

Mémoire de Projet de Fin d'Études

Pour l'Obtention du Titre

D'Ingénieur d'État en Informatique

Sujet

Conception et Développement d'un Module de Formation Personnalisé pour le Site Web LGP

Soutenu par :

ADIOUANE Anas

Sous la direction de :

M ABENAOU Abdenbi

M EL BAQIR Mohamed

Soutenu le 22/06/2024, devant la commission du jury :

M ABENAOU Abdenbi

M TOUMANARI Ahmed

M BNOUHACHEM Abdellah

Année Universitaire 2023-2024

Dédicace

Chers membres de nos familles et chers amis,

Je souhaite humblement dédier ce travail à chacun d'entre vous, qui avez été une source inestimable de soutien et de motivation tout au long de cette expérience. Vos encouragements constants, votre confiance indéfectible en mes capacités et vos précieux conseils ont joué un rôle déterminant dans ma réussite.

J'exprime ma profonde gratitude pour votre patience, votre compréhension et votre générosité tout au long de ce parcours. Vous avez été mon moteur lors des moments difficiles, m'encourageant à surmonter les obstacles et les défis qui se sont dressés devant moi. Grâce à votre présence bienveillante, j'ai pu avancer avec confiance et persévérance.

J'espère sincèrement que ce travail répondra à vos attentes et vous plaira. J'ai investi un effort considérable pour produire un travail de qualité, et cela n'aurait pas été possible sans votre soutien inconditionnel. Votre confiance en mes capacités m'a inspiré et motivé à donner le meilleur de moi-même.

Je tiens à vous exprimer ma gratitude la plus profonde pour votre soutien indéfectible. Je suis extrêmement reconnaissant d'avoir pu compter sur votre présence à mes côtés tout au long de ce périple. J'espère sincèrement que ce travail sera une source de satisfaction pour vous, car votre bonheur et votre satisfaction sont pour moi une récompense inestimable.

Encore une fois, du fond du cœur, je vous remercie pour votre amour, votre encouragement et votre soutien sans faille. Je suis bénî de vous avoir dans ma vie et espère que ce travail témoignera de mon appréciation sincère.

Remerciements

Je suis extrêmement reconnaissant de pouvoir exprimer mes sincères remerciements et ma profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à la réalisation de mon projet de fin d'année.

Tout d'abord, j'aimerais adresser mes plus sincères remerciements à M. ABENAOU Abdenbi, mon professeur et encadrant au sein de l'école. Je suis extrêmement reconnaissant pour le temps qu'il m'a consacré, les précieux conseils qu'il m'a prodigués et sa volonté de partager son expertise avec moi. Son soutien inconditionnel tout au long du projet a été d'une valeur inestimable.

Je tiens également à remercier chaleureusement M. AAROUJI Mohamed Amine, le Directeur général de l'entreprise, pour m'avoir accueilli et avoir rendu possible la réalisation de ce projet. Sa vision et son leadership ont été une source d'inspiration pour moi tout au long du processus.

Je souhaite également exprimer ma profonde reconnaissance à EL BAQIR Mohamed, mon encadrant au sein de l'entreprise et mon maître de stage. Je suis très reconnaissant de l'opportunité qui m'a été offerte de mener à bien ce projet au sein de leur société. La confiance qu'ils ont placée en moi et leur soutien constant ont été des éléments clés de ma réussite.

Je souhaite exprimer ma sincère gratitude envers AMKHAO Mohamed, qui m'a apporté une aide précieuse tout au long de ce projet. Sa générosité et son soutien ont été d'une importance capitale pour sa réussite. Je suis profondément reconnaissant envers lui pour son assistance et sa disponibilité, qui ont contribué de manière significative à l'avancement de ce travail.

Je voudrais exprimer ma sincère gratitude envers Monsieur TOUKAD Abdelaziz, Responsable des Ressources Humaines, pour son précieux soutien dans la compréhension de la logique métier du projet.

Enfin, je souhaite exprimer ma profonde gratitude envers tous les membres du jury pour l'honneur qu'ils me font en acceptant de juger mon travail. Leur expertise et leurs commentaires constructifs vont toujours contribuer à mon apprentissage et à mon développement professionnel.

Je suis réellement reconnaissant envers chacune de ces personnes pour leur contribution inestimable à mon projet. Leur soutien, leur mentorat et leur confiance ont été essentiels à mon succès. Je suis profondément reconnaissant et honoré d'avoir eu l'opportunité de travailler avec eux et de bénéficier de leur expertise.

Résumé

Le présent mémoire décrit le travail réalisé dans le cadre de mon projet de fin d'études, effectué au sein de la société INFOSAT. Mon projet a consisté à développer un site web pour INFOSAT, une entreprise qui utilise déjà un logiciel appelé LGP depuis 2008. Cette initiative s'inscrit dans le cadre d'une stratégie de modernisation de l'entreprise, avec pour objectif de transformer le logiciel existant en une plateforme web tout en ajoutant un nouveau module dédié à la gestion des formations pour les entreprises clientes d'INFOSAT. Au cours de ce projet, j'ai eu l'opportunité de mettre en pratique mes connaissances académiques dans un environnement professionnel, en travaillant en étroite collaboration avec l'équipe d'INFOSAT pour atteindre les objectifs fixés.

Pour mener à bien ce projet, on a commencé par effectuer une analyse approfondie des besoins en utilisant le langage de modélisation UML. Cette phase m'a permis de comprendre en détail les exigences fonctionnelles et non fonctionnelles de la future plateforme, ce qui a posé les bases solides nécessaires pour la suite du développement.

En ce qui concerne les outils et les technologies, ceux-ci ont été déterminés par l'entreprise INFOSAT, en fonction de ses préférences, de ses ressources disponibles et de son infrastructure existante. Mon rôle a été d'assimiler rapidement ces outils et technologies et de les utiliser de manière efficace pour répondre aux exigences du projet.

Le développement du site web a été une étape complexe mais enrichissante. On a travaillé en étroite collaboration avec les développeurs et les autres membres de l'équipe pour assurer une intégration fluide du nouveau module de gestion des formations avec le logiciel existant. Des itérations fréquentes et des rétroactions régulières avec les parties prenantes ont permis d'ajuster et d'améliorer continuellement le produit en cours de développement.

Ce processus détaillé de conception et de développement reflète mon engagement à fournir un produit final de qualité, répondant aux besoins spécifiques de l'entreprise INFOSAT. Mon objectif est de fournir une solution moderne et efficace pour la gestion des formations des entreprises, tout en réalisant mon projet de fin d'étude avec succès.

Mots-clés : LGP, UML.

Abstract

This present thesis describes the work carried out as part of my final year project, conducted within the company INFOSAT. My project involved developing a website for INFOSAT, a company that has been using a software called LGP since 2008. This initiative is part of a company modernization strategy, aiming to transform the existing software into a web platform while adding a new module dedicated to managing training for INFOSAT's corporate clients. Throughout this project, I had the opportunity to apply my academic knowledge in a professional environment, working closely with the INFOSAT team to achieve the set objectives.

To successfully execute this project, I began by conducting a thorough needs analysis using the Unified Modeling Language (UML). This phase allowed me to fully understand the functional and non-functional requirements of the future platform, laying the necessary solid foundation for further development.

Regarding tools and technologies, these were determined by INFOSAT based on its preferences, available resources, and existing infrastructure. My role was to quickly assimilate these tools and technologies and use them effectively to meet the project requirements.

The development of the website was a complex yet rewarding stage. I collaborated closely with developers and other team members to ensure smooth integration of the new training management module with the existing software. Frequent iterations and regular feedback from stakeholders allowed for continuous adjustment and improvement of the product throughout the development process.

This detailed process of design and development reflects my commitment to delivering a quality final product that meets the specific needs of INFOSAT. My goal is to provide a modern and efficient solution for corporate training management while successfully completing my final year project.

Keywords : LGP, UML.

Liste des abréviations

Abréviation	Signification
PFE	Projet de Fin d'Etudes
LGP	Logiciel de Gestion de Personnel
UML	Unified Method Language
HTML	Hypertext Markup Language
HTTP	Hypertext transfer protocol
IDE	Integrated Development Environment
URL	Uniforme resource locator
SGBD	Système de gestion de base de données
VS CODE	Visual Studio Code
JWT	JSON Web Tokens
CSS	Cascading Style Sheets
OAS	OpenAPI Specification
POJO	Plain Old Java Object
DTO	Data Transfer Object
SPA	Single Page Application
VCS	Version Control System

Table des figures

Figure 1 : Organigramme de la société INFOSAT	15
Figure 2 : Localisation de l'entreprise.....	16
Figure 3 : Méthode en cascade	25
Figure 4 : Diagramme de Gantt	26
Figure 5 : Diagramme d'héritage entre les acteurs	29
Figure 6 : Diagramme de cas d'utilisation du responsable de formation, partie paramétrage	29
Figure 7 : Diagramme de cas d'utilisation du responsable de formation, partie gestion de formation ..	30
Figure 9 : Diagramme de cas d'utilisation du formateur et du collaborateur	30
Figure 10 : Diagramme de cas d'utilisation du super administrateur	31
Figure 11 : Diagramme de classe, partie sécurité	32
Figure 12 : Diagramme de classe, module de formation	32
Figure 13 : Diagramme de séquence d'authentification	33
Figure 14 : Diagramme de séquence cas du super admin en relation avec le collaborateur	34
Figure 15 : Diagramme de séquence cas du responsable de formation en relation avec le formateur ..	35
Figure 16 : Diagramme de séquence cas du responsable en relation avec les planifications	36
Figure 17 : Diagramme de séquence cas du formateur en relation avec les évaluations	37
Figure 18 : Diagramme d'état-transition de la réalisation d'une planification	38
Figure 19 : Diagramme d'état-transition de la présence des participants	38
Figure 19 : Diagramme d'état-transition d'une évaluation	39
Figure 20 : le fonctionnement de l'architecture MVC	42
Figure 21 : le fonctionnement de DispatcherServlet	43
Figure 22 : l'architecture global du projet	43
Figure 23 : logo d'API REST	44
Figure 25 : logo de Java EE	45
Figure 26 : logo de Maven	45
Figure 27 : logo de Spring	46
Figure 28 : logo de Spring Boot	46
Figure 29 : logo de Spring Security	46
Figure 30 : logo de Spring Data	46
Figure 31 : logo de JPA	46
Figure 32 : logo de Hibernate	46
Figure 33 : logo de JWT	47
Figure 34 : logo de Lombok	47
Figure 35 : logo de Swagger	47
Figure 36 : logo de IntelliJ IDEA	47
Figure 37 : logo de Postman	47
Figure 38 : logo de Visual Studio Code	48
Figure 39 : logo de Angular 17	48
Figure 40 : logo de HTML 5	48
Figure 41 : logo de CSS 3	48
Figure 42 : logo de TypeScript	49
Figure 43 : logo de Angular Material	49
Figure 44 : logo de RxJS	49
Figure 45 : logo de Zone.js	49
Figure 46 : logo de jsPDF	49
Figure 47 : logo de Iconify	49
Figure 48 : logo de Visual Paradigm	50
Figure 49 : logo de Modelio	50
Figure 50 : logo de GIT	50
Figure 51 : logo de GitHub	50
Figure 52 : logo de MySQL..	51
Figure 53 : logo de Canva	51

Figure 54 : Structure global du projet_ backend	53
Figure 55 : Structure global du projet_ frontend	56
Figure 56 : Page du login	58
Figure 57 : Interface de gestion des utilisateurs	58
Figure 58 : Interface de création d'un utilisateur	59
Figure 59 : Interface de gestion des rôles	59
Figure 60 : Interface de création d'un rôle	60
Figure 61 : Interface de choix des privilèges pour un rôle	60
Figure 62 : Exemple des privilèges de sécurité	60
Figure 63 : Interface de choix des permissions pour un rôle	61
Figure 64 : Interface des collaborateurs de l'entreprise	61
Figure 65 : Interface d'ajout d'un collaborateur	62
Figure 66 : Page affichant une fenêtre de confirmation de succès	62
Figure 67 : Interface de modification d'un collaborateur	62
Figure 68 : Page de la liste des domaines de formation	63
Figure 69 : Interface d'Ajout d'un domaine	63
Figure 70 : Page de la liste des modules de formation	64
Figure 71 : Interface d'Ajout d'un module	64
Figure 72 : Interface de modification d'un module	64
Figure 73 : Page de la liste des thèmes de formation	65
Figure 74 : Page de la liste des lieux de formation	65
Figure 75 : Page de la liste des questions de formation	66
Figure 76 : Page de la liste des formateurs de formation	66
Figure 77 : Interface d'Ajout d'un formateur	67
Figure 78 : Interface de modification d'un formateur	67
Figure 79 : Page de la liste des formations	68
Figure 80 : Interface d'Ajout d'une formation	68
Figure 81 : Page des détails d'une formation	69
Figure 82 : Page de la liste des planifications de formation	69
Figure 83 : Interface d'Ajout d'une planification	70
Figure 84 : Interface de modification d'une planification	70
Figure 85 : Page des détails d'une planification	71
Figure 86 : Interface de la liste de présence	71
Figure 87 : Page de choix de planification pour ajouter des participants	72
Figure 88 : Page de choix des participants	72
Figure 89 : Page de la liste des évaluations pour un collaborateur	73
Figure 90 : Page répertoriant toutes les questions de la formation sélectionnée	73
Figure 91 : Résultat lors de la sélection d'une évaluation non autorisée	74
Figure 92 : Page de la liste des évaluations complétées et envoyées au responsable de formation	74
Figure 93 : Page de la liste des réponses d'un collaborateur	74
Figure 94 : Exemple d'une évaluation validée	75
Figure 95 : Exemple du thème sombre	75

Liste des tableaux

Tableau 1 : Fiche Technique de la société INFOSAT	14
Tableau 2 : Technologies de développement backend	47
Tableau 3 : Outils de développement backend	47
Tableau 4 : Technologies de développement frontend	49
Tableau 5 : Outils de modélisation	50
Tableau 6 : Outils et technologies de contrôle de versions	50

Table des matières

Liste des abréviations	6
Table des figures	7
Liste des tableaux	9
Introduction générale	12
Chapitre 1 : Présentation de l'entreprise.....	13
Introduction	14
Présentation de l'entreprise d'accueil	14
Fiche d'identité	14
Organigramme de fonctionnement	15
Localisation de l'entreprise	15
Les services offerts par l'entreprise	16
Les missions de l'entreprises	17
Conclusion du chapitre	17
Chapitre 2 : Cahier De Charges	18
Objectif principal	19
Objectifs spécifiques	19
Les acteurs de l'application	19
Exigences Fonctionnelles	20
Exigences non fonctionnelles	24
Démarche de travail	25
Diagramme de Gantt	26
Conclusion du chapitre	26
Chapitre 3 : Analyse et Conception	27
Introduction	28
Diagramme des Cas d'Utilisation	28
Diagramme de classe	31
Diagramme de séquences	33
Diagramme d'état et transition	37
Conclusion	39
Chapitre 4 : Etude Techniques	40
Introduction	41
Outils et Technologies de Développement	41
Conclusion	51

Chapitre 5 : Implémentation	52
Introduction	53
La structure du projet	53
Scénario d'utilisation	57
Conclusion	75
Conclusion générale et perspectives	76
Webographie	77

Introduction générale

Dans le paysage technologique en constante évolution d'aujourd'hui, le rôle des systèmes d'information dans l'amélioration de l'efficacité organisationnelle et de la compétitivité est plus crucial que jamais. Avec les entreprises s'appuyant de plus en plus sur des solutions numériques pour rationaliser leurs opérations et améliorer l'expérience client, la demande de plates-formes logicielles innovantes ne cesse de croître. Cet environnement dynamique prépare le terrain pour mon projet de fin d'études, réalisé au sein de l'organisation estimée INFOSAT.

Ce projet constitue un véritable pont entre ma formation académique et le monde professionnel, offrant une opportunité unique d'appliquer les connaissances théoriques acquises au cours de mes études à un projet concret au sein d'une entreprise en activité.

Le projet a pour objectif de répondre à un besoin essentiel de modernisation de l'entreprise INFOSAT, qui souhaite transformer son logiciel existant, LGP, en une plateforme web dynamique. En outre, l'ajout d'un nouveau module dédié à la gestion des formations pour les entreprises clientes représente une extension stratégique de l'offre de services d'INFOSAT.

Cette introduction générale posera le cadre pour explorer en détail le processus de conception, de développement et de mise en œuvre de ce projet ambitieux, ainsi que les défis rencontrés et les solutions proposées tout au long du parcours.

Chapitre 1

Présentation de l'Entreprise

1 Introduction

Avant d'aller plus loin dans notre rapport, il est important de présenter notre projet en commençant par présenter brièvement l'entreprise d'accueil, citer ces quelques missions et s'arrêter sur ses services offerts, pour introduire par la suite la problématique, suivie par les objectifs du projet et une analyse détaillée des besoins, avant de présenter en dernier point la méthode de conduite du projet adoptée.

2 Présentation de l'entreprise d'accueil

INFOSAT est une société de services orientée vers les nouvelles technologies au service de la télécommunication et de l'informatique portant sur les domaines Internet / Intranet, les bases de données en ligne, le commerce électronique, le développement informatique, les réseaux informatiques et téléphoniques.

3 Fiche d'identité

RAISON SOCIALE	INFOSAT
FORME JURIDIQUE	S.A.R.L
ACTIVITES	DEVELOPPEMENT DE LOGICIEL, INFORMATIQUE, BUREAUTIQUE, ELECTRONIQUE, TELEPHONIE.
CAPITAL	100 000 .00 DH
ADRESSE	18, RUE 486 LES AMICALES AGADIR.
REGISTRE DE COMMERCE D'AGADIR N°	7243.
PATENTE N°	49226643
IF N°	06926240
C/B	ATTIJARI WAFABANK N° : 184C000677
TEL/FAX	0528 23 24 62

Tableau 1 : Fiche Technique de la société INFOSAT

4 Organigramme de fonctionnement

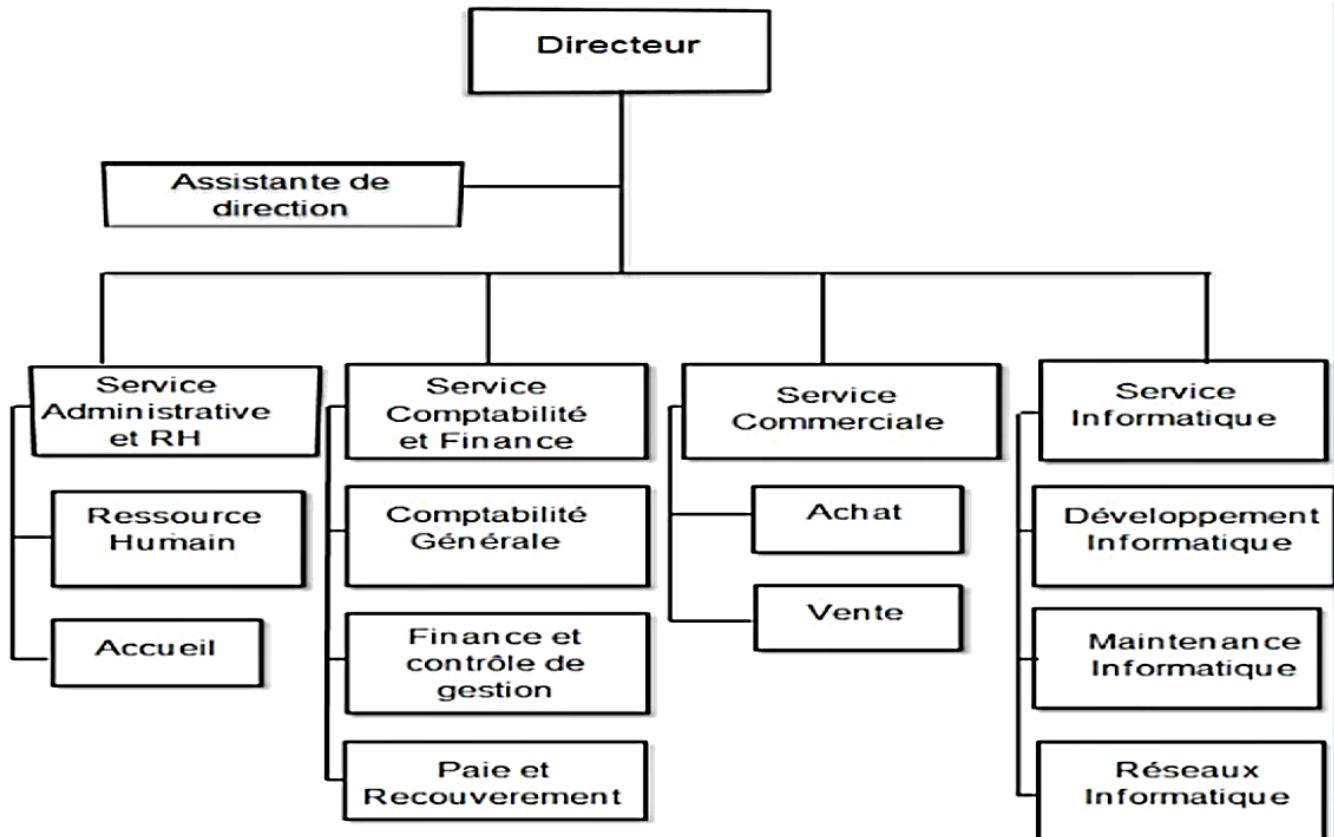


Figure 1 : Organigramme de la société INFOSAT

Cet organigramme illustre la structure hiérarchique d'une organisation dirigée par un directeur M. AAROUJI Mohamed Amine, assisté par une assistante de direction. Il comprend quatre services principaux : le Service Administrative et RH, qui gère le personnel, le recrutement, la formation et l'accueil ; le Service Comptabilité et Finance, responsable de la gestion financière, des comptes, des budgets, des salaires et du recouvrement des créances ; le Service Commerciale, chargé des achats, des approvisionnements et des ventes ; et le Service Informatique, qui développe et maintient les logiciels, gère la maintenance des équipements informatiques et s'occupe des réseaux et de la connectivité internet. Chaque service est subdivisé en sections spécialisées pour assurer un fonctionnement optimal de l'organisation.

5 Localisation de l'entreprise

Voici l'adresse de l'entreprise : Infosat Adresse 18, rue 486 les Amicales, Souss-Massa-Drâa, Agadir, Maroc



Figure 2 : Localisation de l'entreprise

6 Les services offerts par l'entreprise

Les activités principales de l'entreprise INFOSAT :

RÉSEAUX INFORMATIQUES

- Etude et conception d'architectures et solutions réseaux (LAN- INTRANET-WIFI...)
- Installation de serveurs LINUX /Windows Server (contrôleur de domaine, serveur de fichiers, messagerie, site web, impression...)
- Etablissement de stratégies de sécurité réseau (Proxy, Firewall, sauvegardes, antivirus...)
- Travaux de câblage Ethernet (CAT5, CAT6, Fibre optique), armoires informatiques, Antennes et Points d'accès WIFI....

RÉSEAUX TELEPHONIQUES

- Travaux de câblage téléphonique (prises, moulure et accessoires...)
- Equipement, configuration et installation de standards téléphoniques ainsi que la Téléphonie IP.

DÉVELOPPEMENT INFORMATIQUE

- Logiciel, Etude, conception et intégration de solutions informatiques standards, spécifiques et Open source sous les environnements Linux et Windows.
- Développement d'applications Web.

COMPETENCE DE DEVELOPPEMENT

- Langages de programmation : WinDev, VB, Delphi, Java (J2EE), C++
- Bases de données : Access, SQL Server, Interbase/Firbird, MySQL...
- Méthodes d'analyse et modélisation : Merise, UML.

MAINTENANCE ET REPARATION

- SAV et Contrats de maintenance sur tous les produits (matériels et logiciels) fournis par INFOSAT
- Maintenance et réparation soft et hard informatique, téléphonique et électronique.

VENTE

- SYSTÈMES DE POINTAGE (Pointeuses, cartes de proximité, badges, logiciel de gestion de présence)
- LOGICIELS (LGP Gestion de pointage, antivirus, gestion pharmacie, gestion commerciale, gestion paie)
- MATERIEL ET ACCESSOIRES INFORMATIQUE (Micro-ordinateurs, Imprimantes, Traceur, Onduleur...)
- EQUIPEMENTS DE TELECOMUNICATION (routeurs, Switch, modems, équipements WIFI, standards et appareils téléphoniques)
- SOLUTIONS DE TÉLÉSURVEILLANCE (caméras de surveillance, caméras IP, cartes DVR...)
- FOURNITURES BUREAUTIQUE ET CONSOMMABLES INFORMATIQUE

7 Les missions de l'entreprises

Offrir aux clients des produits et des services de qualité en fonction de leurs besoins et leurs budgets dans chacun des secteurs suivants :

- ➔ Développement logiciel
- ➔ Réseau informatique
- ➔ Sécurité et vidéosurveillance
- ➔ Electronique
- ➔ Téléphonique

Être un fournisseur d'informatique reconnu comme un leader pour la qualité de ses produits et de ses services, et pour le professionnalisme de ses ressources.

8 Conclusion

Ce chapitre a permis de présenter de manière concise et détaillée l'entreprise d'accueil, ses services, son domaine d'expertise et ses partenaires. En examinant ces informations, il devient évident que l'environnement dans lequel nous avons travaillé sur notre projet est extrêmement propice à la réussite et à l'innovation. L'entreprise d'accueil possède une solide réputation dans son domaine et est soutenue par un réseau de partenaires influents. Sa vaste gamme de services témoigne de son engagement à répondre aux besoins variés de ses clients.

Chapitre 2

Cahier De Charges

1. Objectif principal

Créer un site web de gestion des formations pour simplifier la planification, le suivi et l'évaluation des formations internes.

2. Objectifs spécifiques

- Automatiser le processus de planification des sessions de formation en permettant à l'administrateur de créer facilement des sessions, d'assigner des formateurs et de définir des dates et des lieux.
- Donner l'accès à toutes parties de l'entreprise d'utiliser l'application même la partie opérationnelle en pratiquant une interface visuelle intuitive qui minimise la dépendance à la lecture (cela peut inclure l'utilisation d'icônes, de symboles et de couleurs pour représenter les différentes fonctionnalités et options du logiciel).
- Permettre aux utilisateurs de consulter les sessions de formations disponibles, en fonction de leurs besoins de développement professionnel et des recommandations de leurs responsables.
- Faciliter le suivi de la participation des employés aux sessions de formation et de leur progression dans les programmes de développement.

3. Les acteurs de l'application

- Super administrateur

Le "super administrateur" est le plus haut niveau d'accès et de contrôle au sein du système. En tant que tel, il détient des priviléges étendus qui lui permettent d'accomplir toutes les tâches et fonctions disponibles sur le site web. Il est responsable de la gestion globale du système, y compris la création de rôles et d'utilisateurs, l'attribution des autorisations appropriées à chaque utilisateur, et la supervision de toutes les activités sur la plateforme. En outre, le super administrateur est souvent le point de contact principal pour toute question technique ou de sécurité, et il est chargé de garantir le bon fonctionnement et la sécurité de l'ensemble du système. En résumé, le super administrateur est le gardien ultime du site web, doté du pouvoir et de la responsabilité de prendre des décisions critiques pour son bon fonctionnement.

- Responsable de formation (admin)

Ce responsable est chargé de planifier, organiser et superviser les activités de formation au sein de l'entreprise. Il est responsable de la définition des objectifs de formation, du choix des formateurs, de la coordination des sessions de formation et de l'évaluation de l'efficacité des programmes de formation.

- Collaborateurs :

Ce sont les employés de l'entreprise qui participent aux sessions de formation pour acquérir de nouvelles compétences, améliorer leurs performances ou se familiariser avec de nouveaux processus ou technologies. Ils ont également un compte avec lequel peuvent se connecter à l'application.

- Formateurs :

Ces utilisateurs sont chargés de dispenser les sessions de formation. Ils transmettent les connaissances et les compétences nécessaires aux participants pour atteindre les objectifs de formation définis par l'entreprise. Il est possible qu'ils soient des employés internes ou des personnes extérieurs.

4. Exigences Fonctionnelles

Planification des formations

1. Création d'une nouvelle planification de formation

- Le responsable de formation accède à la fonction de planification des formations depuis le tableau de bord.
- Il remplit un formulaire détaillé en spécifiant la date de début et de fin de la formation, ainsi que le lieu où elle se déroulera.
- Le responsable sélectionne un formateur parmi ceux disponibles pour cette période.
- Chaque planification a un statut initial (nouveau).
- Après avoir vérifié les détails, il enregistre la planification.

2. Validation des planifications

- Les planifications nouvellement créées sont soumises à la validation.
- Le responsable de formation examine chaque planification et peut les valider ou les rejeter en fonction de leur pertinence et de leur faisabilité.
- Il se peut qu'elle soit validée dès le départ.

3. Modification des planifications existantes

- Le responsable de formation sélectionne la planification qu'il souhaite modifier.
- Il met à jour les détails tels que la date, le lieu et le formateur associé selon les besoins.
- Une fois les modifications apportées, il enregistre les changements.

4. Suppression des planifications

- Le responsable de formation peut supprimer une planification en sélectionnant l'option de suppression.
- Un message de confirmation s'affiche pour éviter les suppressions accidentielles.
- Une fois confirmée, la planification est supprimée de la base de données.

Définition des formations

1. Création de nouvelles formations

- Le responsable de formation crée une nouvelle formation en spécifiant un titre, un contenu détaillé, le nombre d'heures prévues et en l'associant à un thème qui fait partie d'un module, ce dernier est inclus dans un domaine de formation.
- Il peut également définir les prérequis nécessaires à la participation à la formation.
- Une fois les détails saisis, il enregistre la nouvelle formation.

2. Modification des formations existantes

- Le responsable de formation accède à la liste des formations existantes.
- Il sélectionne la formation à modifier et met à jour les détails comme le contenu, la durée ou les associations avec d'autres éléments.
- Après avoir apporté les modifications nécessaires, il enregistre les changements.

3. Suppression des formations

- Le responsable de formation peut supprimer une formation obsolète ou inutile en sélectionnant l'option de suppression.
- Une confirmation est demandée avant la suppression définitive de la formation.

Gestion des formateurs

1. Ajout de nouveaux formateurs

- Les administrateurs accèdent à la section de gestion des formateurs depuis le tableau de bord.
- Ils remplissent un formulaire pour ajouter un nouveau formateur en saisissant ses informations personnelles telles que le nom, le prénom, le sexe, la date de naissance, la civilité et le CIN.
- Des informations supplémentaires telles que les compétences et les domaines d'expertise peuvent également être ajoutées.
- Une fois les informations saisies, l'administrateur enregistre le nouveau formateur dans le système.

2. Modification des informations des formateurs

- Les administrateurs accèdent à la liste des formateurs existants.
- Ils sélectionnent le formateur dont ils souhaitent modifier les informations et accèdent à son profil.
- Ils mettent à jour les informations nécessaires telles que les coordonnées, les compétences ou les domaines d'expertise.
- Après avoir apporté les modifications, ils enregistrent les changements.

3. Suppression des formateurs

- En cas de besoin, les administrateurs peuvent supprimer un formateur en accédant à son profil.
- Ils sélectionnent l'option de suppression et confirment leur choix.
- Une fois confirmée, le profil du formateur est supprimé de la base de données.

Gestion des collaborateurs

1. Ajout de nouveaux collaborateurs

- Les administrateurs accèdent à la section de gestion des collaborateurs depuis le tableau de bord.

- Ils remplissent un formulaire pour ajouter un nouveau collaborateur en saisissant ses informations personnelles telles que le nom, le prénom, le sexe, la date de naissance, la civilité, le CIN, le matricule de pointage et le matricule de paie.
- D'autres détails tels que le département, le poste, ou les informations de contact peuvent également être ajoutés.
- Une fois les informations saisies, l'administrateur enregistre le nouveau collaborateur dans le système.

2. Modification des informations des collaborateurs

- Les administrateurs accèdent à la liste des collaborateurs existants.
- Ils sélectionnent le collaborateur dont ils souhaitent modifier les informations et accèdent à son profil.
- Ils mettent à jour les informations nécessaires telles que les coordonnées, le département ou le poste.
- Après avoir apporté les modifications, ils enregistrent les changements.

3. Suppression des collaborateurs

- En cas de besoin, les administrateurs peuvent supprimer un collaborateur en accédant à son profil.
- Ils sélectionnent l'option de suppression et confirment leur choix.
- Une fois confirmée, le profil du collaborateur est supprimé de la base de données.

Gestion des présences

1. Enregistrement des présences

- Les responsables de formation accèdent à la liste des sessions de formation planifiées.
- Pour chaque session, ils peuvent ajouter les collaborateurs intéressés par la formation.
- Ils peuvent sélectionner les collaborateurs à partir d'une liste préexistante ou en ajoutant manuellement de nouveaux noms.
- Pour chaque session, ils saisissent le statut de présence de chaque collaborateur (présent, absent, non défini).

2. Modification des présences

- En cas de besoin, les responsables de formation peuvent modifier les statuts de présence des collaborateurs.
- Ils accèdent à la liste des présences enregistrées pour une session donnée et mettent à jour les informations selon les changements survenus.

Gestion des questions et réponses

1. Ajout de questions

- Le responsable de formation accède à la section dédiée à la gestion des questions (partie paramétrage).
- Il crée de nouvelles questions.
- Les questions sont ensuite associées à la formation correspondante et enregistrées dans la base de données.

2. Modification des questions

- Le responsable de formation peut modifier les questions existantes en accédant à la liste des questions associées à chaque formation.
- Il met à jour le contenu de la question selon les besoins et enregistre les modifications.

3. Suppression des questions

- En cas de questions obsolètes ou inutiles, le responsable de formation peut les supprimer en sélectionnant l'option de suppression.
- Une confirmation est demandée avant la suppression définitive.

4. Ajout des réponses

- Chaque collaborateur voit la liste des planifications qui le concernent.
- Pour chacune, il existe une section d'évaluation permettant de voir les questions de la formation ainsi que les champs à remplir pour y répondre.
- Toutes les réponses sont envoyées au formateur et aussi aux responsables sauf que le formateur peut valider ou non les réponses des collaborateurs participés !
- Chaque collaborateur peut voir la liste des planifications donc avec le statut de l'évaluation qui a passé.

5. Exigences non fonctionnelles

Les besoins non fonctionnels, également connus sous le nom de contraintes ou exigences de qualité, regroupent les aspects essentiels à prendre en compte lors du développement et de l'exploitation efficace du système. Ils englobent des éléments tels que la performance, la sécurité, la fiabilité, l'évolutivité, l'interopérabilité et l'utilisabilité. Ces besoins jouent un rôle crucial dans la conception et l'implémentation du système, car ils définissent les attentes et les critères de réussite liés à son fonctionnement global. Ils fournissent un cadre pour garantir que le système répond aux normes de performance, de sécurité et de convivialité attendues, tout en tenant compte des contraintes spécifiques de l'environnement dans lequel il sera déployé. En accordant une attention adéquate aux besoins non fonctionnels, les développeurs peuvent s'assurer que le système répondra aux exigences globales de qualité et offrira une expérience utilisateur satisfaisante.

a. Convivialité et flexibilité

L'application doit offrir une interface conviviale, flexible, et ergonomique, afin de faciliter son utilisation par l'utilisateur. Cela comprend la mise en place de fonctionnalités intuitives, une disposition claire des éléments, et une navigation fluide pour une expérience utilisateur

optimale. En privilégiant ces caractéristiques, l'application deviendra plus accessible et conviviale, permettant à l'utilisateur de l'exploiter de manière aisée et efficace.

b. Rapidité

L'application doit optimiser les processus afin de garantir des temps de réponse raisonnables, assurant ainsi une expérience utilisateur fluide et réactive. Cela implique une optimisation du code, une gestion efficace des requêtes et une utilisation judicieuse des ressources système.

c. Efficacité

L'application doit fonctionner de manière optimale en termes d'utilisation des ressources, telles que le CPU et la mémoire, afin de garantir une performance fluide et efficace. Il est crucial d'optimiser le code et d'adopter des pratiques de programmation efficientes pour minimiser la charge sur les ressources. Cela peut inclure des techniques telles que la mise en cache des données fréquemment utilisées, la gestion appropriée des connexions réseau et la limitation des opérations gourmandes en ressources. En utilisant les ressources de manière efficiente, l'application pourra fonctionner de manière plus réactive, éviter les ralentissements et les goulots d'étranglement, et améliorer globalement l'expérience utilisateur.

d. Maintenabilité

Le code de l'application doit être soigneusement structuré, compréhensible et lisible, dans le but de faciliter les évolutions du système et de permettre une localisation et une correction aisées des erreurs. Pour cela, il est important d'adopter des conventions de nommage claires, d'utiliser des commentaires pertinents et d'organiser le code en modules logiques. En favorisant la lisibilité du code, les développeurs pourront comprendre plus facilement son fonctionnement, le maintenir et le mettre à jour efficacement au fil du temps.

6. Démarche de travail

Afin d'assurer une représentation claire et compréhensible de la structure et du comportement de notre projet, nous avons fait le choix d'utiliser l'outil UML (Unified Modeling Language). En utilisant les diagrammes et les notations standardisées de l'UML, nous pouvons modéliser les différentes entités, leurs relations et les interactions entre elles, ce qui facilite la communication et la compréhension du projet. Cette approche nous permet de visualiser de manière systématique les différents aspects du projet, favorisant ainsi une meilleure collaboration et une gestion efficace des tâches.

6.1 Présentation du langage UML

Le langage UML (langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie. Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

6.2 Cycle de vie de développement

Un projet informatique se démarque par son cycle de vie, qui est une succession des différentes phases de développement du logiciel, de l'établissement des besoins du client jusqu'à l'achèvement du logiciel en tant que produit commercial.

Comme pour toutes les fabrications, il est important d'avoir un processus de fabrication du logiciel bien défini. En génie logiciel il existe pas mal de 14 modèles à savoir le modèle en cascade, le modèle en Y, le modèle 2TUP (modèle en Y itératif), le modèle en V et bien d'autres. En raison de sa simplicité, son contrôle facile, sa facilité de planification des étapes et des délais, ainsi que son idéalité pour les petits projets, nous avons choisi le modèle en cascade. Le modèle de cycle de vie en cascade est développé dans les années 1970 par W. ROYCE. Ce modèle comporte 7 phases : analyse des besoins, spécifications, conception de l'architecture, conception détaillée, implémentation, tests (validation) et enfin installation, et une phase n'est entamée que quand la phase précédente est à 100% finie et validée. Ci-dessous, le modèle du cycle de vie en cascade :

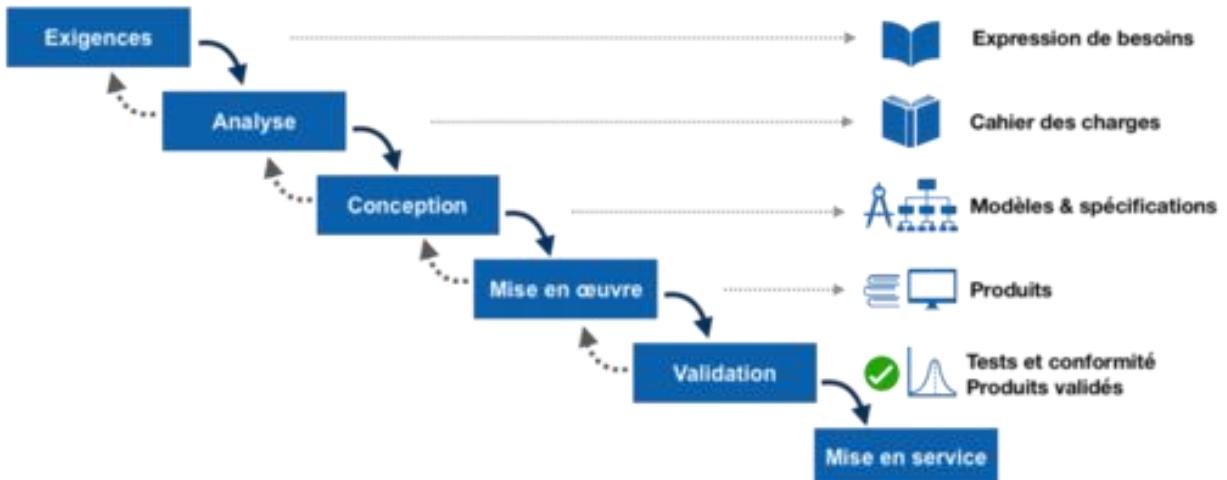


Figure 3 : Méthode en cascade

7. Diagramme de Gantt

Le diagramme de Gantt a été utilisé pour planifier et répartir les tâches de notre projet en fonction du temps. Toutefois, en raison de la complexité des technologies utilisées lors de la phase de développement et des contraintes temporelles, nous n'avons pas pu respecter strictement les délais initialement définis dans le diagramme.

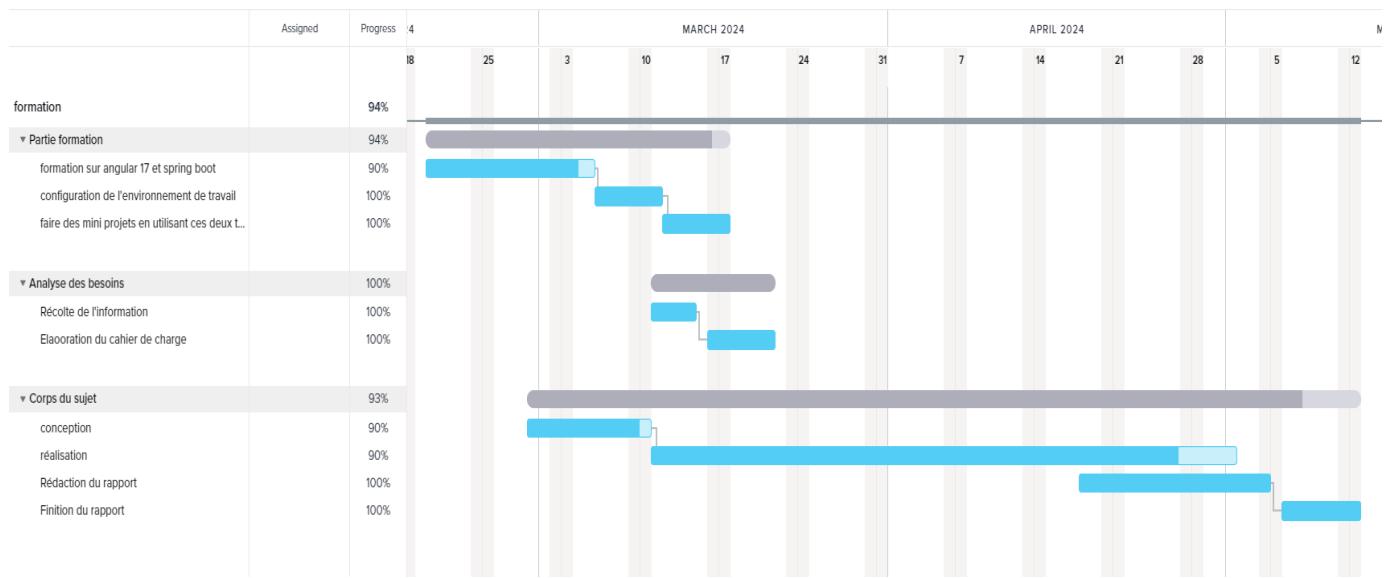


Figure 4 : Diagramme de Gantt [1]

8. Conclusion du chapitre

En conclusion, ce chapitre aborde les éléments essentiels de la spécification et de l'analyse des besoins dans le projet. Nous avons examiné en détail le cadre général du projet, la méthodologie de travail, le diagramme de Gantt et la méthode en cascade. Nous avons également exploré l'utilisation de Trello comme outil de gestion de projet, l'UML pour la modélisation, ainsi que les différents types de besoins, à savoir les besoins fonctionnels et non fonctionnels. Enfin, nous avons identifié et étudié les utilisateurs de l'application, afin de prendre en compte leurs besoins et attentes.

Chapitre 3

Analyse et Conception

Introduction

Après avoir élaboré les aspects fondamentaux de mon projet, j'entame maintenant la phase cruciale du cycle de vie, à savoir la conception. Cette étape vise à décrire en détail le fonctionnement de l'ensemble de l'application. Dans ce chapitre, j'aborde la conception et la modélisation de mon projet.

1. Diagramme des Cas d'Utilisation

Les diagrammes des cas d'utilisation sont des outils essentiels en ingénierie logicielle, fournissant une vue d'ensemble du comportement fonctionnel du système. Ils servent non seulement à communiquer efficacement avec la direction et les parties prenantes, mais également à guider le développement en identifiant les interactions essentielles entre les utilisateurs et le système.

1.1. Définitions

Cas d'Utilisation

Ces cas décrivent l'interaction entre l'utilisateur (ou l'acteur) et le système, mettant en lumière les objectifs que l'utilisateur cherche à atteindre en utilisant le système. Chaque cas d'utilisation représente une unité significative de travail.

Acteurs

Ce sont des entités externes interagissant avec le système, qu'il s'agisse d'humains ou de machines. Il est crucial de décrire les acteurs en fonction de leur rôle, reflétant ainsi leurs besoins et leurs capacités dans le contexte du système.

Relations

La norme UML définit trois types de relations, chacune étant graphiquement représentée par des notations spécifiques :

Inclusion

Un cas d'utilisation de base peut explicitement incorporer un autre cas, de manière obligatoire, à des points spécifiques de son déroulement.

Extension

Un cas d'utilisation de base peut implicitement inclure un autre cas, de manière optionnelle, à des points indirects de son déroulement.

Généralisation

Les cas d'utilisation descendants héritent des descriptions de leurs parents communs. Bien qu'ils partagent des aspects généraux, ils peuvent également inclure des relations spécifiques avec d'autres acteurs ou cas d'utilisation.

1.2. Diagrammes

Les acteurs de système

Mon système se compose de quatre parties : un super administrateur qui réalise toutes les tâches du système, un responsable de formation qui s'occupe de la planification des formations et de la gestion des listes de présence, un formateur qui dispense les formations et

valide ou non les évaluations, et un collaborateur qui participe aux formations et passe les évaluations.

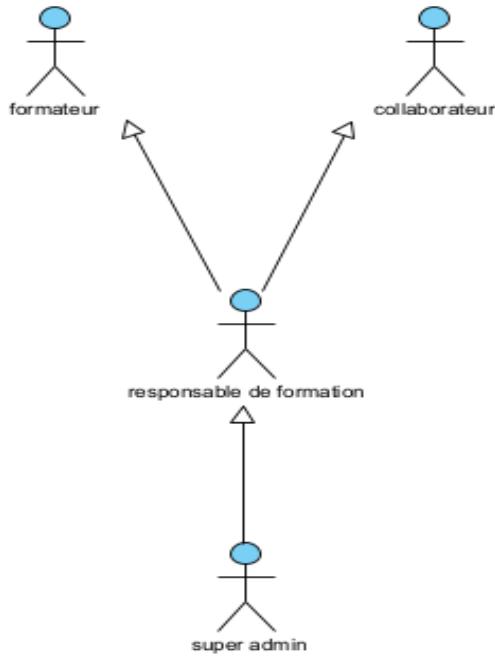


Figure 5 : Diagramme d'héritage entre les acteurs

Cas du responsable de formation

- Partie paramétrage

Le diagramme suivant illustre la partie paramétrage du système, où sont spécifiés les éléments des formations : le thème, le module, le domaine et le lieu.

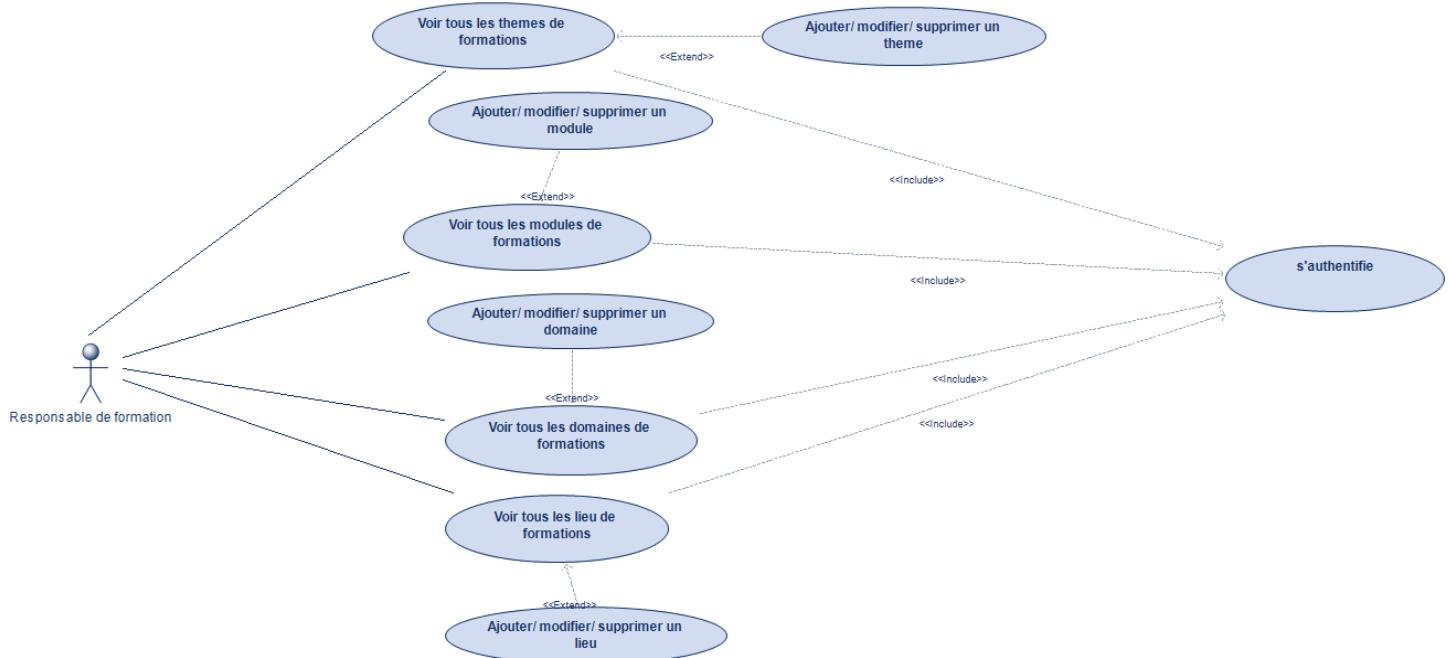


Figure 6 : Diagramme de cas d'utilisation du responsable de formation, partie paramétrage

- Partie de la gestion de formation

Cette section concerne la planification des formations et la gestion de la présence des collaborateurs.

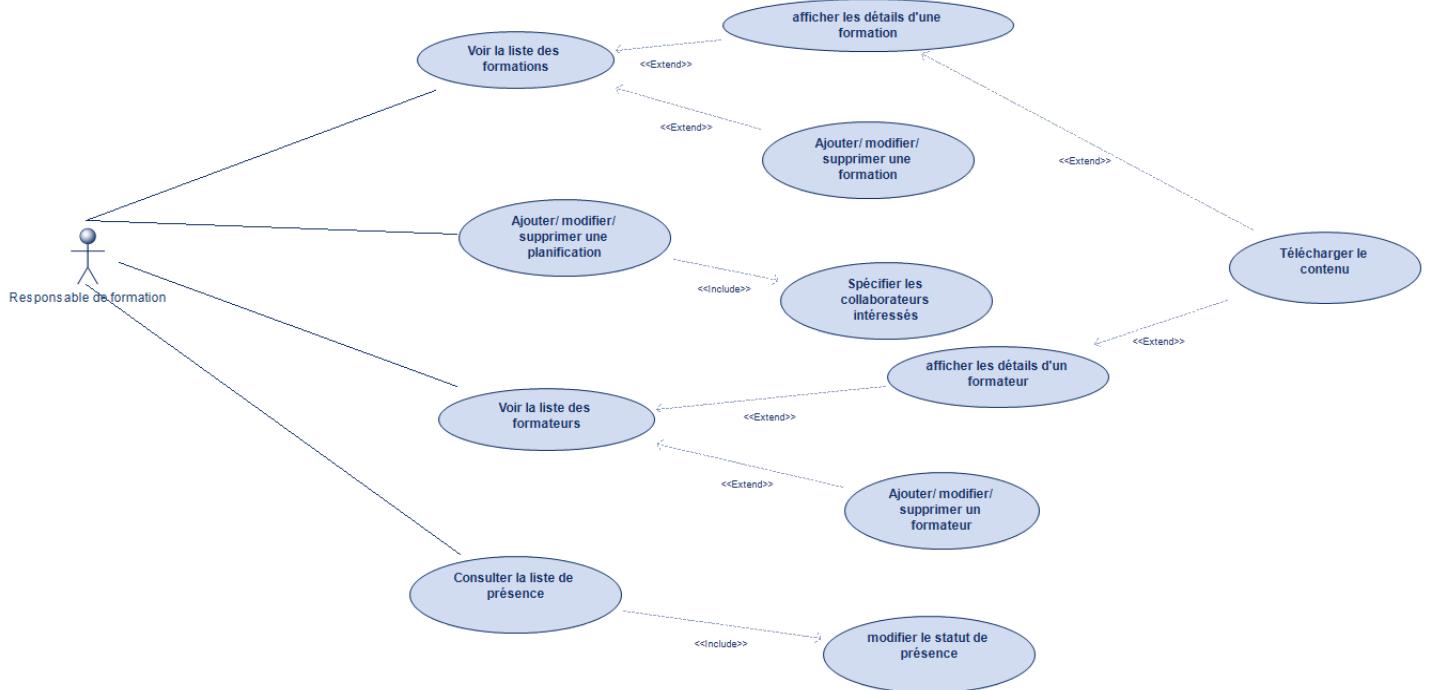


Figure 7 : Diagramme de cas d'utilisation du responsable de formation, partie gestion de formation

Cas de formateur et collaborateur

Ce diagramme de cas d'utilisation présente les actions effectuées par le collaborateur et le formateur

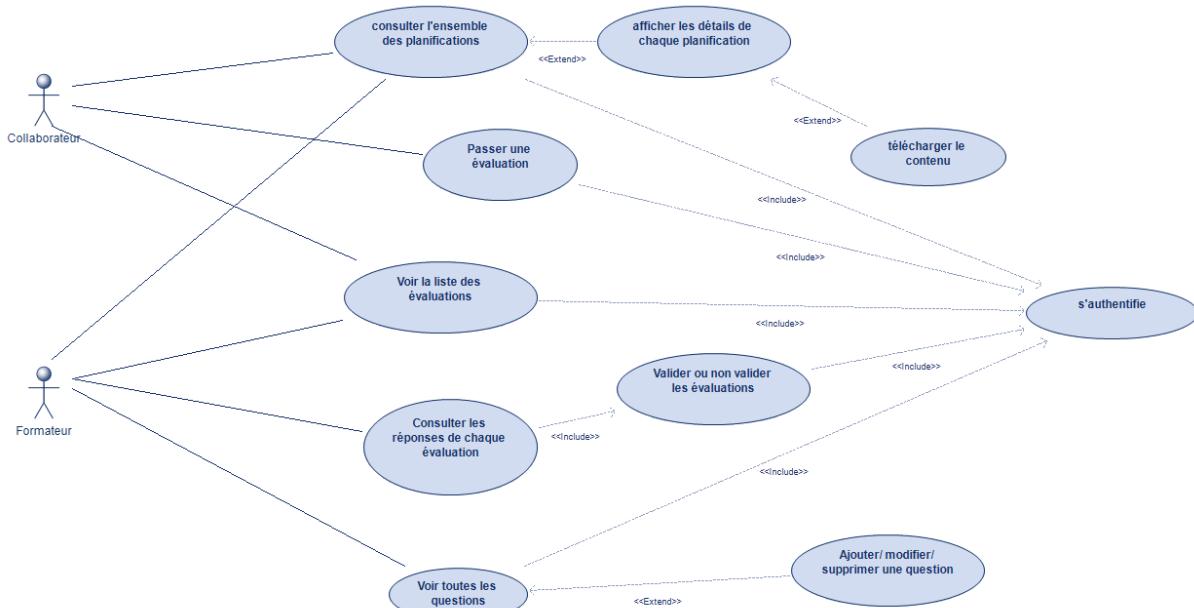


Figure 9 : Diagramme de cas d'utilisation du formateur et du collaborateur

Cas de super administrateur

Le super administrateur accomplit toutes les tâches effectuées par les autres acteurs. Il crée les utilisateurs en définissant pour chacun un rôle. Dans notre cas, nous avons trois rôles : responsable de formation, formateur et collaborateur, sans oublier qu'il a la possibilité de créer d'autres rôles.

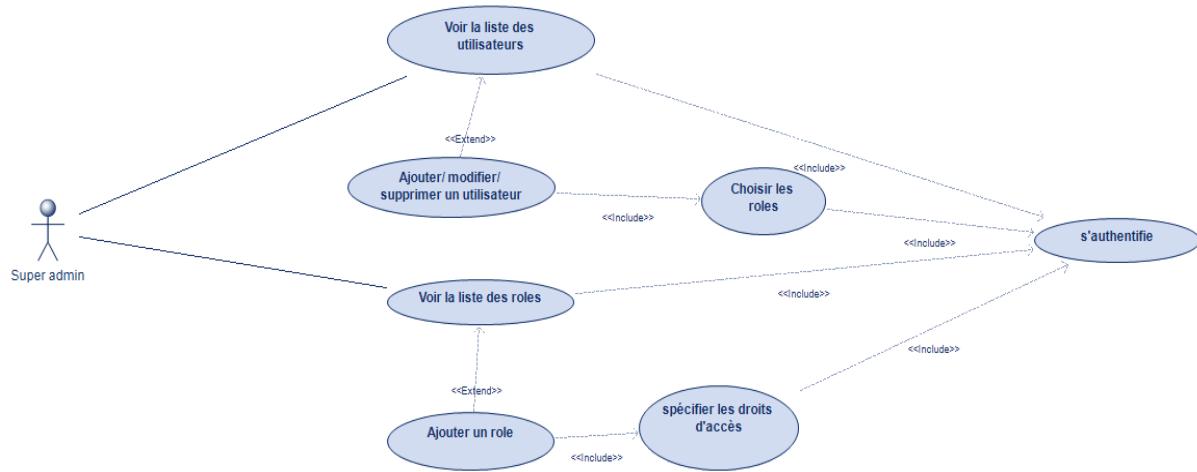


Figure 10 : Diagramme de cas d'utilisation du super administrateur

2. Diagramme de classe

2.1 Définition

Le diagramme de classes est un outil essentiel en génie logiciel pour représenter visuellement les classes, les interfaces et les relations entre elles dans un système. Il offre une vue d'ensemble structurée des composants d'un programme, permettant aux développeurs de mieux comprendre l'organisation et les interactions entre les différentes parties du système. Une classe est un élément fondamental de la programmation orientée objet. Elle regroupe des fonctions et des attributs liés qui décrivent le comportement et les caractéristiques d'un objet. Les fonctions définies dans une classe, appelées méthodes, encapsulent les actions que l'objet peut effectuer, tandis que les attributs représentent les données associées à l'objet. En utilisant des classes, les développeurs peuvent modéliser un programme de manière plus intuitive, en divisant une tâche complexe en plusieurs modules plus simples et réutilisables.

2.2 Diagrammes

Partie sécurité

Dans ce diagramme de classe, la partie sécurité du système est représentée. En effet, la création des utilisateurs n'est pas effectuée par eux-mêmes, mais par le super administrateur. Celui-ci attribue à chaque utilisateur un rôle, et chaque rôle ayant ses propres priviléges et permissions.

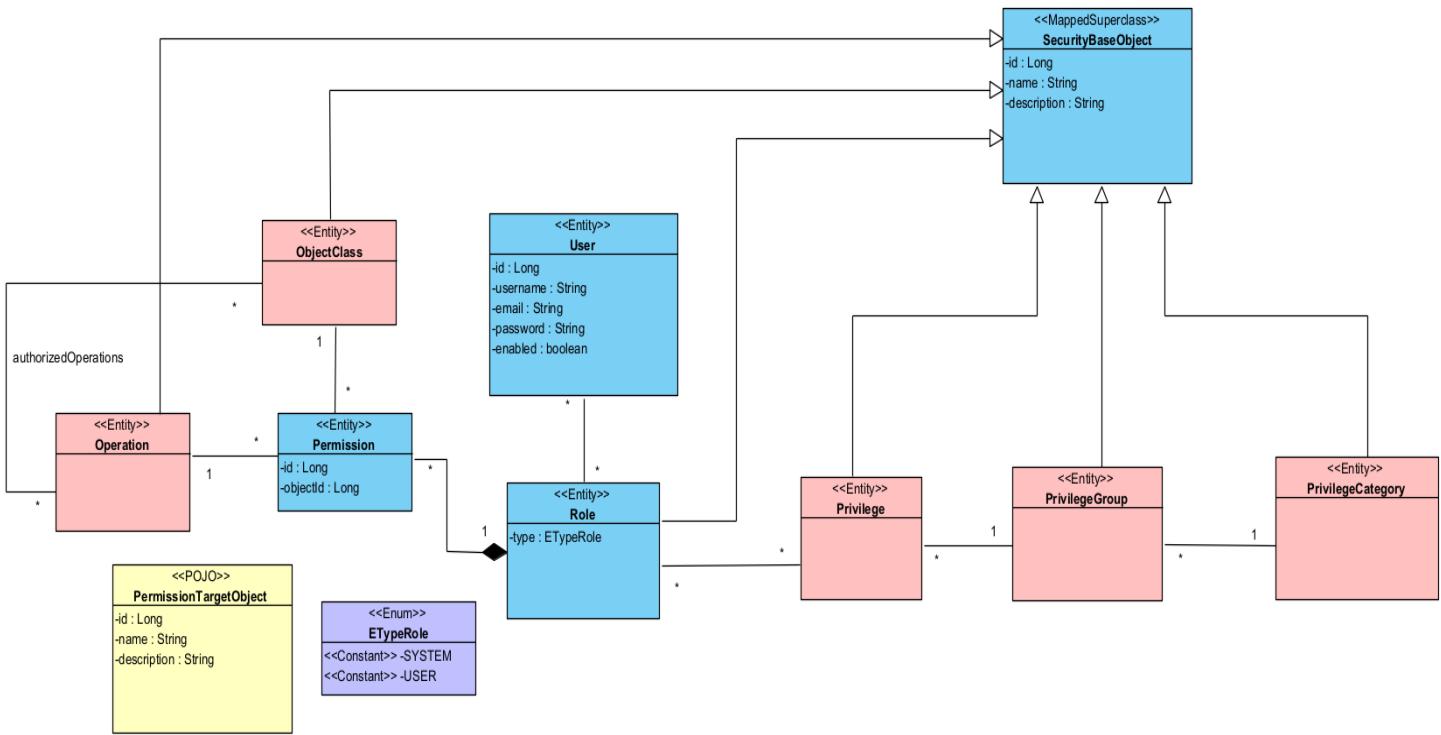


Figure 11 : Diagramme de classe, partie sécurité

Module de formation :

Ce diagramme présente toutes les classes de notre système de formation. Notez que toutes les entités héritent de la classe " BaseEntity ", qui agit comme un enregistreur d'historique.

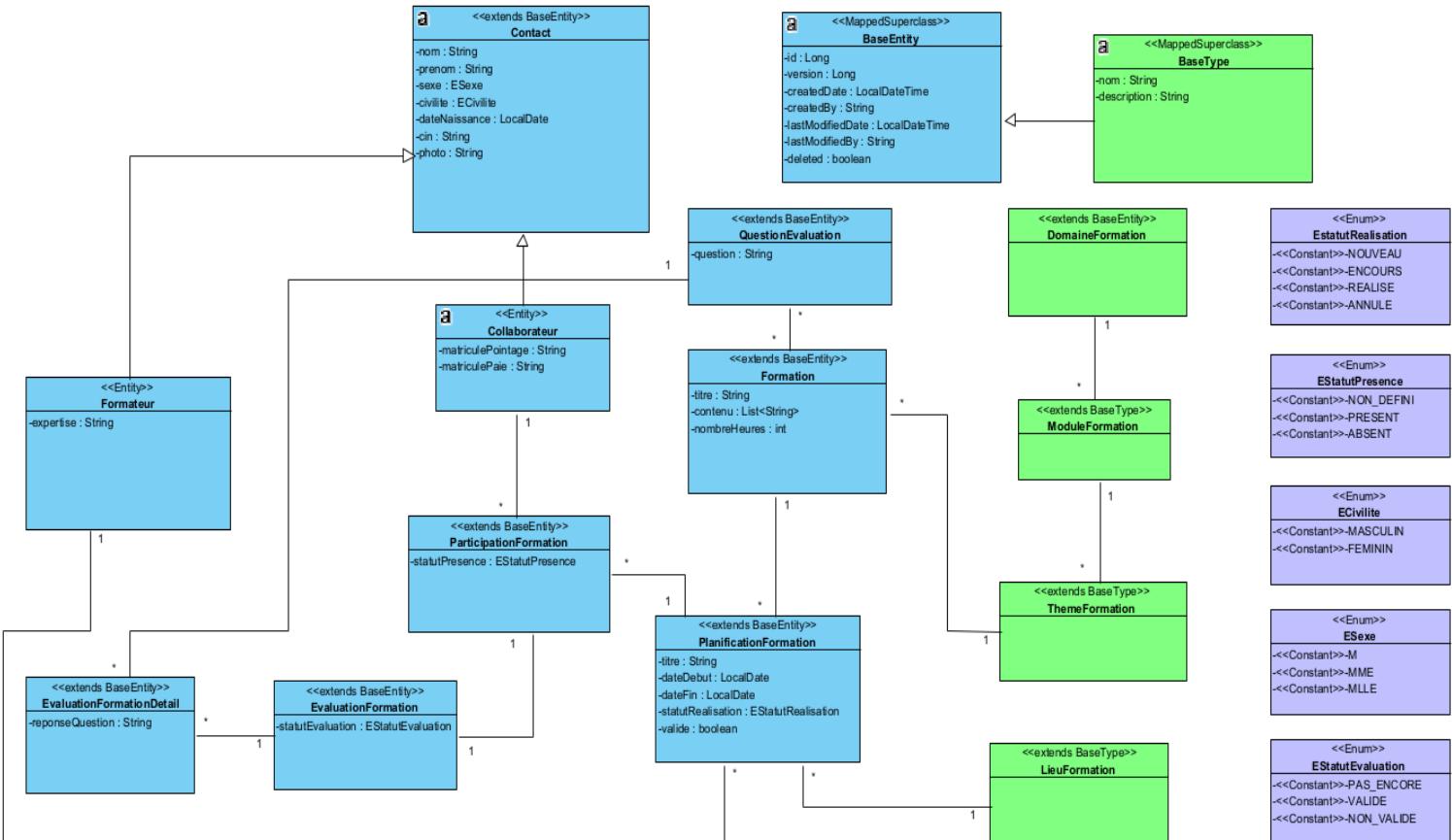


Figure 12 : Diagramme de classe, module de formation

Concernant les couleurs, nous les utilisons pour simplifier le diagramme. Au lieu d'avoir de multiples liaisons qui compliquent la lecture, nous employons des couleurs distinctes. Par exemple, les énumérations sont en violet ; au lieu de les relier aux entités appropriées, nous les distinguons simplement par cette couleur. De plus, les entités en vert sont associées à la partie paramétrage du site web.

3. Diagramme de séquences

3.1 Définition

Un diagramme de séquence est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquence sont organisés en fonction du temps. Le temps s'écoule au fur et à mesure que vous parcourez la page. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence de messages.

Cas d'authentification

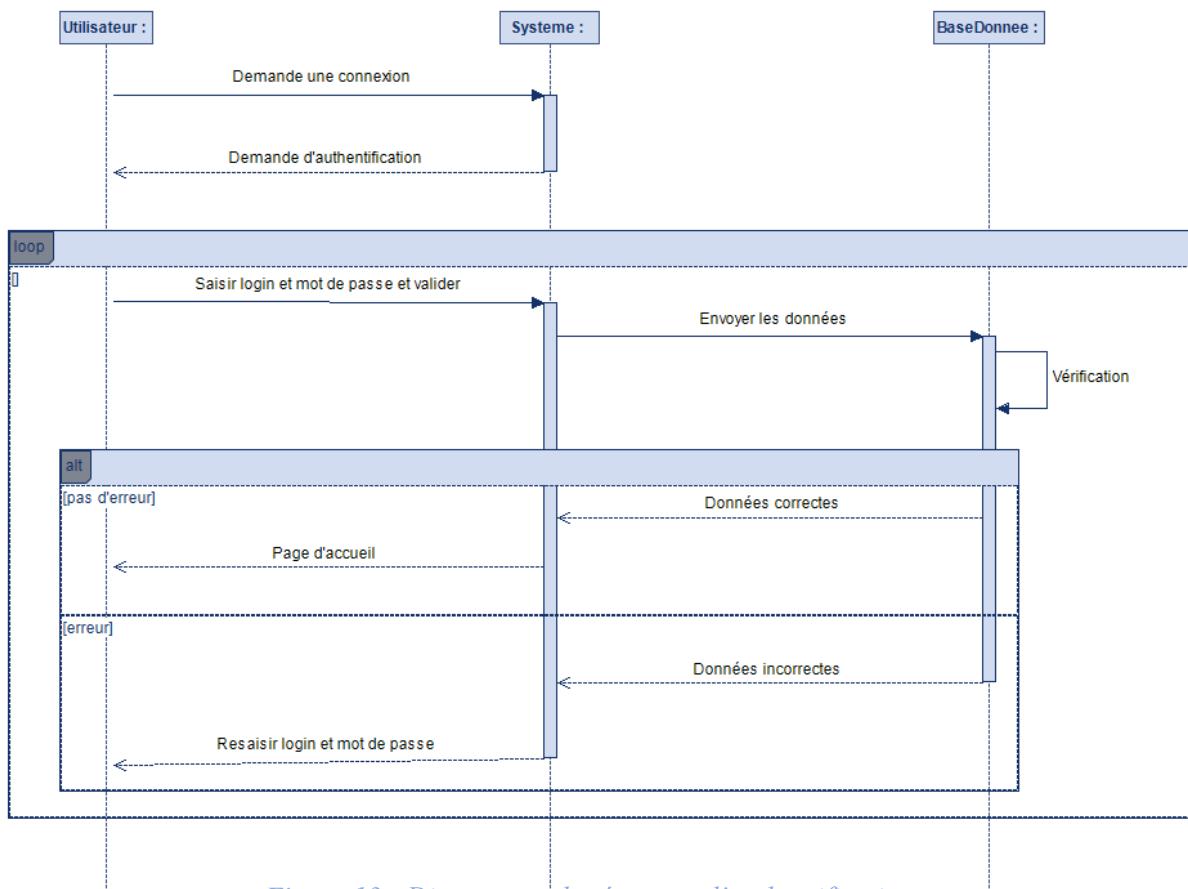


Figure 13 : Diagramme de séquence d'authentification

L'authentification est considérée comme un moyen important pour sécuriser notre application. D'où l'obligation pour l'administrateur de s'authentifier afin d'y accéder. L'utilisateur donc saisie son login et mot de passe, ils sont vérifiés par le système en les comparant avec les ensembles (login, MDP) dans la base de données. Si les données saisies sont justes le système retourne un message de succès et renvoie l'utilisateur à la page principale, sinon il affiche un message d'erreur et demande l'utilisateur de re-saisir son login et mot de passe.

Cas du super admin en relation avec le collaborateur

Déjà c'est le collaborateur qui est intéressé par la formation, tandis que c'est le super administrateur qui s'occupe de sa création, de sa modification et de sa suppression.

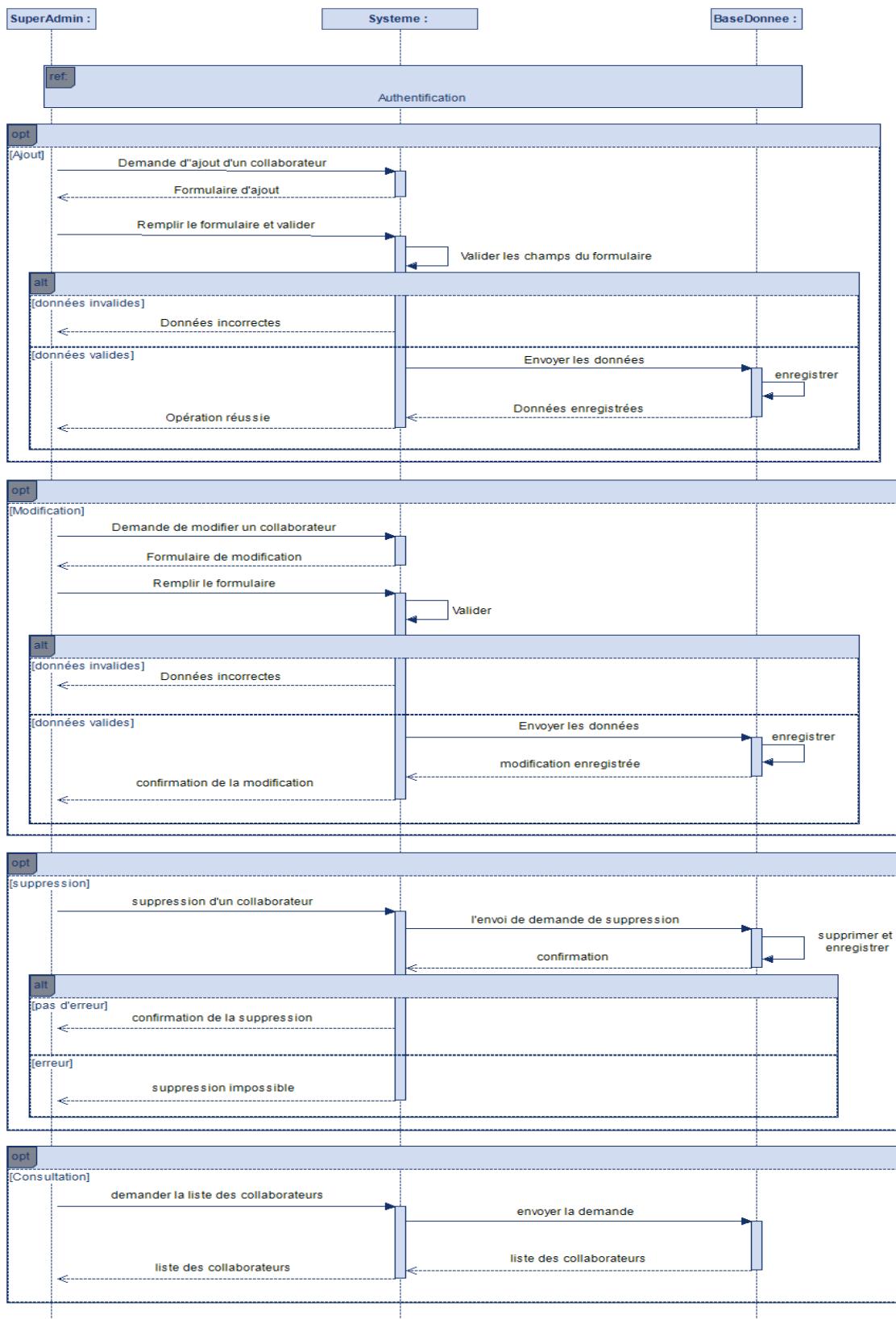


Figure 14 : Diagramme de séquence cas du super admin en relation avec le collaborateur

Cas du responsable de formation en relation avec le formateur

Avant de planifier une formation, il est nécessaire de désigner le formateur chargé de la dispenser. Cela relève de la responsabilité du responsable de la formation.

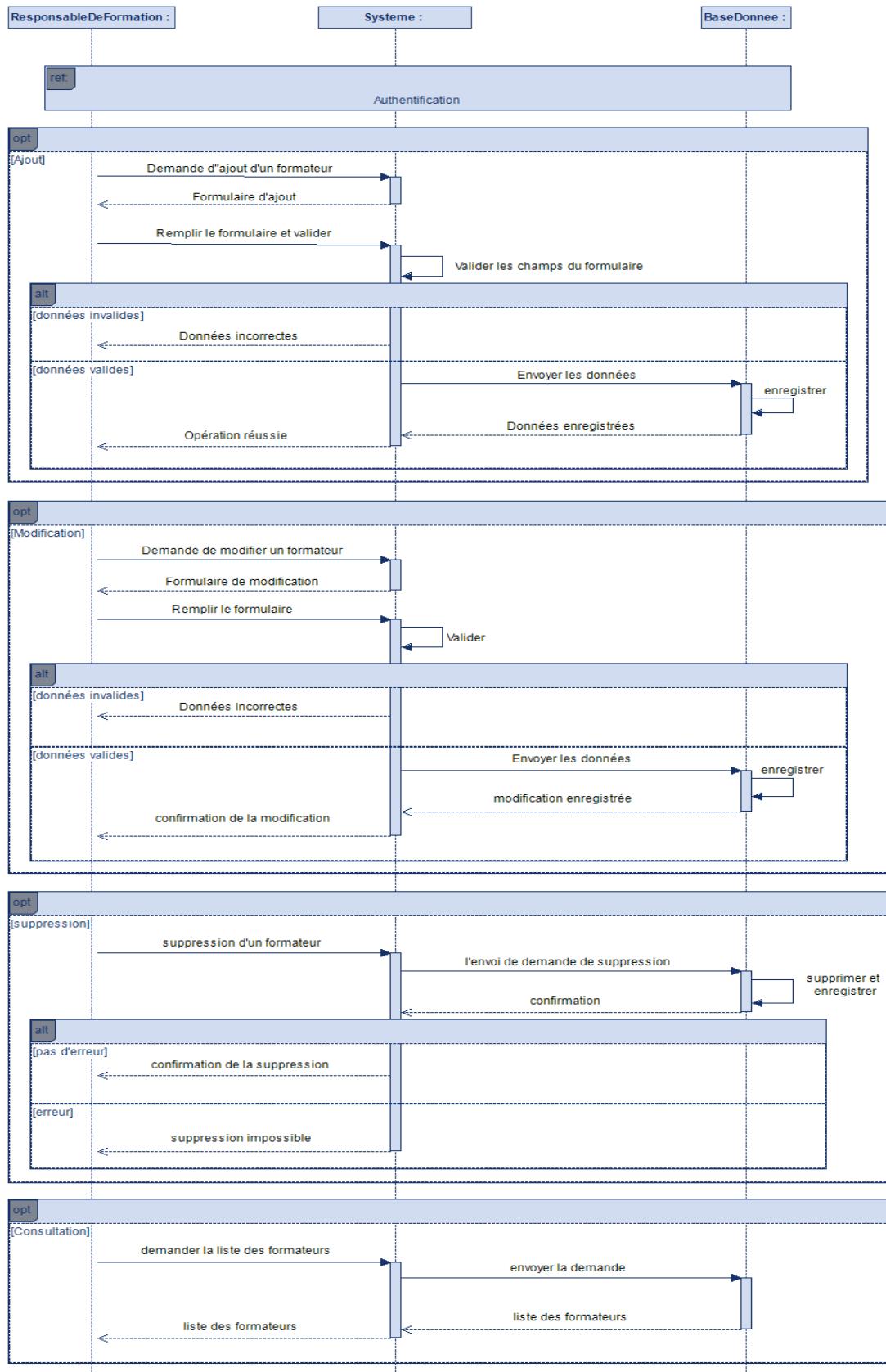


Figure 15 : Diagramme de séquence cas du responsable de formation en relation avec le formateur

Cas du responsable de formation en relation avec la planification de la formation

Lorsque le responsable de département identifie un besoin en matière de formation des collaborateurs, il organise une session pour le combler.

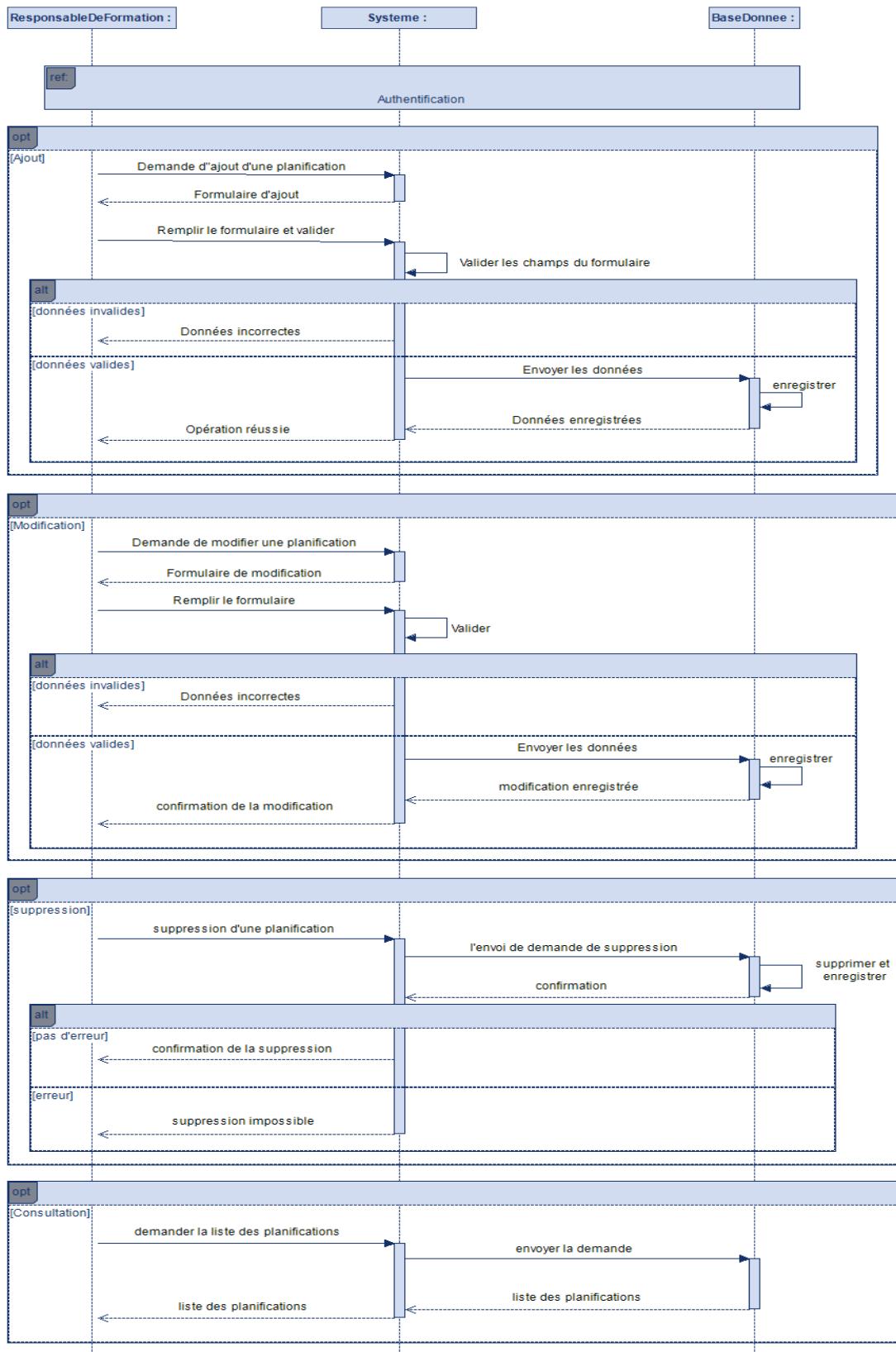


Figure 16 : Diagramme de séquence cas du responsable de formation en relation avec les planifications

Cas du formateur en relation avec les évaluations

Dans ce diagramme de séquence, nous simplifions la manière dont le formateur approuve ou refuse les évaluations de chaque collaborateur.

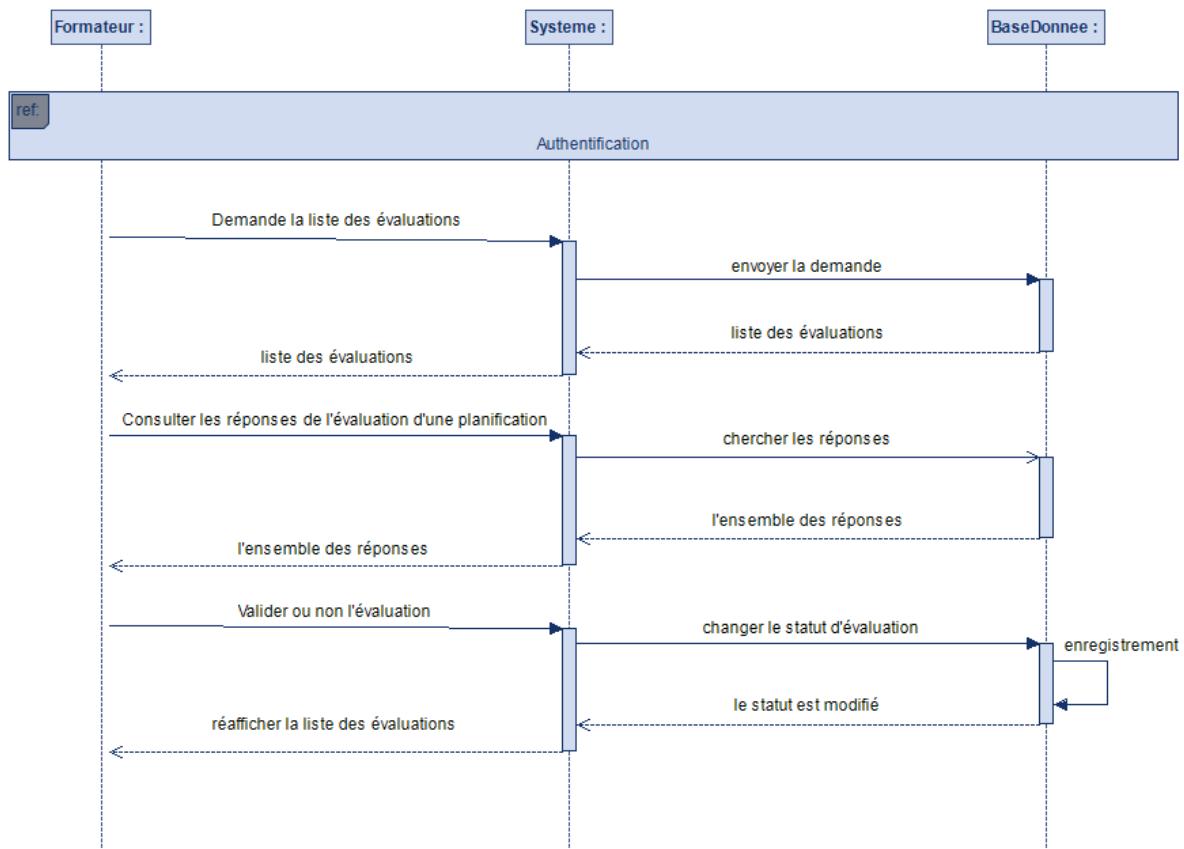


Figure 17 : Diagramme de séquence cas du formateur en relation avec les évaluations

4. Diagramme d'état et transition

4.1 Définition

Un diagramme d'état et de transition est une représentation graphique qui modélise le comportement d'un système en spécifiant ses différents états et les transitions entre ces états en réponse à des événements. Il est utilisé pour décrire le cycle de vie d'un objet ou d'un système, en mettant en évidence les changements d'état et les conditions qui les déclenchent. Ces diagrammes sont largement utilisés en ingénierie logicielle, en systèmes embarqués et dans d'autres domaines pour concevoir et comprendre le fonctionnement des systèmes dynamiques.

4.2 Diagrammes

Cas du statut de réalisation

Ce diagramme montre clairement le flux de la planification d'une formation depuis sa création jusqu'à sa réalisation ou son annulation. Voici les états qu'on peut avoir :

- **Nouvelle** : La formation est nouvellement créée et pas encore commencée.
- **En cours** : La formation est en cours de réalisation.
- **Réalisée** : La formation a été complétée avec succès.
- **Annulée** : La formation a été annulée, soit avant qu'elle ne commence, soit pendant sa

réalisation.

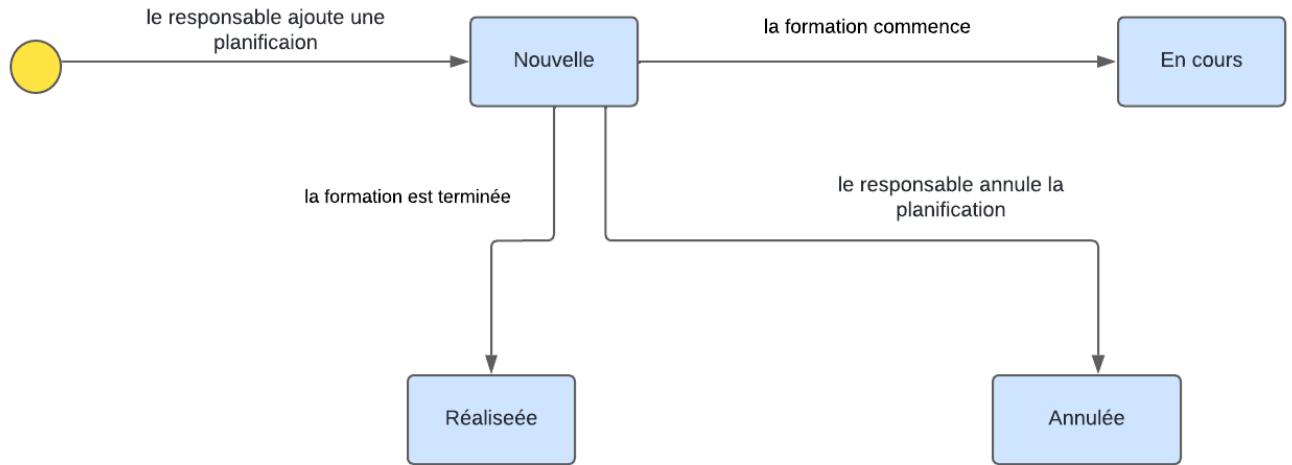


Figure 18 : Diagramme d'état-transition de la réalisation d'une planification

Cas du statut de présence des participants

Ce diagramme montre clairement le flux de la présence des participants depuis leur ajout à la planification jusqu'à la validation de leur présence ou absence par le responsable.

Voici l'explication de chaque état :

- Planifié** : État initial lorsque le collaborateur est planifié pour la formation.
- Non définie** : État après la planification mais avant que la présence ou l'absence ne soit validée.
- Présent** : Le collaborateur est marqué comme présent après validation par le responsable.
- Absent** : Le collaborateur est marqué comme absent après validation par le responsable.

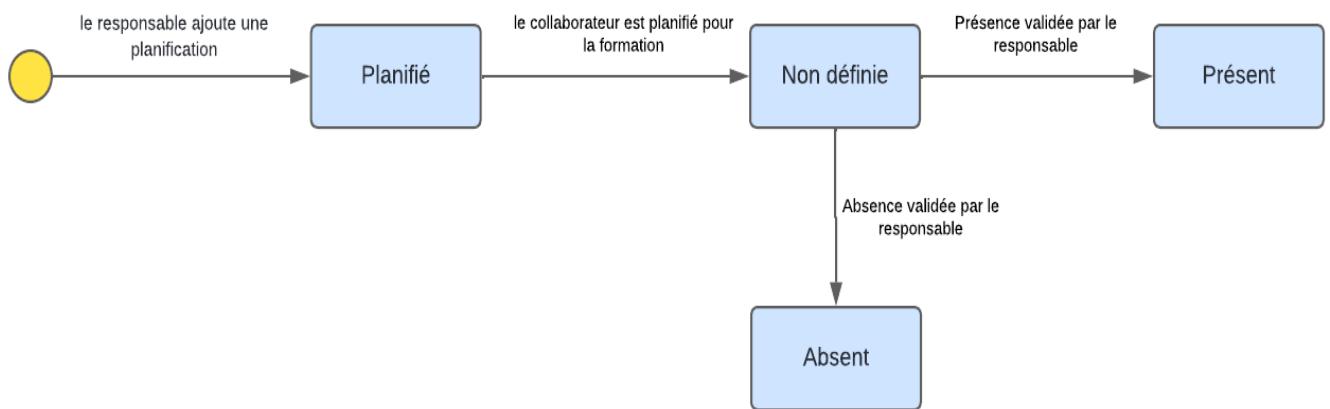


Figure 19 : Diagramme d'état-transition de la présence des participants

Cas du statut d'évaluation

Ce diagramme montre clairement le flux de l'évaluation depuis sa planification jusqu'à sa validation ou son rejet par le formateur.

Voici une explication de chaque état :

- **Planifiée** : L'évaluation est dans cet état dès qu'elle est créée et planifiée.
- **Réponses reçues** : Une fois que les participants répondent à l'évaluation, elle passe à cet état.
- **En attente de validation** : Lorsque toutes les réponses sont reçues, l'évaluation attend la validation par le formateur.
- **Validée** : Si le formateur valide l'évaluation, elle passe à cet état.
- **Non validée** : Si le formateur refuse de valider l'évaluation, elle passe à cet état.

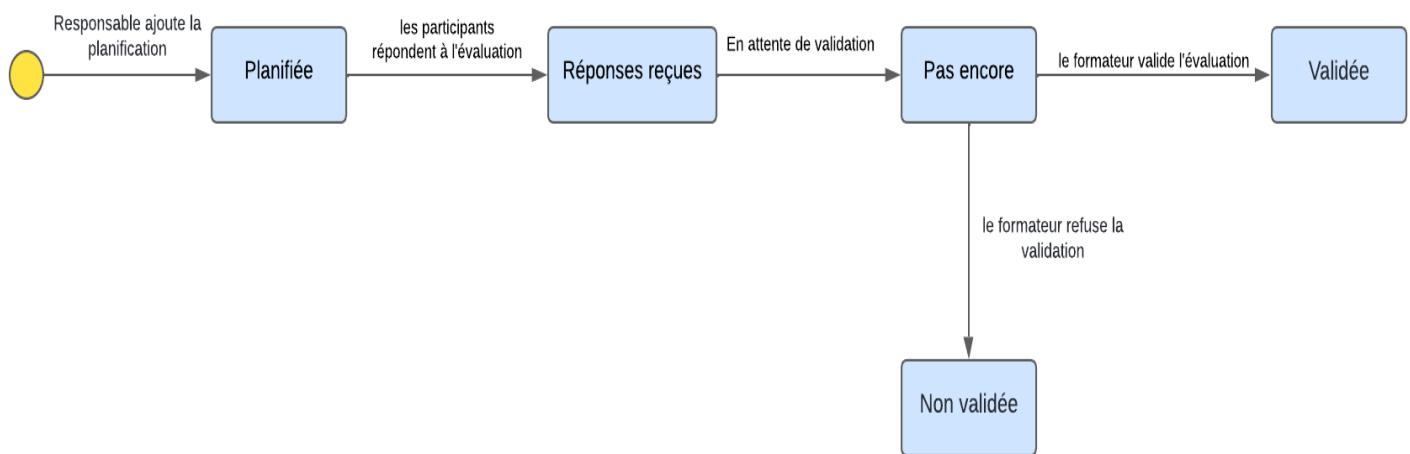


Figure 20 : Diagramme d'état-transition d'une évaluation

Conclusion

À la fin de ce chapitre consacré à la modélisation et à la conception de notre projet, nous avons exploré une variété de diagrammes. Ces représentations nous ont offert une vision claire et organisée de la logique métier. En tant qu'outils essentiels, ces diagrammes facilitent grandement notre compréhension et notre communication du projet.

Chapitre 4

Etude Techniques

Introduction

Dans ce chapitre consacré à l'étude technique, je vais présenter l'environnement de travail que j'ai utilisé pour la réalisation de ce projet. Cependant, contrairement à la structure habituelle, je ne m'attarderai pas sur l'environnement matériel, mais je me concentrerai exclusivement sur l'environnement logiciel. Celui-ci englobe les outils et les programmes utilisés pour le développement, le test et le déploiement de ma solution. Je détaillerai les choix technologiques effectués ainsi que les raisons qui ont motivé ces sélections, en mettant en lumière leur impact sur la conception et l'implémentation de ma solution.

Outils et Technologies de Développement

Le choix des technologies utilisées dans un projet joue un rôle crucial dans la qualité du travail réalisé. Des outils appropriés peuvent rendre le processus de développement plus fluide et moins laborieux. Le succès ou l'échec d'un projet dépend en partie des technologies employées. Il est important de noter qu'aucune technologie n'est universellement supérieure à une autre ; l'adéquation dépend fortement du type de projet concerné. Dans notre cas, certaines technologies ont été définies par l'entreprise, tandis que d'autres ont été choisies par moi-même pour mieux répondre aux besoins spécifiques du projet.

1. Architecture logique

Les patrons de conception (Design Patterns) sont des solutions éprouvées qui résolvent des problèmes de conception bien identifiés. Leur utilisation améliore la qualité du développement et en réduit la durée. En appliquant ces modèles, on peut diminuer le couplage au sein d'une application, apporter de la flexibilité, faciliter la maintenance et, de manière générale, encourager le respect des bonnes pratiques de développement. Parmi les patrons de conception que nous avons utilisés dans notre projet, on peut citer :

1.1. Inversion de contrôle

L'inversion de contrôle (IoC) est un principe architectural commun à tous les frameworks de développement et d'exécution. Ce principe repose sur le fait que le flux d'exécution d'une application n'est plus géré directement par l'application elle-même, mais par le framework ou la couche logicielle sous-jacente.

IoC est un terme générique qui peut prendre différentes formes en fonction du problème à résoudre, la plus connue étant l'injection de dépendances. Cette approche permet, en programmation orientée objet, de réduire le couplage entre les objets en déléguant la gestion de leurs dépendances à un conteneur.

Le framework Spring, par exemple, utilise un système de "bean". Un bean est un objet qui est instancié, assemblé et géré par le conteneur IoC de Spring. L'inversion de contrôle permet de définir les dépendances d'un objet sans les créer directement. Lors de la création des objets, Spring injecte les beans entre eux pour satisfaire toutes leurs dépendances.

1.2. MVC

Nous avons adopté une architecture couramment utilisée et fortement recommandée : l'architecture MVC (Modèle Vue Contrôleur).

Cette structure est spécialement conçue pour améliorer la séparation entre les données et leur présentation. Le modèle gère les données, la vue affiche ces données, et le contrôleur établit le

lien entre les deux, en agissant comme un médiateur pour les requêtes reçues par l'application. La figure suivante illustre le fonctionnement de l'architecture MVC :

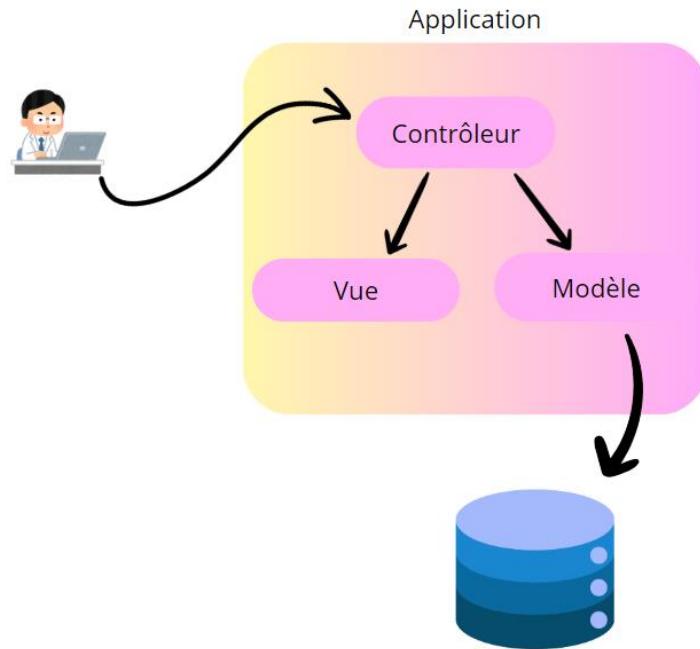


Figure 21 : le fonctionnement de l'architecture MVC

Dans Spring MVC, le DispatcherServlet joue un rôle central en s'assurant que chaque requête entrante est correctement redirigée vers le contrôleur adéquat. Ce composant sert de contrôleur frontal (Front Controller), une approche introduite avec la deuxième version du modèle MVC. Le DispatcherServlet intercepte toutes les requêtes HTTP et les achemine vers les contrôleurs appropriés. La figure suivante montre comment le DispatcherServlet fonctionne :

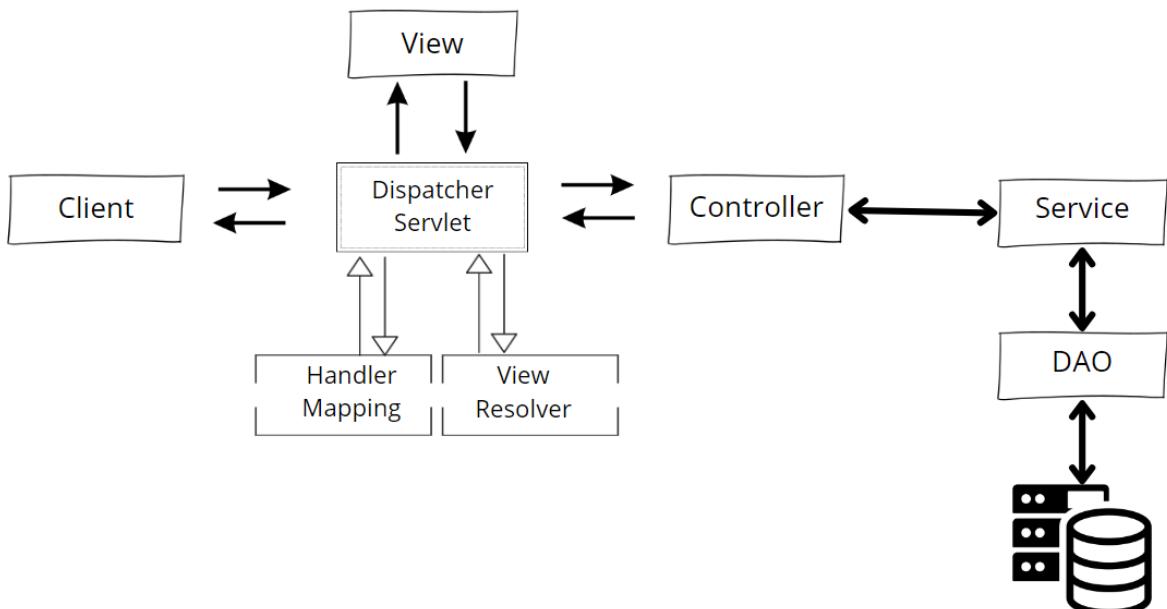


Figure 22 : le fonctionnement de DispatcherServlet

1.3. Data Transfer Object

Un objet de transfert de données (DTO), également appelé Data Transfer Object, est un patron de conception utilisé dans les architectures logicielles orientées objet. Son but est de simplifier les transferts de données entre les sous-systèmes d'une application logicielle. Voici quelques points clés à retenir :

- Transfert de données entre processus : Un DTO est un objet qui transporte des données entre différents processus. Il facilite la communication entre deux systèmes, tels qu'une API et un serveur, sans risquer d'exposer des informations sensibles.
- Confidentialité et découplage : Les DTO permettent de préserver la confidentialité des données en ne transférant que les informations nécessaires. Par exemple, si vous avez besoin du nom et de la photo d'un collaborateur pour lui accorder l'accès à votre entreprise, le DTO peut transmettre uniquement ces informations sans exposer d'autres données sensibles.
- Conception simple et lisible : Un bon DTO doit contenir uniquement des données, sans logique métier. Il s'agit d'un petit objet simple qui ne réalise qu'une seule tâche. Évitez d'utiliser du code standard et créez chaque DTO en partant de zéro.

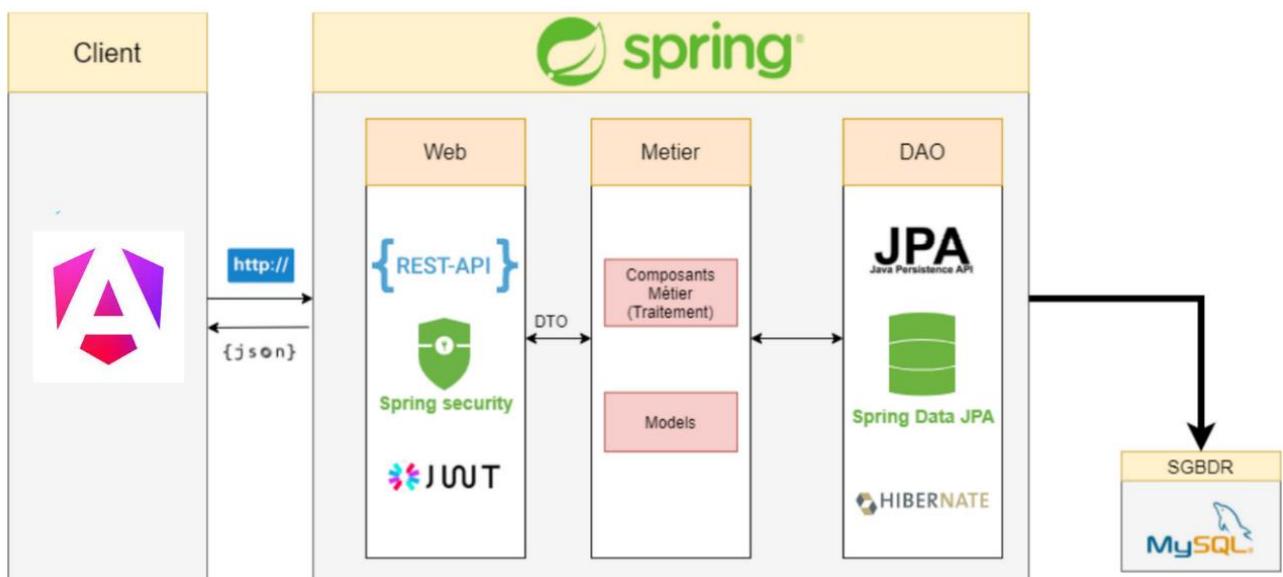


Figure 23 : l'architecture global du projet

Cette architecture favorise une séparation claire des responsabilités, la réutilisabilité et l'évolutivité grâce à l'utilisation du Framework Spring Boot. Afin de bien comprendre cette architecture, nous allons procéder à l'explication de différentes couches :

- Couche Présentation : Elle assure la logique de navigation. Elle se compose d'une partie frontale en Angular pour la représentation graphique des informations et une partie en Back End avec un contrôleur frontal DispatcherServlet fourni par Spring, et assure l'interaction avec les clients et l'échange des données en format JSON. Aucun traitement métier n'est à implémenter dans cette couche.

- Couche Métier : C'est dans cette couche qu'on doit retrouver tous les traitements métiers de l'application. Ces traitements seront implantés sous forme de services Spring. Spring est un conteneur léger permettant de simplifier l'intégration des différentes couches via l'utilisation des interfaces Java, du design pattern IOC de la programmation orientée aspect AOP. En plus, Spring est totalement portable sur tous les serveurs d'application et totalement

intégrables avec toutes les technologies choisies pour le développement de ce projet.

- **Couche Persistance** : C'est la couche qui permet de sauvegarder et faire appel aux données persistantes de l'application. Elle est implémentée via des services d'accès aux données qui feront appel au système de gestion de la base de données MySQL.

2. Choix des outils technologiques du backend

Avant d'entamer le premier sprint afin d'implémenter la première partie de notre application, une étude technologique a été tout d'abord réalisée. Dans le cadre de cette étude, les outils et technologies qui sont proposés, sont comme suit :

2.1. Choix de Modèle de communication entre le client et le serveur

J'ai choisi d'utiliser une API REST car elle est facile à prendre en main et à maîtriser. De plus, c'est actuellement le modèle le plus répandu, et nous pouvons l'apprendre partout. Et déjà l'entreprise travail avec.



Figure 24 : logo d'API REST

2.2. Choix des outils et technologies de développement de Backend

a) Le langage de programmation



Figure 25 : logo de Java EE

Dès le début du projet, l'entreprise où je réalise mon stage m'a demandé de travailler avec Java EE et le Spring Framework, car ce sont des technologies basées sur le langage Java que nous maîtrisons bien. J'ai également suivi des formations préalables pour me familiariser avec ces outils.

b) Le gestionnaire de projet



Figure 26 : logo de Maven

Maven est un outil d'automatisation de la construction de projets en utilisant Java. Il permet de déterminer comment un logiciel spécifique est construit, ainsi que ses différentes dépendances. Voici quelques points clés à retenir :

- **Fichier XML** : Maven utilise un fichier XML appelé pom.xml pour décrire le projet, les dépendances du logiciel, l'ordre de construction et les plugins nécessaires.
- **Cibles prédéfinies** : Maven propose des cibles prédéfinies pour des tâches courantes telles que l'empaquetage (création d'un fichier JAR ou WAR) et la compilation. Cela simplifie le processus de construction.
- **Gestion des dépendances** : L'un des avantages majeurs de Maven est sa gestion des dépendances. En utilisant le fichier pom.xml, vous pouvez spécifier les bibliothèques et les modules dont votre projet a besoin. Maven télécharge automatiquement ces dépendances depuis des référentiels en ligne.
- **Gradle** : Bien que Maven soit largement utilisé, il existe d'autres outils similaires, comme Gradle. Gradle utilise le langage Groovy pour définir la configuration du projet. Cependant, comme vous l'avez mentionné, vous avez opté pour Maven en raison de sa simplicité et de sa familiarité avec le format XML.

c) Autres outils et technologies

Le tableau suivant représente l'ensemble des technologies de développement backend utilisées dans le projet :

Technologie	Description
 The logo for Spring, featuring a green circular icon with a white 'S' shape inside, followed by the word "spring" in a lowercase, bold, sans-serif font with a registered trademark symbol.	<p>Spring est un socle pour le développement d'applications. Il fournit de nombreuses fonctionnalités qui peuvent être configurées et utilisées de plusieurs manières. Ceci laisse le choix au développeur d'utiliser la solution qui lui convient le mieux et/ou qui répond aux besoins. Spring est ainsi un des frameworks les plus répandus dans le monde Java, et sa popularité a grandi grâce à la qualité et la richesse de ses fonctionnalités [2].</p>

Figure 27 : logo de Spring



Figure 28 : logo de Spring Boot

Spring Boot est un framework open source qui simplifie le développement d'applications web et de microservices avec Spring Framework. Il offre une configuration automatique, une approche directive de la configuration et la possibilité de créer des applications autonomes¹. En résumé, Spring Boot accélère le développement en fournissant des fonctionnalités prêtes à l'emploi [2].



Figure 29 : logo de Spring Security

Spring Security est un framework qui fournit l'authentification, l'autorisation et la protection contre les attaques courantes. Il est largement utilisé pour sécuriser les applications basées sur Spring. Que ce soit pour des applications impératives ou réactives, Spring Security est la norme de facto pour la sécurité des applications Spring [2].

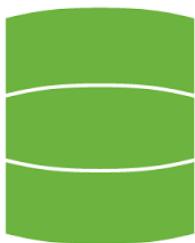


Figure 30 : logo de Spring Data

Spring Data est un projet qui simplifie l'accès aux données en offrant une couche d'abstraction commune à plusieurs sources de données. Il s'interface avec des technologies telles que JPA, Neo4j, MongoDB, etc. Spring Data permet d'écrire plus simplement l'accès aux données en utilisant des interfaces et des annotations [2].



Figure 31 : logo de JPA

JPA est une spécification standard pour la gestion de la persistance des données en Java. Elle permet de mapper les objets Java aux tables de la base de données et d'exécuter des opérations de persistance. Les entités JPA sont de simples POJOs annotés, et l'API fournit des fonctionnalités pour les manipuler (ajout, modification, suppression, recherche) [2].



Figure 32 : logo de Hibernate

Hibernate est un framework Java qui simplifie le développement d'applications Java pour interagir avec la base de données. Il s'agit d'un outil de mapping objet-relationnel léger, qui implémente les spécifications de JPA pour la persistance des données [3].



Figure 33 : logo de JWT

JSON Web Token (JWT) est un standard ouvert qui permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Ces jetons sont principalement utilisés pour partager des données d'authentification entre applications, et ils sont transparents pour les utilisateurs [4].



Figure 34 : logo de Lombok

Lombok est une API dont le but est de générer à la compilation, du code Java. Tout se fait à l'aide de simples annotations à poser dans les classes [5].



Figure 35 : logo de Swagger

Swagger 3, également connu sous le nom d'OpenAPI Specification (OAS), est un langage de description d'interface qui permet de décrire des API à l'aide de JSON. Il facilite la conception, la documentation et l'utilisation d'API tout en fournissant une description standardisée des services web [6].

Tableau 2 : Technologies de développement backend

Le tableau suivant représente l'ensemble des outils de développement backend utilisés dans le projet :

Outil	Description
	IntelliJ IDEA est un environnement de développement intégré (IDE) développé par JetBrains pour la programmation. Principalement destiné à Java, cet IDE permet de maximiser la productivité des développeurs en automatisant les tâches répétitives. Il offre des fonctionnalités telles que la complétion intelligente du code, l'analyse statique du code et les refactorings, ce qui rend le développement logiciel non seulement productif, mais aussi agréable [7].
	Postman est un client API qui permet aux développeurs de créer, partager, tester et documenter facilement un API. Ceci est fait en permettant aux utilisateurs de créer et d'enregistrer des requêtes HTTP simples et complexes, ainsi que de lire leurs réponses [8].

Tableau 3 : Outils de développement backend

3. Choix des outils technologiques du backend

Tout d'abord, **Visual Studio Code** est l'outil privilégié pour le développement front-end, car il offre une interface conviviale et des fonctionnalités puissantes.

Visual Studio Code, également couramment appelé VS Code, est un éditeur de code source développé par Microsoft pour Windows, Linux, macOS et les navigateurs web¹. Il offre des fonctionnalités telles que la mise en évidence de la syntaxe, l'achèvement intelligent du code, les extraits de code, le refactoring du code et le contrôle de version intégré avec Git [9].

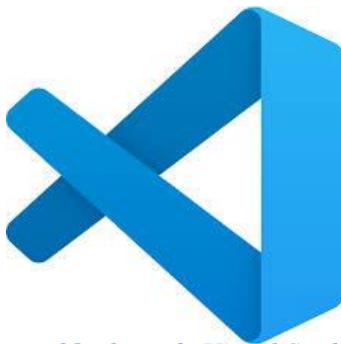


Figure 38 : logo de Visual Studio Code

Le tableau ci-dessous présente toutes les technologies de développement frontend utilisées dans le projet :

Technologie	Description
 <i>Figure 39 : logo de Angular 17</i>	Angular est un framework de conception d'applications et une plateforme de développement permettant de créer des applications à page unique (SPA) efficaces et sophistiquées. Développé par Google, Angular offre des fonctionnalités puissantes qui facilitent la création d'applications robustes. Concernant mon projet j'ai utilisé la dernière version [10].
 <i>Figure 40 : logo de HTML 5</i>	HyperText Markup Language (HTML) est un langage de balisage utilisé pour la création de pages web. Il est considéré comme étant le langage de base pour la création des sites internet. Il permet de structurer et de mettre en forme le contenu des pages [11].
 <i>Figure 41 : logo de CSS 3</i>	Les feuilles de style en cascade (CSS) sont un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage tel que HTML [12].



Figure 42 : logo de TypeScript

TypeScript est un langage de programmation libre et open source développée par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un surensemble syntaxique strict de JavaScript, ce qui signifie que tout code JavaScript correct peut être utilisé avec TypeScript [13].

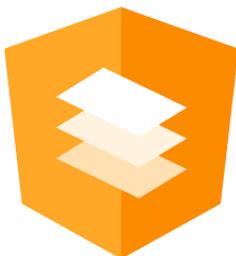


Figure 43 : logo de Angular Material

Angular Material est une bibliothèque de composants d'interface utilisateur qui met en œuvre les principes du Material Design pour les applications web Angular. Elle permet de créer des interfaces utilisateur réactives, cohérentes et esthétiques pour vos projets web. Les composants d'Angular Material sont bien testés, accessibles et personnalisables tout en respectant les spécifications du Material Design [14].



Figure 44 : logo de RxJS

RxJS est une bibliothèque de programmation réactive utilisant des Observables, ce qui facilite la composition de code asynchrone ou basé sur des rappels. Elle est largement utilisée dans le développement d'applications Angular et d'autres projets JavaScript [15].



Figure 45 : logo de Zone.js

Zone.js est une bibliothèque JavaScript qui joue un rôle essentiel dans le fonctionnement d'Angular. Elle crée et gère des contextes d'exécution pour les opérations asynchrones telles que setTimeout, Promise, et autres événements.



Figure 46 : logo de jsPDF

jsPDF est une bibliothèque JavaScript qui permet de générer des fichiers PDF directement depuis le navigateur. Elle offre une solution simple pour créer des documents PDF à partir de contenu HTML, de graphiques, de tableaux et d'autres éléments [16].



Figure 47 : logo de Iconify

Iconify est un framework d'icônes pour les développeurs web. Il permet d'inclure des icônes dans des applications web en utilisant une syntaxe simple et intuitive, et offre une bibliothèque d'icônes gratuite et open source [17].

Tableau 4 : Technologies de développement frontend

4. Outils de modélisation

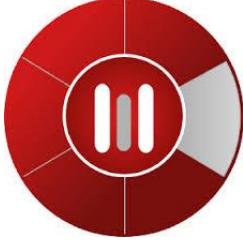
Outil	Description
 Visual Paradigm <p>Figure 48 : logo de Visual Paradigm</p>	<p>Visual Paradigm est un puissant outil de modélisation UML et de génération de code. Il permet aux développeurs de concevoir, visualiser et documenter leurs projets logiciels. Visual Paradigm offre une suite complète d'outils pour créer des diagrammes UML, générer du code à partir de ces modèles, collaborer avec d'autres membres de l'équipe et gérer l'ensemble du cycle de vie du développement logiciel [18].</p>
 <p>Figure 49 : logo de Modelio</p>	<p>Modelio est un outil de modélisation open source développé par Modeliosoft, permet aux développeurs de concevoir, visualiser et documenter leurs projets logiciels. Il offre une large gamme de fonctionnalités pour créer des diagrammes UML, générer du code à partir de ces modèles, collaborer avec d'autres membres de l'équipe et gérer l'ensemble du cycle de vie du développement logiciel [19].</p>

Tableau 5 : Outils de modélisation

- Pour les diagrammes de classe et de cas d'utilisation, l'entreprise m'a demandé d'utiliser Visual Paradigm, car il est compatible avec leurs outils existants. Quant aux autres types de diagrammes, j'ai choisi Modelio, car j'ai déjà de l'expérience avec cet outil et je le maîtrise bien.

5. Outils et technologies de contrôle de versions

Outil	Description
 <p>Figure 50 : logo de GIT</p>	<p>Git est un système de contrôle de version (VCS) qui permet de gérer et de suivre l'historique du code source. Il permet de créer des instantanés de vos projets au fil du temps, ce qui facilite la collaboration et la gestion des modifications [20].</p>
 <p>Figure 51 : logo de GitHub</p>	<p>GitHub est un service d'hébergement basé sur le cloud qui vous permet de gérer vos référentiels Git. C'est là que vous pouvez héberger, partager et collaborer sur vos fichiers de code. GitHub utilise Git en interne, ce qui signifie que vous travaillez avec Git lorsque vous utilisez GitHub [21].</p>

Tableau 6 : Outils et technologies de contrôle de versions

→ J'ai principalement utilisé ces outils pour organiser mon travail personnel. Chaque jour, je définis des tâches que je dois commettre. De plus, ils m'ont été très utiles lorsque j'ai rencontré un problème ou quelque chose qui ne se voyait pas directement : je pouvais alors éliminer les nouvelles modifications.

6. Outils et technologies de gestion de base des données



Figure 52 : logo de MySQL

MySQL est un système de gestion de base de données relationnel open source. Il stocke les données dans des tables composées de lignes et de colonnes. Les utilisateurs peuvent définir, manipuler, contrôler et interroger les données à l'aide du langage de requête structuré, plus communément appelé SQL. MySQL est largement utilisé pour sa facilité d'utilisation, sa sécurité, ses performances et sa flexibilité. Il est souvent intégré dans des piles technologiques telles que LAMP (Linux, Apache, MySQL et Perl/PHP/Python) et est utilisé par de nombreuses applications web populaires.

Le choix d'utilisation sera bénéfique à la fois pour l'entreprise et pour moi, car je maîtrise bien ce système [22].

7. Outils de design



Canva est un outil de design et de communication visuelle en ligne dont la mission est de permettre à tout le monde de concevoir et de publier selon ses envies [23].

Figure 53 : logo de Canva

Conclusion

Dans ce chapitre, nous avons présenté divers outils et technologies utilisés tout au long du projet. Les outils de conception, tels que les logiciels de modélisation, ont permis une planification visuelle des étapes. Les langages de programmation, frameworks et bibliothèques ont été utilisés pour le développement des fonctionnalités. Les outils de gestion de projet, comme les plateformes de suivi des tâches, ont assuré une gestion efficace. Ces outils et technologies ont été indispensables pour mener à bien chaque aspect du projet, en garantissant une conception solide, un développement robuste et une gestion optimale des ressources.

Chapitre 5

Implémentation

Introduction

Dans cette section finale, je suis ravi de vous présenter de manière exhaustive les résultats tangibles de mon travail à travers la visualisation de captures d'écran représentant les interfaces soigneusement conçues. Ces captures d'écran, obtenues à partir d'une variété de scénarios d'utilisation, vous permettront de saisir pleinement l'aspect visuel et l'ergonomie de mon projet, mettant en évidence l'effort considérable que j'ai déployé pour créer une expérience utilisateur immersive et intuitive. Chaque détail, du choix des couleurs à la disposition des éléments interactifs, a été minutieusement pensé pour garantir une navigation fluide et agréable. En outre, pour souligner la rigueur et la profondeur de ma démarche, j'inclurai également des extraits de code pertinents illustrant les étapes cruciales de la mise en œuvre de mon projet. Ces extraits, soigneusement sélectionnés parmi ma base de code complète, mettront en évidence les choix techniques judicieux que j'ai faits pour résoudre des défis spécifiques.

La structure du projet

1. Backend

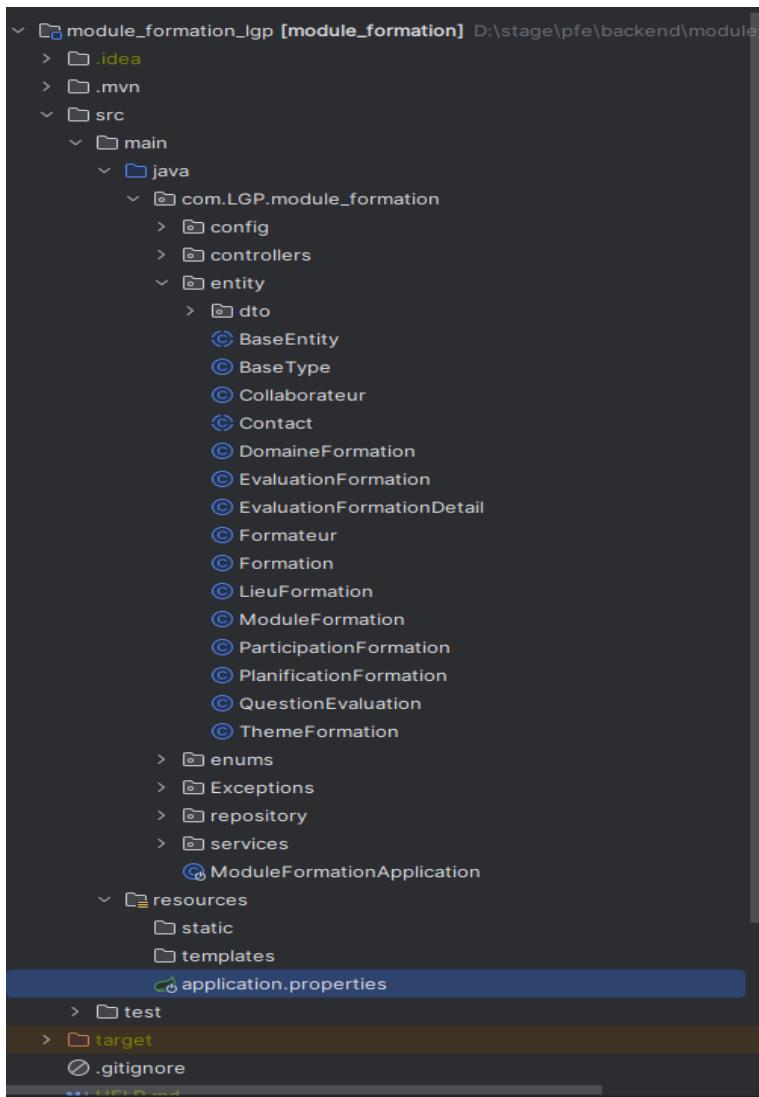


Figure 54 : Structure global du projet_backend

- .idea :

Contient les fichiers de configuration spécifiques à IntelliJ IDEA, l'EDI que vous utilisez. Ces fichiers sont générés automatiquement et gèrent des aspects comme les configurations de projet, les vues et les paramètres de l'EDI.

- .mvn :

Contient les fichiers de configuration Maven spécifiques au projet. Maven est un outil de gestion de projet et de compréhension utilisé principalement pour des projets Java. Ce dossier peut contenir les exécutables de Maven et des configurations spécifiques.

- src :

Répertoire principal contenant tout le code source et les ressources de votre application.

- config :

Contient les classes de configuration de Spring. Par exemple, les configurations de sécurité, les configurations des beans, etc.

- controllers :

Contient les classes contrôleurs de votre application Spring. Les contrôleurs sont responsables de gérer les requêtes HTTP et de retourner les réponses appropriées.

- entity :

Ce package contient les classes d'entités qui représentent les tables de la base de données.

- dto :

Contient les classes utilisées pour transférer des données entre différentes couches de l'application. Les DTO sont souvent utilisés pour encapsuler les données à transférer entre le client et le serveur.

- enums :

Contient les énumérations qui définissent des ensembles de constantes pour les types spécifiques de données utilisées dans l'application.

- exceptions :

Contient les classes d'exception personnalisées. Ces classes définissent les types d'erreurs spécifiques que votre application peut lancer et gérer.

- repository :

Contient les interfaces de repository Spring Data JPA. Les repositories sont utilisés pour interagir avec la base de données et effectuer des opérations CRUD (Create, Read, Update, Delete).

- services :

Contient les classes de service qui implémentent la logique métier de l'application. Ces services sont souvent appelés par les contrôleurs pour traiter les requêtes.

- application.properties :

Fichier de configuration principal pour votre application Spring. Il contient les configurations des bases de données, des paramètres de l'application, etc.

- test :

Contient le code source des tests. Les tests unitaires et d'intégration de votre application sont placés ici.

- **target** :

Répertoire généré par Maven où sont placés les fichiers compilés, les packages et d'autres résultats de la construction du projet.

- **.gitignore** :

Fichier utilisé par Git pour déterminer quels fichiers et répertoires doivent être ignorés dans les contrôles de version. Par exemple, il peut ignorer les répertoires target, les fichiers de configuration de l'EDI, etc.

2. Frontend

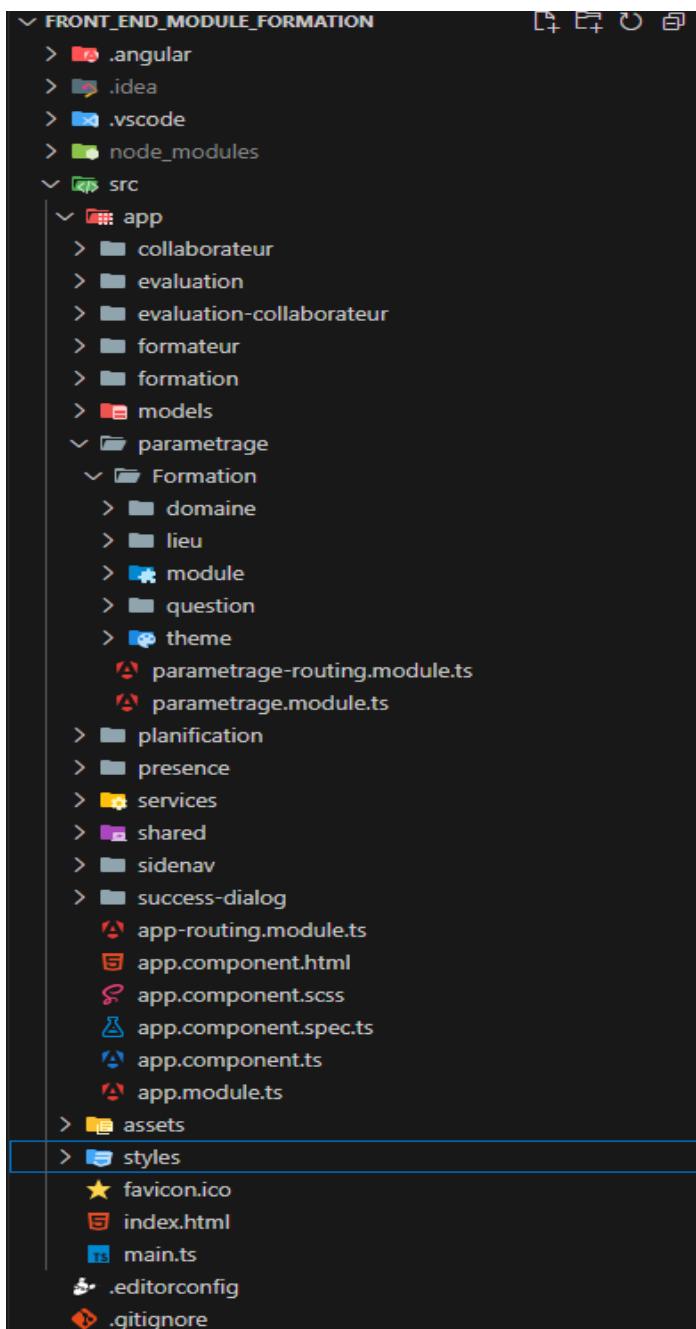


Figure 55 : Structure global du projet_frontend

- .angular :

- Contient des fichiers de configuration et de cache utilisés par Angular CLI pour optimiser les builds et les performances de l'application.

- .vscode :

- Contient les configurations spécifiques à Visual Studio Code, un autre EDI. Cela inclut les paramètres d'espace de travail, les extensions, et d'autres configurations.

- node_modules :

- Contient tous les modules et les dépendances Node.js installés via npm. Ce répertoire est géré automatiquement et ne devrait pas être modifié manuellement.

- app :

Contient le code principal de l'application Angular, y compris les composants, les services et les modules.

- collaborateur, evaluation, evaluation-collaborateur, formateur, formation :

Ces répertoires contiennent probablement des composants, des services et d'autres éléments spécifiques aux fonctionnalités respectives de votre application. Par exemple, collaborateur pourrait gérer les fonctionnalités liées aux collaborateurs, evaluation celles liées aux évaluations, etc.

- models :

Contient les modèles de données TypeScript utilisés pour définir les structures des données dans l'application.

- parametrage :

Contient les composants et les services relatifs aux paramètres de l'application.

- parametrage-routing.module.ts :

Fichier de routage pour le module de paramétrage. Définit les routes spécifiques à ce module.

- parametrage.module.ts :

Fichier de définition du module de paramétrage. Contient les déclarations, les imports, et les providers spécifiques à ce module.

- planification, presence :

Répertoires contenant des composants pour la planification et la gestion des présences respectivement.

- services :

Contient les services Angular qui fournissent des fonctionnalités comme la gestion des données, la logique métier, etc.

- shared :

Contient les composants, les directives et les pipes partagés entre plusieurs parties de l'application.

- side-nav :

Contient les composants relatifs à la navigation latérale de l'application.

- **success-dialog :**

Contient les composants pour les dialogues de succès, probablement utilisés pour afficher des messages de confirmation après certaines actions.

- **app-routing.module.ts :**

Fichier de routage principal de l'application Angular. Définit les routes de l'application.

- **app.component.html :**

Fichier de template HTML pour le composant principal de l'application.

- **app.component.scss :**

Fichier de styles SCSS pour le composant principal de l'application.

- **app.component.spec.ts :**

Fichier de test unitaire pour le composant principal de l'application.

- **app.component.ts :**

Fichier TypeScript pour le composant principal de l'application.

- **app.module.ts :**

Fichier de définition du module principal de l'application. Contient les déclarations, les imports, et les providers de l'application.

- **assets :**

Contient les ressources statiques comme les images, les fichiers JSON, les polices, etc.

- **styles :**

Contient les fichiers de styles globaux de l'application (CSS, SCSS).

- **favicon.ico :**

Icône de l'application visible dans l'onglet du navigateur.

- **index.html :**

Fichier HTML principal qui est la page de démarrage de l'application Angular. Ce fichier contient l'élément <app-root> où l'application Angular est montée.

- **main.ts :**

Fichier principal TypeScript qui bootstrap l'application Angular. C'est le point d'entrée de l'application.

Scénario d'utilisation

Authentification

Cette interface affiche la section d'authentification du système. Il est important de noter que la création de comptes est réservée au super administrateur. Les utilisateurs ordinaires n'ont pas le droit de créer des comptes

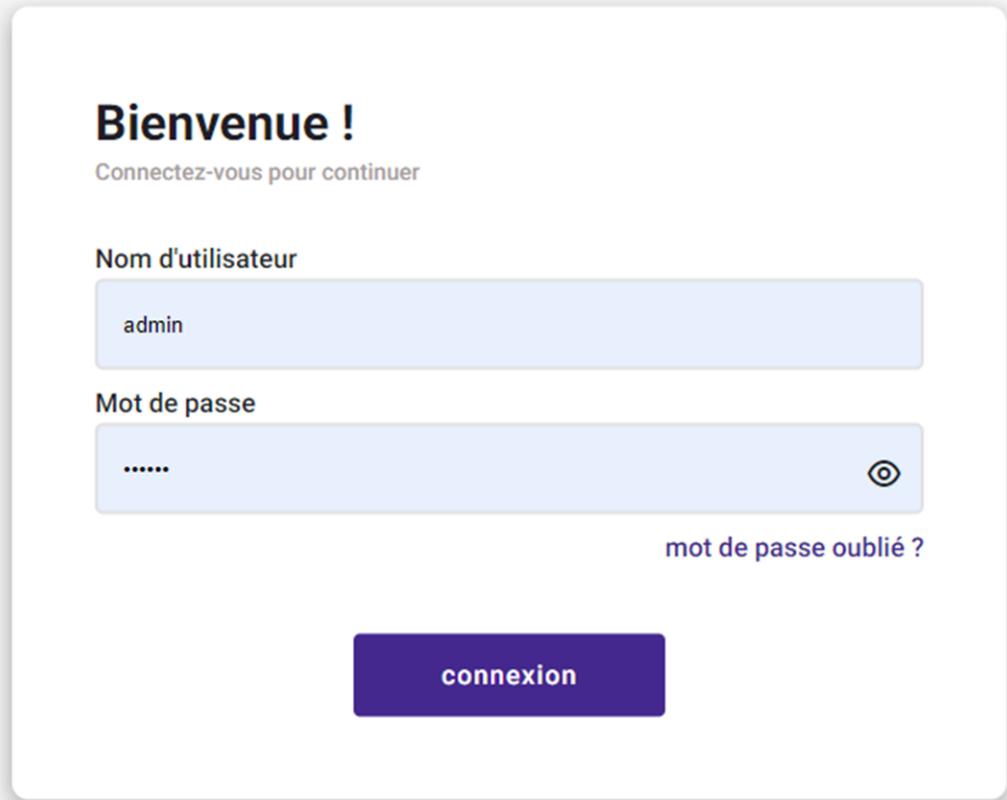


Figure 56 : Page du login

Gestion des comptes (partie concerne le super administrateur)

Voici un aperçu général de la partie paramétrage de notre projet LGP, en mettant l'accent sur la sécurité, notamment la gestion des utilisateurs. C'est là où on peut visualiser tous les comptes créés

	Nom d'utilisateur	E-mail	Rôles	Statut	Actions
admin	m.elbaqr@infosat.ma	Administrateur	Activé		

Figure 57 : Interface de gestion des utilisateurs

Lorsque vous cliquez sur le bouton ‘Ajouter’, vous accédez à cette interface pour insérer un nouvel utilisateur :

Ajouter un nouvelle utilisateur

Nom (*)
saisir ici le nom

E-mail (*)
admin

Mot de passe (*)
.....

Rôles (*)
Choisir des options

Annuler Valider

Figure 58 : Interface de création d'un utilisateur

Chaque utilisateur on lui attribue un rôle que l'on peut créer de manière indépendante voici les rôles qu'on a déjà créé :

Rôles

Nom	Description	Actions
ROLE_ADMIN	Administrateur	
ROLE_MANAGER	Manager	
ROLE_COLLABORATEUR	Collaborateur	
ROLE_ALL_DEPARTMENTS	Tous les départements	
ROLE_ALL_CATEGORIES	Toutes les catégories	

Par page 20 Pages : 1 -> 5 de 5

Précédent Suivant

Figure 59 : Interface de gestion des rôles

Pour ajouter un déjà il faut cliquer sur ajouter pour afficher cette interface suivante :

Figure 60 : Interface de création d'un rôle

Chaque rôle a des priviléges qu'il faut ajouter après cette étape :

Figure 61 : Interface de choix des priviléges pour un rôle

Figure 62 : Exemple des priviléges de sécurité

En prenant l'exemple de la sécurité, si l'on souhaite ajouter un autre administrateur capable de gérer les comptes ou de définir des rôles, il suffit de lui accorder les priviléges nécessaires. Ensuite, nous pouvons passer à l'étape des permissions, que nous pouvons configurer comme suit :

Figure 63 : Interface de choix des permissions pour un rôle

Voici l'affichage de la liste des collaborateurs. Chaque administrateur peut voir cette liste et, s'il en a les droits, il peut également la modifier.

Nom	Prénom	Mat. Pointage	Mat. Pale	Actions
Adiouane	Anas	p214de	e210	
Alami	Houssam	C587	eb145	
Ait alli	Hamid	C587	J589	
Rehal	Mohamed	T5478	J548e	
Goudi	Yassine	E454	J896e	

Figure 64 : Interface des collaborateurs de l'entreprise

Ensuite, en cliquant sur le bouton "Ajouter", vous pouvez créer un nouveau collaborateur comme suit :

Figure 65 : Interface d'ajout d'un collaborateur

Après avoir rempli le formulaire, on obtient la fenêtre suivante :

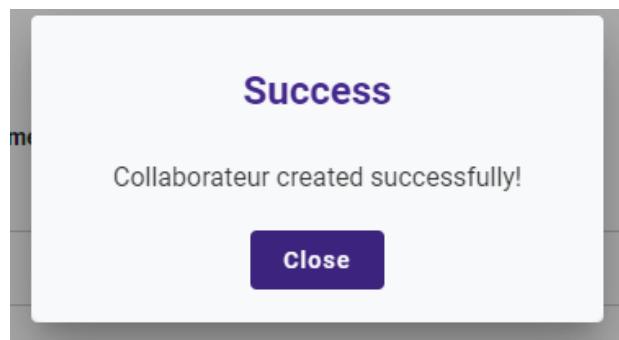


Figure 66 : Page affichant une fenêtre de confirmation de succès

En cliquant sur l'icône de modifier on reçoit cette interface avec les dernières informations saisies :

Figure 67 : Interface de modification d'un collaborateur

Nous débutons dès maintenant le module de formation, en commençant par la section de paramétrage. C'est ici que nous définissons les caractéristiques des formations. C'est le travail donc du responsable de formation qui commence.

Comme chaque formation a un thème, et chaque thème fait partie d'un module, et chaque module appartient à un domaine, la logique donc de création d'un thème commence par la création d'un domaine, suivie de celle d'un module, afin de pouvoir associer le thème créé à ce module. Voici la partie du domaine :

Domaine de Formation		Recherche	+ Ajouter
	Nom	Description	Actions
	Développement des Compétences en Leadership	Ce domaine vise à renforcer les capacités de leadership des collaborateurs, en leur fournissant les outils et les connaissances nécessaires pour diriger efficacement des équipes et gérer des projets.	
	Compétences Techniques et Numériques	Ce domaine couvre les compétences techniques et numériques nécessaires pour évoluer dans un environnement de travail moderne.	
	Développement Personnel et Bien-être	Ce domaine met l'accent sur le bien-être et le développement personnel des collaborateurs. Il vise à améliorer leur qualité de vie au travail et leur efficacité professionnelle en abordant des aspects tels que la gestion du stress.	

Figure 68 : Page de la liste des domaines de formation

Si on veut ajouter un on clique sur le bouton « Ajouter » :

Ajouter un domaine de formation

Nom:

Description:

Figure 69 : Interface d'Add d'un domaine

Il est également possible de modifier et de supprimer un domaine en cliquant sur l'icône correspondante. Lors de la modification, les données précédentes s'affichent. Ensuite, il faut créer un module en l'intégrant à un domaine existant. Voici la liste qui affiche tous les modules.

Nom	Description	Domaine	Actions
Leadership et Gestion d'Équipe	Ce module se concentre sur les compétences essentielles pour diriger et inspirer une équipe. Il couvre divers styles de leadership, la communication avec les membres de l'équipe, et les techniques de gestion des conflits.	Développement des Compétences en Leadership	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Gestion de Projets	Ce module vise à doter les collaborateurs des compétences nécessaires pour gérer des projets de manière efficace et efficiente.	Développement des Compétences en Leadership	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Informatique et Technologies de l'Information	Ce module vise à améliorer les compétences des collaborateurs en matière de technologies de l'information. Il couvre des sujets allant de la sécurité informatique à la programmation de base.	Compétences Techniques et Numériques	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Analyse de Données	Ce module vise à doter les collaborateurs des compétences nécessaires pour analyser et interpréter des données.	Compétences Techniques et Numériques	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Gestion du Stress	Ce module aide les collaborateurs à gérer le stress de manière efficace. Il inclut des techniques de relaxation, des stratégies pour équilibrer vie professionnelle et vie personnelle.	Développement Personnel et Bien-être	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Figure 70 : Page de la liste des modules de formation

Voici ce que l'on voit lors de l'ajout d'un module :

Figure 71 : Interface d'AJout d'un module

Figure 72 : Interface de modification d'un module

Cette capture montre le cas de la modification d'un module. De plus, la section "Domaine" affiche tous les domaines existants.

Nous passons maintenant à la partie de création du thème, que l'on peut lier à la formation souhaitée. La liste des thèmes s'affiche comme ceci :

The screenshot shows a list of training themes. Each theme has a name, a detailed description, a assigned module, and a set of actions (Edit, Delete, Preview) for each row.

Nom	Description	Module	Actions
Styles de Leadership	Exploration des différents styles de leadership (autocratique, démocratique, transformationnel, etc.) et de leur impact sur les équipes.	Leadership et Gestion d'Équipe	
Communication Efficace	Techniques pour améliorer la communication verbale et non-verbale avec les membres de l'équipe, incluant l'écoute active et la transmission claire des messages.	Leadership et Gestion d'Équipe	
Gestion des Conflits	Stratégies pour identifier, comprendre et résoudre les conflits au sein de l'équipe de manière constructive.	Leadership et Gestion d'Équipe	
Initiation à la Gestion de Projets	Introduction aux fondamentaux de la gestion de projets, incluant la définition des objectifs, la planification et le suivi des projets.	Gestion de Projets	
Outils de Gestion de Projets	Utilisation d'outils logiciels (comme Microsoft Project, Trello, etc.) pour planifier, suivre et gérer les projets efficacement.	Gestion de Projets	
Evaluation des Risques	Identification et évaluation des risques potentiels dans les projets, et mise en place de plans de contingence.	Gestion de Projets	
Sécurité Informatique	Principes de base de la sécurité informatique, y compris la protection des données, les menaces courantes et les bonnes pratiques pour sécuriser les systèmes d'information.	Informatique et Technologies de l'information	
Programmation de Base	Introduction à la programmation, couvrant les concepts fondamentaux et les langages de programmation courants comme Python ou JavaScript.	Informatique et Technologies de l'information	
Utilisation des Logiciels de Bureau	Formation sur les applications courantes de la suite bureautique (Word, Excel, PowerPoint) pour améliorer l'efficacité et la productivité au travail.	Informatique et Technologies de l'information	
Introduction à l'Analyse de Données	Concepts fondamentaux de l'analyse de données, incluant la collecte, le nettoyage et l'interprétation des données.	Analyse de Données	

Figure 73 : Page de la liste des thèmes de formation

La partie d'actions se fait de la même chose que le module.

Comme chaque formation se déroule dans un lieu qu'on définit également, cette étape fait aussi partie du paramétrage.

The screenshot shows a list of training locations. Each location has a name, a detailed description, and a set of actions (Edit, Delete, Preview) for each row.

Nom	Description	Actions
Université Ibn Zohr	L'Université Ibn Zohr est l'une des principales institutions académiques de la région, offrant des salles de conférence et des amphithéâtres modernes. C'est un lieu approprié pour les formations académiques et les ateliers de recherche.	
Hôtel Sofitel Agadir Royal Bay Resort	Cet hôtel de luxe dispose de salles de réunion et de conférence bien équipées, ainsi que d'espaces de détente pour les pauses et les activités de team-building. Son emplacement en bord de mer offre un cadre agréable pour les participants.	
INFOSAT	INFOSAT est une entreprise technologique située dans l'immeuble Les Amicales, au cœur d'Agadir. Elle se spécialise dans les solutions de télécommunication, l'informatique et les services numériques.	
Centre de Formation et de Perfectionnement de l'OFPPT Agadir	Ce centre de formation professionnelle offre des salles et des équipements pour diverses formations techniques et professionnelles. Il est particulièrement adapté pour des formations spécialisées et techniques.	
Chambre de Commerce, d'Industrie et de Services d'Agadir (CCIS)	La CCIS d'Agadir propose des salles de formation équipées pour accueillir des sessions de formation en entreprise, des ateliers et des séminaires dans un cadre professionnel.	

Figure 74 : Page de la liste des lieux de formation

Sur la page suivante. On définit également toutes les questions que l'on peut utiliser, avec la possibilité d'en ajouter ou d'en modifier une.



**Logiciel
Gestion
Personnel**

Questions d'Évaluations

Recherche

+ Ajouter

Question	Actions
Qu'est-ce qui différencie les styles de leadership autoritaire et démocratique ?	
Comment identifier le style de leadership le plus adapté à votre équipe ?	
Pouvez-vous donner un exemple où un changement de style de leadership a amélioré les performances d'une équipe ?	
Quels sont les principaux obstacles à une communication efficace dans une équipe ?	
Comment la communication non-verbale influence-t-elle les interactions professionnelles ?	
Quelles techniques utilisez-vous pour assurer une écoute active lors des réunions ?	
Quelle est la première étape pour résoudre un conflit entre collègues ?	
Pouvez-vous décrire une situation où un conflit bien géré a conduit à une amélioration du travail d'équipe ?	
Quels sont les avantages de la médiation dans la gestion des conflits ?	
Quelles sont les étapes clés de la planification d'un projet ?	

Questions par page 1 – 10 of 28 |< < > >|

Actions

Figure 75 : Page de la liste des questions de formation

Pour chaque formation, il y a un formateur. C'est pourquoi nous créons des formateurs, qui peuvent être soit un collaborateur de l'entreprise, soit une personne externe. L'affichage des formateurs déjà se fait de la même manière :



**Logiciel
Gestion
Personnel**

Formateurs

Recherche

+ Ajouter

Nom	Prénom	CIN	Actions
Cherrad	Abdelaziz	JT47887	
El Ghilimi	Tariq	JC98745	

Formateurs par page 1 – 2 of 2 |< < > >|

Actions

Figure 76 : Page de la liste des formateurs de formation

Pour insérer un nouveau formateur il faut juste cliquer sur le bouton « Ajouter ». On reçoit donc ce formulaire à remplir :



Nom (*)

Prénom (*)

Date de Naissance

CIN

Sexe

 Homme Femme

Civilité

Origine

À propos du Formateur

Figure 77 : Interface d'Ajout d'un formateur



Nom (*)

Prénom (*)

Date de Naissance

CIN

Sexe

 Homme Femme

Civilité

Origine

À propos du Formateur

Figure 78 : Interface de modification d'un formateur

La dernière capture montre le cas de la modification d'un formateur. Dans la section "À propos du formateur", vous pouvez ajouter ses informations professionnelles et autres détails. Nous passons maintenant à la partie dédiée à la création de la formation. Voici la liste des formations qu'on a :

Titre	Nombre d'Heures	Thème	Actions
Maîtriser les Styles de Leadership pour une Équipe Performante	2	Styles de Leadership	
Améliorer la Communication Interne pour une Meilleure Collaboration	3	Communication Efficace	
Techniques de Gestion des Conflits pour un Environnement de Travail Harmonieux	2	Gestion des Conflits	
Les Fondamentaux de la Gestion de Projets pour Débutants	2	Initiation à la Gestion de Projets	
Évaluation et Gestion des Risques dans les Projets	3	Évaluation des Risques	

Figure 79 : Page de la liste des formations

Pour ajouter une formation, procédez comme suit :

Figure 80 : Interface d'Addition d'une formation

Comme chaque formation peut avoir un support tel qu'un PDF ou un document Word, la zone de contenu dédiée est prévue à cet effet.

Pour voir le détail d'une formation, cliquez sur l'icône appropriée pour accéder à cette interface.

The screenshot shows a detailed view of a training session. At the top, there are fields for 'Titre' (Title) containing 'Maîtriser les Styles de Leadership pour une Équipe Performante', 'Nombre d'heures' (Number of hours), and 'Thème de formation' (Training theme) containing 'Styles de Leadership'. Below these, under 'Questions de formation' (Formation questions), is a list of ten questions related to leadership styles and their impact on team performance. A context menu is open at the top right, with 'Export as PDF' highlighted.

Titre: Maîtriser les Styles de Leadership pour une Équipe Performante

Nombre d'heures:

Thème de formation: Styles de Leadership

Questions de formation:

- Qu'est-ce qui différencie les styles de leadership autocratique et démocratique ?
- Comment identifier le style de leadership le plus adapté à votre équipe ?
- Pouvez-vous donner un exemple où un changement de style de leadership a amélioré les performances d'une équipe ?
- Le formateur était-il clair et précis dans ses explications ?
- Pensez-vous que cette formation aura un impact positif sur votre performance au travail ?
- Quelle est votre satisfaction globale vis-à-vis de cette formation ?
- Y a-t-il des aspects que vous n'avez pas aimés ou que vous aimeriez voir améliorés ?

Figure 81 : Page des détails d'une formation

Vous pouvez également télécharger ces informations au format PDF.

Dans une étape importante, nous planifions ces formations en définissant la date de début, la date de fin, et d'autres caractéristiques. Voici tout d'abord les planifications déjà définies. Notez que le statut de réalisation est lié aux dates.

Si vous voulez ajouter une planification, cliquez toujours sur le bouton "Ajouter".

The screenshot shows a list of scheduled trainings. The header is 'Planification de Formation'. On the left, there's a sidebar with navigation links: Tableau de bord, Histoire, Accueil, Paramétrage, Collaborateurs, Formateurs, Planification (which is selected and highlighted in pink). Under 'Planification', there are sub-links: Planification Simple, Planification Périodique, Jour de Congé, and Planification de Formation. The main area displays a table of training sessions with columns: Titre (Title), Date Début (Start Date), Date Fin (End Date), Statut Réalisation (Status), Valide (Valid), Formation (Training), and Actions (with icons for edit, delete, and export). The table contains four entries.

Titre	Date Début	Date Fin	Statut Réalisation	Valide	Formation	Actions
Ateliers Pratiques sur les Styles de Leadership	Jun 4, 2024	Jun 5, 2024	REALISE	Oui	Maîtriser les Styles de Leadership pour une Équipe Performante	
Formation Interactive sur la Communication Efficace	Jun 12, 2024	Jun 13, 2024	NOUVEAU	Oui	Améliorer la Communication Interne pour une Meilleure Collaboration	
Techniques de Résolution de Conflits en Milieu Professionnel	Jun 1, 2024	Jun 2, 2024	REALISE	Oui	Techniques de Gestion des Conflits pour un Environnement de Travail Harmonieux	
Atelier sur l'Évaluation et la Gestion des Risques en Projets	May 30, 2024	May 31, 2024	REALISE	Oui	Évaluation et Gestion des Risques dans les Projets	

Formations par page: 5 | 1 - 4 of 4 | < < > >

Figure 82 : Page de la liste des planifications de formation

Ajouter une planification de formation

Titre de la Planification

Date début

Date fin

Lieu de Formation

Formateur

Formation

Validé

Oui Non

Annuler Valider

Figure 83 : Interface d'Ajout d'une planification

Pour ce qui suit, si vous souhaitez modifier une planification, voici comment procéder : cliquez sur l'icône de modification dans la liste des planifications pour afficher les informations précédentes. Puis cliquer sur valider, si les modifications sont effectuées correctement, une boîte de dialogue de succès s'affiche.

Modifier une planification de formation

Titre de la Planification

Date début

Date fin

Lieu de Formation

Formateur

Formation

Validé

Oui Non

Annuler Valider

Figure 84 : Interface de modification d'une planification

Pour le détail des planifications voici un exemple :

The screenshot shows a form for a planning detail. The fields and their values are:

- Titre:** Ateliers Pratiques sur les Styles de Leadership
- Date de Naissance:** 04/06/2024
- Date de Naissance:** 05/06/2024
- Formateur:** Cherrad Abdelaziz
- Formation:** Maîtriser les Styles de Leadership pour une Équipe Performante
- Lieu de formation:** Hôtel Sofitel Agadir Royal Bay Resort
- Statut de réalisation:** REALISE
- Validation de la planification:** true
- Les participants:**
 - Adiouane Anas
 - Alami Houssam
 - Ait ali Hamid

Figure 85 : Page des détails d'une planification

Nous passons maintenant à la section "Présence" pour ajouter les participants à une planification. L'interface suivante montre comment enregistrer la présence : cochez les participants présents. Par défaut, tous les participants ont un statut "absent", sinon, "non défini" si la date de la planification n'est pas encore arrivée.

The screenshot shows the "Liste de présence" (Presence List) interface. The table displays the following data:

	Nom	Prénom	Statut de présence	Planification	Formation	Actions
<input type="checkbox"/>	Adiouane	Anas	PRESENT	Ateliers Pratiques sur les Styles de Leadership	Maîtriser les Styles de Leadership pour une Équipe Performante	
<input type="checkbox"/>	Alami	Houssam	ABSENT	Ateliers Pratiques sur les Styles de Leadership	Maîtriser les Styles de Leadership pour une Équipe Performante	
<input type="checkbox"/>	Ait ali	Hamid	ABSENT	Ateliers Pratiques sur les Styles de Leadership	Maîtriser les Styles de Leadership pour une Équipe Performante	
<input type="checkbox"/>	Mouzar	Leila	NON_DEFINI	Formation Interactive sur la Communication Efficace	Améliorer la Communication Interne pour une Meilleure Collaboration	
<input type="checkbox"/>	Goudi	Yassine	NON_DEFINI	Formation Interactive sur la Communication Efficace	Améliorer la Communication Interne pour une Meilleure Collaboration	

Figure 86 : Interface de la liste de présence

En cliquant sur le bouton "Ajouter", vous pouvez insérer des collaborateurs pour qu'ils deviennent participants à la planification souhaitée.

Figure 87 : Page de choix de planification pour ajouter des participants

C'est ici que vous sélectionnez les participants.

Figure 88 : Page de choix des participants

Une fois validée, chaque collaborateur intégré à la planification verra dans la partie d'évaluation une évaluation concernant cette formation. Voici ce qu'un collaborateur, par exemple, verra.

Figure 89 : Page de la liste des évaluations pour un collaborateur

Si le statut de réalisation est "réalisé" et que le collaborateur était présent, il pourra accéder à l'évaluation. Sinon, il recevra un message, pour l'évaluation se passe comme suivant :

Figure 90 : Page répertoriant toutes les questions de la formation sélectionnée

Supposons qu'il a cliqué sur la deuxième évaluation voici ce qu'il reçoit en retour :

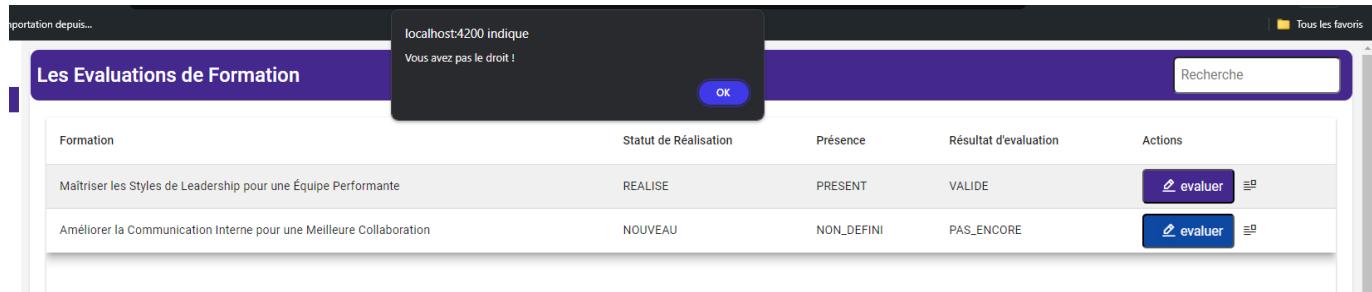


Figure 91 : Résultat lors de la sélection d'une évaluation non autorisée

Après, toutes les réponses sont envoyées au responsable de la formation ainsi qu'au super administrateur.

Figure 92 : Page de la liste des évaluations complétées et envoyées au responsable de formation

En cliquant sur "Voir les réponses", vous pouvez consulter toutes les réponses

Question	Response
Qu'est-ce qui différencie les styles de leadership autocratique et démocratique ?	Le style autocratique implique une prise de décision centralisée avec peu de participation des employés, tandis que le style démocratique encourage la participation et la collaboration des membres de l'équipe dans la prise de décision.
Le formateur était-il clair et précis dans ses explications ?	Oui, le formateur était clair et précis dans ses explications.
Pouvez-vous donner un exemple où un changement de style de leadership a amélioré les performances d'une équipe ?	Un manager est passé d'un style autocratique à un style démocratique, ce qui a encouragé la participation active des membres de l'équipe, augmenté leur motivation et amélioré la productivité.
Pensez-vous que cette formation aura un impact positif sur votre performance au travail ?	Oui, je pense que cette formation aura un impact positif sur ma performance au travail.
Y a-t-il des aspects que vous n'avez pas aimés ou que vous aimeriez voir améliorés ?	J'aimerais que le contenu soit plus interactif et pratique.
Quelle est votre satisfaction globale vis-à-vis de cette formation ?	Je suis globalement satisfait(e) de cette formation.

Figure 93 : Page de la liste des réponses d'un collaborateur

L'évaluation démarre initialement avec le statut "pas encore". Ensuite, le responsable peut la valider ou non en cliquant sur l'icône d'accord ou de refus. Voici un exemple :

Nom	Prénom	formation	Etat d'évaluation	Evaluation
Adiouane	Anas	Maîtriser les Styles de Leadership pour une Équipe Performante	VALIDE	voir les réponses

Figure 94 : Exemple d'une évaluation validée

Enfin, On mentionne que le site web propose deux modes, sombre et clair. Voici un exemple du thème sombre.

The screenshot shows a dark-themed user interface for managing employee profiles. On the left, a sidebar lists various modules: Tableau de bord, Histoire, Accueil, Paramétrage, Collaborateurs (selected), Formateurs, Planification, Formations, Présence, Formation, Pointage, and Evaluations des Formations. The main area displays a user profile placeholder and a form for entering personal information: Nom (*), Prénom (*), Date de Naissance (format jj/mm/aaaa), CIN, Sexe (radio buttons for Homme and Femme), Civilité (dropdown menu), Matricule de Pointage, Matricule de Paie, and Origine. At the bottom right are 'Annuler' and 'Valider' buttons.

Figure 95 : Exemple du thème sombre

Conclusion

En conclusion, ce chapitre a exposé la structure globale du projet, couvrant à la fois le backend et le frontend. Nous avons également présenté les interfaces graphiques qui illustrent le fonctionnement de notre application. Ces informations clés permettent de visualiser l'architecture du projet dans son ensemble, soulignant la cohérence entre les différentes composantes. En combinant une structure backend solide avec des interfaces utilisateur conviviales, notre application offre une expérience utilisateur fluide et intuitive.

Conclusion générale et perspectives

Pour la mise en œuvre de ce projet, nous avons tout d'abord procédé à une étude conceptuelle approfondie afin d'identifier les différents modules nécessaires à cette application, ainsi qu'à une évaluation des outils et technologies pouvant être adaptés à sa réalisation. Ensuite, nous avons réalisé une analyse et une conception du projet en utilisant le formalisme UML. Nous avons élaboré plusieurs diagrammes pour mieux structurer le projet, ce qui a grandement facilité sa mise en œuvre. Enfin, nous avons implémenté les différents modules de l'application. Les résultats obtenus lors de cette dernière phase répondent aux exigences et aux besoins du cahier des charges.

Durant ce stage, nous avons eu l'occasion de nous familiariser avec le monde professionnel et de mettre en pratique nos connaissances, ce qui nous a permis d'acquérir de nouvelles compétences et de maîtriser de nouveaux outils. Travailler sur ce projet nous a offert une vision détaillée de la gestion et de la réalisation de projets.

Grâce à ces différentes étapes, nous avons pu développer une application qui, nous l'espérons, répond aux attentes du cahier des charges. Cette expérience nous a également permis d'approfondir nos compétences en matière de gestion de projet et d'élargir notre expertise dans l'utilisation d'outils technologiques spécifiques. Nous sommes convaincus que les connaissances acquises lors de ce projet seront d'une grande valeur pour nos futures entreprises et projets. Nous sommes fiers du travail accompli et des résultats obtenus grâce à notre engagement et à notre dévouement tout au long de ce processus.

Dans le cadre de mon PFE, l'entreprise a identifié la nécessité de développer un nouveau module dédié à la gestion des formations. Ce module doit traiter à la fois des formations internes et externes. Cependant, en raison de la complexité liée aux formations externes, notamment à cause des régulations étatiques, nous avons décidé de commencer par le volet interne. Pour le volet externe, nous avons opté pour une approche simplifiée où le formateur est une personne individuelle, rémunérée pour dispenser la formation, plutôt qu'une organisation. De plus, il est crucial d'ajouter un filtrage par tous les champs existants dans chaque liste et de tester toutes les fonctionnalités. Ensuite, nous aborderons le cas plus complexe en commençant par la conception jusqu'à la réalisation.

Webographie

- [1] https://clickup.com/?utm_source=youtube&utm_medium=cpc&utm_campaign=yt_cpc_ar_nc_pro_trial_desktop_troas_lp_x_all-departments_x_all-custom-combinations&utm_content=ar_cs-combo_rva_all-industries_all-features_all-use-cases_custom-combination-remarketing-all-visitors&utm_creative=694136182278_AD-4061-team-collaboration-v2_rva_30-60s_mograph_trytoday&qclid=CjOKCQjwjLGyBhCYARIsAPqTz18I5mwKEFpk5kvCfRx5BDxHsid5HNMPrys769rkwfivcr_R98mAf0aAlFqEALw_wcB, consulté le 01/06/2024
- [2] <https://spring.io/>, consulté le 28/02/2024
- [3] <https://hibernate.org/>, consulté le 04/03/2024
- [4] <https://jwt.io/>, consulté le 01/03/2024
- [5] <https://projectlombok.org/>, consulté le 04/03/2024
- [6] <https://swagger.io/>, consulté le 05/03/2024
- [7] <https://www.jetbrains.com/idea/>, consulté le 26/02/2024
- [8] <https://www.postman.com/>, consulté le 06/03/2024
- [9] <https://code.visualstudio.com/>, consulté le 26/02/2024
- [10] <https://angular.dev/>, consulté le 12/03/2024
- [11] https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/HTML_basics, consulté le 15/05/2024
- [12] https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/CSS_basics, consulté le 15/05/2024
- [13] <https://www.typescriptlang.org/>, consulté le 15/05/2024
- [14] <https://material.angular.io/>, consulté le 22/03/2024
- [15] <https://rxjs.dev/>, consulté le 13/03/2024
- [16] <https://www.npmjs.com/package/jspdf>, consulté le 15/03/2024
- [17] <https://iconify.design/>, consulté le 10/03/2024
- [18] <https://www.visual-paradigm.com/>, consulté le 05/03/2024
- [19] <https://www.modelio.org/index.htm>, consulté le 06/03/2024
- [20] <https://git-scm.com/>, consulté le 27/02/2024
- [21] <https://github.com/>, consulté le 27/02/2024
- [22] <https://www.mysql.com/>, consulté le 02/03/2024
- [23] https://www.canva.com/fr_fr/, consulté le 18/04/2024
- [24] <https://stackoverflow.com/>, consulté le 10/03/2024
- [25] <https://www.w3schools.com/>, consulté le 02/03/2024
- [26] <https://www.youtube.com/@mohamedYoussfi>, consulté le 29/02/2024
- [27] <https://www.youtube.com/watch?v=BBh3wMVJaso&list=PL3EibBwUnE37aZ937p2L2VoozXUKcsI76>, consulté le 15/03/2024
- [28] <https://www.bing.com/chat?q=Microsoft+Copilot&FORM=hpcodx>, consulté le 18/03/2024

- [29] <https://images.google.com/>, consulté le 26/05/2024