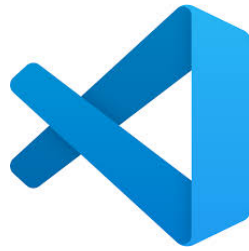


FIGURE 4.2 – Logo Visual Studio Code.

Xampp : est un ensemble de logiciels simplifiant la création d'interfaces web interactives avec des bases de données. Cette suite multiplateforme (X) intègre Apache comme serveur web, MySQL/MariaDB pour la gestion des bases de données, ainsi que PHP et Perl pour le développement côté serveur. Offrant un environnement local prêt à l'emploi, XAMPP permet aux développeurs de concevoir, tester et déboguer des applications web de manière efficace avant leur mise en production. Compatible avec Windows, Linux et macOS, c'est un outil indispensable pour le développement full-stack et l'apprentissage des technologies web [15].



FIGURE 4.3 – Logo XAMPP.

GitHub : GitHub est un service web et cloud qui permet aux développeurs de stocker, gérer et suivre les modifications apportées à leur code grâce au contrôle de version, en s'appuyant sur le système Git. Il facilite la collaboration entre individus ou équipes, en rendant plus accessible l'utilisation de Git, même pour les débutants. Sur GitHub, il est possible de créer des branches pour travailler sur différentes parties d'un projet, de fusionner ces modifications dans la branche principale, de soumettre des pull requests pour proposer des changements, et de forker des projets pour en faire des copies indépendantes. GitHub est largement utilisé pour les projets open source, mais propose aussi des offres privées et des outils pour les entreprises[16].



FIGURE 4.4 – Logo Github.

4.2.2 Langages et frameworks

JavaScript : JavaScript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que Node.js, Apache CouchDB voire Adobe Acrobat. Le code JavaScript est interprété ou compilé à la volée (JIT). C'est un langage à objets utilisant le concept de prototype, disposant d'un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orientée objet [17].

JSX : JSX (JavaScript XML) est une extension de syntaxe utilisée dans React pour décrire l'interface utilisateur sous forme de code JavaScript. Elle permet de mélanger du code JavaScript avec des balises HTML, ce qui rend la création d'interfaces utilisateur en React plus lisible. Il est important d'être à l'aise avec cette syntaxe le plus tôt possible. JSX facilite la création d'interfaces dynamiques et modulaires en combinant la logique et la présentation dans un même fichier[18]

React : React est une bibliothèque JavaScript développée par Meta qui permet de créer des interfaces utilisateur dynamiques et interactives en construisant des composants réutilisables. Elle utilise une syntaxe appelée JSX, qui combine JavaScript et HTML, facilitant ainsi la création d'éléments visuels. React fonctionne avec un DOM virtuel, ce qui optimise les performances en ne mettant à jour que les parties modifiées de la page. Cette approche rend le développement plus simple, rapide et maintenable. React se concentre uniquement sur la partie interface, ce qui le rend flexible et souvent utilisé avec d'autres outils pour créer des applications web modernes et performantes[19].

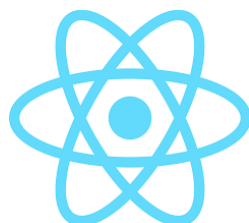


FIGURE 4.5 – Logo React.

Node.js : est un environnement d'exécution JavaScript côté serveur, construit sur le moteur V8 de Google Chrome. Il permet de créer des applications web rapides, évolutives et performantes grâce à une architecture orientée événements et non bloquante, facilitant la gestion de nombreuses requêtes simultanées. Node.js offre un écosystème riche avec son gestionnaire de paquets npm, permettant aux développeurs d'utiliser JavaScript à la fois côté client et côté serveur, ce qui simplifie le développement. Il est particulièrement adapté aux applications en temps réel, comme les chats en ligne ou les API RESTful[20].



FIGURE 4.6 – LogoNode.js.

Express : Express.js est un framework minimaliste et flexible pour Node.js qui fournit un ensemble robuste de fonctionnalités pour développer des applications web et des API. Il simplifie la gestion des routes, des requêtes et des réponses HTTP, et permet d'ajouter facilement des fonctionnalités via des middlewares. Express est largement utilisé pour créer des applications backend évolutives et performantes, notamment des applications web monopages, multipages ou hybrides[21].

Bootstrap5 : est un framework front-end gratuit qui aide les développeurs à créer des sites web responsives et modernes rapidement. Il fournit des styles CSS et des composants prêts à l'emploi, comme des boutons, formulaires et menus, ainsi que des fonctionnalités JavaScript sans dépendance à jQuery. Conçu en mode mobile-first, Bootstrap 5 facilite l'adaptation des sites à tous les types d'écrans, du mobile au bureau[22].

SQL : SQL (Structured Query Language) est un langage informatique utilisé pour communiquer avec une base de données. Il permet aux développeurs de lire, insérer, modifier ou supprimer des données dans une base. Chaque système de gestion de base de données (SGBD) comme MySQL, PostgreSQL ou Oracle peut avoir ses spécificités, mais SQL reste le langage standard pour interagir avec les bases relationnelles. Comprendre le fonctionnement du stockage et de la lecture des données est essentiel pour optimiser les performances des requêtes SQL[23].

4.2.3 Bibliothèques utilisées

cloudinary : est à la fois une plateforme cloud complète de gestion, transformation, optimisation et diffusion d'images et de vidéos, et une bibliothèque (SDK) qui permet aux développeurs d'intégrer facilement ces fonctionnalités dans leurs applications via du code. Ces SDK (Software Development Kits) fournissent des outils et des bibliothèques pour uploader, transformer et livrer des médias optimisés de manière fluide et sécurisée, en s'adaptant à différents environnements de développement (Node.js, React, Android, etc.)[24].

joi : est une bibliothèque JavaScript puissante et flexible dédiée à la validation des données. Elle permet de définir des schémas de validation clairs et déclaratifs pour garantir que les données reçues par une application (par exemple via des formulaires ou des API) respectent des règles précises (types, formats, contraintes, etc.). Joi facilite la création de règles complexes avec un minimum de code, supporte la validation asynchrone, et fournit des messages d'erreur personnalisables pour une meilleure gestion des erreurs. Elle est très utilisée dans les projets Node.js pour assurer l'intégrité et la qualité des données avant traitement[25].

nodemailer : est une bibliothèque Node.js qui facilite l'envoi d'e-mails depuis un serveur en créant un objet transporteur SMTP permettant d'envoyer des messages en texte brut, HTML, avec pièces jointes, à un ou plusieurs destinataires. Elle offre une API simple pour configurer, sécuriser et gérer l'envoi d'e-mails, compatible avec plusieurs plateformes et services de messagerie [26].

4.3 Structure du code source

Dans le cadre du projet, nous avons adopté une architecture claire et modulaire en séparant le **backend** (côté serveur) et le **frontend** (côté client), chacun organisé en sous-dossiers selon les responsabilités fonctionnelles.

Backend

- **Controllers (Contrôleurs)** : Composants simples qui reçoivent les requêtes HTTP (comme POST, GET...) et les transmettent aux services pour traitement. Ils ne contiennent pas de logique métier.
- **Services (Services)** : Contiennent la logique métier, c'est-à-dire le cœur du fonctionnement de l'application (calculs, règles, traitements...).
- **Models (Modèles)** : Gèrent les interactions avec la base de données (lecture, ajout, modification, suppression). Ils représentent les structures de données.
- **Middleware** : Gèrent des fonctionnalités transversales, c'est-à-dire qui interviennent entre la requête et la réponse (par exemple : vérification de l'authentification, gestion des erreurs, journalisation...).
- **Utils (Utilitaires)** : Fonctions d'aide réutilisables dans différentes parties du code (exemples : formatage de dates, génération de mots de passe, etc.).

Frontend

- **Pages** : Chaque route de l'application correspond à une page (exemples : `/home`, `/login`). Chaque page est un composant principal affiché selon l'URL.
- **Components (Composants)** : Petites parties de l'interface utilisateur (UI), réutilisables dans plusieurs pages (par exemple : boutons, champs de formulaire, cartes...).
- **Hooks** : Fonctions personnalisées dans React qui permettent de gérer l'état, les effets secondaires, ou d'extraire de la logique réutilisable (comme `useAuth`, `useForm`...).
- **API Services** : Fonctions centralisées pour envoyer des requêtes au backend (par exemple : récupérer des données, envoyer un formulaire, etc.).
- **Utils (Utilitaires)** : Fonctions d'aide côté client, comme le formatage d'un texte, la validation d'un champ de formulaire, etc.

Cette organisation a permis une meilleure séparation des responsabilités, un code plus maintenable, et une meilleure collaboration entre les différentes parties du développement.

4.4 Présentation des interfaces de notre plateforme

Dans ce qui suit, nous présentons quelques interfaces de notre plateforme.

4.4.1 Interface d'Authentification

L'accès à l'application est sécurisé par une authentification (email et mot de passe) pour tous les profils utilisateurs : Administrateur, Responsable de formation, Direction,

DRH et Employés. La capture d'écran 4.7 montre l'interface de connexion sur PC et smartphone.

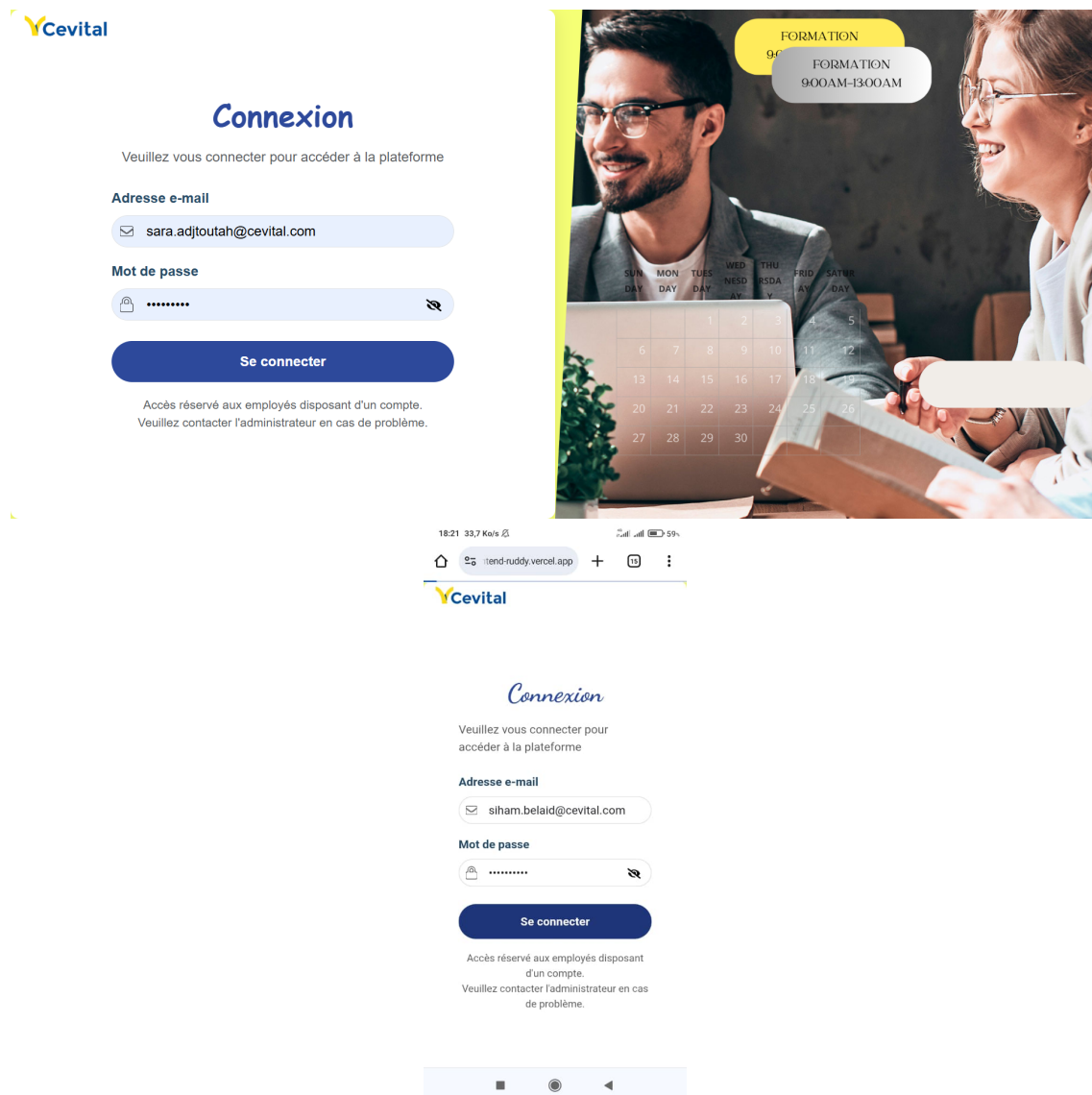


FIGURE 4.7 – Interface d'authentification

4.4.2 Interface des statistiques de formation

La figure 4.8 représente le tableau de bord du système de gestion de la formation. Il offre une vue synthétique sur les indicateurs clés tels que le nombre de formations, d'utilisateurs, de besoins enregistrés, ainsi que le suivi des évaluations.

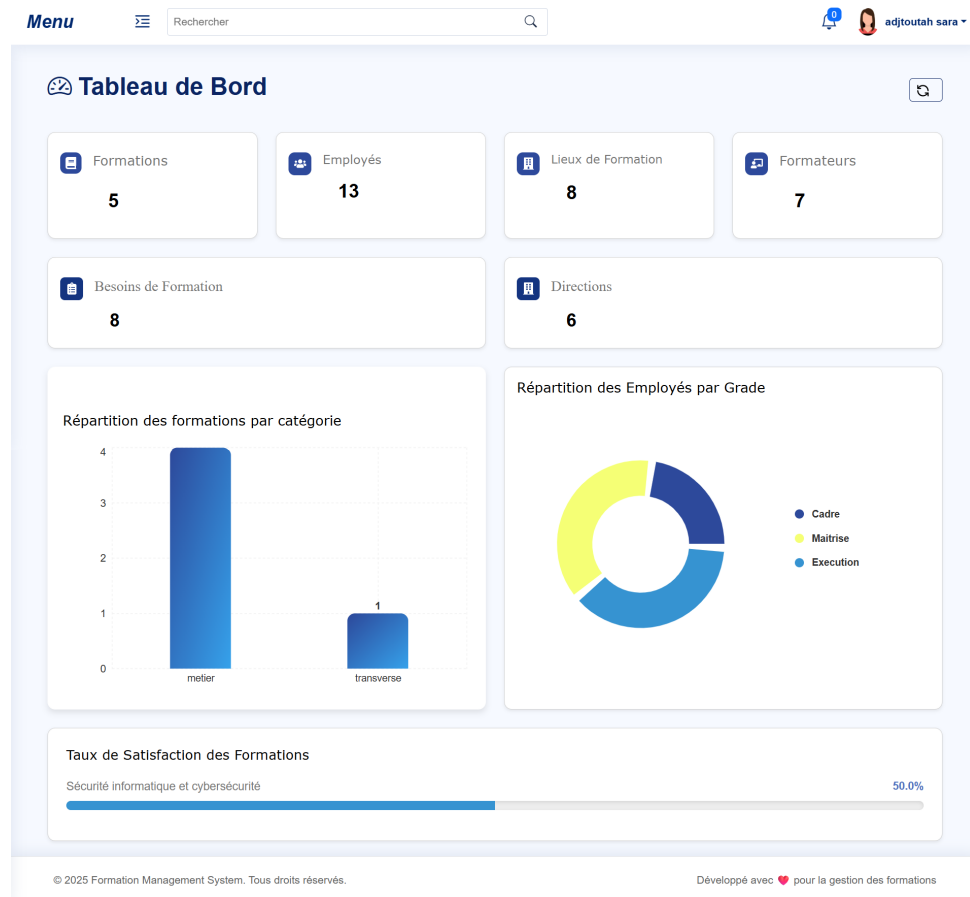


FIGURE 4.8 – Tableau de bord analytique du système de formation

4.4.3 Interface des besoins en formation

La figure 4.9 présente l'interface Besoin en formation, destinée au responsable de direction. Celle-ci lui permet de renseigner et de consulter les besoins en formation de son service.

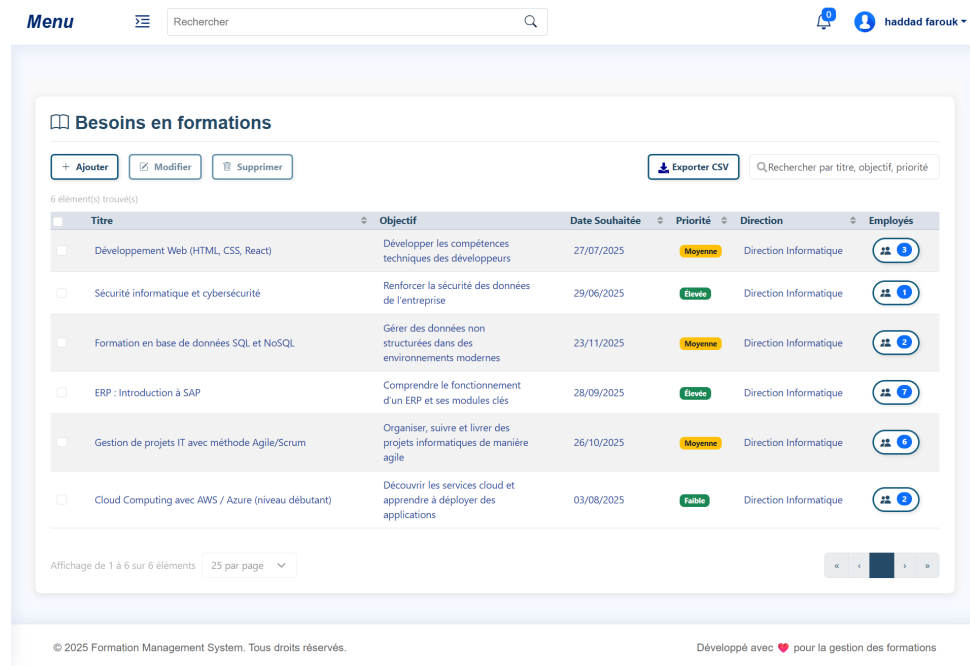


FIGURE 4.9 – Interface Besoin en formations

Formulaire d'ajout d'un besoin

La figure 4.10 illustre la fenêtre modale utilisée pour l'ajout d'un besoin en formation.

FIGURE 4.10 – Formulaire d'ajout d'un besoin en formation.

4.4.4 Interface de planification des formations

L'interface de planification offre au service formation la capacité de créer, éditer et supprimer des sessions de formation. Elle permet une gestion structurée des propositions avant la validation finale du plan annuel. La capture d'écran 4.11 illustre l'interface.

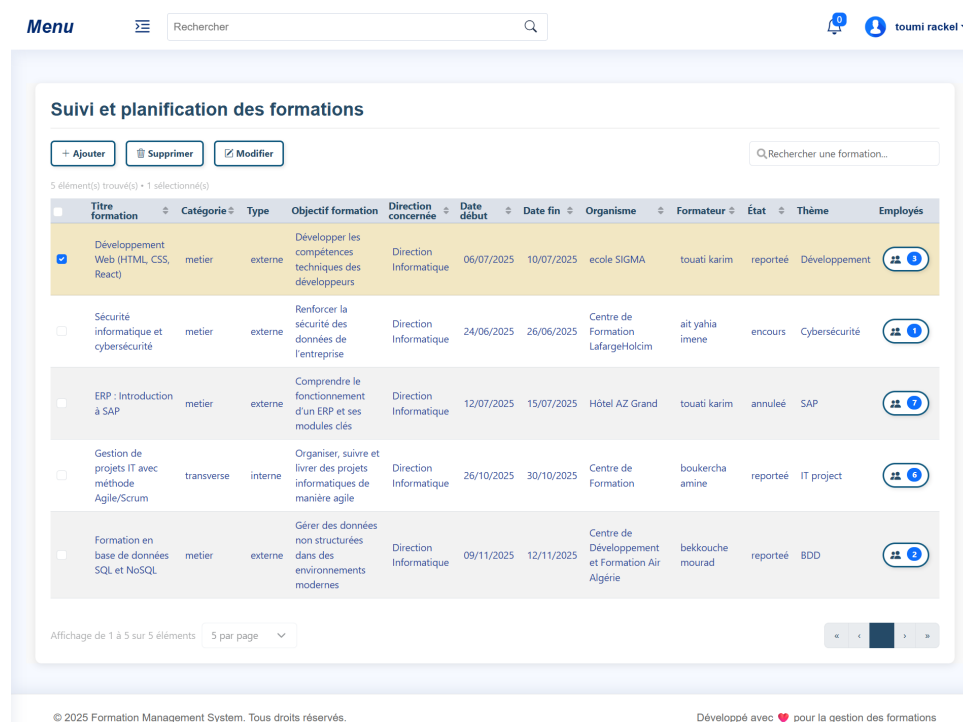


FIGURE 4.11 – Interface Suivi et planification des formations

4.4.5 Interface du plan annuel

La figure 4.12 représente l'interface qui permet au service formation d'établir le plan annuel en ajoutant les formations et les budgets associés.

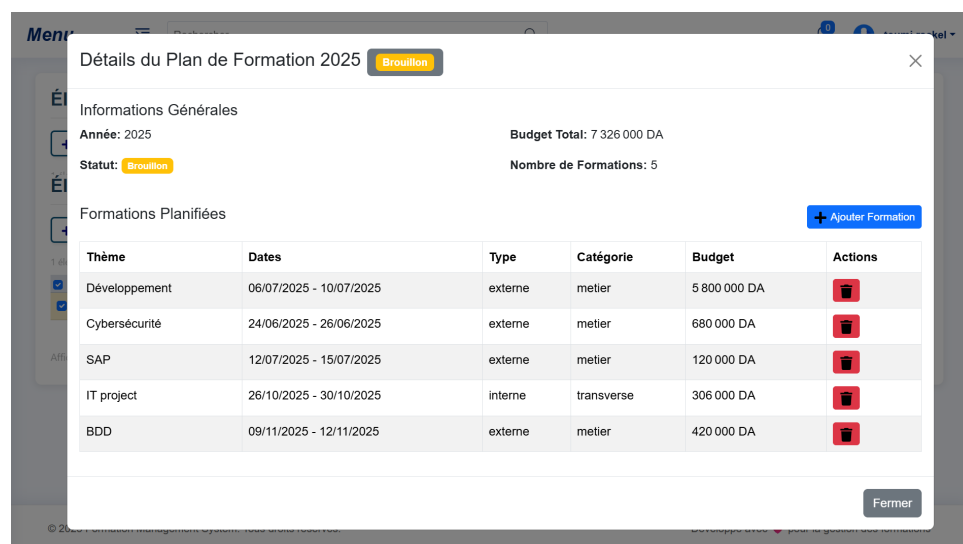


FIGURE 4.12 – Interface du plan annuel de formation

4.4.6 Interface de validation du plan annuel

La figure 4.13 représente l'interface dédiée à la validation du plan annuel de formation. Elle permet au Directeur Général (DG) de consulter les détails du plan proposé, puis de prendre une décision en le validant ou en le rejetant. En cas de rejet, un champ

de commentaire est prévu afin que le DG puisse justifier sa décision et transmettre ses remarques au service formation.

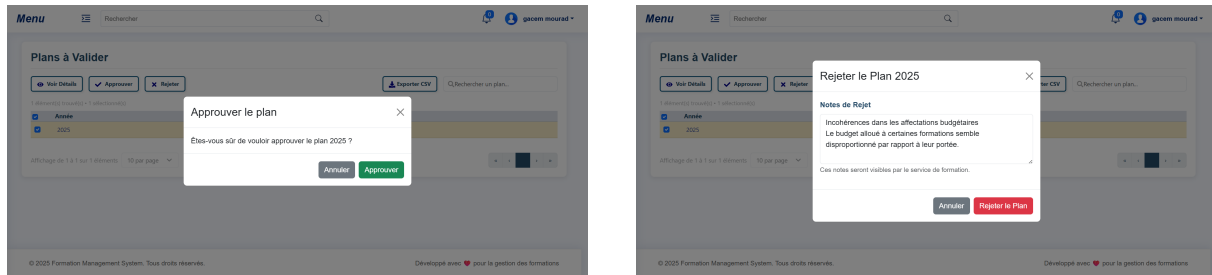


FIGURE 4.13 – Interface de validation du plan annuel

4.4.7 Interface d'évaluation à froid

La figure 4.14 représente l'interface de l'évaluation à froid, qui permet au responsable de direction d'évaluer l'impact des formations.

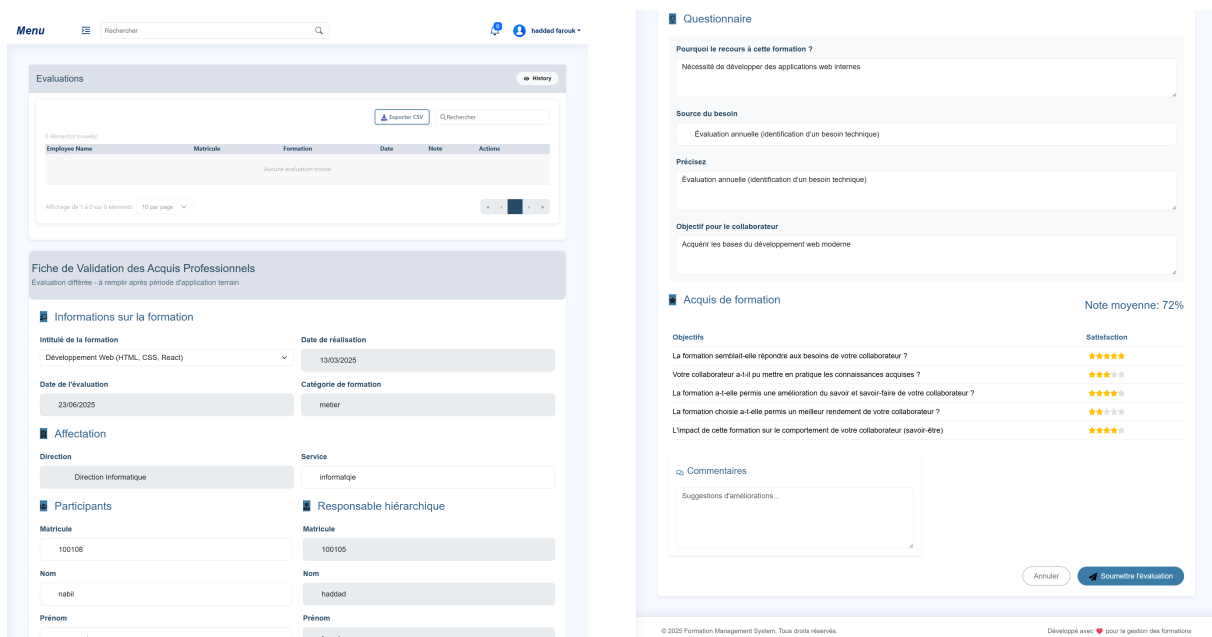


FIGURE 4.14 – Interface de l'évaluation à froid

4.4.8 Interface d'évaluation à chaud

La figure 4.15 représente l'interface de l'évaluation à chaud, qui permet à l'employé d'évaluer la formation juste après y avoir participé, en donnant son avis sur son déroulement, son contenu et l'efficacité du formateur.

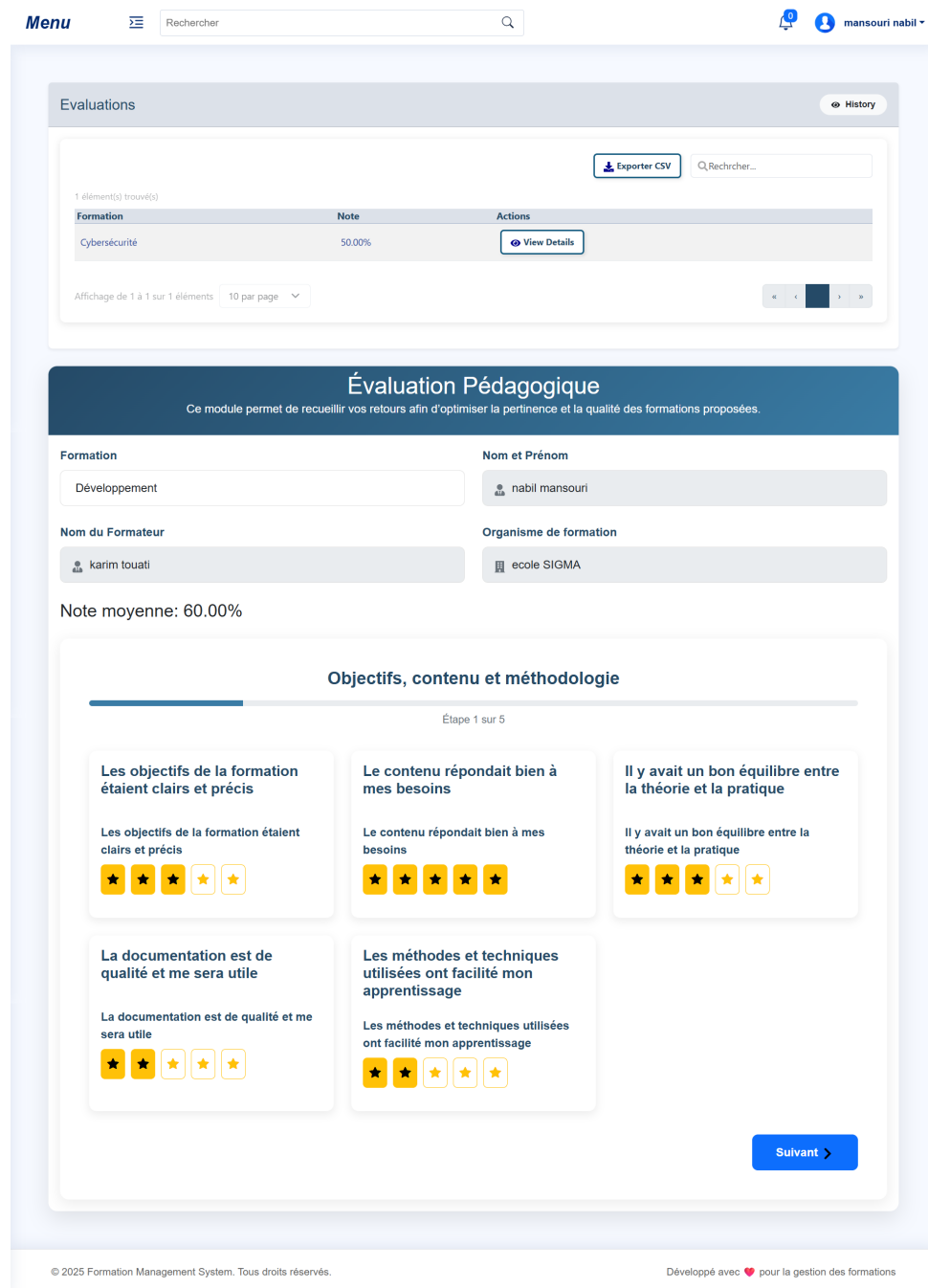


FIGURE 4.15 – Interface de l'évaluation à chaud

4.5 Conclusion

ce chapitre a permis de présenter les choix technologiques et les outils essentiels ayant servi à la réalisation de notre plateforme. Après avoir détaillé les logiciels, frameworks et langages de programmation utilisés, nous avons illustré ces éléments par la présentation de plusieurs interfaces clés. Cette approche a permis de mieux comprendre les fondations techniques et fonctionnelles sur lesquelles repose notre application.

Conclusion générale

Ce mémoire représente le fruit d'une exploration approfondie dans le domaine de la conception et du développement d'une application web dédiée à la gestion des formations au sein de l'entreprise Cevital. L'objectif principal était de créer un outil performant et intuitif permettant de faciliter la planification, le suivi et l'évaluation des formations, contribuant ainsi à l'amélioration continue des compétences des collaborateurs. Cette solution vise à optimiser la gestion des ressources humaines en offrant une meilleure organisation des parcours de formation, tout en répondant aux besoins spécifiques de l'entreprise.

Dans un premier temps, une étude préalable a permis de présenter en détail l'entreprise où s'est déroulé notre stage, ce qui nous a offert l'opportunité de côtoyer le monde professionnel afin de bien comprendre son organisation, son fonctionnement et son contexte spécifique lié à la gestion des formations. Cette étape a constitué la base nécessaire pour appréhender les enjeux et les contraintes du projet. Par la suite, l'analyse des besoins a conduit à l'identification des acteurs clés impliqués dans le système, ainsi qu'à la définition précise des besoins fonctionnels et non fonctionnels auxquels notre solution devait répondre. Pour structurer et piloter efficacement le projet, nous avons adopté la méthodologie du Processus Unifié (UP), reconnue pour son approche itérative et centrée sur les cas d'utilisation. Par ailleurs, le recours au langage de modélisation UML a facilité la formalisation des différents aspects du système, à travers des diagrammes clairs et compréhensibles, assurant ainsi une meilleure communication entre les parties prenantes et une conception rigoureuse. Cette démarche méthodologique a permis d'organiser le développement en plusieurs phases successives, garantissant une progression maîtrisée et adaptée aux besoins identifiés.

L'application conçue offre une gestion optimisée et sécurisée des formations, assurant une planification, un suivi et une consultation simplifiés. Elle favorise une collaboration harmonieuse entre les différents utilisateurs, garantissant ainsi la fluidité des échanges et la transparence des informations. Chaque élément technique a fait l'objet d'une analyse approfondie, avec une attention soutenue portée aux obstacles rencontrés et aux solutions techniques déployées pour assurer une performance fiable et une expérience utilisateur optimale.

Enfin, ce mémoire illustre notre volonté de mettre à profit les compétences en ingénierie informatique pour améliorer les processus de gestion des formations, contribuant ainsi au développement des compétences et à la performance des organisations. Malgré les contraintes de temps qui ont limité l'ampleur du projet, cette expérience ouvre la voie à de nombreuses améliorations possibles, telles que l'ajout de fonctionnalités complémentaires comme la diffusion automatisée des attestations de formation, la génération de rapports détaillés, ou encore l'intégration de modules d'analyse avancée.

Ces évolutions permettraient d'optimiser davantage la gestion des ressources humaines et d'accompagner efficacement la montée en compétences dans un environnement professionnel en constante évolution. Ce travail se veut ainsi une source d'inspiration pour de futurs projets visant à renforcer et à enrichir les outils dédiés à la formation.

Bibliographie

- [1] CEVITAL, *L'histoire du groupe*, 2025. Disponible en ligne : <https://www.cevital.com/lhistoire-du-groupe/>. [Consulté le 04 02 2025].
- [2] CEVITAL Agro-Industrie, *Qui sommes-nous ?*, 2025. Disponible en ligne : <https://www.cevital-agro-industrie.com/qui-sommes-nous/>. [Consulté le 05 02 2025].
- [3] Savoir Plus, *Besoins fonctionnels et non fonctionnels*, 2025. Disponible en ligne : <https://savoir.plus/besoins-fonctionnels-non-fonctionnels/>. [Consulté le 10 02 2025].
- [4] Lucidchart, *Langage UML*. Disponible en ligne : <https://www.lucidchart.com/pages/fr/langage-uml>. [Consulté le 15 02 2025].
- [5] L. Audibert, *Cours UML - Diagramme de cas d'utilisation*, Disponible en ligne : <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation#L2-2-1>, [Consulté le 02 03 2025]
- [6] Miro, *Diagramme contextuel – Modèles*, disponible sur : <https://miro.com/fr/modeles/diagramme-contextuel/>, [consulté le 05 03 2025].
- [7] IBM, *Diagrams Use Case*. Disponible en ligne : <https://www.ibm.com/docs/fr/dmrt/9.5?topic=diagrams-use-case>. [Consulté le 10 03 2025].
- [8] *Diagrams Use Case*. Disponible en ligne : <https://creately.com/fr/diagram-type/cas-d-utilisation/>. [Consulté le 13 03 2025].
- [9] *def diagramme sequence*. Disponible en ligne : <https://miro.com/fr/diagramme/qu-est-ce-qu-un-diagramme-de-sequence-uml/>. [Consulté le 15 03 2025].
- [10] Embarcadero Technologies. *Définition des diagrammes de classes UML 1.5*. Disponible en ligne : https://docwiki.embarcadero.com/.../Définition_des_diagrammes_de_classes_UML [Consulté le 20 04 2025].
- [11] Maxicours. *Comprendre le modèle relationnel d'une base de données*. Disponible en ligne : <https://www.maxicours.com/se/cours/comprendre-le-modele-relationnel-d-une-base-de-donnees/> [Consulté le 23 04 2025].
- [12] Miro. *Qu'est-ce qu'un diagramme de séquence UML ?* Disponible en ligne : <https://miro.com/.../diagramme-de-sequence-uml>. [Consulté le 29 04 2025].
- [13] Lucid Software. *Lucidchart : Créez des diagrammes UML en ligne*. Disponible en ligne : <https://lucid.co/fr/produit/lucidchart>. [Consulté le 28 05 2025].
- [14] Bility. *Définition Visual Studio Code*. Disponible en ligne : <https://bility.fr/definition-visual-studio-code/>. [Consulté le 28 05 2025].
- [15] Université de Liège. (2014). *Introduction XAMPP*. Disponible en ligne : https://services.montefiore.uliege.be/verif/cours/bd/repet2014/tp7_slides.pdf. [Consulté le 30 05 2025].

- [16] Kinsta. *Base de connaissances GitHub*. Disponible en ligne : <https://kinsta.com/fr/base-de-connaissances/base-de-connaissances-github/>. [Consulté le 30 05 2025].
- [17] MDN Web Docs. *JavaScript – Documentation MDN*. Disponible en ligne : <https://developer.mozilla.org/fr/docs/Web/JavaScript>. [Consulté le 02 06 2025].
- [18] Bensaid, R. *React – Développez le front-end de vos applications web et mobiles avec JavaScript (Nouvelle édition) : Découvrir le JSX*. Éditions ENI. Disponible en ligne : <https://www.editions-eni.fr/livre/react-developpez-le-front-end-de-vos-applications-web-et-mobiles-avec-javascript-nouvelle-edition-9782409043260/decouvrir-le-jsx>. [Consulté le 02 06 2025].
- [19] Hostinger. *Qu'est-ce que React ?*. Disponible en ligne : <https://www.hostinger.com/fr/tutoriels/quest-ce-que-react>. [Consulté le 02 06 2025].
- [20] Node.js. *À propos de Node.js*. Disponible en ligne : <https://nodejs.org/fr/about>. [Consulté le 12 06 2025].
- [21] Kinsta. *Qu'est-ce qu'Express.js ?*. Disponible en ligne : <https://kinsta.com/fr/base-de-connaissances/qu-est-express-js/>. [Consulté le 04 06 2025].
- [22] W3Schools. *Bootstrap 5 Get Started*. Disponible en ligne : https://www.w3schools.com/bootstrap5/bootstrap_get_started.php. [Consulté le 04 06 2025].
- [23] SQL.sh. *Cours et tutoriels sur le langage SQL*. Disponible en ligne : <https://sql.sh/>. [Consulté le 12 06 2025].
- [24] Cloudinary. *Cloudinary SDKs Documentation*. Disponible en ligne : https://cloudinary.com/documentation/cloudinary_sdks. [Consulté le 12 06 2025].
- [25] Cursa. *Validation des données avec le package Joi*. Disponible en ligne : <https://curse.app/fr/page/validation-des-donnees-avec-le-package-joi>. [Consulté le 12 06 2025].
- [26] Nodemailer. *Official Nodemailer Documentation*. Disponible en ligne : <https://nodemailer.com>. [Consulté le 15 06 2025].

Annexe A

Diagramme de cas d'utilisation – Acteur

Diagramme de cas d'utilisation – Acteur : « Admin »

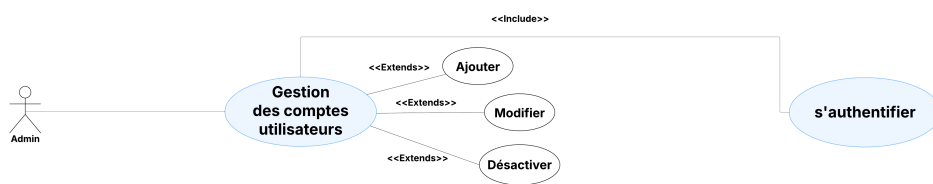


FIGURE 4.16 – Diagramme de cas d'utilistion de l'administrateur

Diagramme de cas d'utilisation – Acteur : « Responsable direction »

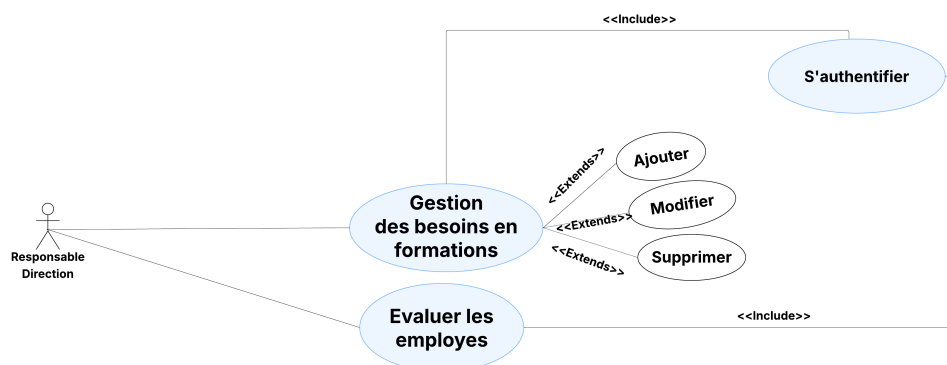


FIGURE 4.17 – Diagramme de cas d'utilistion de Responsable direction

Diagramme de cas d'utilisation – Acteur : « Responsable de formation »

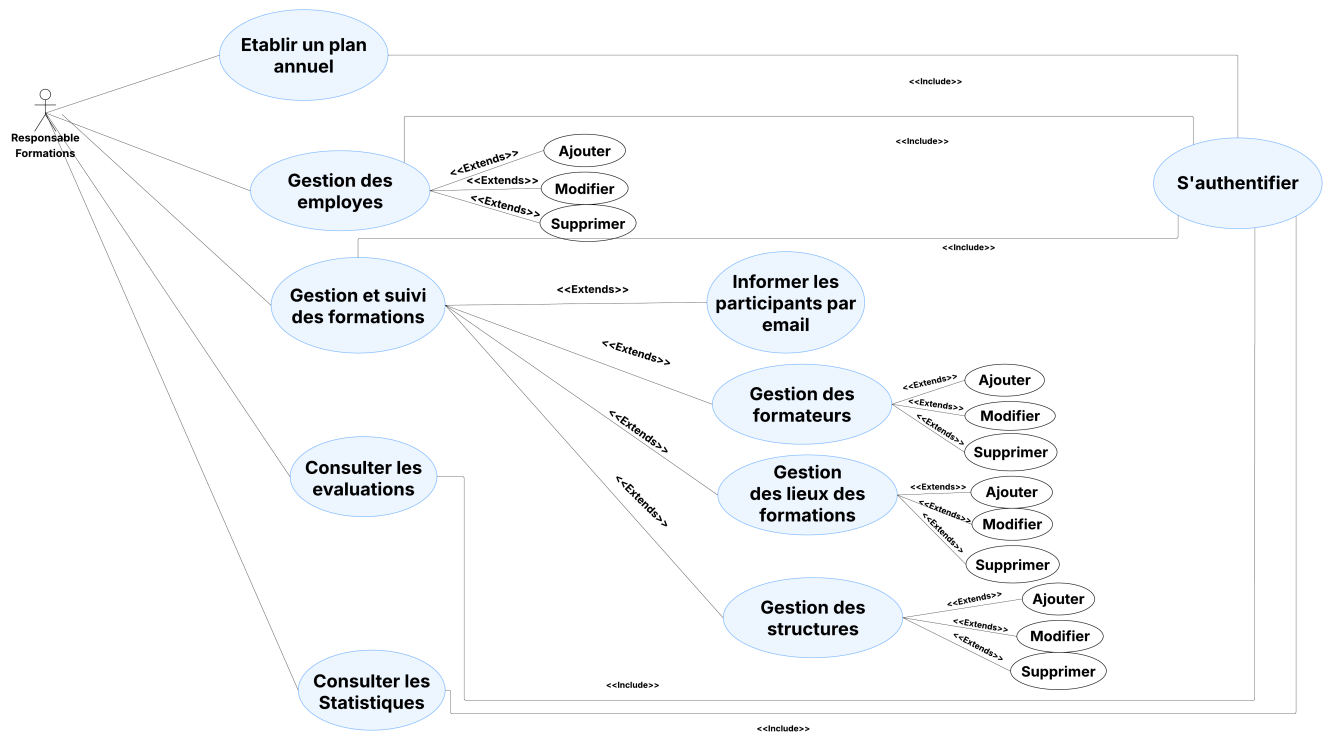


FIGURE 4.18 – Diagramme de cas d'utilisation de Responsable de formation

Diagramme de cas d'utilisation – Acteur : «DRH »

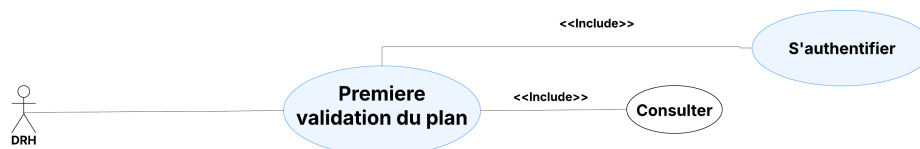


FIGURE 4.19 – Diagramme de cas d'utilistion du DRH

Diagramme de cas d'utilisation – Acteur : «DG »

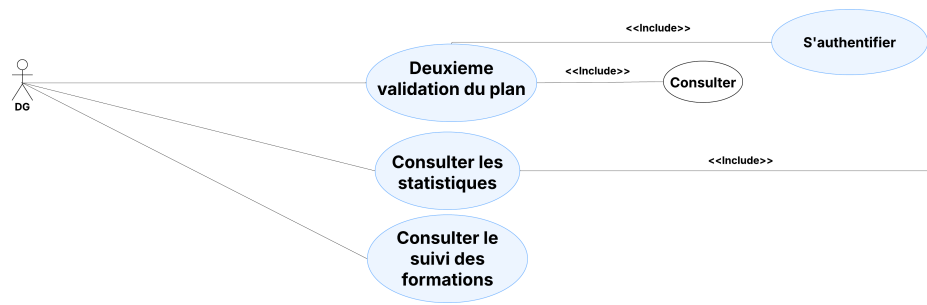


FIGURE 4.20 – Diagramme de cas d'utilistion du Directeur General

Diagramme de cas d'utilisation – Acteur : «Employe »



FIGURE 4.21 – Diagramme de cas d'utilistion de l'employe

Annexe B

Tests de l'Application

Afin d'assurer le bon fonctionnement de notre application web dédiée à la gestion des formations, nous avons recours à Postman pour réaliser des tests d'intégration. Ces tests permettent de vérifier que l'ensemble des composants de notre application — API, base de données et contrôleurs — interagissent correctement entre eux.

Postman est un outil puissant pour tester les API. Il permet de formuler des requêtes HTTP de manière simple et de vérifier les réponses obtenues. Parmi ses fonctionnalités principales, la formulation de requêtes HTTP (GET, POST, etc.), la gestion des environnements qui permet de configurer différents environnements (développement, test, production) avec des variables spécifiques, et la collaboration, ce qui en fait un outil collaboratif permettant de documenter et de partager les collections de requêtes avec l'équipe.



FIGURE 4.22 – Logo Postman

Nous avons testé deux types de requêtes HTTP : GET et POST. Pour la requête GET, nous avons vérifié que l'API peut récupérer correctement les besoins en formation stockés dans la base de données. Nous avons envoyé une requête GET à l'endpoint `http://localhost:5000/api/besoins/` sans fournir de corps (body vide), puisque les paramètres sont passés directement dans l'URL. La réponse retournée est une liste des besoins en formation au format JSON, avec un code HTTP 200 (OK), ce qui confirme le bon fonctionnement de l'API. La figure ci-dessous illustre ce test réalisé avec le logiciel Postman.

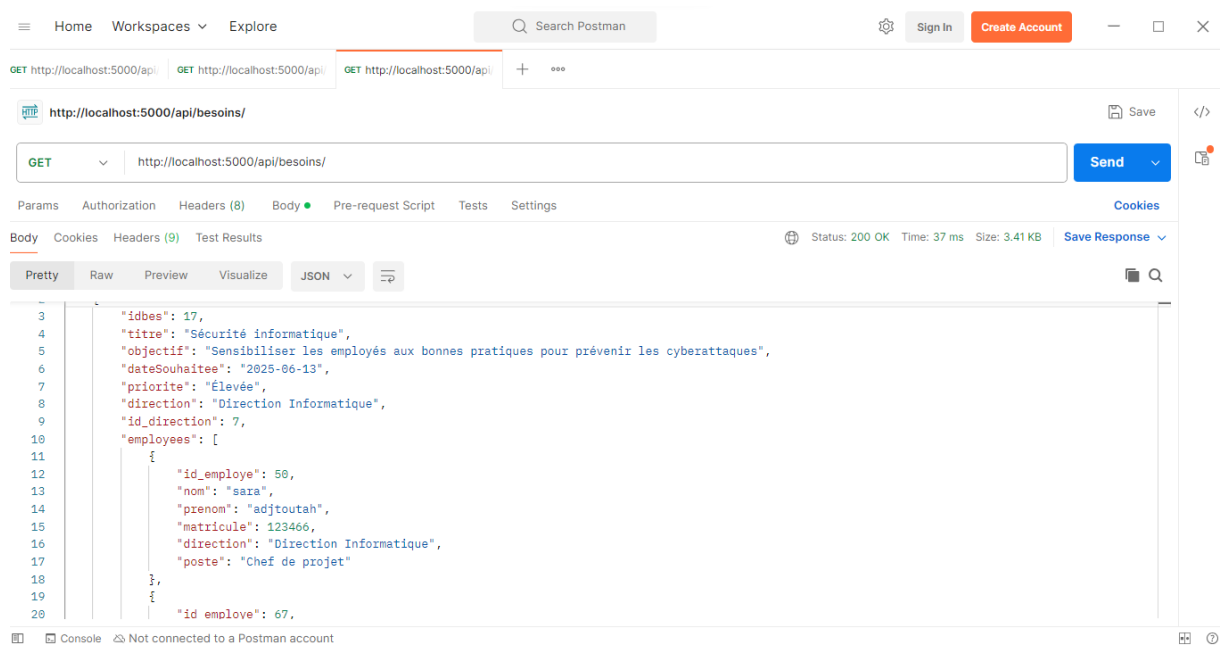


FIGURE 4.23 – La requête GET dans le logiciel Postman

Pour la requête POST, nous avons vérifié que l'API peut recevoir et traiter les données nécessaires à la création d'un lieu destiné à une formation. Nous avons envoyé une requête POST à l'URL `http://localhost:5000/api/lieu` avec les champs requis, tels que le nom, adresse, numéro de téléphone et budget du lieu. La requête a été transmise via le corps de la requête (body), au format JSON. Nous avons confirmé que la réponse retournée indiquait une création réussie du lieu, avec un statut HTTP 201 (Created), ainsi que les données du lieu nouvellement enregistré. La figure ci-dessous illustre ce test réalisé à l'aide du logiciel Postman.

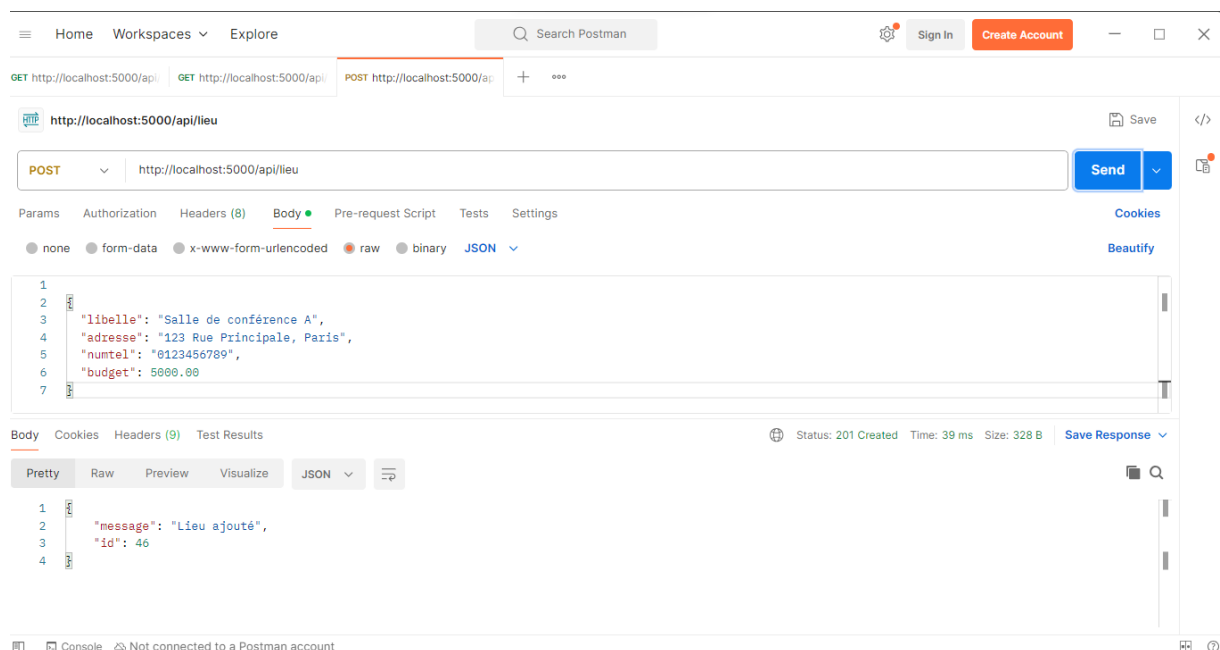


FIGURE 4.24 – La requête POST dans le logiciel Postman

Annexe C

Mail automatique lors de la création d'un compte utilisateur

Un e-mail est automatiquement envoyé à l'utilisateur une fois son compte créé par l'administrateur. Ce mail contient ses informations de connexion (adresse e-mail et mot de passe initial).

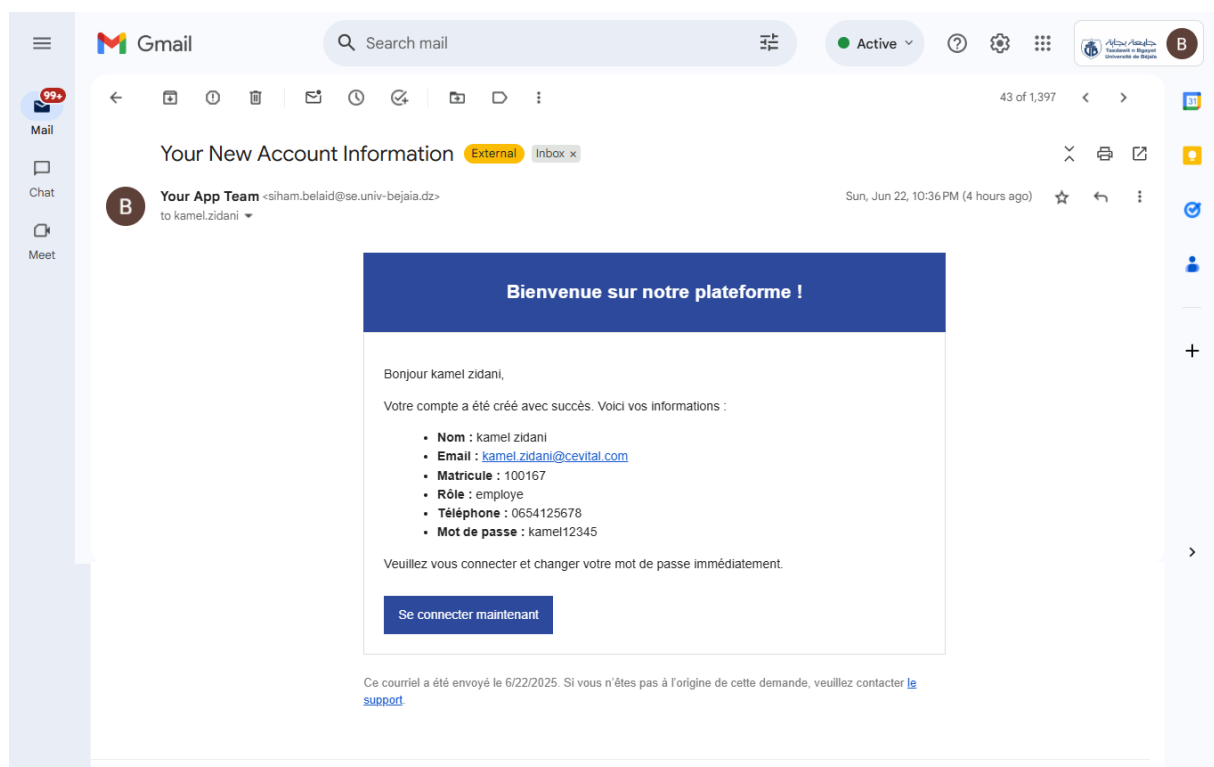


FIGURE 4.25 – Email envoyé suite à la création d'un compte

Mail automatique d'inscription à une formation

Lorsqu'un employé est ajouté à une formation, il reçoit un e-mail de notification l'informant de son inscription, avec les détails de la session : thème de la formation, date, lieu, etc.

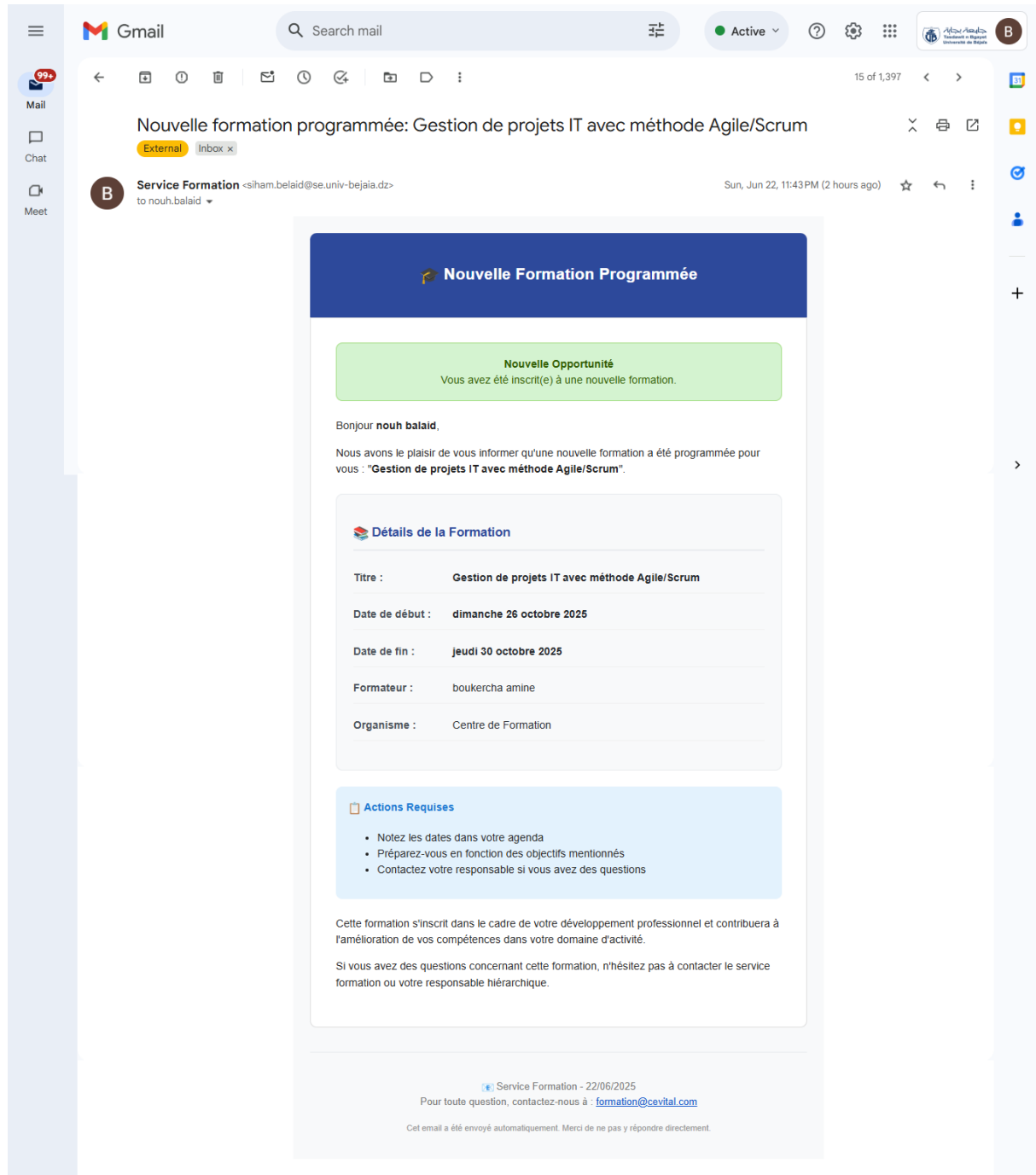


FIGURE 4.26 – Inscription confirmée à une session de formation

Fonctionnalité d'exportation des données

La figure montre la fonctionnalité d'exportation, permettant de télécharger facilement les listes du système sous format CSV.

	A	B	C	D	
1	Titre	Objectif	Date Souhaitée	Priorité	Direction
2	Développement Web (HTML, CSS, React)	Développer les compétences techniques des développeurs	27/07/2025	Moyenne	Direction Informatique
3	Sécurité informatique et cybersécurité	Renforcer la sécurité des données de l'entreprise	29/06/2025	Élevée	Direction Informatique
4	Formation en base de données SQL et NoSQL	Gérer des données non structurées dans des environnements modernes	23/11/2025	Moyenne	Direction Informatique
5	ERP : Introduction à SAP	Comprendre le fonctionnement d'un ERP et ses modules clés	28/09/2025	Élevée	Direction Informatique
6	Gestion de projets IT avec méthode Agile/Scrum	Organiser, suivre et livrer des projets informatiques de manière agile	26/10/2025	Moyenne	Direction Informatique
7	Cloud Computing avec AWS / Azure (niveau débutant)	Découvrir les services cloud et apprendre à déployer des applications	08/03/2025	Faible	Direction Informatique
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					

FIGURE 4.27 – Fonctionnalité d'exportation des données

Résumé

Les formations constituent un levier puissant pour renforcer les compétences, la créativité et l'engagement des collaborateurs, en particulier au sein des grandes entreprises. Ainsi, la mise en place d'outils efficaces s'avère indispensable pour assurer la planification, le suivi et l'évaluation de ces formations. Cette problématique a motivé, au cours de notre projet de fin d'études (PFE), la conception et la mise en œuvre d'un outil approprié : une application web de gestion des formations destinée au groupe CEVITAL. Pour réaliser ce projet, nous avons suivi la méthode UP en nous appuyant sur la modélisation UML. L'application a été développée avec React pour le front-end, Express.js pour le back-end, MySQL comme système de gestion de base de données, et des API assurant la communication entre les différentes couches de l'architecture.

Mots-clés : gestion des formations, application web, digitalisation, méthode UP, React, Express.js, MySQL, Cevital.

Abstract

Training is a powerful lever for boosting the skills, creativity and commitment of employees, particularly in large companies. Effective tools are essential for planning, monitoring and evaluating training. During our final year project (PFE), this issue led us to design and implement an appropriate tool: a web-based training management application for the CEVITAL group. To carry out this project, we followed the UP method, based on UML modeling. The application was developed using React for the front-end, Express.js for the back-end, MySQL as the database management system, and APIs to ensure communication between the different layers of the architecture.

Keywords : training management, web application, digitalization, UP method, React, Express.js, MySQL, Cevital