

CURRICULUM VITAE

MALALANIRINA Felaniaina Nicolette

Né le 09 Octobre 2002 à Tambohobe Fianarantsoa

Email : felaniainanicolette@gmail.com

N° Téléphone :0343510587

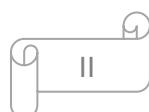


Formations et Diplômes

- **2023-2024** : Troisième Année de Licence Professionnelle à l’ École de Management d’Innovation Technologique.
- **2022-2023** : Deuxième Année de Licence Professionnelle à l’ École de Management d’Innovation Technologique.
- **2021-2022** : Première Année de Licence Professionnelle à l’ École de Management d’Innovation Technologique.
- **2020-2021** : Lycéenne en Terminale D au Lycée Saint Joseph de Cluny Fianarantsoa à l’obtention du diplôme Baccalauréat avec Mention Assez Bien.

Expérience

- **Juillet 2024-Octobre 2024** :Stagiaire au sein de la Société Miezaka Fianarantsoa durée trois mois pour thème « Conception et Réalisation d’une Application de Gestion d’Absence des Employés de la Société Miezaka » développé avec NodeJs et VueJS.
- **2023-2024** : Projet d’étude EMIT Crédation d’Application de Gestion de CyberCafe(Econnect) en CodeIgniter.Création d’Application de Gestion de Stock de Pharmacie en Spring Boot.Création d’ Application mobile.
- **Juin-Août 2023** : Stagiaire au sein de la Service Régionale des Entreprises Fianarantsoa durée deux mois avec le thème « Conception et Réalisation d’une Application de Gestion de Transfert Trésorerie du Service » technologies utilisés Php.



Connaissance en Informatique

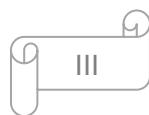
- **Système d'exploitation :** Windows, Linux
- **Méthodes et langages de Conception :** MERISE, UML
- **Outils de conception :** Microsoft Access, Lucidchart.
- **Outils de développement :** Visual Studio Code, NetBeans.
- **Programmation Web :** HTML, JavaScript
- **SGBD :** MS Access, MySQL, PostgreSQL.

Connaissances Linguistiques

Langues	Aptitude à :			
	Comprendre	Lire	Écrire et Rédiger	Parler et Communiquer oralement
Malagasy	Très Bien	Très Bien	Très Bien	Très Bien
Français	Bien	Bien	Bien	Assez Bien
Anglais	Assez Bien	Assez Bien	Assez Bien	Assez Bien

Divers

- Sport: Basket-bal.
- Sens du travail d'équipe, sens de la communication.
- Lire des livres, regarder des documentaires et films



AVANT PROPOS

L'objectif principal de la formation universitaire est de préparer les étudiants au monde professionnel. Les métiers qu'ils exercent nécessitent des compétences techniques, de la motivation, du dynamisme, ainsi qu'un haut degré de professionnalisme. Il est donc essentiel que les futurs cadres soient à la hauteur de ces exigences. Dans un contexte où les technologies évoluent rapidement. Grâce aux opportunités offertes par ces avancées, l'informatisation des systèmes de travail devient perspective de plus en plus réalisable.

Chaque année, les étudiants de troisième année de l'École de Management d'Innovation Technologique sont amenés à réaliser un stage de trois mois. Ce stage constitue une période temporaire d'apprentissage et de formation pratique, permettant aux étudiants de mettre en pratique leurs connaissances et compétences dans différents secteurs professionnels. En intégrant une entreprise ou une organisation, les stagiaires acquièrent une expérience concrète, confrontant ainsi leur savoir théorique aux réalités du terrain.

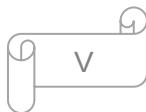
À l'issue de cette période, les étudiants rédigent un mémoire ou un rapport de stage, en lien avec le thème qu'ils ont choisi au cours de leur stage. Ce travail constitue le fruit de trois mois d'efforts, d'analyse et de création, synthétisant leur apprentissage et leurs réalisations au sein de l'entreprise.

Pour les étudiants inscrits dans la mention informatique, le choix du thème s'oriente généralement vers la conception et la réalisation d'une application web ou mobile, intégrant la méthode UML (Unified Modeling Language) pour la modélisation, ou vers la mise en place et la sécurisation d'un réseau informatique. Ces projets sont conçus pour démontrer non seulement les compétences techniques acquises au cours de leurs études, mais aussi leur capacité à résoudre des problèmes concrets, à proposer des solutions innovantes et à contribuer de manière significative à l'entreprise qui les a accueillis.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué à la réussite de ce travail. Leur soutien, leur conseil et leur encouragement ont été essentiels tout au long de ce projet.

- Le Docteur HDR HAJALALAINA Aimée Richard, président de l'Université de Fianarantsoa .
- Le Docteur RAKOTONIRAINY Hasina Lalaina,, Directeur de l'École de Management et d'Innovation Technologique pour m'avoir chaleureusement accueilli au sein de son établissement.
- Madame RABEZANAHARY Hobiniaina, Responsable de la Mention Informatique cycle de licence, mérite également mes remerciements pour son accompagnement tout au long de mes études.
- Monsieur HOUSSENALY Amine qui a généreusement accepté ma demande de stage et m'a offert l'opportunité de mettre en pratique mes connaissances dans un cadre professionnel.
- Monsieur RAZAFINDRAKOTO Todisoa Brunel, responsable du département informatique au sein de la Société MIEZAKA, pour son encadrement précieux et son soutien tout au long de ce projet.
- Monsieur MAMINIRINA Honoré Philibert mon encadreur professionnel, pour son assistance inestimable. Son accompagnement, ses critiques constructives, et son expertise ont enrichi mon expérience et m'ont permis de développer des compétences pratiques essentielles.
- A tous mes enseignants durant mes études.
- A tous les personnels de la Société Miezaka..
- Enfin, je voudrais exprimer toute ma gratitude à mes parents pour leur soutien inconditionnel et leurs encouragements constants, ainsi qu'à mes camarades du parcours DA2I niveau L3 et mes amis, pour leur aide précieuse et leur solidarité tout au long de ce parcours. Leur présence à mes côtés a été une source de motivation et de réconfort.
- À vous tous, je vous adresse mes plus sincères remerciements.



LISTES DES FIGURES

Figure 1-1 Offre de formation à l'EMIT	17
Figure 4-1 Évolution de l'UML.....	34
Figure 4-2 Représentation des acteurs.....	38
Figure 4-3 Représentation de cas d'utilisation.....	39
Figure 4-5 Formalisme du diagramme de séquence.....	42
Figure 4-6 Formalisme d'une classe.....	43
Figure 4-7 Représentation des nœuds d'activités.....	45
Figure 6-1 Représentation graphique du diagramme de cas d'utilisation global.....	56
Figure 6-2 Diagramme de cas d'utilisation s'authentifier.....	57
Figure 6-3 Représentation du diagramme de séquence de l'authentification.....	58
Figure 6-4 Diagramme d'activité de l'authentification	59
Figure 6-5 Diagramme de cas d'utilisation d'ajout d'un élément.....	61
Figure 6-6 Diagramme de séquence d'ajout d'un élément(employeur, département,poste)	62
Figure 6-7 Diagramme d'activité d'ajout d'élément(employé,département,poste).....	63
Figure 6-8 Diagramme de cas d'utilisation d'ajout demande d'absence.....	65
Figure 6-9 Diagramme de séquence d'ajout de demande d'absence.....	66
Figure 6-10 Diagramme d'activité d'ajout de demande d'absence.....	68
Figure 6-11 Diagramme de cas d'utilisation de modification.....	69
Figure 6-12 Diagramme de séquence de modification.....	70
Figure 6-13 Diagramme d'activité de modification	71
Figure 6-14 Diagramme de cas d'utilisation de suppression.....	72
Figure 6-15 Diagramme de séquence de suppression.....	73
Figure 6-16 Diagramme d'activité de suppression.....	74
Figure 6-17 Diagramme de classe en globale.....	75
Figure 8-1 Modèle de l'entité employé	86
Figure 8-2 Route /login pour connecter à l'application.....	87
Figure 8-3 Un serveur Express pour une API RESTful.....	88
Figure 9-1 Page d'authentification.....	89
Figure 9-2 Page d'accueil	90
Figure 9-3 Page à propos de l'employé.....	91
Figure 9-4 Page a propos de demande d'absence.....	91
Figure 9-5 Formulaire d'ajout demande absence	92
Figure 9-6 Le modèle de fiche d'absence	93

LISTES DES TABLEAUX

Tableau 1-1 Présentation des mentions et parcours de l'EMIT	15
Tableau 2-1 Présentation de la fiche d'identification de la Société Miezaka	26
Tableau 3-1Planning du déroulement du stage.....	30
Tableau 4-1 Les types de Visibilité.....	42
Tableau 4-2 Les différents types de relations dans les diagrammes de classes	43
Tableau 6-1 Comparaison entre Ludichart et Visual Paradigm.....	51
Tableau 6-2 Dictionnaire de données	54
Tableau 6-3 Description du scénario de l'authentification	58
Tableau 6-4 Description du scenario d'ajout d'un élément(employé,département,poste)	62
Tableau 6-4 Description de scénario d'ajout de demande d'absence	66
Tableau 6-5 Description du scénario de modification	70
Tableau 6-6 Description de scénario de diagramme de séquence de suppression.....	73
Tableau 7-1Avantages et inconvénients des quatre SGBD	77
Tableau 7-2Comparaison de Nodejs Express et Django	79
Tableau 7-3Comparaison de Vuejs avec d'autres frameworks	81

LISTE DES ABRÉVIATIONS

AES : Administration Économique et Social

API : Application Programming Interface

CIGSI : Conception, Intégration et Gestion des Systèmes d'Information

CM : Communication Multimédia

CNMP : Communication Numérique et Management de Projet

CUFP : Centre Universitaire de Formation Professionnaliste

DAII : Développement d'Application Internet et Intranet

EMIT : Ecole de Management et d'Innovation Technologique

IGTI : Ingénierie Géospatiale et Technologies de l'Information

JS : JavaScript

JSON: JavaScript Object Notation

JWT : JSON Web Token (Jeton Web JSON)

LMD : Licence Master Doctorat

MD : Management Décisionnel

MEDA : Management d'Entreprises et de Développement des Affaires

MVC : Model-View-Controller

NPM : Node Package Manager

ORM : Object-Relational Mapping

REST : Representational State Transfer

RH : Ressources Humaines

RPM : Relations Publiques et Multimédia

SDIA : Sciences de Données et Intelligence Artificielle

SGBD : Système de Gestion de Base de Données

SIGD : Système d'Information, Géomatique et Décision



SQL : Structured Query Language

UML : Unified Modeling Language

UF : Université de Fianarantsoa



GLOSSAIRE

Algorithme : Ensemble d'instructions ou de règles permettant de résoudre un problème ou d'exécuter une tâche.

Architecture Client-Serveur : Modèle de réseau où un serveur central fournit des ressources ou des services à des clients.

Base de données : Ensemble organisé de données stockées et structurées, accessible et manipulable à l'aide d'un logiciel (ex. : MySQL, PostgreSQL).

API (Application Programming Interface) : Interface de programmation qui permet à différentes applications de communiquer entre elles. Les API sont souvent utilisées pour intégrer des fonctionnalités spécifiques d'un système dans un autre.

HTTP (HyperText Transfer Protocol) : Protocole de communication utilisé pour le transfert de données sur le Web.

MVC : Modèle architectural utilisé pour structurer les applications logicielles. Il sépare l'application en trois parties interconnectées : le modèle (données), la vue (interface utilisateur), et le contrôleur (logique de traitement).

ORM (Object-Relational Mapping) : Technique de programmation permettant de convertir des données entre des systèmes incompatibles en utilisant la programmation orientée objet. Dans une application, un ORM est utilisé pour interagir avec une base de données relationnelle à l'aide de classes et d'objets.

UML (Unified Modeling Language) : Langage de modélisation standardisé utilisé pour spécifier, visualiser, et documenter les systèmes logiciels. UML est souvent utilisé pour représenter les structures de données et les processus dans une application.

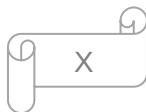


TABLE DES MATIÈRES

CURRICULUM VITAE	2
AVANT PROPOS.....	4
LISTES DES FIGURES.....	6
LISTES DES TABLEAUX.....	7
LISTE DES ABRÉVIATIONS.....	8
GLOSSAIRE.....	10
TABLE DES MATIÈRES.....	11
Introduction générale	17
Partie I : PRÉSENTATION GÉNÉRALE.....	19
Chapitre 1: Présentation de l'Ecole de Management et d'Innovation Technologique	20
1.1. Historique	20
1.2. Missions	21
1.3. Formations existantes.....	22
 1.3.1. Cycle Licence.....	22
 1.3.1.1 Mention Management, Parcours Administration Economique et Sociale	22
 1.3.1.2.Mention Informatique, Parcours Développement d'Application Internet Intranet et Parcours Conception, Intégration et Gestion des Systèmes d'Information	23
 1.3.1.3. Mention Relations Publiques et Multimédia, Parcours Communication Multimédia et Relations Publiques et Communication Organisationnelle	23
 1.3.2. Cycle Master.....	23
 1.3.2.1 Mention Management, Parcours Management Décisionnel.....	24
 1.3.2.2. Mention Management, Parcours Management d'Entreprises et de Développement des Affaires (MEDA)	24
 1.3.2. 3. Mention Informatique, Parcours Système d'Information, Géomatique et Décision et Parcours Modélisation et Ingénierie Informatique	25
 1.3.2.4. Mention Informatique, Parcours Ingénierie Géospatiale et Technologies de l'Information (IGTI).....	25
 1.3.2.5. Mention Informatique, Parcours Sciences de Données et Intelligence Artificielle (SDIA)	

.....	25
1.3.2.6. Mention Informatique, Parcours Modélisation et Ingénierie Informatique	26
1.3.2.7. Mention Relation publique et multimédia, Parcours Relations Publiques et Communication Organisationnelle (RPCO).	26
1.3.2.8. Mention Relation publique et multimédia, Parcours Communication Multimédia (CM)	26
1.3.2.9. Mention Relation publique et multimédia, Parcours Communication Numérique et Management de Projet (CNMP).....	27
1.4. Partenaires de l'EMIT.....	27
1.5. Organigramme de l'EMIT.....	28
Chapitre 2 : Présentation de la Société Miezaka.....	29
2.1. Historique de la Société MIEZAKA.....	29
2.2. Développement et Croissance	29
2.2.1 Partenariats Stratégiques avec Divers Fournisseurs.....	30
2.2.2. Collaboration avec des Fournisseurs Internationaux.....	31
2.4.3. Accords Commerciaux et Facilités d'Importation.....	31
2.3. Fiche d'identification.....	32
2.4. Implantation d'une usine d'efflichage.....	33
Chapitre 3 : Présentation du projet.....	34
3.1. Origine du projet.....	34
3.2. Problématique.....	34
3.3. Conduite de projet.....	34
3.4. Cahier des charges	35
3.4.1. Définitions	35
3.4.2. Objectifs du projet.....	35
3.4.3. Planification du projet.....	36
PARTIE II : ANALYSE ET CONCEPTION	37
Chapitre 4: Méthodes et Notations Utilisés.....	38
4.1. L'UML.....	38
4.1.1. Historique.....	38

4.1.2. Définitions et objectifs.....	39
4.1.3. UML face à la programmation.....	39
4.1.4. Points forts et points faibles de l'UML	39
4.1.4.1. Points forts.....	39
4.1.4.2. Points faibles.....	40
4.1.5. Diagramme de cas d'utilisation.....	42
4.1.5.1. Acteur	42
4.1.5.2.Cas d'utilisation.....	43
4.1.5.3. Relation entre cas d'utilisation.....	43
4.1.5.4. Formalisme d'un diagramme de cas d'utilisation.....	44
4.1.6. Diagrammes des séquences	45
4.1.6.1. Ligne de vie.....	45
4.1.6.2. Message	45
4.1.6.3. Formalisme d'un diagramme de séquence.....	45
4.1.8. Le diagramme d'activité.....	48
4.1.8.1. Composition d'un diagramme d'activités	49
Chapitre 5 Analyse du Projet.....	50
5.1. Analyse de besoins.....	50
5.2. Analyse de faisabilité.....	51
5.2.1. Faisabilité Technique.....	51
5.2.2. Faisabilité Opérationnelle.....	51
5.3. Analyse de l'existant.....	51
5.4. Critique de l'existant.....	52
5.5. Propositions des solutions.....	52
5.6. Solutions retenues.....	53
5.7. Analyse de l'architecture.....	53
Chapitre 6 Conception du Projet.....	55
6.1. Présentation de l'outil de conception.....	55



6.1.1. Lucidchart.....	55
6.1.2. Visual Paradigm.....	55
6.2. Choix d'outils de conception.....	56
6.3. Règles de gestion	57
6.4. Dictionnaire de données.....	58
6.5. Modelisation.....	60
6.5.1. Schéma de diagramme de cas d'utilisation globale	60
6.5.2. Étude du cas d'utilisation : « S'authentifier ».....	61
6.5.2.1. Description du cas d'utilisation s'authentifier.....	61
6.5.2.2. Diagramme de séquence de l'authentification.....	62
6.5.2.3. Diagramme d'activité d'authentification	64
6.5.3. Étude du cas d'utilisation « Ajout des éléments(employer,département,poste)»	64
6.5.3.1. Description du cas d'utilisation	64
6.5.3.2. Diagramme de séquence ajout d'un élément(employer,département,poste).....	66
.....	66
6.5.3.3. Diagramme d'activité d'un ajout élément(employé,département,poste)	67
6.5.4. Étude du cas d'utilisation : « Ajout Demande absence d'un employé »	68
6.5.4.1.. Description du cas d'utilisation	68
6.5.4.2. Diagramme de séquence de ajout absence d'un employé.....	69
6.5.4.3. Diagramme de d'activité de ajout absence d'un employé	72
6.5.5. Étude du cas d'utilisation : Modification	72
6.5.5.1. Description du cas d'utilisation	72
6.5.5.2. Diagramme de séquence de modification	73
6.5.5.2. Diagramme d'activité de modification	75
6.5.6. Étude de cas d'utilisation : Suppression	76
6.5.6.1. Description de cas d'utilisation	76
6.5.6.2. Diagramme de séquence de suppression	77
6.5.5.3 Diagramme d'activité de suppression.....	78

6.5.7. Diagramme de classe en globale	79
Partie III : Réalisation du projet	81
Chapitre 7 Spécification des outils de réalisation.....	82
7.1. Le système de gestion de base de données.....	82
7.1.1. Définition d'une base de données	82
7.1.2. Utilité d'une base de données	82
7.2. Caractéristiques des SGBD.....	82
7.3. Points forts et faibles des SGBD.....	83
7.4. Présentation de Node.js et Express.js.....	84
7.4.1. Introduction	84
7.4.2. Avantages de Node.js et Express.js.....	85
7.4.3. Inconvénients de Node.js et Express.js	85
7.4.4. Comparaison avec Django	85
7.5. Présentation du framework VueJs.....	86
7.5.1. Introduction	86
7.5.2. Avantages de Vue.js.....	87
7.5.3. Inconvénients de Vue.js	88
7.5.4.Comparaison avec d'autres frameworks	89
7.6. Présentation des outils de développement	89
7.6.1. Visual Studio Code (VSCode)	89
7.6.2. Docker.....	90
7.6.3. Postman.....	90
7.6.4. Git.....	90
Chapitre 8 : Mise en œuvre et implémentation	91
8.1. Architecture Logicielle	91
8.1.1. Frontend (Vue.js).....	91
8.1.2. Backend (Node.js avec Express).....	91
8.1.3. Base de Données (MySQL).....	92

8.2. Architecture Matérielle	92
8.3. Extrait de code	93
Chapitre 9 : Présentation de l'application développée	96
9.1. Fonctionnalités de l'application	96
9.2. Présentation de quelques pages de l'application	96
9.2.1. Page d'authentification	96
9.2.2. Page d'authentification	97
9.3.2. Page à propos de l'employé	98
9.2.4. Page de demande absence	98
9.2.5. Demande absence	99
CONCLUSION GENERALE	101
BIBLIOGRAPHIE	10
WEBOGRAPHIE	11
ANNEXE	12
ABSTRACT	13

Introduction générale

La Société Miezaka est une entreprise dynamique et innovante, spécialisée dans la vente de balles de friperie provenant de l'Union Européenne. Elle joue un rôle clé dans le marché local, en offrant des produits de qualité à ses clients. Pour assurer le bon fonctionnement de ses activités, l'entreprise emploie de nombreuses personnes, chacune apportant des compétences et des talents uniques. Grâce à la diversité de son personnel et à un environnement de travail collaboratif, Miezaka parvient à répondre efficacement aux demandes de ses clients, tout en favorisant le développement professionnel de ses employés.

Dans cette entreprise, les employés ont la possibilité de s'absenter et de prendre des congés selon leurs besoins et préférences. Cela leur permet de mieux équilibrer leur vie professionnelle et personnelle. Cependant, avec un grand nombre de demandes d'absence à traiter chaque mois, le service des ressources humaines (RH) peut parfois rencontrer des difficultés. Ces défis peuvent entraîner des erreurs dans le calcul des jours d'absence et des soldes de congés. De telles erreurs peuvent générer des malentendus entre les employés et le service RH, ce qui nuit à la confiance dans le système de gestion des absences.

Il est clair que l'informatique est une grande révolution qui a transformé notre vie moderne. Les logiciels et les nouvelles méthodes de développement ont introduit des technologies que nous utilisons chaque jour. Grâce à ces avancées, nous pouvons accomplir de nombreuses tâches plus facilement et plus rapidement, transformant ainsi notre quotidien et nos façons de travailler. Pour cela, est-il possible de créer une application qui puisse gérer efficacement les absences des employés, en calculant automatiquement les jours d'absence et en mettant à jour les soldes de congés en temps réel, tout en répondant aux besoins spécifiques de l'entreprise en matière de gestion des absences ? Ce mémoire présente le travail accompli dans le cadre de ce projet de fin d'études. L'objectif est de concevoir et de développer une application web pour améliorer la gestion des absences des employés de la Société Miezaka.

Pendant le stage, la première étape a été d'étudier ce qui existait déjà et de créer un cahier des charges pour définir les besoins de l'application de gestion des absences. Ensuite, nous avons cherché la meilleure solution qui respecte ces besoins et commencé à mettre en place l'application. Après avoir collecté des informations utiles, nous avons imaginé comment l'application fonctionnerait, son organisation, ainsi que les outils et technologies à utiliser. Ce

mémoire explique toutes les étapes que nous avons suivies pour concevoir et réaliser cette application de gestion des absences pendant le stage.

Au fil de ce mémoire, nous allons étudier en trois parties comment optimiser la gestion des absences au sein de la Société Miezaka. La première partie sera consacrée à une présentation générale de l'EMIT et de la société. La deuxième partie portera sur l'analyse et la conception du projet, où nous détaillerons les besoins, la méthode choisie et le langage utilisé pour modéliser la solution. Enfin, la troisième partie se focalisera sur la réalisation du projet, en présentant les outils employés, la mise en œuvre de l'application, et son fonctionnement final.

Partie I : PRÉSENTATION GÉNÉRALE

Chapitre 1: Présentation de l'Ecole de Management et d'Innovation Technologique

1.1. Historique

L'École de Management et d'Innovation Technologique (EMIT) est une école universitaire publique pluridisciplinaire, rattachée à l'Université de Fianarantsoa. La grande maturité au niveau de l'enseignement et la compétence des étudiants sortant de l'établissement ont permis aux dirigeants sous l'approbation du Ministère la conversion du Centre en École au sein de l'Université de Fianarantsoa par le Décret N°2016-1394 du 15 Novembre 2016. L'EMIT prépare d'une part le diplôme de Master en deux mentions en trois parcours et d'autre part le diplôme de Licence en trois mentions en cinq parcours.

Auparavant, elle a été connue sous le nom du Centre Universitaire de Formation Professionnaliste (CUFP), créé par le Décret N°2005-205 du 26 Avril 2005 et dispensait le diplôme de Licence professionnelle en Administration ainsi qu'en Informatique. Mais avant cela, elle a été connue également sous le nom du Centre de Formation Continue (CFC), créé par l'Arrêté Rectoral N°99-23/UF/R du 10 Mars 1999 qui formait de diplôme de Technicien Supérieur.

L'EMIT a été sélectionnée « Meilleur Etablissement » pendant le Salon de la Recherche organisé par l'Organisation Internationale du Travail les 5 et 6 Juillet 2017. Depuis l'année universitaire 2013-2014, l'école est basculée totalement vers le système Licence, Master et Doctorat (LMD). Toutes les offres de formation dispensée à l'EMIT sont habilitées par le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique. L'école propose huit parcours répartis en trois mentions, représentés sur le tableau 1.1 :

Cycle	Mention Management	Mention Informatique	Mention Relations Publiques et Multimédia
Licence	Administration Economique et Sociale	Développement d'Application Internet/Intranet	Communication Multimédia
		Conception, Intégration et Gestion des Systèmes d'Information	Relations Publiques et Communication Organisationnelle
Master	Management Décisionnel	Système d'Information, Géomatique et Décision(SIGD)	Relations Publiques et Communication Organisationnelle (RPCO).
		Ingénierie Géospatiale et Technologies de l'Information (IGTI)	Communication Multimédia (CM)
	Management d'Entreprises et de Développement des Affaires (MEDA)	Sciences de Données et Intelligence Artificielle (SDIA)	Communication Numérique et Management de Projet (CNMP)
		Modélisation et Ingénierie Informatique	

Tableau 1-1 Présentation des mentions et parcours de l'EMIT

1.2. Missions

L'école a pour mission, d'abord de dispenser de formations initiales et continues en informatique, en administration et en relations publiques et multimédia. Ensuite, elle offre des services connexes à l'informatique. Puis, elle forme des techniciens opérationnels immédiatement au sein des entreprises. Enfin, elle assure le perfectionnement professionnel des étudiants, des demandeurs d'emplois, des employés et des cadres d'entreprises.

1.3. Formations existantes

L'école présente actuellement deux cycles : Licence et Master. Chaque cycle possède plusieurs parcours assurés par un chef de mention. Cette figure offre une vue d'ensemble sur les formations dispensées au sein de l'EMIT.

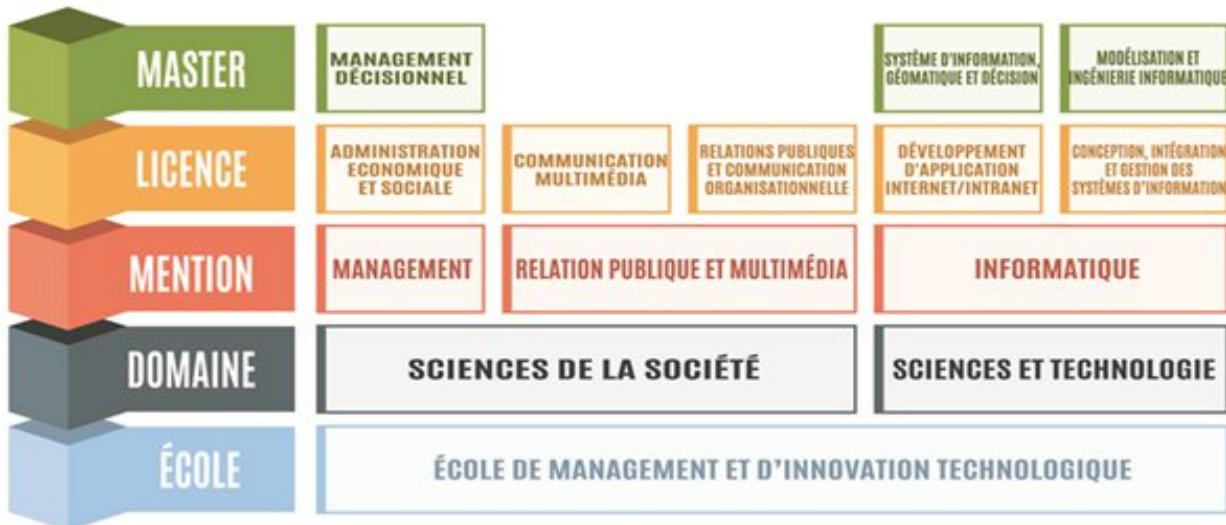


Figure 1-1 Offre de formation à l'EMIT

1.3.1. Cycle Licence

L'école possède trois (03) mentions pour cinq (05) parcours en cycle Licence. Les étudiants toutes mentions confondues doivent effectuer un voyage d'études d'insertion en entreprise pour les premières années de Licence (L1), un stage de réalisation en entreprise avec un rapport de stage soutenu pour les étudiants en deuxième année de Licence (L2) et un stage de fin d'études suivi de la soutenance d'un mémoire pour les étudiants en troisième année de Licence (L3). La formation dure trois années universitaires et à la fin de la formation, les étudiants obtiennent des diplômes de Licence.

1.3.1.1 Mention Management, Parcours Administration Economique et Sociale

La condition d'accès en première année de Licence se fait par voie de concours pour les titulaires d'un Baccalauréat général de toutes séries (A, C, D) ou d'un Baccalauréat technique G1 et G2. A l'issue de la formation, les étudiants ont les compétences de :

- Assister le Directeur Général, le Directeur des Ressources Humaines et le Directeur Administratif et Financier ;
- Gérer les Ressources Humaines ;
- Gérer une entreprise ou un projet.

1.3.1.2. Mention Informatique, Parcours Développement d'Application Internet Intranet et Parcours Conception, Intégration et Gestion des Systèmes d'Information

La condition d'accès en première année de Licence se fait par voie de concours pour les élèves titulaires du diplôme de Baccalauréat Général Scientifique (Série C et série D), Baccalauréat Technique Professionnelle et Baccalauréat Technique Technologique (Filière Industrielle, Maintenance Automobile, Ouvrage Bois, Ouvrage Métallique, Génie Civile). A l'issue de la formation, les étudiants sont compétents en :

- Administration des bases de données ;
- Administration des réseaux et systèmes informatiques ;
- Développement d'application client/serveur.

1.3.1.3. Mention Relations Publiques et Multimédia, Parcours Communication Multimédia et Relations Publiques et Communication Organisationnelle

La condition d'accès en première année de Licence se fait par voie de concours pour les titulaires d'un Baccalauréat général de toutes séries (A, C, D) ou d'un Baccalauréat technique G1 et G2. A l'issue de la formation, les étudiants ont les compétences de :

- Rédiger un article dans un journal ;
- Occupier un poste d'un technicien de presse ;
- Travailler dans la revue de presse.

1.3.2. Cycle Master

L'Ecole de Management et d'Innovation Technologique (EMIT) possède trois (03) mentions pour neuf (09) parcours en Master Professionnel qu'en Master Recherche. La durée de la formation est quatre semestres c'est-à-dire deux années universitaires.

1.3.2.1 Mention Management, Parcours Management Décisionnel

La condition d'accès en première année de Master (M1) en S7 se fait par sélection de dossier après l'obtention du diplôme de Licence en Administration Economique et Sociale, en Gestion ou en Economie.

La mention Management propose un parcours Management Décisionnel ayant pour objectif de former et d'équiper les apprenants à la maîtrise des outils d'aide à la décision en matière de management et de leur donner les compétences requises dans ce domaine. Comme le management a besoin de se conformer en permanence aux diverses nouvelles exigences du marché, l'enseignement doit alors toujours viser pour mettre à jour les connaissances de l'apprenant par la formulation de programmes de cours qui tiennent compte de ces nouveautés. Ainsi, les objectifs principaux peuvent se résumer à former des acteurs de haut niveau en management décisionnel, de préparer des cadres capables de gérer et de créer un projet de développement économique régional et national.

Les sortants peuvent travailler dans les secteurs privés et publics des différentes régions de Madagascar en tant que chefs de conduite de travaux d'enquêtes communautaires, concepteurs de projets, chefs de services ou directeurs d'entreprises. Dans ce cas, les étudiants sortants sont capables de créer une petite entreprise, de monter un projet de développement rural et de gérer un grand projet.

1.3.2.2. Mention Management, Parcours Management d'Entreprises et de Développement des Affaires (MEDA)

Le master en management d'entreprises et développement des affaires prépare les étudiants à exceller dans la création et la constitution d'entreprises. Les diplômés acquièrent des compétences clés pour élaborer des plans d'affaires solides, identifier les opportunités de marché et gérer efficacement les ressources. Les étudiants sont bien positionnés pour fonder leur propre entreprise en tant qu'entrepreneurs prospères ou occuper des rôles clés dans des start-ups. Les débouchés incluent la création d'entreprises innovantes, la gestion de projets entrepreneuriaux et la consultation stratégique pour les nouvelles initiatives commerciales. Ce master offre ainsi une voie dynamique vers l'entreprenariat et le succès dans le domaine des affaires.

1.3.2. 3. Mention Informatique, Parcours Système d'Information, Géomatique et Décision et Parcours Modélisation et Ingénierie Informatique

La condition d'accès en première année de Master (M1) en S7 se fait par sélection de dossier après l'obtention du diplôme de Licence en Informatique et en Mathématique et Informatique pour les Sciences Sociales (MISS). Le Recrutement en S9 se fait par validation des crédits acquis.

Le parcours Système d'Information, Géomatique et Décision a pour objectif de donner un panorama des recherches actuelles et émergentes en termes de système d'aide à la décision. En effet, les systèmes informatiques et la géomatique sont en plein essor, par les grilles de calcul et les multiples appareils mobiles intégrant des systèmes informatiques de plus en plus performants et complexes. Ces systèmes informatiques intégrant un parallélisme massif ou /et une mobilité des composants représente un défi pour le génie logiciel qui doit fournir de nouvelles méthodes et des outils de production de logiciel pour la description de l'architecture de ces systèmes complexes et pour leur validation et/ou certification. De plus, l'effervescence des techniques en géomatique qui fournissent des données spatiales et temporelles dans différents domaines représentent des moyens efficaces pour prendre les bonnes décisions.

1.3.2.4. Mention Informatique, Parcours Ingénierie Géospatiale et Technologies de l'Information (IGTI)

Ce parcours met particulièrement l'accent sur le développement de solutions en géomatique en exploitant une expertise avancée dans la conception de systèmes d'information géographique, la télédétection et l'analyse spatiale. Les étudiants qui suivent ce parcours sont formés pour être capables de proposer des solutions informatiques adaptées aux besoins des utilisateurs en géomatique. Cela inclut la capacité à concevoir des systèmes d'information géographique efficaces, à interpréter et à analyser des données spatiales, ainsi qu'à appliquer des techniques de télédétection pour obtenir des informations précises et utiles. En somme, ce parcours vise à former des professionnels capables de répondre aux défis et aux besoins spécifiques liés à la géomatique dans divers domaines d'application.

1.3.2.5. Mention Informatique, Parcours Sciences de Données et Intelligence Artificielle (SDIA)

Ce parcours vise à former des experts en informatique de haut niveau, spécialisés en intelligence artificielle et science des données. Les étudiants auront l'occasion de développer des compétences clés dans des domaines essentiels tels que la conception et le développement de

logiciels pour l'analyse automatique de grandes masses de données. Cela inclut également la programmation pour le Big Data et l'IA, ainsi que la modélisation et l'automatisation du raisonnement sur les données. En outre, les étudiants apprendront à concevoir des architectures d'entreprise efficaces, à appliquer des méthodes statistiques avancées et à maîtriser les techniques de gestion d'entreprise. L'utilisation de technologies pour l'exploitation de données réelles sera également un aspect important de la formation, permettant aux étudiants de mettre en pratique leurs compétences dans des environnements professionnels réels.

1.3.2.6. Mention Informatique, Parcours Modélisation et Ingénierie Informatique

Ce parcours vise à développer des compétences avancées en modélisation de systèmes complexes, en architecture logicielle et en analyse des besoins. Il met particulièrement l'accent sur l'adoption des méthodologies agiles et sur les bonnes pratiques de développement logiciel. Les étudiants apprendront à concevoir des modèles pour des systèmes complexes, à créer des architectures logicielles robustes et à analyser de manière approfondie les besoins des utilisateurs. En intégrant les méthodologies agiles, les étudiants seront également formés à travailler de manière collaborative, à s'adapter rapidement aux changements et à fournir des solutions logicielles de haute qualité dans des délais stricts.

1.3.2.7. Mention Relation publique et multimédia, Parcours Relations Publiques et Communication Organisationnelle (RPCO).

Un parcours axé sur le développement de compétences stratégiques et opérationnelles. Les étudiants bénéficient d'une formation complète, couvrant la gestion de l'image de marque, la planification d'événements, la communication interne et externe, afin de les préparer à exceller dans le monde dynamique et exigeant des relations publiques.

1.3.2.8. Mention Relation publique et multimédia, Parcours Communication Multimédia (CM)

Le secteur de la communication multimédia est à la fois diversifié et concurrentiel, reposant sur la recherche et le traitement de l'information avant sa publication, que ce soit sur des sites web, à la radio ou à la télévision. Il est donc crucial de posséder une solide formation dans ces domaines pour réussir dans cette industrie en constante évolution.

Le double diplôme proposé entre le parcours Cultures et Métiers du Web (CMW) de l'Université Gustave Eiffel et le parcours Communication Multimédias (CM) de l'Université de Fianarantsoa offre une opportunité unique. Il permet d'obtenir simultanément un diplôme de master en CMW et un diplôme de master en CM, enrichis de deux suppléments de diplôme. Cette combinaison offre aux étudiants une formation complète et diversifiée, couvrant à la fois les aspects techniques et créatifs de la communication multimédia, tout en les préparant aux défis du marché du travail dans ce secteur en pleine expansion.

1.3.2.9. Mention Relation publique et multimédia, Parcours Communication Numérique et Management de Projet (CNMP)

La formation en Communication Numérique et Management de Projet de l'EMIT a pour objectif de former des professionnels polyvalents capables de gérer des projets numériques, de diriger des équipes et de comprendre les enjeux des médias numériques. Ce programme est développé en collaboration avec l'Université Bordeaux Montaigne en France, ce qui garantit une expertise internationale et une approche complète des défis liés à la communication et au management dans le contexte numérique actuel. Les étudiants acquerront des compétences essentielles pour réussir dans ce domaine en constante évolution, en combinant des connaissances techniques et managériales.

1.4. Partenaires de l'EMIT

L'EMIT travaille en collaboration avec plusieurs laboratoires de recherche, d'entreprises et d'autres écoles et universités. Parmi les organisations partenaires, citons à titre d'exemple les laboratoires de recherche tels que le LIMAD, LIMOS, IRD, CNRE, SPAD, Espace Dev, LRI et l'UPR-Green à travers le CIRAD.

Pour ce qui est des écoles et des universités partenaires, il y a entre autres : l'EDMI, l'Université de Toulouse Paul Sabatier, Université de Montpellier 2, l'ENI, GOUVSOMU, IOGA, l'Université de Clermont Auvergne, ESMIA, Université de Mahajanga, ISSTM et l'Université de Fianarantsoa.

L'Ecole est également en partenariat avec plusieurs entreprises, notamment dans le cadre des stages à effectuer à travers chaque parcours, telles que les entreprises Etech consulting, Orange, Lazan'i Betsileo, STAR, Alliance Française de Fianarantsoa, TELMA, BFV-SG, Bank of

Africa (BOA), BNI Madagascar, Nelli Studio, YMAGOO, PREMIYA, JIRAMA, les assurances NY HAVANA, MAMA et ARO.

Des organismes gouvernementaux sont également partenaires de l'EMIT : la Région Haute Matsiatra, le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, le Ministère de l'Education Nationale, le Ministère des Travaux Publics, le Ministère des Finances et du Budget, le Ministère du Tourisme, le Ministère des Transports et de la Météorologie, le Ministère de la Poste, de la Télécommunication et des Technologies Numériques, la Banque Centre de Madagascar, le Foibe Taosaritanin'i Madagasikara (FTM), l'INSTAT.

1.5. Organigramme de l'EMIT

La structure hiérarchique au sein de l'EMIT se compose d'un conseil (scientifique ou d'établissement), une direction, un collège des enseignants, des chefs de mentions et des services présents au sein de l'école. Cette structure est représentée de manière générale sur la figure 1-2.

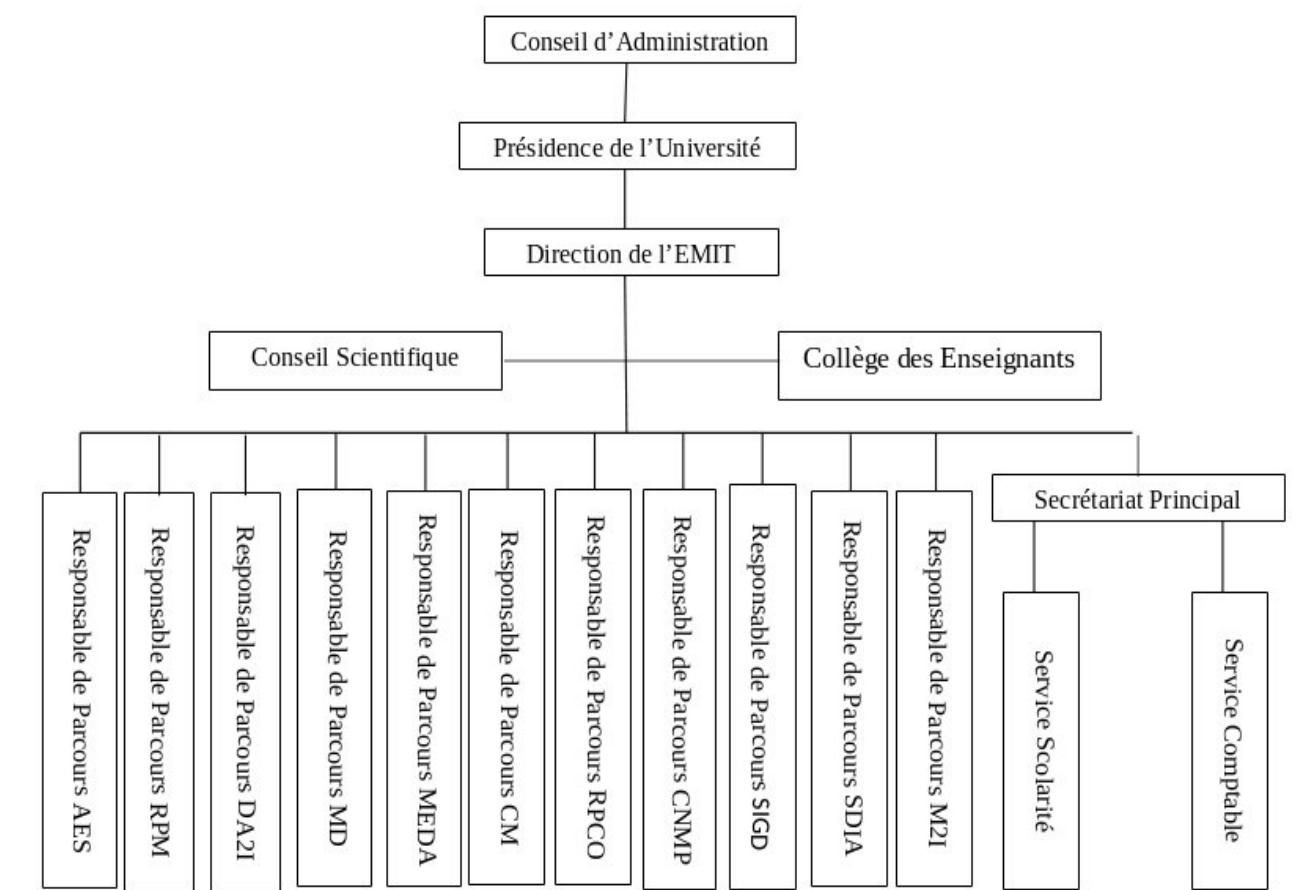


Figure 1-2 Organigramme de l'EMIT

Chapitre 2 : Présentation de la Société Miezaka

2.1. Historique de la Société MIEZAKA

L'EURL MIEZAKA est une entreprise créée il y a maintenant une trentaine d'années par Monsieur HOUSSENALY Amine. Au départ, elle était dénommée « TRANOMBAROTRA MIEZAKA » avec seulement 08 salariés. Le capital de départ investi était de 2 000 000 Ariary, aujourd'hui celui-ci s'élève à 100 000 000 Ar. Le siège de l'entreprise est situé à Fianarantsoa ville du centre de Madagascar à 400 km de la capitale Antananarivo. L'activité de cette entreprise est l'importation et la vente de balles de friperie provenant de l'Union européenne. Elle travaille aujourd'hui avec une dizaine de fournisseurs. Madagascar étant un pays sous développé, Monsieur HOUSSENALY a créé cette entreprise en lui donnant une dimension sociale, avec pour vocation d'aider la population et de répondre à ses besoins. Société MIEZAKA est une entreprise qui a beaucoup de succès comme nous pouvons le voir avec son chiffre d'affaires qui s'élève à plus de dix milliards d'Ariary (le SMIC malgache est de 200 000 Ar). A l'échelle du pays, ce chiffre d'affaires est très élevé et permet de placer l'entreprise dans le top 100 des grandes entreprises malgaches.

Il est vrai que Société MIEZAKA est un leader national dans l'importation et la vente de balles de friperie. Implantée à Fianarantsoa depuis maintenant une trentaine d'années, l'entreprise a su développer son activité et atteindre un haut niveau de compétence. Comme le dit l'expression du XVème siècle « on n'a rien sans rien », Monsieur HOUSSENALY, ferveur du travail, a développé d'années en années son activité pour le construire en véritable empire de l'importation et la vente de balles de friperie. C'est pourquoi, en observant le fonctionnement de cette entreprise, on peut se rendre compte que son succès n'est pas anodin et qu'il repose sur différents points qui ont été réfléchis et exploités par la direction.

2.2. Développement et Croissance

Tout d'abord, comme dit précédemment, la société s'est développée dans un contexte politique et économique instable faisant baisser le pouvoir d'achat de la population. Ces crises ont été des points négatifs pour le pays mais un point positif pour Société MIEZAKA qui a su profiter de l'environnement qui l'entourait pour évoluer. En effet, se lançant dans une activité qui a

pour but de répondre aux besoins de la population malgache, ces crises cycliques ont permis un essor considérable de ses affaires.

En effet, durant les années 2000 jusque 2008, pour la société, la vente de balles de friperie a littéralement explosé. Cela s'explique par le fait que la population avait besoin de se vêtir mais à faible coût au vu de leur manque énorme de moyen mais aussi qu'elle devait générer une source de revenu pour survivre et subsister à la crise. La friperie était donc la solution alternative la plus adaptée, permettant aux personnes de s'habiller à moindre prix, avoir un large choix d'habillement et y ajouter en plus de cela la qualité. La vente d'articles de friperie étant un travail facile ne consistant qu'à acheter et vendre, c'était une aubaine pour la population de se lancer dans cette activité en installant leur étale les jours de marchés. Aujourd'hui la demande est de plus en plus forte et il est plus que courant de voir à chaque coin de rue de chaque quartier de la ville de Fianarantsoa un ou plusieurs vendeurs de friperie sur les bords de route.

De plus, depuis 2009 le succès de Société MIEZAKA ne cesse d'accroître. L'histoire est telle que, Le Relais France étant le 1er fournisseur de balle de friperie de Société MIEZAKA, celui-ci a proposé à Monsieur Amine HOUSSENALY de créer un Le Relais Madagasikara à Fianarantsoa (ville où est implantée Société MIEZAKA). Monsieur HOUSSENALY a accepté la proposition, l'infrastructure où est situé actuellement le site a été trouvé et appartient aujourd'hui à Société MIEZAKA.

2.2.1 Partenariats Stratégiques avec Divers Fournisseurs

Crée en 2008, Le Relais Madagasikara s'inscrit dans une volonté de développer des activités socio-économiques dans le but de créer des emplois durables pour des personnes en grande pauvreté. Ses activités se déclinent dans plusieurs domaines, par exemple : atelier de construction automobile « SOATAO », complexe hôtelier éco-touristique « MORINGA ». Elle emploie en totalité environ 400 personnes. Ici, l'activité qui nous intéresse est le centre de tri « TARATRA ». Effectivement, le Relais Madagascar est un prestataire de tri pour Société MIEZAKA. Ce secteur de l'entreprise génère une centaine d'emploi. Parmi les six activités présentent au Relais Madagascar, le tri textile se positionne en tant que vache à lait pour le Relais. La particularité de la relation entre ces deux entreprises est que le contrat établi entre Le Relais Madagascar et Société MIEZAKA en ce qui concerne le tri textile est un contrat exclusif. Les deux enseignes y gagnent leurs propres avantages. Pour Société MIEZAKA ces avantages sont :

- Les différentes friperies sont collectées par le Relais France sur tout le territoire français. Elles sont ensuite conditionnées dans des balles de 400 kg à 600 kg et acheminées dans le port de départ. Leur voyage continue en traversant les mers et océans pour arriver au port de Tamatave (Port de Madagascar). Elles sont après acheminées vers Fianarantsoa et dédouanées là-bas. Viens à la suite le tri des balles fait au Relais afin de les éclater et dispatcher dans les 8 ateliers d'habillement (exemple : pantalon homme, linge de maison), ainsi que la presse et le conditionnement pour les faire sortir en balles de 45 kg. Pour finir, l'ultime étape est le transfert dans les dépôts de Société MIEZAKA. Le fait que les balles soient triées à Madagascar représente un point fort pour Société MIEZAKA car elle bénéficie d'un faible coût de mains d'œuvre par rapport à un tri fait en France et permet alors de proposer des balles moins chères par rapport à ses concurrents.
- Société MIEZAKA détient un entrepôt dans l'enceinte du centre de tri TARATRA, Le Relais s'occupe de l'entreposage des marchandises ainsi que de la gestion des stocks de celles-ci. Le fait d'avoir un dépôt là-bas est également un avantage car cela permet d'avoir un lieu en plus pour stocker la marchandise et avoir encore plus d'offre pour répondre à la forte demande.

2.2.2. Collaboration avec des Fournisseurs Internationaux

De cet avantage en découle une autre clé de succès pour la Société MIEZAKA, malgré une concurrence assez présente localement ainsi que nationalement, le critère de différenciation de celle-ci repose réellement sur le tri fait à Madagascar. Grâce à cela la qualité des marchandises qui arrivent est garantie et le prix est moins cher. La société présente pour les clients un réel avantage qualité/prix par rapport à ses concurrents.

Un point qui renforce également la réussite de la société comme nous avons pu le voir avec Le Relais c'est la relation forte que Société MIEZAKA instaure avec ses différents fournisseurs. MIEZAKA travaille avec une dizaine de fournisseurs, les plus gros étant les suivants : Le Relais France, SOEX (Allemagne), BADROTEX (Belgique), TTR (La Réunion).

2.4.3. Accords Commerciaux et Facilités d'Importation

Il est important de noter également que grâce à son chiffre d'affaires et à ses nombreuses réceptions de containers au port de Tamatave, un contrat de performance a été signé entre

Société MIEZAKA et la Douanes Malgache. Aujourd’hui, l’entreprise fait face à un contexte douanier très favorable pour le déroulement de ses affaires. En effet, depuis cette année grâce à sa situation financière saine (200/200 points) qui montre que la société est solvable, celle-ci fait partie des 21 entreprises malgaches éligibles à la Procédure Accélérée de Dédouanement (PAD) c’est-à-dire qu’elle bénéficie du circuit vert qui lui permet l’enlèvement immédiat des marchandises après le dépôt de la déclaration en douane, par enlèvement immédiat on entend par là aucun contrôle de la marchandise un paiement de la douane s’effectuant dans les jours suivants. Être bénéficiaire de cette PAD est un véritable atout pour Société MIEZAKA car cela lui permet entre autres une réduction des délais et coûts des libéralisations de ses marchandises ou encore une transparence et traçabilité de ses opérations douanières.

Autre que la Procédure Accélérée de Dédouanement mais toujours dans le contexte douanier, Société MIEZAKA est également éligible à l’Autorisation Globale de Mise en Entrepôt Privé (AGLOMEP). « L’entrepôt privé est le régime douanier qui permet le stockage des marchandises sous contrôle douanier dans les locaux agréés par l’Administration des douanes en suspension des droits et taxes et des mesures économiques en attendant l’assignation à un autre régime douanier » (définition selon les Douanes Malagasy). Avoir l’AGLOMEP lui permet également faciliter et sécuriser le dédouanement de ses marchandises.

2.3. Fiche d’identification

RAISON SOCIAL	SOCIETE MIEZAKA
FORME JURIDIQUE	EURL
CAPITAL SOCIAL	AR 10000000
ADRESSE	Lot : 111 3608 SAHALAVA FIANARATSOA
TELEPHONE	7551655
FAX	7552125
EMAIL	siege@miezaka.com
SIEGE SOCIAL	Lot 111MA/3608sahalava

Tableau 2-1 Présentation de la fiche d’identification de la Société Miezaka

2.4. Implantation d'une usine d'efflichage

Actuellement, Société MIEZAKA est sur le point d'implanter une usine effilocheuse défibreuse pour recyclage de matières textiles, sise à Ampahipotsy, Fokontany Anjaninoro – Fianarantsoa.

La création de l'usine d'Effilochage répond à un besoin de notre économie locale (géotextiles, isolants, coussins, matelas, et autres).

Le projet industriel contribue à la lutte pour la protection de l'environnement car il permet la transformation de produits classés déchets antérieurement. Cette valorisation des déchets constitue pour nous une opération indispensable pour réduire l'impact de l'homme sur son environnement en donnant une seconde vie aux déchets.

Forte de ses 32 années d'expériences dans l'importation de friperies, la société MIEZAKA a fait le choix de prioriser l'importation de grosses balles pour être traitées dans une usine de tri à MADAGASCAR, créée à cet effet à Fianarantsoa.

Ce choix a généré des emplois faisant vivre plusieurs centaines de personnes à Fianarantsoa et les environs immédiats. Cette contribution au renforcement du tissu économique et à la valorisation de la main d'œuvre locale est le fruit d'un travail long et continu.

Chapitre 3 : Présentation du projet

3.1. Origine du projet

Dans la Société Miezaka, chaque employé a son propre solde d'absence ou de congé. Ce solde augmente chaque mois de 2,5 jours, ce qui représente le droit de l'employé, et diminue lorsque l'employé prend des absences. Il y a des congés spéciaux, comme le congé de maternité, le congé de décès, et d'autres, qui ne réduisent pas le solde de congés. Ces congés sont traités séparément, permettant aux employés de bénéficier de temps libre sans affecter leur solde mensuel.

Chaque employé a un plafonnement qui détermine le nombre maximum de jours de congé qu'il peut prendre en un mois. Ce plafonnement est variable et peut être différent pour chaque employé, selon les politiques de l'entreprise ou les conditions spécifiques liées à son poste. Certains congés, comme le congé de maternité ou le congé de décès, ne sont pas soumis à cette limite.

Les responsables des ressources humaines (RH) sont chargés de gérer ce processus. Ils suivent les soldes d'absence et de congé des employés, ainsi que les augmentations et diminutions de ces soldes, à l'aide d'Excel. Ce tableau permet de maintenir un enregistrement précis et à jour des congés, garantissant ainsi une gestion efficace des absences au sein de l'entreprise.

3.2. Problématique

Dans un environnement où la gestion des ressources humaines est cruciale pour le bon fonctionnement de l'entreprise, comment la société Miezaka peut-elle améliorer le suivi et la gestion des absences de ses employés tout en respectant les règles concernant les congés réguliers et spéciaux ? Quelles solutions peuvent être mises en place pour aider les responsables RH à gérer efficacement les absences et garantir que chaque employé puisse profiter de ses droits sans créer de problèmes administratifs ?

3.3. Conduite de projet

Dans un environnement professionnel où la gestion des ressources humaines est cruciale, la société Miezaka souhaite améliorer son système de gestion des congés pour garantir que chaque employé puisse bénéficier de ses droits tout en maintenant une bonne organisation. Actuellement, la gestion des absences repose sur des méthodes manuelles, ce qui peut entraîner des erreurs et des retards.

3.4. Cahier des charges

Dans le cadre de notre projet de fin d'études, la société m'a demandé de concevoir et de réaliser une application capable de gérer efficacement les absences des employés. Cette application aura pour but de simplifier le suivi des congés, de faciliter la gestion des différents types d'absences (annuelles, spéciales, etc.), et d'automatiser les processus de calcul du solde de congés, tout en respectant les plafonds et règles spécifiques de l'entreprise.

3.4.1. Définitions

Le cahier des charges, tel que défini par Pierre G. (2005) et Bakari M. (2016), est un document rédigé par le maître d'œuvre à l'attention du maître d'ouvrage. Il sert à formaliser les attentes du client de manière claire et compréhensible, en évitant les termes techniques complexes. Bien que le maître d'œuvre soit en charge de proposer la solution technique, le cahier des charges présente les besoins du projet sous un angle purement fonctionnel, sans entrer dans les détails de l'implémentation. Ce document joue également un rôle clé en assurant que les livrables correspondent aux attentes du maître d'ouvrage. De plus, il permet au client d'ajuster ses demandes au fur et à mesure de l'avancement du projet, en intégrant parfois des fonctionnalités nouvelles non prévues initialement.

3.4.2. Objectifs du projet

L'objectif principal de ce projet est de développer une application qui automatisera la gestion des absences au sein de la société Miezaka. Cette application devra :

1. **Automatiser le calcul des soldes de congés** : Chaque mois, le solde des congés des employés sera automatiquement crédité de 2,5 jours. Le solde sera ajusté en temps réel selon les jours de congé pris, sans affecter les congés spéciaux.
2. **Gérer différents types de congés** : L'application devra être capable de gérer les congés annuels ainsi que les congés spéciaux, tels que les congés de maternité ou de décès, qui ne doivent pas diminuer le solde des employés.
3. **Mettre en place des plafonnements individuels** : Chaque employé a un plafond de congés mensuel qui peut varier en fonction des politiques internes. L'application devra prendre en compte ces spécificités pour s'assurer que les employés ne dépassent pas leurs plafonds mensuels, à l'exception des congés spéciaux.

4. **Améliorer l'efficacité du service RH** : En automatisant le processus de gestion des absences, l'application réduira la charge administrative liée aux suivis manuels, minimisera les erreurs, et permettra aux RH de se concentrer sur d'autres tâches à plus forte valeur ajoutée.

3.4.3. Planification du projet

	Juillet	Août	Septembre
Semaine(01)	Intégration de l'entreprise	Installations des outils utilisés	Développement du backend et fronted
Semaine(02)	Recueil des données	Développement du backend	Test de l'application
Semaine(03)	Conception du projet : Elaboration des différents diagrammes	Développement du backend	Rédaction du mémoire
Semaine(04)	Choix des technologies utilisés	Développement du backend et fronted	Rédaction du mémoire

Tableau 3-1Planning du déroulement du stage

PARTIE II : ANALYSE ET CONCEPTION

Chapitre 4: Méthodes et Notations Utilisés

Cette approche permet aux chercheurs et étudiants d'obtenir les informations nécessaires à leurs projets. Son objectif est d'assurer une conception optimale du système d'information tout en simplifiant la programmation et la supervision du processus. Bien qu'il existe diverses méthodes de conception, nous privilégierons ici une méthode largement adoptée par la communauté des développeurs, à savoir la méthode UML (Unified Modeling Language).

4.1. L'UML

4.1.1. Historique

Né de la convergence des trois méthodes orientées objet les plus dominantes (OMT, Booch et OOSE) et officialisé par l'OMG en 1997, UML s'est rapidement imposé comme un standard essentiel dans le domaine de la modélisation [Rumbaugh J., Jacobson I. et Booch G., 1999]. Bien qu'il ne soit pas à l'origine des concepts orientés objet, UML leur apporte une définition plus rigoureuse et intègre une dimension méthodologique qui manquait aux approches orientées objet. Cette présentation n'a pas pour but de promouvoir UML ni de réduire son utilisation à sa seule notation graphique. En réalité, la vraie valeur d'UML réside dans son application ! Il est essentiel de comprendre que maîtriser la notation graphique ne constitue pas un objectif en soi. L'important est d'appliquer judicieusement les concepts orientés objet et de suivre la démarche d'analyse appropriée.

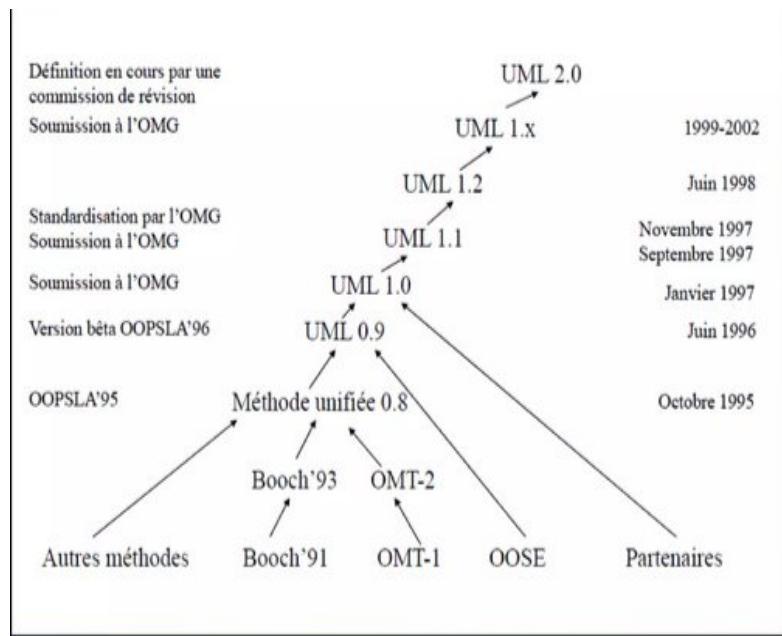


Figure 4-1 Évolution de l'UML

4.1.2. Définitions et objectifs

L'UML (Unified Modeling Language) est un langage de modélisation visuel standardisé, utilisé pour concevoir et documenter des systèmes logiciels, mais aussi des processus métier ou d'autres types de systèmes complexes. Il permet de représenter graphiquement les différentes composantes d'un système, notamment sa structure, son comportement, et ses interactions, à l'aide de diagrammes.

UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.

L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions. Il a été conçu comme un ensemble de notation qui vise à être utilisées dans différentes étapes de la modélisation objet, de l'expression des besoins jusqu'à l'implantation. Il s'agit de la notation synthétique et intuitive et accessible par différentes intervenants de la modélisation.

UML est donc un langage de modélisation avec plusieurs objectifs qui font un véritable outil de communication [Laurent A., 2010]:

- Comprendre et décrire les besoins ;
- Spécifier un système ;
- Établir l'architecture logicielle.

4.1.3. UML face à la programmation

Avant de se lancer directement dans l'écriture du code pour développer une application, il est essentiel de structurer ses idées, de les formaliser et de planifier le processus de réalisation. Cela inclut la définition des différents modules et l'organisation des étapes du développement. L'une des forces de l'UML réside dans la diversité des diagrammes qu'il propose, permettant de représenter ces aspects de manière claire et méthodique.

4.1.4. Points forts et points faibles de l'UML

4.1.4.1. Points forts

- Standardisation et universalité : UML est un langage de modélisation standardisé, adopté et utilisé internationalement. Il fournit un cadre commun aux équipes pour décrire les systèmes, quel que soit le langage de programmation utilisé.

- Visibilité et communication : Les diagrammes UML facilitent la communication entre les développeurs, les chefs de projet et les autres parties prenantes en fournissant une représentation visuelle claire des systèmes complexes.
- Flexibilité : UML s'applique à divers types de systèmes, y compris les systèmes logiciels, les processus métier et même les systèmes non informatiques.
- Outils d'analyse et de conception puissants : Les différents types de diagrammes (structurels et comportementaux) permettent d'aborder plusieurs aspects d'un système, de sa structure (classes, objets, composants) à son comportement dynamique (séquence, états, interactions).
- Amélioration de la qualité du code : En modélisant les systèmes avant de les coder, les équipes identifient plus facilement les incohérences ou erreurs conceptuelles avant la phase d'implémentation, réduisant ainsi les risques et améliorant la qualité du code final.
- Documentation complète : UML permet de documenter de manière exhaustive un projet tout au long de son cycle de vie, facilitant ainsi la maintenance et la compréhension du système à long terme.

4.1.4.2. Points faibles

- Complexité : UML peut être difficile à appréhender pour les débutants ou dans des projets de petite envergure. Il nécessite une bonne connaissance préalable pour être utilisé efficacement, ce qui peut freiner certaines équipes.
- Temps et ressources : La modélisation UML, notamment pour des projets complexes, peut être chronophage. Les équipes doivent consacrer du temps à la création de diagrammes détaillés avant de commencer la phase de développement, ce qui peut allonger les délais de livraison.
- Manque de flexibilité pour les petites applications : Pour de petits projets, l'utilisation de l'UML peut être perçue comme excessive. Des systèmes plus simples peuvent ne pas nécessiter autant de formalismes ou de documentation.
- Pas toujours adapté aux méthodes agiles : Bien que UML puisse être intégré dans des processus agiles, certaines équipes le trouvent rigide et difficile à adapter dans un contexte où les spécifications évoluent constamment. Les méthodes agiles privilégient des documents plus légers et des itérations rapides.
- Manque d'outils de rétro-ingénierie : Même si certains outils peuvent générer du code à partir de diagrammes UML, la création d'UML à partir de code existant (rétro-ingénierie) reste complexe et limitée dans certains environnements.

- Ambiguïtés d'interprétation : En fonction de l'outil ou de la personne qui le manipule, certaines parties d'UML peuvent être interprétées de manière différente, ce qui peut entraîner des incohérences dans la compréhension entre les membres de l'équipe.

4.1.5. Les grandes lignes essentielles de l'UML

L'UML, ou « Langage de modélisation unifiée », est utilisé pour clarifier et simplifier la conception et la représentation d'un système ainsi que son fonctionnement. Il s'agit d'un langage privilégié pour illustrer de manière compréhensible, tant pour l'homme que pour la machine, les différentes étapes essentielles de ce système.

La modélisation UML se divise en deux types de vues :

- Les vues statiques ;
- Les vues dynamiques.

Les vues statiques représentent le système physiquement qui comprend les diagrammes suivants :

- Diagramme de classe ;
- Diagramme de cas d'utilisation ;
- Diagramme de composant ;
- Diagramme de déploiement.

Les vues dynamiques montrent le fonctionnement du système qui comprend les diagrammes suivants :

- Diagramme de séquence ;
- Diagramme d'état transition ;
- Diagramme d'activité.

Dans le cadre de la conception de notre application, nous avons choisi d'utiliser uniquement ces diagrammes, car ils offrent les meilleures solutions pour répondre aux besoins de l'utilisateur dans ce nouveau système.

Nous allons maintenant détailler les diagrammes spécifiques que nous avons utilisés pour la modélisation de notre application.

4.1.5. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont essentiels pour structurer les besoins des utilisateurs et définir les objectifs d'un système. Ils mettent l'accent sur les attentes des utilisateurs en se concentrant sur leurs interactions avec le système. L'idée centrale est que les objectifs du système sont directement motivés par les utilisateurs. Cependant, identifier et comprendre ces besoins peut s'avérer complexe, car les parties prenantes sont souvent submergées par un excès d'informations. Il devient donc crucial de clarifier et d'organiser ces besoins en les modélisant de manière structurée.

Les cas d'utilisation permettent ainsi de :

- Identifier les acteurs qui interagissent avec le système ;
- Définir leurs interactions avec le système ;
- Classer les acteurs et organiser les objectifs du système.

Une fois cette structuration réalisée, les cas d'utilisation offrent plusieurs avantages :

- Ils délimitent le périmètre du système en précisant les objectifs à atteindre ;
- Ils mettent en évidence les fonctionnalités clés du système, en se concentrant sur celles qui sont essentielles.

L'objectif des cas d'utilisation n'est pas de proposer des solutions techniques ou des détails d'implémentation, mais plutôt de s'assurer que l'on reste focalisé sur les besoins des utilisateurs, sans tomber dans une approche purement fonctionnelle qui se contenterait de lister une série de fonctions à réaliser.

4.1.5.1. Acteur

La première étape de la modélisation consiste à délimiter le périmètre du système et à modéliser son cadre organisationnel. Toute entité extérieure à ce cadre, mais en interaction avec le système, est qualifiée d'acteur en UML.

Un acteur est une abstraction stéréotypée qui se situe à l'extérieur du système modélisé. Il représente un rôle spécifique joué par une personne ou un élément externe interagissant avec le système. Il est important de noter qu'une même personne physique peut être représentée par plusieurs acteurs en fonction des différents rôles qu'elle endosse.

Pour identifier ces acteurs, il faut donc s'attarder sur les rôles joués par les entités extérieures au périmètre du système.



Figure 4-2 Représentation des acteurs

4.1.5.2. Cas d'utilisation

Un cas d'utilisation est une unité cohérente qui représente une fonctionnalité observable de l'extérieur. Il fournit un service complet, avec un point de départ, un déroulement et une conclusion, destiné à l'acteur qui l'initie. En d'autres termes, un cas d'utilisation modélise un service offert par le système, sans pour autant spécifier comment ce service est mis en œuvre.

Graphiquement, un cas d'utilisation est représenté par une ellipse contenant le nom de ce cas, généralement exprimé sous la forme d'un verbe à l'infinitif. Il peut également inclure, au-dessus du nom, un stéréotype indiquant des informations supplémentaires. La figure 4-3 présente un exemple classique de cette représentation.



Figure 4-3 Représentation de cas d'utilisation

4.1.5.3. Relation entre cas d'utilisation

UML propose trois types de relations standardisées entre les cas d'utilisation, qui sont décrites ci-dessous :

- Relation de généralisation : Elle reflète une hiérarchie entre cas d'utilisation, où un cas d'utilisation plus spécifique hérite du comportement d'un cas d'utilisation plus général.
- Relation d'inclusion : Représentée par le mot-clé « include », elle signifie qu'un cas d'utilisation inclut systématiquement le comportement d'un autre cas d'utilisation pour accomplir sa tâche
- Relation d'extension : Représentée par le mot-clé « extend », elle décrit une extension conditionnelle d'un cas d'utilisation par un autre, activée sous certaines conditions ou circonstances spécifiques.

4.1.5.4. Formalisme d'un diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de visualiser les fonctionnalités que le système doit offrir. Il aide également à délimiter le système et à montrer les interactions entre celui-ci et son environnement, que ce soit avec les utilisateurs ou d'autres systèmes

La représentation formelle d'un diagramme de cas d'utilisation, illustrée par la figure 4-4 montre les relations entre les acteurs et les cas d'utilisation, mettant ainsi en évidence les fonctionnalités principales et les interactions extérieures du système.

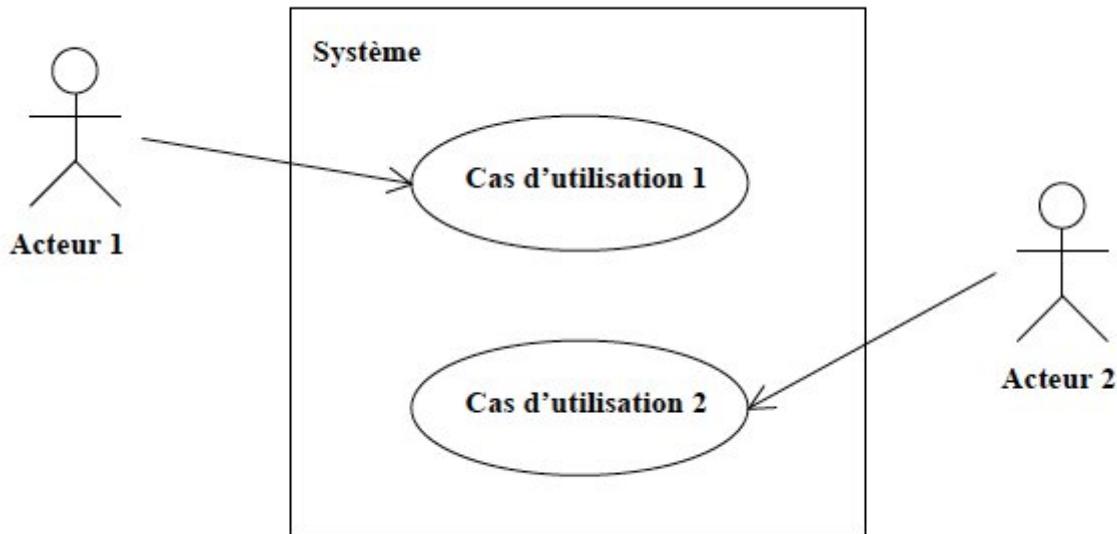


Figure 4-4 Formalisme du diagramme de cas d'utilisation

4.1.6. Diagrammes des séquences

Le diagramme de séquence représente l'ordre chronologique des actions effectuées par un acteur. Il montre les objets que l'acteur manipule ainsi que les opérations qui permettent de passer d'un objet à un autre.

C'est une représentation graphique des interactions entre les acteurs et le système dans un ordre chronologique, selon la notation UML. Utilisé pour illustrer un cas d'utilisation, ce diagramme met l'accent sur l'enchaînement des messages échangés, tout en soulignant la temporalité des interactions.

4.1.6.1. Ligne de vie

La ligne de vie représente un participant, qu'il s'agisse d'un objet ou d'un acteur, impliqué dans une interaction. Elle symbolise la durée pendant laquelle cet acteur ou objet est actif au cours de l'échange dans le diagramme de séquence. Chaque ligne de vie montre ainsi le rôle du participant dans le déroulement chronologique des interactions.

4.1.6.2. Message

Le message dans un diagramme de séquence décrit la communication spécifique qui s'effectue entre des lignes de vie, qu'il s'agisse d'objets ou d'acteurs. Selon UML, différents types de messages (ou actions) peuvent transiter entre ces participants :

- Message simple : Une communication de base qui ne spécifie aucune caractéristique particulière d'envoi ou de réception.
- Message minuté (timeout) : L'expéditeur attend un certain temps que le récepteur prenne en compte le message. Si cela n'est pas fait dans le délai imparti, l'expéditeur est libéré.
- Message synchrone : L'expéditeur est bloqué jusqu'à ce que le récepteur prenne en charge le message. Une fois le message pris en compte, le contrôle passe du premier au second, rendant l'émetteur passif et le récepteur actif.
- Message asynchrone : L'expéditeur n'est pas interrompu dans son exécution, et le récepteur peut traiter ou ignorer le message à tout moment.
- Message dérobant : Le message n'interrompt pas l'exécution de l'expéditeur, mais est mis en attente dans une file de traitement côté récepteur pour être traité ultérieurement.

4.1.6.3. Formalisme d'un diagramme de séquence

La figure 4-5 présente une illustration graphique suivant le formalisme du diagramme de séquence.

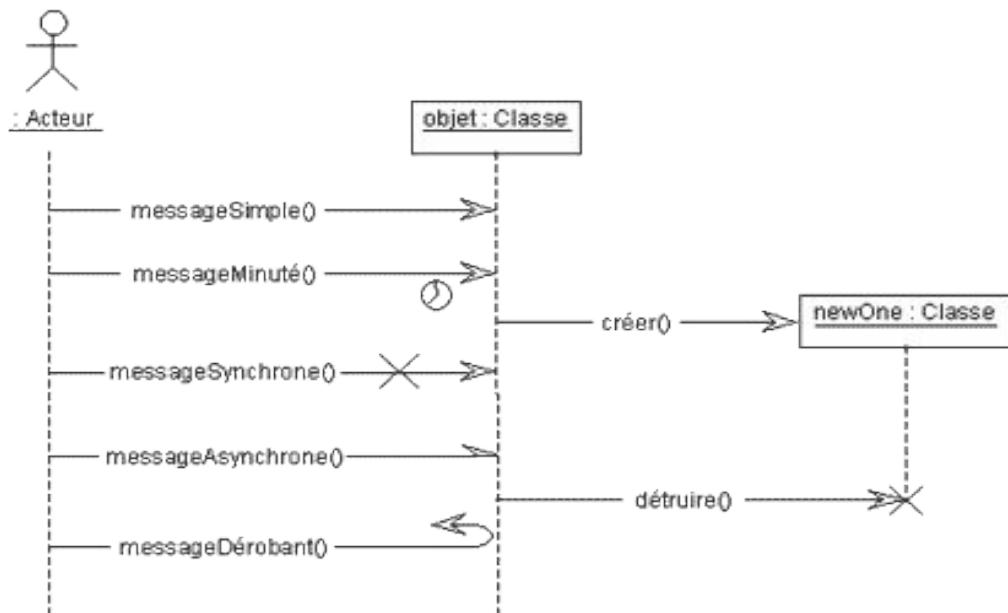


Figure 4-5 Formalisme du diagramme de séquence

4.1.7. Le diagramme de classe

Le diagramme de classes représente la structure statique d'un système, en montrant les classes et les relations entre elles. Il permet de modéliser les entités du système d'information, c'est-à-dire les objets qui le composent. Ce diagramme structure ces objets en regroupant les informations associées sous forme de classes. Il met également en évidence les relations qui peuvent exister entre ces classes.

Dans un diagramme de classes, on distingue 6 éléments principaux :

- Classe : C'est une représentation d'une entité avec ses caractéristiques (attributs) et ses comportements (opérations).
 - Relation : Les liens ou associations entre les classes qui montrent comment elles interagissent entre elles.
 - Attribut : Les propriétés ou données associées à une classe qui définissent son état.
 - Généralisation : Une relation hiérarchique entre des classes où une classe dérivée hérite des propriétés d'une classe de base.
 - Identifiant : Un attribut ou un groupe d'attributs qui permet d'identifier de manière unique un objet d'une classe.
 - Opération : Les actions ou comportements qu'une classe peut effectuer, souvent appelés méthodes.

La figure 4-6 nous montre la notation d'une classe.

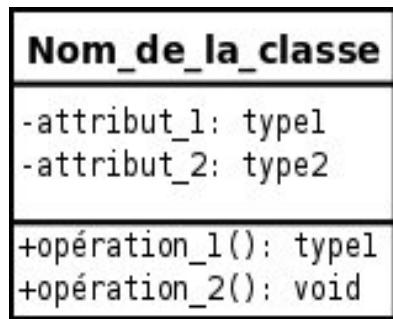


Figure 4-6 Formalisme d'une classe

Les différents types de visibilité sont montrés dans le tableau

Modificateur	Visibilité
Privé (-)	Seuls les membres de la classe uniquement
Publique (+)	Accessibles par toutes les classes
Package(~/)	Accessibles par toutes les classes de même package
Protégé (#)	A l'intérieur de la classe et dans la sous classe

Tableau 4-1 Les types de Visibilité

Les relations dans les diagrammes de classes permettent de montrer comment les différentes classes interagissent entre elles. Il existe plusieurs types de relations que l'on peut utiliser pour exprimer ces interactions. Le tableau 4.2 présente un résumé de ces types de relations.

Type de relation	Représentation graphique
------------------	--------------------------

<p>Association : Elles représentent un lien durable ou ponctuel entre deux objets, une appartenance, ou une collaboration. L'association permet à A d'atteindre B.</p>	
<p>Généralisation et héritage : B est la classe de base et A est la classe dérivée. La classe A peut posséder toutes les caractéristiques de sa classe parent B.</p>	
<p>Agrégation : Lorsqu'un objet en contient d'autres, on parle d'agrégation. A est inclus dans B.</p>	
<p>Composition (agrégation composite) : Quand le composite (B) est détruit, le composant (A) l'est aussi</p>	
<p>Implémentation : Une classe peut implémenter une interface. Les interfaces se différencient des autres classes par le stéréotype <<interface>>.</p>	
<p>Dépendance : C'est unidirectionnelle. La modification de B modification de A.</p>	

Tableau 4-2 Les différents types de relations dans les diagrammes de classes

4.1.8. Le diagramme d'activité

Le diagramme d'activité illustre les processus tels qu'ils ont été conçus lors de la phase de préparation à la modélisation. Il montre la séquence des activités qui participent à un processus. Ce type de diagramme met l'accent sur les actions ou traitements effectués. Il est donc particulièrement utile pour modéliser le flux de contrôle et le flux de données. Il permet de représenter visuellement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Un diagramme d'activités peut être lié à n'importe quel élément de modélisation pour visualiser, spécifier, construire ou documenter son comportement.

4.1.8.1. Composition d'un diagramme d'activités

Un diagramme d'activité se compose des éléments suivants :

- Activité : Représente l'exécution d'un ensemble d'étapes séquentielles.
- Garde : Condition de transition entre deux activités, symbolisée par un losange.
- Synchronisation : Barre épaisse utilisée pour coordonner plusieurs transitions venant de différentes activités vers une activité commune.
- Nœud d'activité : Élément abstrait marquant les étapes dans le flux d'une activité, incluant :
 - Nœuds d'actions
 - Nœuds objets
 - Nœuds de contrôle

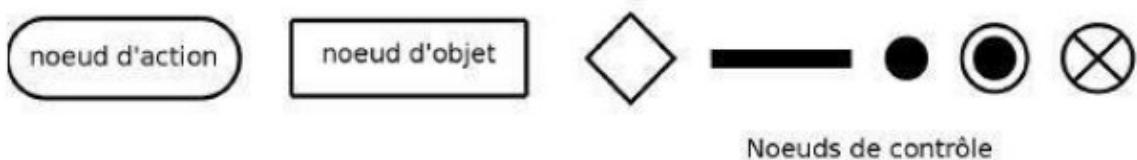


Figure 4-7 Représentation des nœuds d'activités

Chapitre 5 Analyse du Projet

Après avoir présenté les méthodes et notations utilisées dans notre projet, ce chapitre nous permet d'analyser en détail chaque aspect du futur système, notamment l'étude préalable,

l'existant, l'analyse des besoins, l'étude de faisabilité, ainsi que les objectifs généraux de l'application, tout en proposant des solutions aux critiques identifiées.

5.1. Analyse de besoins

L'analyse des besoins a pour objectif d'identifier les exigences fonctionnelles et techniques nécessaires à la mise en place d'un système de gestion des absences efficace. Voici les principaux besoins identifiés :

- Une application capable d'ajouter et de mettre à jour les informations des employés, y compris leur département, leur fonction, et d'autres données si nécessaire.
- Une application permettant d'ajouter les types d'absence conformément aux règles de la société, tels que les congés de maternité, les congés réguliers, etc.
- Un système pour automatiser l'augmentation du solde d'absence ou solde de congé des employés de 2,5 jours chaque mois.
- Développer une interface pour permettre aux employés de soumettre leurs demandes de congé.
- Une interface permettant de consulter le solde d'absence restants pour un employé.
- Intégrer des règles spécifiques pour les congés spéciaux (maternité, décès, etc.) qui ne diminuent pas le solde d'absence.
- Un système pour gérer les règles de plafonnement des absences des employés, qui s'applique uniquement aux congés réguliers, tandis que les congés de maternité et d'autres congés spéciaux ne sont pas soumis à cette restriction.
- Mettre en place un système pour calculer automatiquement les jours restants après chaque congé pris.
- Offrir la possibilité d'obtenir une liste des demandes d'absence par mois, par année, ou selon d'autres critères, et permettre l'exportation de ces données au format Excel.
- Une application destinée à fournir des statistiques adaptées aux besoins spécifiques de l'entreprise.
- Possibilité d'imprimer la demande immédiatement après sa soumission.
- Implémenter un système d'authentification robuste, afin que les utilisateurs doivent se connecter pour accéder à l'application.
- Mettre en place des niveaux d'accès pour protéger les informations sensibles, en veillant à ce que certaines routes ne soient accessibles qu'aux administrateurs et certains.

5.2. Analyse de faisabilité

L'analyse de faisabilité pour le projet de gestion des absences de la société Miezaka vise à évaluer la possibilité de développer une application efficace qui réponde aux exigences spécifiques des utilisateurs.

5.2.1. Faisabilité Technique

- Utilisation de Node.js pour le développement du backend et de Vue.js pour le développement du frontend.
- Base de données MySQL pour le stockage des informations relatives aux employés et aux demandes de congé.
- Visual Studio comme éditeur de code, et Postman facilite les tests des API.
- Une connexion Internet stable est disponible au sein de l'entreprise pour le développement et la recherche.

5.2.2. Faisabilité Opérationnelle

L'application sera utilisée par deux types d'utilisateurs : les administrateurs et les utilisateurs(RH). Les administrateurs auront accès à toutes les fonctionnalités, tandis que les utilisateurs pourront uniquement soumettre des demandes de congé.

5.3. Analyse de l'existant

Actuellement, la société Miezaka utilise des tableaux Excel pour gérer ses processus, notamment ceux liés à la gestion des absences. Bien que cette approche permette de suivre les données de manière basique, elle présente plusieurs limites qui nuisent à l'efficacité globale du système.

- Risque d'erreurs : La saisie manuelle des demandes d'absence et la mise à jour des soldes de congés de l'employé sont sujets à des erreurs humaines. Cela peut entraîner des incohérences dans les données et rendre difficile le suivi précis des absences.
- Perte de temps : Le processus de gestion des congés est lent et nécessite beaucoup de travail administratif, notamment pour le responsable chargé de suivre les absences et calculer les jours restants manuellement.
- Manque de visibilité : Les responsables n'ont pas une vue d'ensemble claire des absences des employés. Il est difficile d'obtenir des statistiques globales ou des rapports précis sur les jours pris par département ou par employé.

- Complexité des règles de congés spéciaux : Certains types de congés, comme les congés de maternité ou de décès, ne sont pas soumis aux mêmes règles que les congés normaux. Le système manuel actuel ne permet pas de gérer efficacement ces spécificités, ce qui ajoute à la confusion.
- Absence d'automatisation : Les augmentations automatiques du solde de congés de 2,5 jours par mois pour chaque employé ne sont pas mises en place dans l'outil actuel. Cela demande une mise à jour manuelle et régulière des soldes de chaque employé.

5.4. Critique de l'existant

Les demandes d'absence dans la société Miezaka sont traitées manuellement via des tableaux Excel, ce qui entraîne des risques d'erreurs et un suivi peu fiable. De plus, il n'existe pas de mécanisme pour augmenter automatiquement le solde d'absence des employés chaque mois, rendant la gestion des droits d'absence complexe.

De plus, la façon dont le solde diminue automatiquement après chaque absence régulière n'est pas bien gérée, ce qui peut causer des erreurs dans le suivi des droits d'absence. Les règles pour les absences spéciales, qui devraient empêcher la diminution du solde d'absence, ne sont pas bien intégrées, ce qui peut créer des confusions sur le nombre de jours d'absence restants.

5.5. Propositions des solutions

Après avoir examiné les problèmes de gestion des absences au sein de la société Miezaka, voici quelques solutions pour améliorer le système :

- Créer une application permettra aux responsables RH de gérer facilement les demandes d'absence des employés et de suivre leur solde d'absence.
- Implémenter un système qui calcule automatiquement le solde d'absence des employés. Chaque mois, le solde augmentera de 2,5 jours, et il diminuera automatiquement après chaque absence régulière.
- Mettre en place des règles claires stipulant que le nombre de jours d'absence qu'un employé peut prendre ne doit pas dépasser un certain plafond par mois.
- Les règles concernant les absences spéciales doivent être bien intégrées afin que ces absences n'affectent pas le solde d'absence des employés. Cela réduira les confusions sur le nombre de jours disponibles.

- Mettre en place des niveaux d'accès pour protéger les informations sensibles. Une bonne authentification garantira que seules les personnes autorisées, comme les responsables RH, peuvent accéder à l'application.
- Organiser des formations pour s'assurer que le personnel des RH comprend comment utiliser efficacement l'application et gérer les demandes d'absence.

5.6. Solutions retenues

Après l'évaluation des problèmes existants et des propositions de solutions, les mesures suivantes ont été retenues pour optimiser la gestion des absences :

- ✓ Création d'une Application de Gestion des Absences ;
- ✓ Automatisation des Calculs de Solde d'Absence ;
- ✓ Mise en Place de Règles de Plafonnement ;
- ✓ Gestion des Absences Spéciales ;
- ✓ Formation du Personnel RH ;
- ✓ Sécurisation de l'Application.

5.7. Analyse de l'architecture

L'architecture de l'application suit un modèle client-serveur qui sépare clairement les responsabilités entre le frontend et le backend.

- **Frontend** : L'interface utilisateur est conçue pour offrir une expérience fluide et réactive. Elle utilise des composants modulaires, ce qui facilite la réutilisation du code et la maintenance. La communication avec le backend s'effectue via des API, permettant l'échange de données de manière efficace.
- **Backend** : Le backend est responsable de la gestion de la logique métier et des opérations sur les données. Il expose une API qui permet au frontend de réaliser des opérations de création, lecture, mise à jour et suppression (CRUD). Des mécanismes de gestion des requêtes, de validation et d'authentification sont intégrés pour assurer la sécurité et l'intégrité des données.
- **Base de données** : Les données de l'application, telles que les informations sur les employés et les demandes d'absence, sont stockées dans une base de données. L'utilisation d'un ORM simplifie l'interaction avec la base de données, rendant les opérations plus sûres et efficaces.

L'application est déployée sur un serveur cloud, offrant plusieurs avantages :

- **Scalabilité** : Le serveur cloud permet une adaptation facile à la charge des utilisateurs, garantissant des performances optimales même lors des pics d'utilisation.
- **Disponibilité** : En s'appuyant sur des services cloud, l'application bénéficie de garanties de disponibilité, réduisant le risque d'interruption de service.

Chapitre 6 Conception du Projet

La conception d'un logiciel comprend plusieurs étapes, allant de la demande initiale d'automatisation d'un processus (qui peut être une simple demande orale ou un document détaillé) jusqu'à la création, le développement et les tests du programme. Le logiciel est ensuite livré à la personne ou à l'entreprise qui en a fait la demande.

6.1. Présentation de l'outil de conception

Il existe divers outils de conception permettant de créer des systèmes d'information. Parmi ces outils, nous avons utilisé des solutions telles que Visual Paradigm, Lucidchart, Win'Design, Paradigm Plus, Poseidon for UML et Visual UML. Nous allons présenter deux d'entre eux en détail.

6.1.1. *Lucidchart*

Lucidchart est un outil de création de diagrammes en ligne qui permet aux utilisateurs de concevoir facilement des schémas, des organigrammes et des diagrammes UML. Sa simplicité d'utilisation est un atout majeur, car il offre une interface intuitive par glisser-déposer.

Lucidchart favorise la collaboration en temps réel, permettant à plusieurs utilisateurs de travailler sur le même document simultanément. De plus, il est intégré à diverses applications comme Google Drive et Microsoft Office, facilitant ainsi l'importation et l'exportation des diagrammes.

Avec ses nombreux modèles prédéfinis et ses fonctionnalités de personnalisation, Lucidchart s'avère être un choix idéal pour la conception graphique de projets.

6.1.2. *Visual Paradigm*

Visual Paradigm est un outil complet de création de diagrammes pour la programmation, permettant une modélisation UML étendue. Il offre divers outils pour créer des schémas tels que les diagrammes d'exigences et de cas d'utilisation, tout en permettant une personnalisation fine des éléments.

L'un de ses atouts est la capacité de générer du code source (Java, C++, etc.) à partir de modèles UML, et inversement, de créer un modèle à partir de code existant. L'interface est intuitive, avec la possibilité de manipuler les diagrammes facilement via la souris, facilitant la création et la connexion d'éléments.

Visual Paradigm supporte toutes les actions nécessaires, telles que le traçage, la commande et la gestion des liens entre les composants.

6.2. Choix d'outils de conception

Dans le cadre du développement de l'application de gestion des absences, j'ai opté pour l'utilisation de Lucidchart comme outil de conception. Lucidchart se distingue par sa facilité d'utilisation et ses fonctionnalités de collaboration en temps réel,

Voici un tableau comparatif qui met en lumière les points forts de Lucidchart par rapport à Visual Paradigm :

Critères	Lucidchart	Visual Paradigm
Facilité d'utilisation	Interface intuitive et facile à prendre en main.	Peut nécessiter un temps d'apprentissage plus long.
Types de diagrammes	Large éventail de diagrammes disponibles (UML, organigrammes, etc.).	Offre également une bonne variété de diagrammes, notamment en UML.
Accessibilité	Accessible via le web, aucune installation requise.	Nécessite une installation sur l'ordinateur.
Intégrations	Intégrations faciles avec des outils tels que Google Drive, Microsoft Office, etc.	Intégrations disponibles, mais moins nombreuses.
Collaboration	Excellent fonctionnalités de collaboration en temps réel.	Collaboration possible mais moins fluide.

Tableau 6-1 Comparaison entre Ludichart et Visual Paradigm

À partir de ce tableau, nous pouvons conclure que Lucidchart se révèle être un choix plus avantageux pour la conception de notre application de gestion des absences.

6.3. Règles de gestion

Une application dynamique doit respecter un ensemble de règles de gestion. Ces règles se situent au niveau conceptuel et définissent le "QUOI" de l'entreprise. En d'autres termes, elles sont recueillies à partir des interviews, notamment auprès de la direction, afin de refléter les objectifs existants ou nouveaux, ainsi que les contraintes associées.

Avant d'entamer la conception du système, voici les principales règles de gestion à considérer :

RG1 : Un employé doit faire une demande d'absence avant de prendre des jours d'absence.

RG2 : Un employé ne peut pas dépasser le nombre maximum de jours d'absence autorisés par mois.

RG3: Un employé est identifié par ses attributs, qui incluent son identifiant unique, son nom, son prénom, son sexe, son poste ,son département, son solde d'absence, son plafonnement, l'indication de l'application du plafonnement, et la date de la dernière mise à jour de son solde.

RG4 : Un type d'absence est identifié par son id_absence, qui est une clé primaire auto-incrémentée, et il est associé à des attributs tels que le nom_absence, la durée, le type.

RG5: Chaque type d'absence a ses propres règles.

RG7: Les absences spéciales ne comptent pas dans le plafonnement mensuel.

RG8: Le solde d'absence de chaque employé augmente de 2,5 jours le 1er du mois.

RG9 :Le solde est réduit automatiquement après chaque absence prise, sauf pour les absences spéciales.

RG10 :Une demande d'absence appartient à un employé, qui est identifié par son id_employé, et elle est associée à des attributs tels que la date de début, la date de fin, le type d'absence, ainsi que le motif de la demande.

6.4. Dictionnaire de données

Le dictionnaire de données répertorie toutes les informations qui seront utilisées pendant la conception et la réalisation de l'application. Cette étape conduit à un dictionnaire de données qui doit être exempt de synonymes et de données calculées. Pour cette application, les données sont présentées dans le tableau 6.2 :

Rubrique	Définition	Type	Taille	Format	Etat
code_departement	Code du département	AN	255		E
date_debut	Date début de l'absence	D	10	yyyy -mm-dd	E
date_demande	Date de demande	D	10	yyyy -mm-dd	E

date_fin	Date fin de l'absence	D	10	yyyy -mm-dd	E
date_retour	Date de retour de l'employé après son absence	D	10	yyyy -mm-dd	E
duree	Durée de type de congé	N	10		E
email	Email de l'utilisateur	AN	255		E
id_abs	Identifiant de type de congé	N	10		E
Id_département	Identifiant du département	N	255		E
id_employé	Identifiant de l'employé	N	255		E
Id_poste	Identifiant poste	N	10		E
Id_user	Identifiant de l'utilisateur	N	10		E
isSpecial	C'est un type spécial ou non	B	5		E
jours_abs	Nombre de jours d'absence de l'employé				E
last_update_solde	La dernière mise à jour du solde de l'employé	D	10	yyyy -mm-dd	E
matricule	Matricule de l'employé	AN	255		E
mdp	Mot de passe de l'utilisateur	AN	255		E
motif	Motif de la demande	A	255		

	d'absence d'un employé				E
nom_absence	Nom du congé	AN	255		E
nom_employé	Nom de l'employé	AN	255		E
nom_user	Nom de l'utilisateur	AN	255		E
pfafonnementbole an	Qu'un employé peuvent dépasser sont plafonnement ou non	B	5		E
plafonnement	Limite qu'un employé peuvent prendre en un mois	N	10		E
poste	Poste de l'employé	A	255		E
prenom_employé	Prénom de l'employé	A	255		E
role	Rôle de l'utilisateur	AN	255		E
sexe	Sexe de l'employé	AN	255		E
solde_employe	Solde de l'employé	N	10		E
type	Type de congé	AN	255		E

Tableau 6-2 Dictionnaire de données

Légende : alphanumérique (AN), entier long (N), date (D), Boolean(B)

6.5. Modelisation

Modéliser un système avant sa réalisation aide à comprendre son fonctionnement et à en gérer la complexité tout en garantissant sa cohérence. C'est un outil précieux pour partager une vision commune entre les membres de l'équipe, facilitant la communication grâce à un langage clair

et précis. Dans le cadre de l'ingénierie logicielle, un modèle permet de mieux organiser le travail et d'automatiser certaines tâches, ce qui contribue à réduire les coûts et les délais de développement. Il joue également un rôle crucial dans l'amélioration de la qualité et dans la maintenance du logiciel. Pour notre projet, nous utiliserons le langage UML (Unified Modeling Language) pour modéliser l'application en élaborant les différents diagrammes UML nécessaires.

6.5.1. Schéma de diagramme de cas d'utilisation globale

Pour concevoir un diagramme de cas d'utilisation, il est essentiel de suivre une approche structurée. Dans un premier temps, il convient d'identifier les acteurs, qui représentent les utilisateurs ou systèmes externes interagissant avec l'application. Ensuite, il s'agit de définir les cas d'utilisation, c'est-à-dire les différentes actions que le système permet aux acteurs de réaliser. Enfin, en utilisant un outil de modélisation tel que Lucidchart, on peut représenter graphiquement ces interactions, comme illustré dans la figure 6-1 qui montre la structure de notre projet à travers ce diagramme.

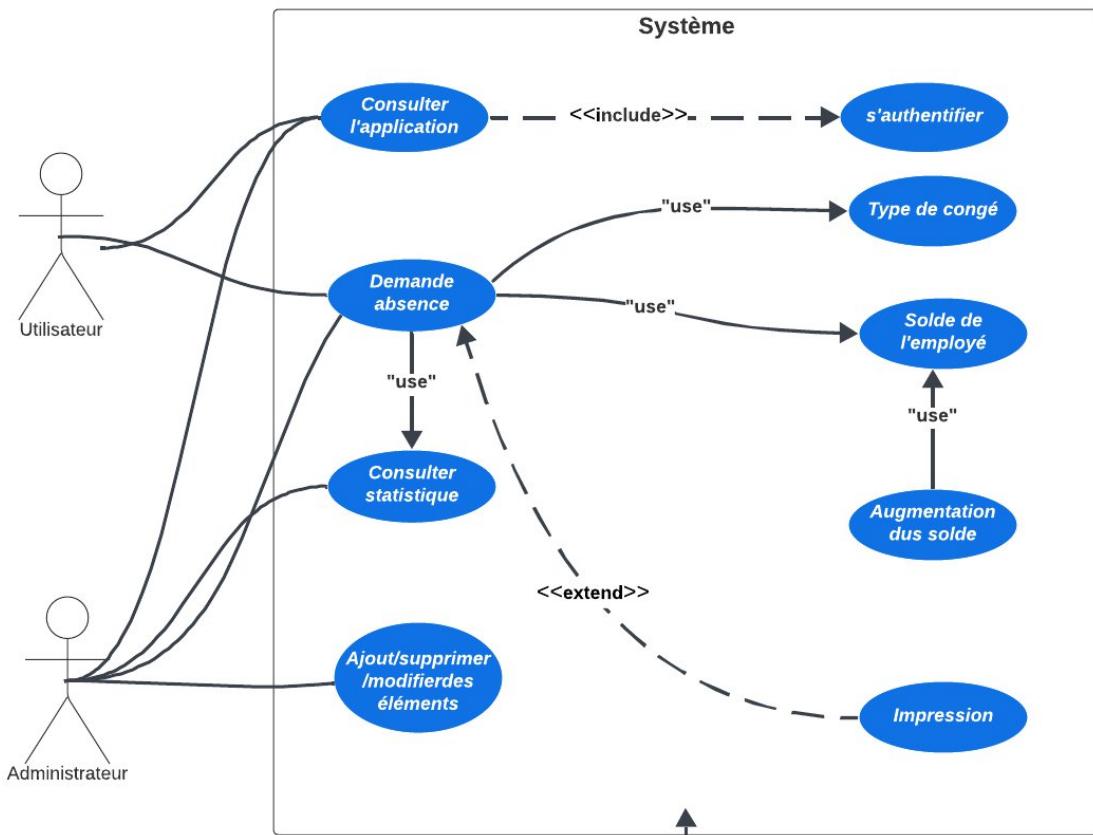


Figure 6-1 Représentation graphique du diagramme de cas d'utilisation global

6.5.2. Étude du cas d'utilisation : « S'authentifier »

6.5.2.1. Description du cas d'utilisation s'authentifier

Titre : S'authentifier.

Objectif : Se connecter pour pouvoir accéder à l'application.

Résumé : Les utilisateurs rentrent sur login pour se connecter.

Acteur : Le Rh et l'administrateur.

Pré-condition : Les utilisateurs arrivent dans l'application

Post-condition : Les utilisateurs arrivent dans la vue Accueil.

Scénario nominal :

1. L'utilisateur ouvre l'application.
2. L'application affiche la page de connexion.
3. L'utilisateur entre son identifiant et mot de passe.
4. L'application vérifie l'utilisateur dans la base de données.
5. Le système redirige vers la page d'accueil.

Scénario alternatif :

3a. L'utilisateur n'a pas de compte et déclenche "Créer compte".

4a. Si l'utilisateur est inconnu, l'application affiche une erreur et retourne à l'étape 4.

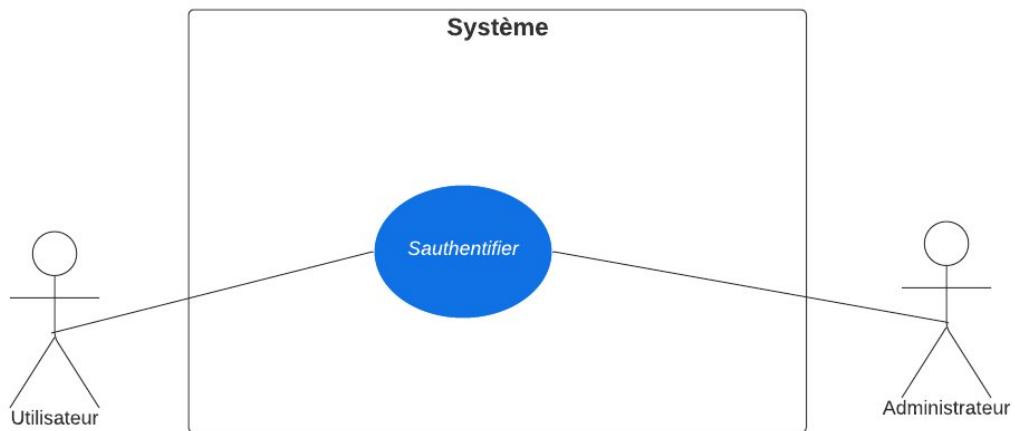


Figure 6-2 Diagramme de cas d'utilisation s'authentifier

6.5.2.2. Diagramme de séquence de l'authentification

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'une utilisation.

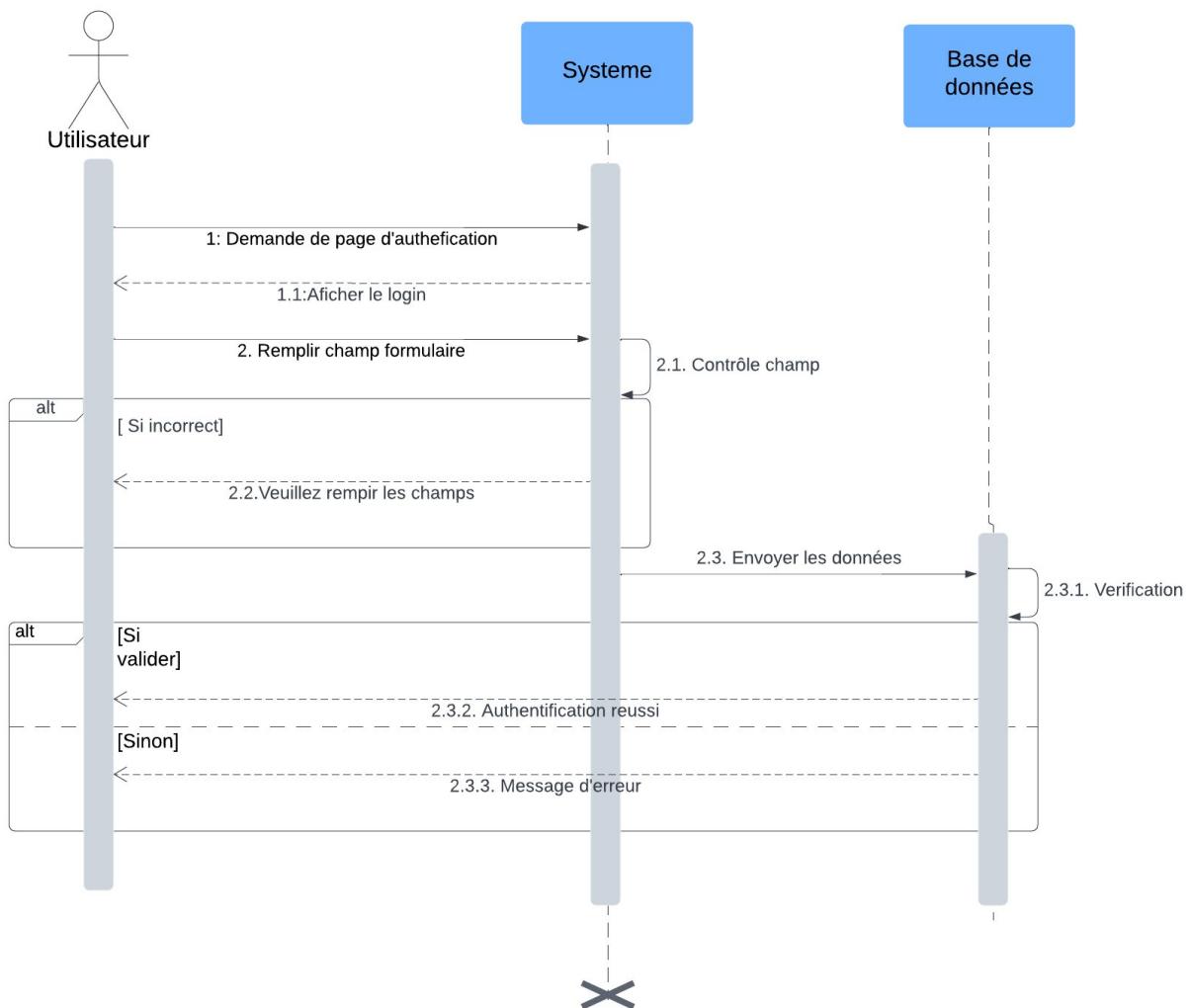


Figure 6-3 Représentation du diagramme de séquence de l'authentification

Description du scénario

N° enchaînement	Action
1	Demande page d'authentification
1.1	Affichage de formulaire d'authentification
2	Permet de remplir le champ formulaire
2.1	Faire contrôle champ
2.2	Message si les champs ne sont pas remplir
2.3	Envoie des données dans la base
2.3.1	Vérification dans la base
2.3.2	Authentification réussi
2.3.3	Message d'erreur

Tableau 6-3 Description du scénario de l'authentification

6.5.2.3. Diagramme d'activité d'authentification

La figure 6-4 représente le diagramme d'activité du cas d'utilisation «Authentification».

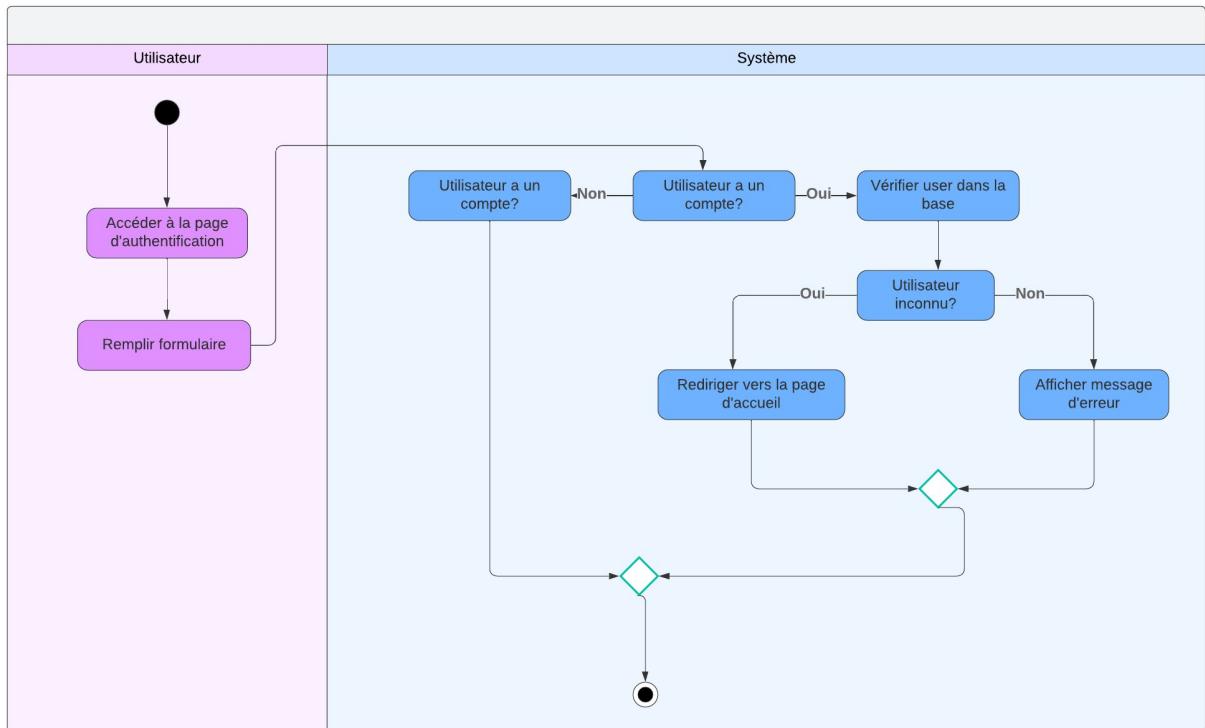


Figure 6-4 Diagramme d'activité de l'authentification

6.5.3. Étude du cas d'utilisation « Ajout des éléments(employé,département,poste)»

6.5.3.1. Description du cas d'utilisation

Titre : Ajouter un élément

Objectif : Enregistrer un nouvel élément (employé, département ou poste) dans l'application.

Résumé : L'utilisateur remplit et soumet le formulaire pour ajouter un nouvel élément.

Acteur : RH et Administrateur.

Pré-condition : L'utilisateur est authentifié et accède à la page d'ajout correspondante.

Post-condition : L'élément est ajouté dans la base de données et une confirmation s'affiche.

Scénario nominal :

1. L'utilisateur accède à la page d'ajout.
2. L'utilisateur remplit le formulaire avec les informations nécessaires.
3. L'utilisateur soumet le formulaire.

4. L'application vérifie les informations.
5. Si les données sont valides, l'élément est ajouté à la base de données.
6. Le système affiche un message de confirmation.

Scénarios alternatifs :

4a. Élément existe déjà : L'application affiche un message d'erreur indiquant que l'élément est déjà enregistré.

4b. Formulaire incomplet : L'application affiche un message d'erreur demandant de remplir tous les champs requis.

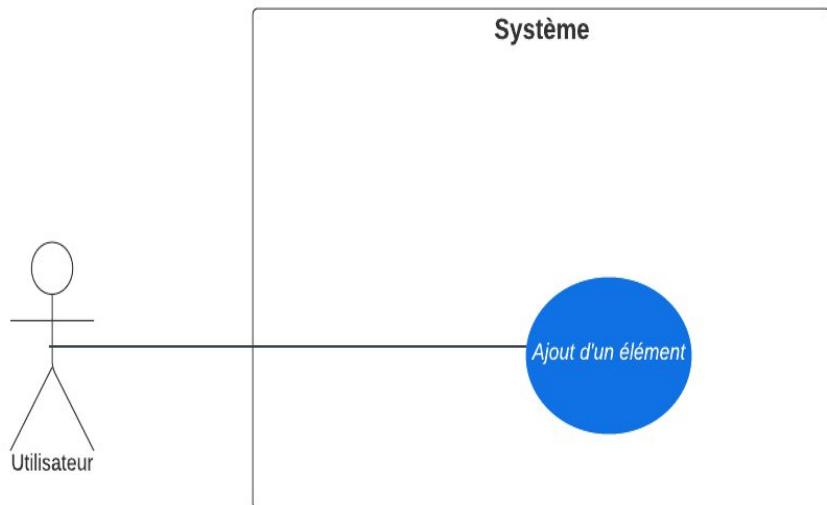


Figure 6-5 Diagramme de cas d'utilisation d'ajout d'un élément

6.5.3.2. Diagramme de séquence ajout d'un élément(employer,département,poste)

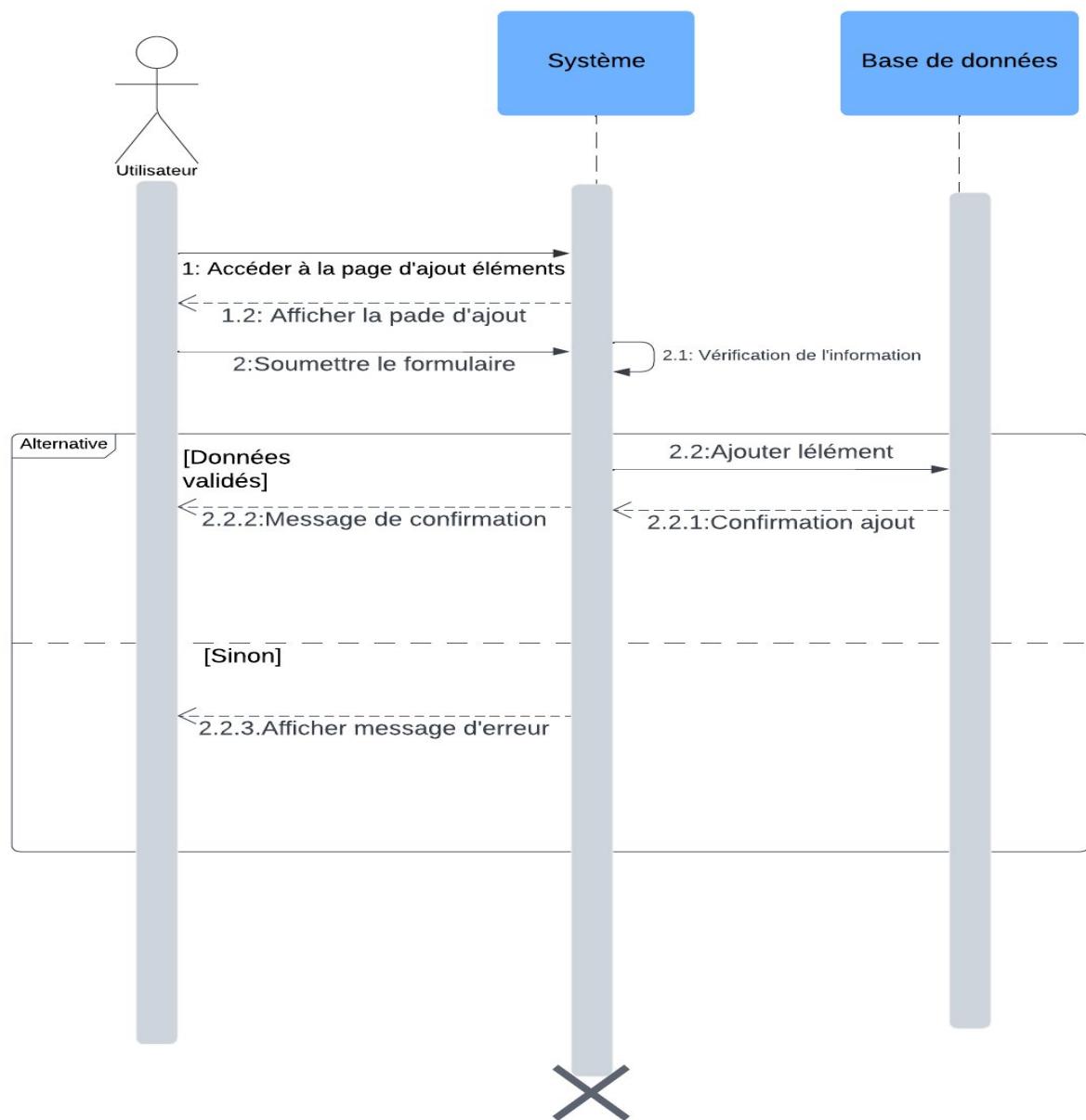


Figure 6-6 Diagramme de séquence d'ajout d'un élément(employer, département, poste)

Description du scénario

N° enchaînement	Action
1	Accéder à la page d'ajout d'éléments
1.2	Affichage de la page d'ajout
2	Soumettre le formulaire
2.1	Vérification de l'information
2.2	Ajout d'un élément dans la base de donnée
2.2.1	Confirmation de l'ajout
2.2.2	Message de confirmation
2.2.3	Affichage message d'erreur

Tableau 6-4 Description du scenario d'ajout d'un élément(employé,département,poste)

6.5.3.3. Diagramme d'activité d'un ajout élément(employé,département,poste)

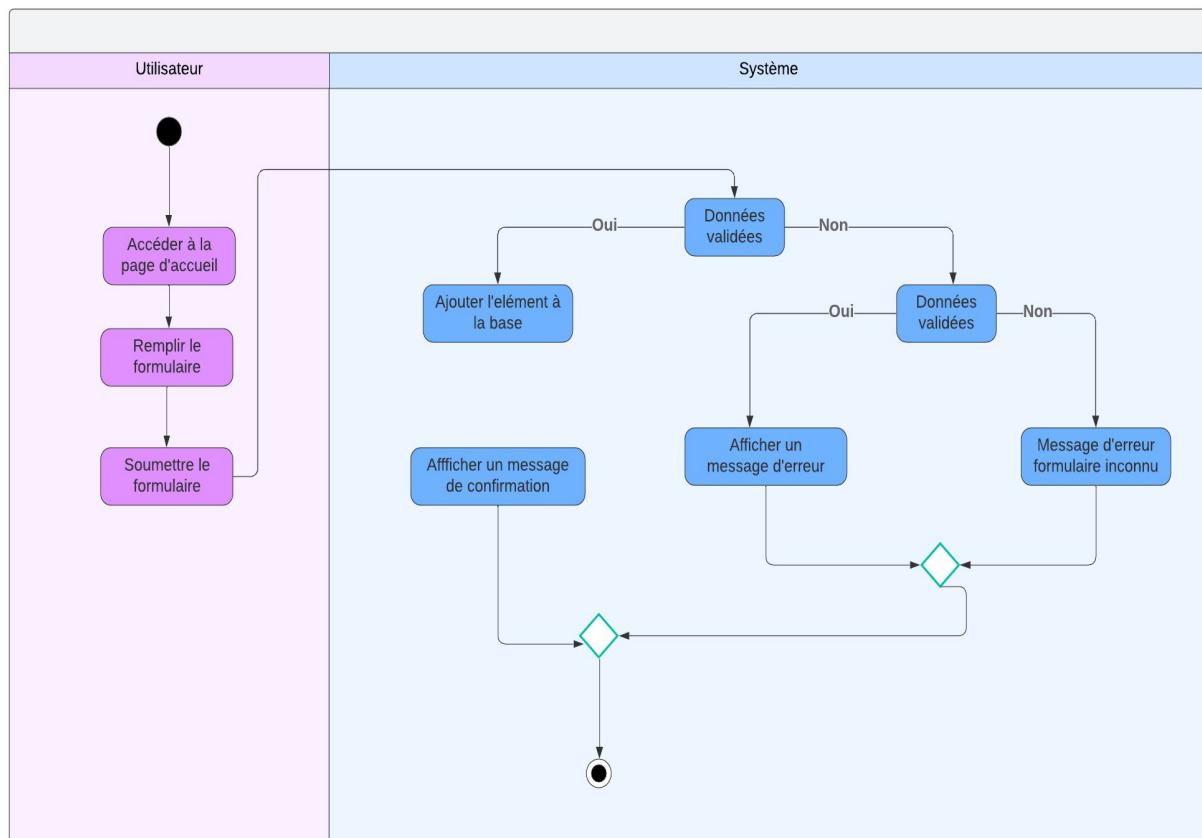


Figure 6-7 Diagramme d'activité d'ajout d'élément(employé,département,poste)

6.5.4. Étude du cas d'utilisation : « Ajout Demande absence d'un employé »

6.5.4.1.. Description du cas d'utilisation

Titre : Ajouter une demande d'absence

Objectif : Permettre au responsable RH d'enregistrer une nouvelle demande d'absence pour un employé, tout en respectant les règles de gestion.

Résumé : Le responsable RH soumet une demande d'absence via un formulaire, en respectant les règles de l'entreprise.

Acteur : Responsable RH, administrateur.

Pré-condition : Le responsable RH est connecté et accède à la page d'ajout de demande d'absence.

Post-condition : La demande est ajoutée à la base de données, le solde de l'employé est mis à jour, et les règles de gestion sont respectées.

Scénario nominal :

1. Le responsable RH accède à la page d'ajout de demande d'absence.
2. Il sélectionne l'employé et remplit le formulaire (type d'absence, dates, motif, etc.).
3. Le système vérifie :
 - Si le solde d'absence est suffisant.
 - Si la demande respecte le plafonnement mensuel (ex. : maximum de 10 jours pour les absences régulières).
 - Si toutes les règles sont respectées, la demande est soumise et le solde de l'employé est mis à jour.
4. Le système confirme l'ajout de la demande.
5. Imprimer la demande

Scénario alternatif :

3a. Plafonnement dépassé : Si la demande dépasse le plafonnement mensuel, le système affiche un message d'erreur et bloque la soumission.

4a. Solde insuffisant : Si le solde est insuffisant, le système affiche une erreur indiquant le solde insuffisant.

5a. Absence spéciale : Pour une absence spéciale, le système calcule automatiquement la durée et ajuste les dates. La demande est acceptée même si elle dépasse le plafonnement.

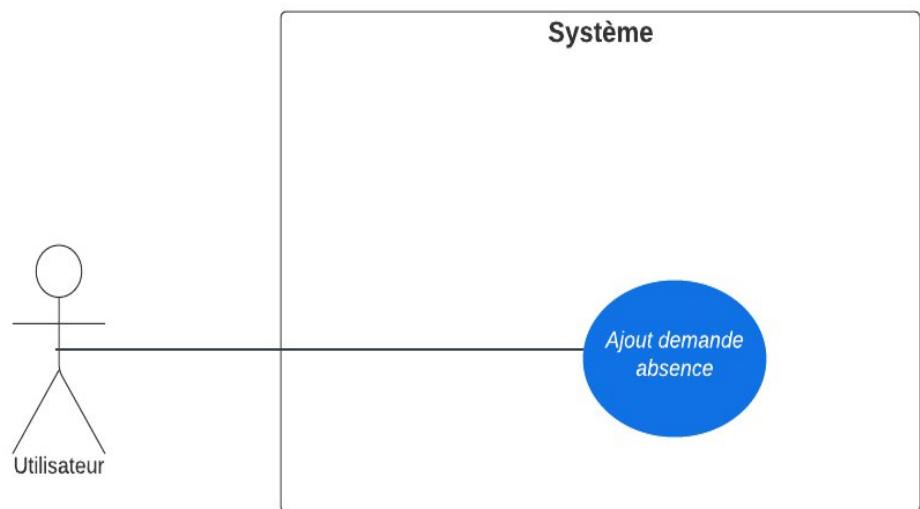


Figure 6-8 Diagramme de cas d'utilisation d'ajout demande d'absence

6.5.4.2. Diagramme de séquence de ajout absence d'un employé

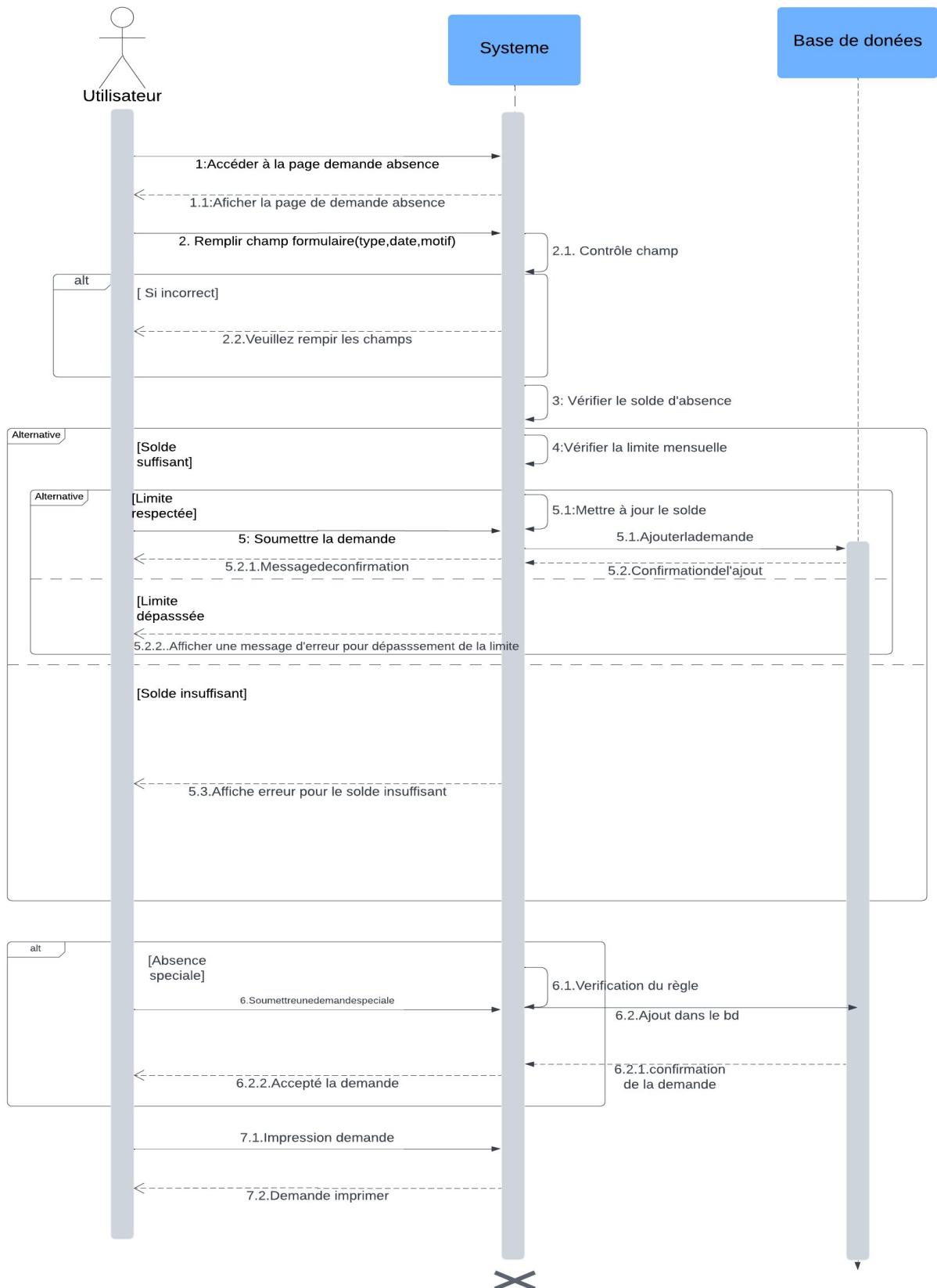


Figure 6-9 Diagramme de séquence d'ajout de demande d'absence

Description du scénario

N° Enchaînement	Action
1	Accéder à la page demande absence
1.1	Affichage de la page demande absence
2	Remplissage du champ formulaire
2.1	Contrôle champ
2.2	Message qu'il doit remplir les champs
3	Vérification du solde absence
4	Vérification du limite mensuelle
5	Soumission de la demande
5.1	Mise à jour du solde
5.1	Ajout de la demande
5.2	Confirmation de l'ajout
5.2.1	Message de confirmation
5.2.2	Affichage d'une message d 'erreur pour le dépassement de la limite
5.3	Affichage d'erreur pour le solde insuffisant
6	Soumettre une demande spéciale
6.1	Vérification du règle
6.2	Ajout à la base de donnée
6.2.1	Confirmation de la demande
6.2.2	Acceptation de la demande
7.1	Impression de la demande
7.2	Demande imprimer

Tableau 6-5 Description de scénario d'ajout de demande d'absence

6.5.4.3. Diagramme de d'activité de ajout absence d'un employé

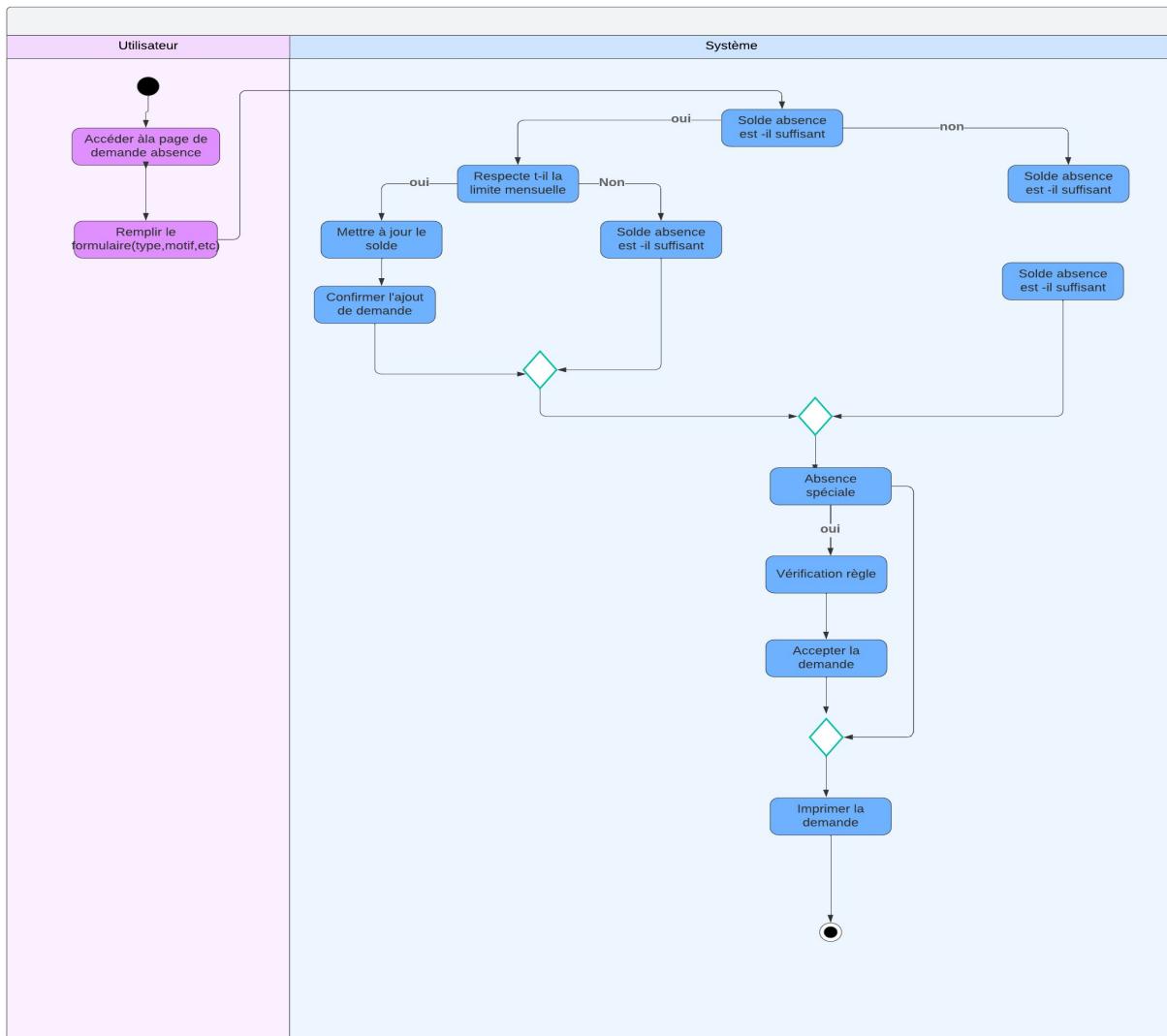


Figure 6-10 Diagramme d'activité d'ajout de demande d'absence

6.5.5. Étude du cas d'utilisation : Modification

6.5.5.1. Description du cas d'utilisation

Titre : Modifier un Élément

Objectif : Mettre à jour les informations d'un employé, d'un département ou d'un poste.

Résumé : L'administrateur modifie les informations et soumet le formulaire.

Acteur : Administrateur.

Pré-condition : L'utilisateur est connecté et accède à la page de modification.

Post-condition : Les informations sont mises à jour dans la base de données.

Scénario nominal :

1. L'utilisateur accède à la page de modification.
2. Il sélectionne l'élément à modifier.
3. Il met à jour les informations dans le formulaire.
4. Il soumet le formulaire.
5. L'application vérifie les informations.
6. Si valides, les informations sont mises à jour.
7. Un message de confirmation s'affiche.

Scénarios alternatifs :

- 4a. Élément non trouvé : Message d'erreur affiché.
- 4b. Formulaire incomplet : Message d'erreur demandé pour compléter les champs.

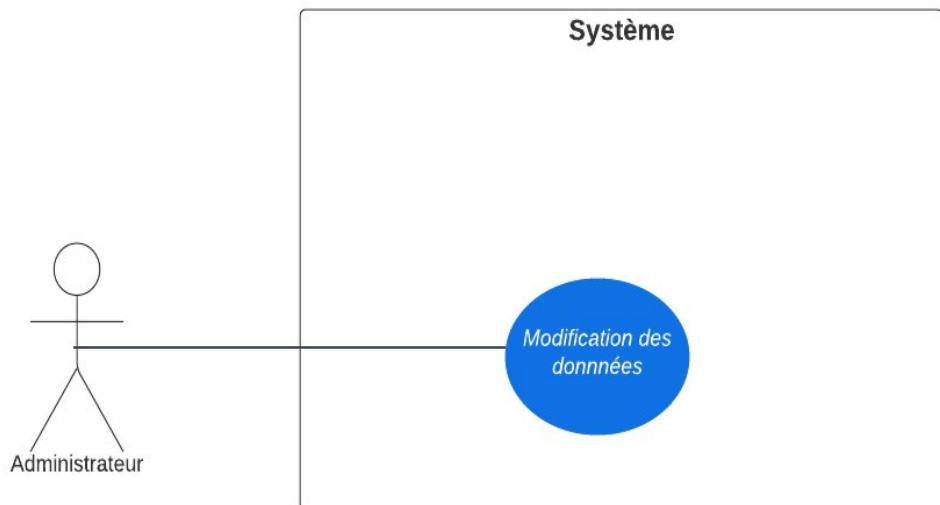


Figure 6-11 Diagramme de cas d'utilisation de modification

6.5.5.2. Diagramme de séquence de modification

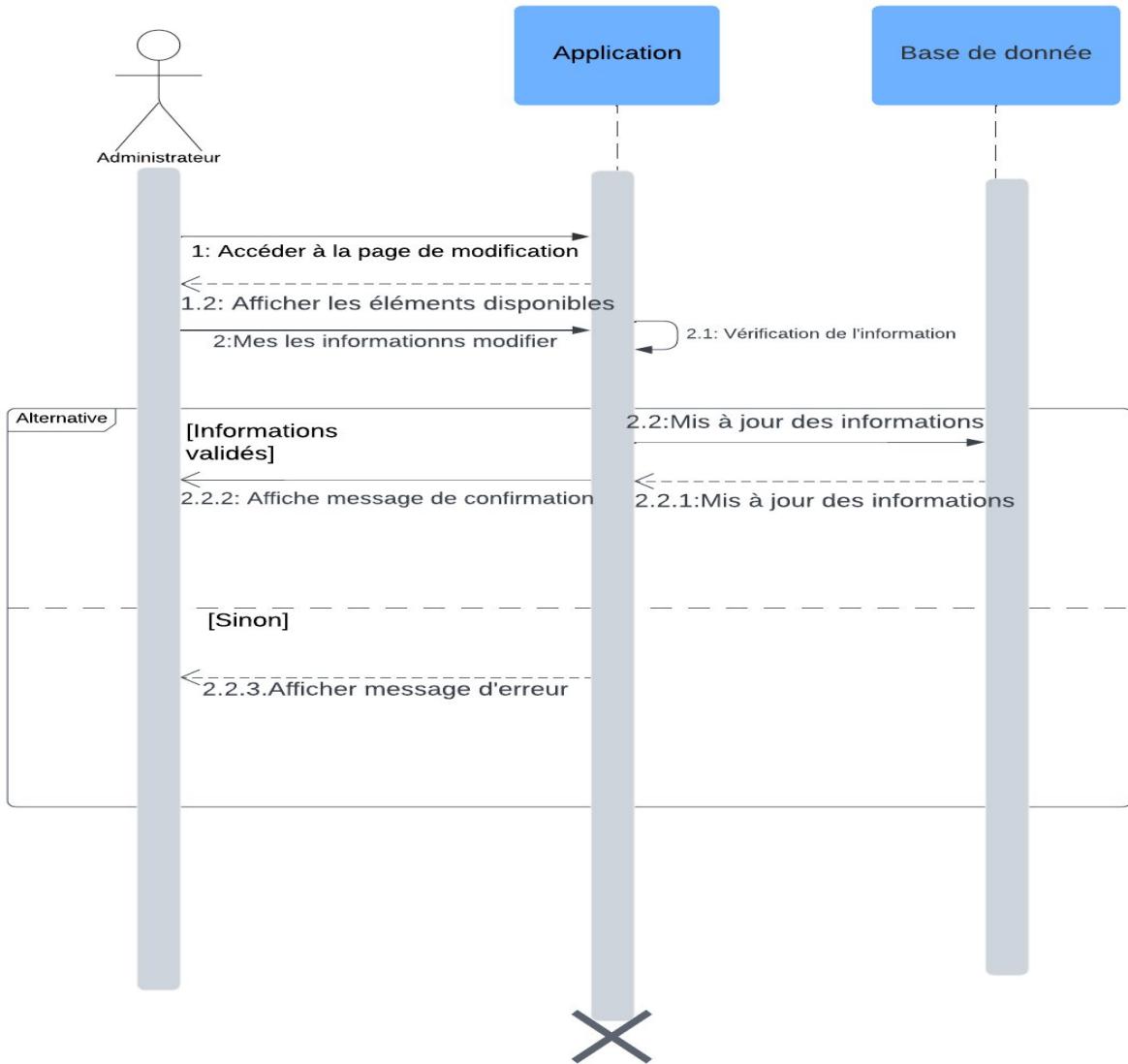


Figure 6-12 Diagramme de séquence de modification

Description du scénario

N° Enchaînement	Action
1	Accéder à la page de modification
1.2	Affichage des éléments
2	Les informations à modifier
2.1	Vérification de l'information
2.2	Mis à jour des informations
2.2.1	Confirmation de modification
2.2.2	Message de confirmation
2.2.3	Message d'erreur

Tableau 6-5 Description du scénario de modification

6.5.5.2. Diagramme d'activité de modification

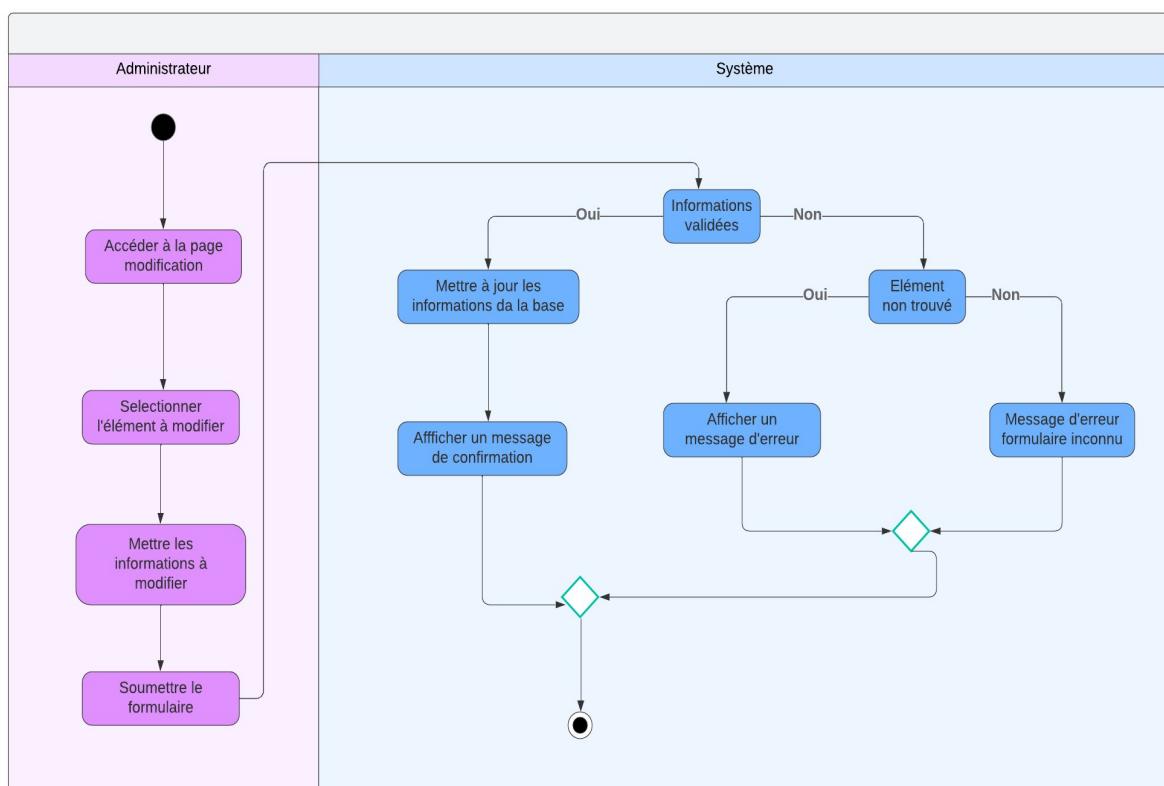


Figure 6-13 Diagramme d'activité de modification

6.5.6. Étude de cas d'utilisation : Suppression

6.5.6.1. Description de cas d'utilisation

Titre : Supprimer un Élément

Objectif : Retirer un élément de l'application.

Résumé : L'administrateur sélectionne et confirme la suppression de l'élément.

Acteur : Administrateur

Pré-condition : L'utilisateur est connecté.

Post-condition : L'élément est supprimé.

Scénario nominal :

L'utilisateur accède à la gestion des éléments.

1. Il sélectionne l'élément à supprimer.
2. Une confirmation s'affiche.
3. L'utilisateur confirme la suppression.
4. L'élément est supprimé de la base de données.
5. Un message de confirmation apparaît.

Scénarios alternatifs :

- 3a. Annulation : L'utilisateur annule la suppression.
- 5a. Erreur : Un message d'erreur s'affiche si la suppression échoue.

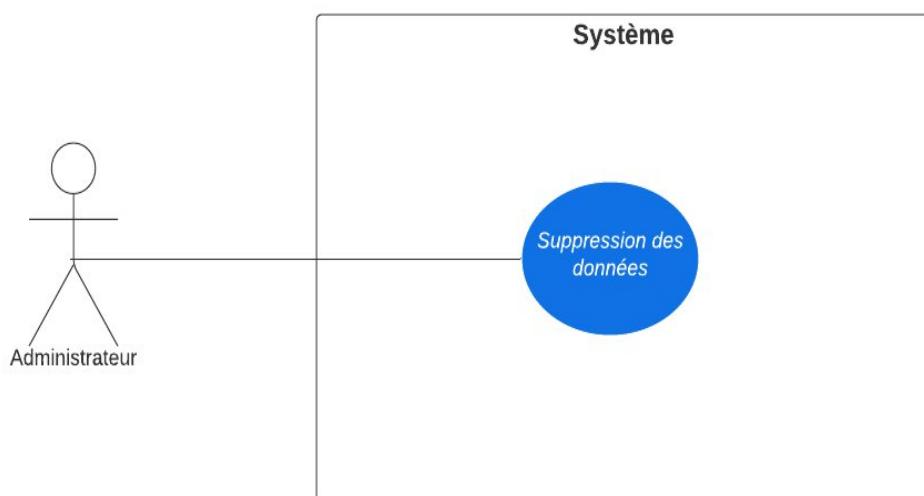


Figure 6-14 Diagramme de cas d'utilisation de suppression

6.5.6.2. Diagramme de séquence de suppression

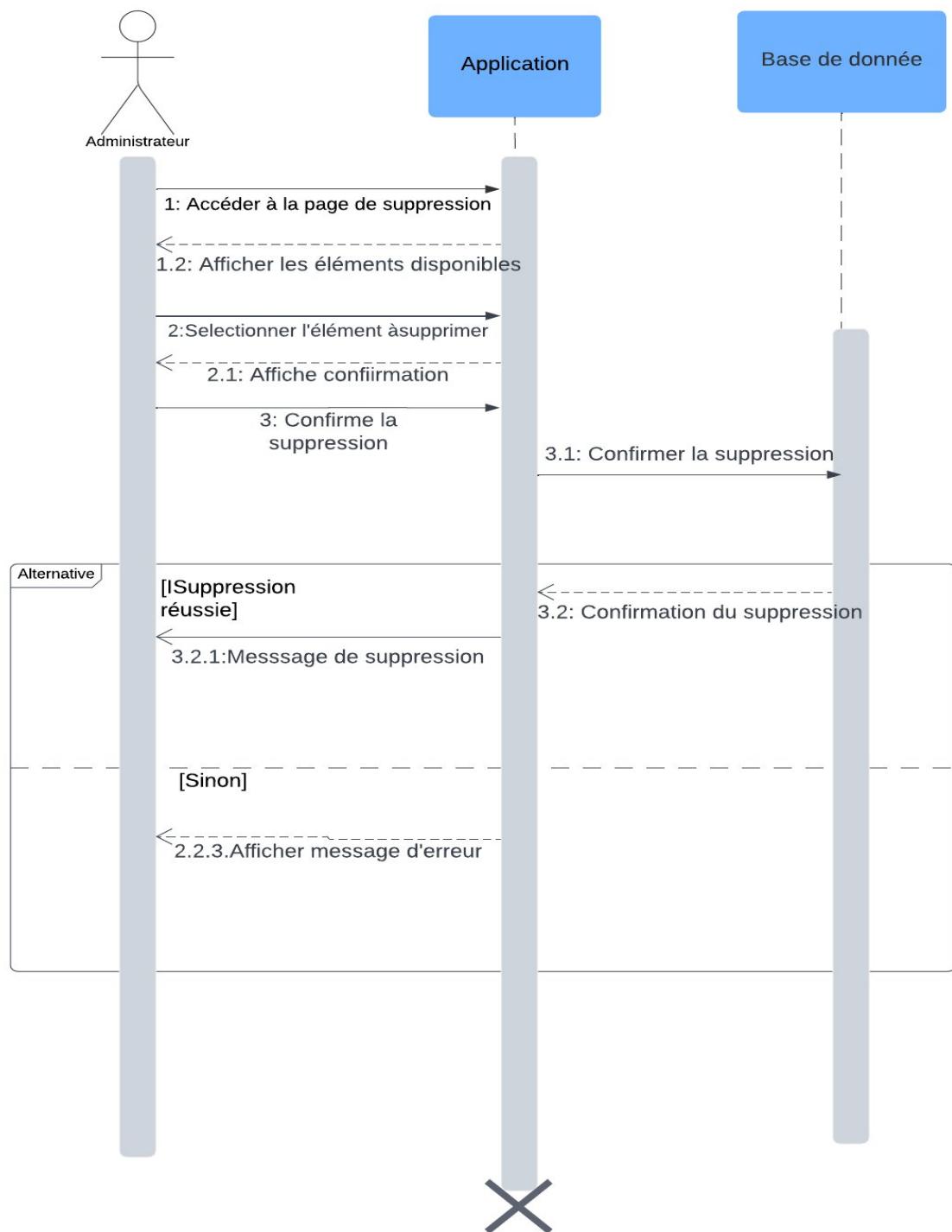


Figure 6-15 Diagramme de séquence de suppression

Description de scénario

N° Enchaînement	Action
1	Accéder à la page de suppression
1.2	Affichage des éléments disponibles
2	Selectionner l'élément à supprimer
2.1	Affichage du confirmation
3	Confirme la suppression
3.1	Confirmer du suppression
3.2	Confirmation du suppression
3.2.1	Message de suppression
3.2.3	Message d'erreur

Tableau 6-6 Description de scénario de diagramme de séquence de suppression

6.5.5.3 Diagramme d'activité de suppression

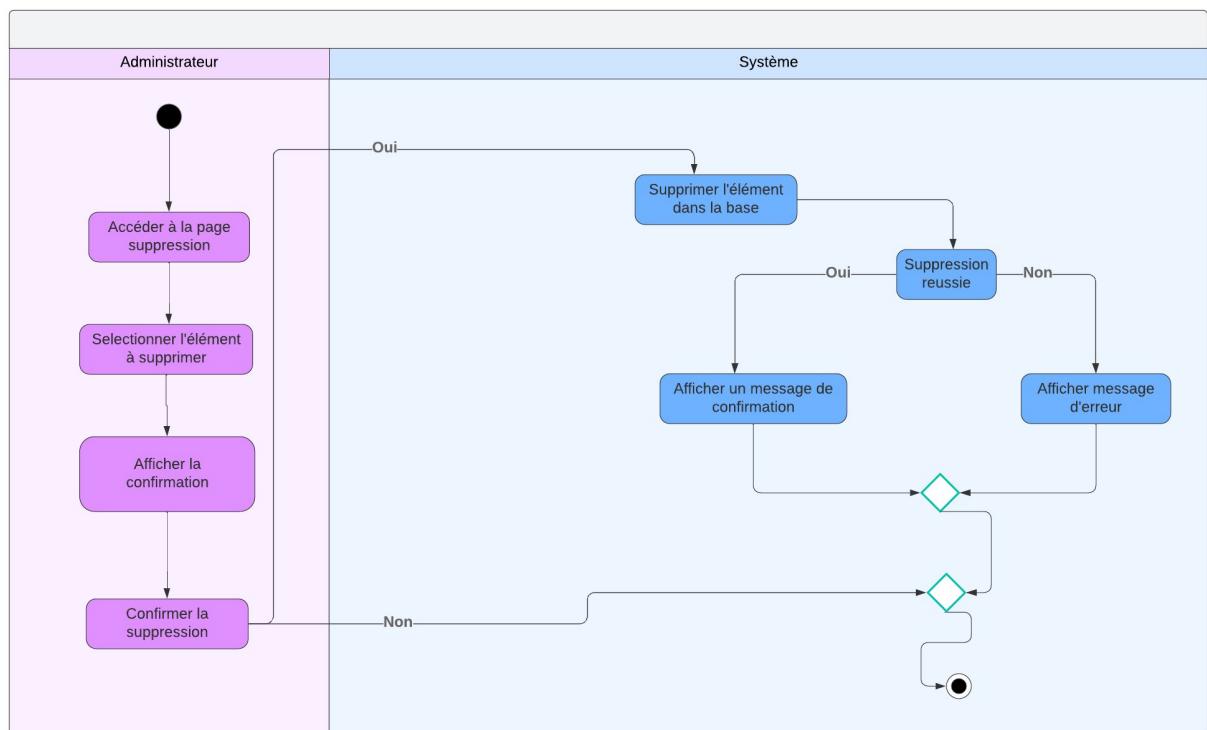


Figure 6-16 Diagramme d'activité de suppression

6.5.7. Diagramme de classe en globale

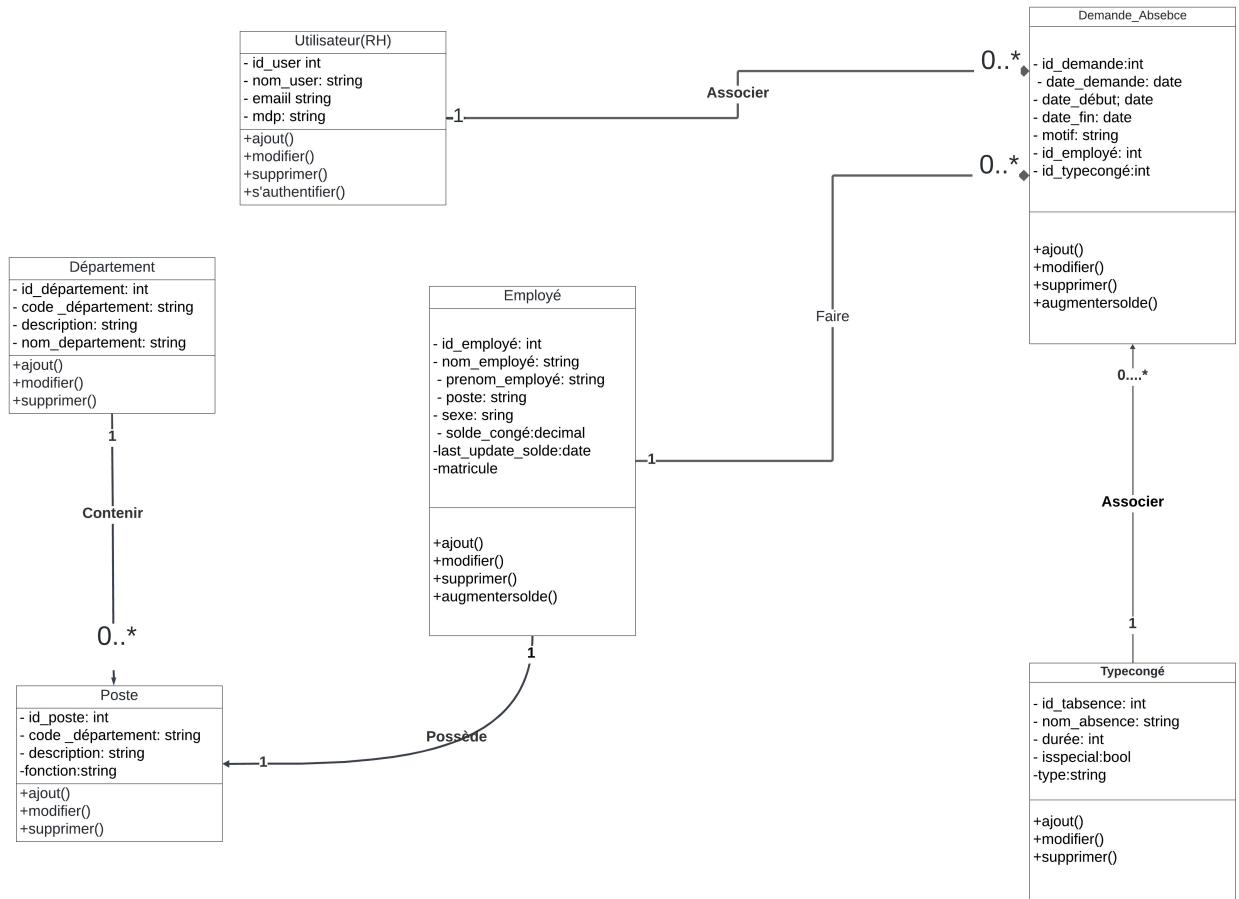


Figure 6-17 Diagramme de classe en globale

Partie III : Réalisation du projet

Chapitre 7 Spécification des outils de réalisation

7.1. Le système de gestion de base de données

Un logiciel permettant d'interagir avec une base de données est appelé un Système de Gestion de Base de Données (SGBD). La gestion d'une base de données consiste à contrôler et organiser les données stockées sous forme de tables. Pour cela, il est nécessaire de choisir un SGBD qui facilite l'accès aux données, autorise l'accès simultané à plusieurs utilisateurs, et permet la manipulation de ces données, qu'il s'agisse de leur insertion, modification ou suppression.

7.1.1. Définition d'une base de données

Une base de données est une entité qui permet de stocker des informations de manière structurée tout en minimisant la redondance. Ces données doivent être accessibles et exploitables par différents programmes et utilisateurs. Ainsi, la notion de base de données est souvent associée à celle de réseau, car elle permet de centraliser et de partager ces informations, ce qui explique l'origine du terme "base". On parle généralement de système d'information pour désigner l'ensemble des moyens mis en place afin de faciliter le partage et la gestion des données.

7.1.2. Utilité d'une base de données

Une base de données permet de centraliser des informations dans un même enregistrement, ce qui est particulièrement utile à mesure que les volumes de données augmentent. Elle peut être locale, c'est-à-dire accessible par un seul utilisateur sur une machine, ou répartie, lorsque les données sont stockées sur des serveurs distants et accessibles via un réseau. L'un des principaux avantages d'une base de données est la possibilité pour plusieurs utilisateurs d'y accéder simultanément, facilitant ainsi la collaboration et la gestion partagée des informations.

7.2. Caractéristiques des SGBD

L'architecture à trois niveaux, définie par la norme ANSI/SPARC, permet d'assurer une indépendance entre les données et les traitements. De manière générale, un SGBD doit posséder les caractéristiques suivantes :

- Indépendance physique : Le niveau physique peut être modifié indépendamment du niveau conceptuel. Cela signifie que les détails matériels de la base de données sont transparents pour l'utilisateur, qui n'a accès qu'à une représentation structurée des informations.

- Indépendance logique : Le niveau conceptuel doit pouvoir être modifié sans affecter le niveau physique. L'administrateur de la base doit pouvoir la faire évoluer sans perturber les utilisateurs ni impacter les performances de la base de données.
- Rapidité d'accès : Le système doit être capable de fournir des réponses aux requêtes de manière rapide, ce qui nécessite l'utilisation d'algorithmes de recherche optimisés.
- Administration centralisée : Le SGBD doit minimiser les redondances d'informations afin d'éviter à la fois le gaspillage de l'espace mémoire et les risques d'erreurs de données.
- Sécurisation des données : Le SGBD doit inclure des mécanismes de gestion des droits d'accès pour garantir que seules les personnes autorisées puissent consulter ou manipuler les données.

7.3. Points forts et faibles des SGBD

Prenons en considération quatre SGBD parmi les plus couramment utilisés : MS Access, MySQL, PostgreSQL, et Oracle Database. Leurs avantages et inconvénients respectifs sont résumés dans le tableau 7.1

SGBD	Avantages	Inconvénients
MS Access	<ul style="list-style-type: none"> - Facile à utiliser pour les débutants et petites applications - Interface graphique intuitive 	<ul style="list-style-type: none"> - Limité en termes de gestion de grandes bases de données - Faible évolutivité et performance en réseau
MySQL	<ul style="list-style-type: none"> - Open-source et gratuit - Supporte des bases de données volumineuses - Bonne rapidité 	<ul style="list-style-type: none"> - Manque certaines fonctionnalités avancées (transactions complexes, etc.) comparé à PostgreSQL et Oracle
PostgreSQL	<ul style="list-style-type: none"> - Open-source et gratuit - Très robuste et riche en fonctionnalités avancées (transactions ACID) 	<ul style="list-style-type: none"> - Configuration et optimisation plus complexes pour les utilisateurs débutants
Oracle DB	<ul style="list-style-type: none"> - Très puissant et évolutif - Excellentes performances pour les grandes bases de données - Fonctionnalités avancées de sécurité et gestion 	<ul style="list-style-type: none"> - Coût élevé des licences - Complexité d'installation et de gestion, nécessite une expertise avancée

Tableau 7-1Avantages et inconvénients des quatre SGBD

7.4. Présentation de Node.js et Express.js

7.4.1. Introduction

Node.js est une plateforme open-source qui permet d'exécuter du JavaScript côté serveur, construite sur le moteur V8 de Google Chrome. L'architecture de Node.js repose sur un modèle d'E/S non-bloquant, ce qui lui confère une grande capacité à gérer des connexions simultanées avec une faible latence. Cela en fait un choix populaire pour le développement d'applications web en temps réel, telles que les applications de chat, les jeux en ligne et les systèmes de notification.

Express.js, créé pour faciliter le développement d'applications web et d'API sur Node.js, est un framework minimaliste mais puissant. Il permet aux développeurs de créer des applications rapidement en fournissant une série de fonctionnalités robustes pour la gestion des routes, des

middlewares et des requêtes HTTP. Grâce à sa flexibilité, Express.js permet de concevoir des architectures sur mesure, adaptées aux besoins spécifiques de chaque projet.

7.4.2. Avantages de Node.js et Express.js

- **Performance élevée** : Grâce à son modèle non-bloquant et asynchrone, Node.js est idéal pour gérer un grand nombre de requêtes simultanées.
- **Utilisation de JavaScript** : Permet aux développeurs d'utiliser JavaScript à la fois côté client et serveur, facilitant le développement et la collaboration.
- **Écosystème riche** : NPM offre une vaste bibliothèque de modules et de packages, accélérant le développement.
- **Flexibilité** : Express.js est minimaliste et permet une grande flexibilité dans la structure et l'architecture des applications.
- **Support communautaire** : Une communauté active offre de nombreuses ressources, bibliothèques et solutions aux problèmes rencontrés.

7.4.3. Inconvénients de Node.js et Express.js

- **Complexité du code asynchrone** : Le modèle asynchrone peut rendre le code difficile à lire et à maintenir, entraînant des problèmes de "callback hell".
- **Performance pour les tâches CPU-intensive** : Pas idéal pour les opérations lourdes en CPU, pouvant bloquer le thread principal.
- **Écosystème immature** : Certaines bibliothèques sur NPM peuvent être mal maintenues, soulevant des problèmes de sécurité.
- **Sécurité** : Nécessite une attention particulière à la sécurité, car il n'impose pas de mécanismes par défaut.
- **Limites de scalabilité** : Difficulté à gérer l'état dans des applications nécessitant une forte scalabilité, avec des solutions externes nécessaires.

7.4.4. Comparaison avec Django

Critères	Node.js/Express.js	Django
Langage	JavaScript	Python
Type	Framework minimaliste et flexible	Framework complet avec de nombreuses fonctionnalités
Architecture	Basé sur des middlewares et des routes, offre flexibilité	Architecture basée sur le modèle MVC (Modèle-Vue-Contrôleur)
Performance	Haute performance pour les applications en temps réel	Bonne performance, mais peut être plus lent pour les E/S
Écosystème	NPM avec une multitude de bibliothèques	PIP avec une grande variété de paquets Python
Base de données	Support pour plusieurs bases de données via des ORM comme Sequelize	ORM intégré avec une gestion facile des bases de données
Facilité d'apprentissage	Peut être plus simple pour les développeurs JavaScript	Facile à apprendre pour ceux qui connaissent déjà Python
Scalabilité	Très scalable grâce à son modèle non-bloquant	Scalable, mais peut nécessiter des ajustements supplémentaires pour des applications très lourdes
Sécurité	Nécessite une configuration manuelle pour certaines pratiques de sécurité	Offre des fonctionnalités de sécurité prêtes à l'emploi, comme la protection CSRF
Utilisation	Idéal pour les applications en temps réel et les APIs légères	Excellent option pour les applications web complètes avec une logique métier

Tableau 7-2Comparaison de Nodejs Express et Django

7.5. Présentation du framework VueJs

7.5.1. Introduction

Vue.js est un framework JavaScript progressif, développé par Evan You et lancé en 2014, qui s'est rapidement imposé comme l'un des outils préférés des développeurs pour la création d'interfaces utilisateur et d'applications web monopages (SPA). À la différence des frameworks

traditionnels, Vue.js adopte une approche modulaire et réactive, ce qui signifie qu'il permet aux développeurs de construire des composants réutilisables qui interagissent de manière fluide avec le modèle de données.

Vue.js se distingue par sa courbe d'apprentissage relativement douce, ce qui en fait une option attrayante pour les débutants tout en restant suffisamment robuste pour les développeurs expérimentés. Son architecture flexible permet d'intégrer progressivement Vue dans des projets existants, ce qui en fait un choix idéal pour les équipes cherchant à moderniser leurs applications sans une refonte complète.

En outre, la communauté croissante de Vue.js a contribué à un écosystème riche, avec de nombreuses bibliothèques et outils complémentaires qui facilitent le développement. Que ce soit pour la gestion de l'état avec Vuex, la navigation avec Vue Router ou l'intégration d'animations avec Vue Transition, Vue.js offre une gamme de solutions qui permettent de répondre aux besoins divers des projets modernes.

7.5.2. Avantages de Vue.js

- **Simplicité et facilité d'apprentissage** : Vue.js possède une syntaxe claire et intuitive qui permet aux développeurs, même aux débutants, de comprendre rapidement les concepts de base et de commencer à créer des applications. La documentation officielle est bien structurée et facile à suivre, offrant un excellent point de départ.
- **Réactivité** : Le système de réactivité de Vue.js permet de lier facilement le modèle de données à l'interface utilisateur. Les changements dans les données se reflètent automatiquement dans l'interface, ce qui réduit le besoin d'écrire du code de manipulation du DOM et rend le développement plus efficace.
- **Modularité et réutilisation des composants** : Vue.js repose sur une architecture de composants, ce qui signifie que les développeurs peuvent créer des éléments encapsulés et réutilisables. Cela favorise la modularité, simplifie la maintenance du code et encourage les bonnes pratiques de développement.
- **Écosystème riche** : Vue.js dispose d'un écosystème florissant avec des outils et des bibliothèques complémentaires, tels que Vue Router pour la gestion de la navigation et Vuex pour la gestion de l'état centralisé. Ces outils facilitent le développement d'applications complexes tout en maintenant la simplicité.

- **Performance** : Vue.js utilise un moteur de rendu optimisé qui garantit d'excellentes performances, même pour des applications de grande envergure. Sa capacité à effectuer des mises à jour rapides et efficaces contribue à une expérience utilisateur fluide.

7.5.3. Inconvénients de Vue.js

- **Manque de fonctionnalités par défaut** : Bien que Vue.js soit un framework flexible, il peut manquer de certaines fonctionnalités intégrées que l'on trouve dans des frameworks plus complets comme Angular. Cela signifie que les développeurs doivent parfois intégrer manuellement des outils pour gérer des fonctionnalités avancées.
- **Communauté et ressources** : Bien que la communauté de Vue.js soit en croissance, elle reste moins étendue que celles de frameworks comme React ou Angular. Par conséquent, il peut être plus difficile de trouver des ressources, des plugins ou des réponses à des questions spécifiques.
- **Flexibilité excessive** : La flexibilité de Vue.js, bien que bénéfique, peut également conduire à des incohérences architecturales si les conventions de codage ne sont pas respectées. Les développeurs doivent être vigilants pour maintenir une structure de projet cohérente.
- **Difficulté de tests** : Tester des composants complexes dans Vue.js peut parfois s'avérer difficile, surtout si les développeurs ne sont pas familiers avec les outils de test disponibles. Cela nécessite souvent une courbe d'apprentissage supplémentaire pour mettre en place des tests efficaces.

7.5.4. Comparaison avec d'autres frameworks

Caractéristiques	Vuejs	React	Angular
Type	Framework progressive	Bibliothèque UI	Framework complet
Apprentissage	Facile	Modéré	Difficile
Modularité	Composants	Composants	Composants
Écosystème	Riche	Très riche	Très riche
Routing intégré	Oui	Non	Oui
Gestion d'état	Avec Vuex	Avec Redux ou Context API	Avec NgRx
Performance	Excellente	Excellente	Bonne
Utilisation	SPA, projets de petite à moyenne taille	SPA, projets de toutes tailles	Applications complexes

Tableau 7-3Comparaison de Vuejs avec d'autres frameworks

7.6. Présentation des outils de développement

Pour le développement de l'application de gestion des absences, plusieurs outils ont été utilisés afin de faciliter le processus de développement, de gestion des versions et de déploiement.

7.6.1. Visual Studio Code (VSCode)

Visual Studio Code (VSCode) est un éditeur de code source léger mais puissant, développé par Microsoft. Il prend en charge plusieurs langages de programmation grâce à ses extensions, et intègre des fonctionnalités comme l'autocomplétion, le débogage intégré et le contrôle Git. Il est particulièrement utile pour les développeurs front-end et back-end avec une prise en charge efficace des frameworks modernes comme Vue.js et Node.js.

Dans ce projet, VSCode a été utilisé pour écrire, organiser et gérer tout le code source de l'application, incluant les fichiers frontend et backend. L'intégration avec Git permet également de suivre facilement l'évolution du projet.

7.6.2. Docker

Docker est une plateforme de conteneurisation qui permet d'exécuter des applications dans des environnements isolés appelés conteneurs. Dans ce projet, Docker est utilisé pour héberger la base de données MySQL, assurant ainsi une gestion efficace des données sans avoir à installer MySQL directement sur la machine de développement.

Cette approche garantit que la base de données est toujours dans un état prévisible, ce qui facilite le développement et la collaboration. Grâce à Docker, il est possible de déployer l'application et la base de données facilement sur n'importe quel environnement.

7.6.3. Postman

Postman est un outil utilisé pour tester les API en envoyant des requêtes HTTP vers un serveur. Il permet de simuler et d'automatiser des requêtes RESTful comme les requêtes GET, POST, PUT, et DELETE, facilitant ainsi les tests de l'API du projet. Dans ce projet, Postman a été utilisé pour tester les différents endpoints de l'application backend, développée avec Node.js et Express. Il permet de vérifier que les données sont bien envoyées et reçues par le serveur, de tester la validation des formulaires de demande de congé, et d'assurer le bon fonctionnement des routes associées aux fonctionnalités de l'application, comme la gestion des employés et des absences.

7.6.4. Git

Git est un système de contrôle de version distribué, largement utilisé pour suivre les modifications dans le code source au cours du développement. Il permet de travailler sur plusieurs fonctionnalités en parallèle, de revenir à des versions antérieures du projet si nécessaire, et de collaborer avec d'autres développeurs de manière efficace.

Dans ce projet, Git a été utilisé pour gérer les différentes versions du code. Chaque fonctionnalité ou correctif est développé sur une branche distincte, ce qui permet de gérer le projet de manière organisée et d'éviter les conflits. De plus, en utilisant Git, il est possible de partager facilement le code avec d'autres collaborateurs et de suivre les contributions.

Chapitre 8 : Mise en œuvre et implémentation

Dans la mise en œuvre, nous allons parler de l'architecture de l'application, qui est la manière dont l'application interagit avec les composants matériels et logiciels, elle décrit d'une manière symbolique et schématique les différents composants d'un ou plusieurs systèmes informatiques, elle se subdivise en 2 parties : l'architecture matérielle et l'architecture logicielle. Ensuite, nous allons voir quelques extraits de code important de notre application.

8.1. Architecture Logicielle

L'architecture logicielle de l'application de gestion des absences est organisée autour d'un modèle client-serveur qui divise clairement les responsabilités entre le frontend et le backend.

8.1.1. Frontend (Vue.js)

Le frontend est développé en utilisant Vue.js, un framework JavaScript progressif qui permet de créer des interfaces utilisateur interactives et réactives. Les principales caractéristiques de l'architecture frontend incluent :

- **Composants Réutilisables** : L'application est construite à partir de composants modulaires, ce qui facilite la réutilisation du code et améliore la maintenabilité.
- **Gestion des État** : L'état de l'application, tel que les soldes d'absence des employés, est géré de manière centralisée, permettant des mises à jour dynamiques de l'interface utilisateur en fonction des actions de l'utilisateur.
- **API RESTful** : Le frontend communique avec le backend via des requêtes HTTP pour récupérer et envoyer des données. Cette interaction se fait généralement en utilisant Axios, un client HTTP basé sur des promesses.

8.1.2. Backend (Node.js avec Express)

Le backend est développé avec Node.js et le framework Express, ce qui permet de créer une API RESTful robuste et efficace. Les principales caractéristiques incluent :

- **Gestion des Routes** : Express permet de définir des routes pour les différentes opérations de l'application, comme la création, la lecture, la mise à jour et la suppression des demandes d'absence.

- **Middleware** : L'utilisation de middlewares pour la gestion des requêtes et l'authentification assure une architecture modulaire, permettant d'ajouter facilement des fonctionnalités telles que la validation des entrées et la protection des routes.
- **Sécurité** : Des mécanismes d'authentification, comme JSON Web Tokens (JWT), sont mis en place pour sécuriser les sessions des utilisateurs et protéger les données sensibles.

8.1.3. Base de Données (MySQL)

La base de données MySQL est utilisée pour stocker les données de l'application. Les principales caractéristiques comprennent :

- **Tables Structurées** : Les données sont organisées dans des tables qui définissent les relations entre les employés, les demandes d'absence, et les types de congés.
- **ORM Sequelize** : L'utilisation d'un ORM comme Sequelize facilite l'interaction avec la base de données, permettant des opérations CRUD simplifiées tout en assurant la sécurité et l'intégrité des données.

8.2. Architecture Matérielle

L'architecture matérielle de l'application repose sur un serveur, généralement un serveur cloud, qui offre plusieurs avantages :

- **Scalabilité** : Le serveur cloud permet de dimensionner les ressources en fonction des besoins, assurant ainsi que l'application peut gérer une augmentation du nombre d'utilisateurs sans compromettre les performances.
- **Disponibilité** : En utilisant un fournisseur de services cloud, l'application bénéficie d'une infrastructure fiable, garantissant une haute disponibilité et une continuité de service.

8.3. Extrait de code

Cet extrait de code est un modèle de l'entité « employé » dans le backend de l'application,

```
models/employe.js [8] Employe > last_sold_update
1 const { DataTypes } = require('sequelize');
2 const DB = require('../db');
3 const sequelize = require('../db');
4
5 /**
6  * Definition des models
7  */
8 const Employe = DB.define('Employe', {
9     id_employe: {
10         type: DataTypes.INTEGER(10),
11         primaryKey: true,
12         autoIncrement: true,
13         allowNull: false
14     },
15     nom_employe: {
16         type: DataTypes.STRING(255),
17         allowNull: false
18     },
19     pre_employe: {
20         type: DataTypes.STRING(255),
21         allowNull: true
22     },
23     matricule: {
24         type: DataTypes.STRING(255),
25         allowNull: false,
26         unique: true
27     },
28     sexe: {
29         type: DataTypes.STRING(50),
30         allowNull: false,
31     },
32     motif_employe: {
33         type: DataTypes.STRING(100),
34         allowNull: false
35     },
36     solde_employe: {
37         type: DataTypes.FLOAT,
38         allowNull: true,
39         defaultValue: 0.0
40     },
41     plafonnement: {
42         type: DataTypes.FLOAT,
43         allowNull: true,
44     },
45     plafonnementboolean: {
46         type: DataTypes.BOOLEAN,
47         allowNull: true,
48     },
49     last_solde_update: [
50         type: DataTypes.DATE,
51         allowNull: true
52     ]
53 });
54 sequelize.sync()
55 module.exports = Employe;
```

Figure 8-1 Modèle de l'entité employé

L'extrait suivant présente la route /login permet aux utilisateurs de se connecter à l'application en saisissant leur adresse email et leur mot de passe.

```
//login des admins et utilisateurs
router.post('/Login', (req, res) => {
  const { email, mdp } = req.body;

  if (!email || !mdp) {
    return res.status(400).json({ message: 'Email ou mot de passe manqu' });
  }

  Utilisateur.findOne({ where: { email: email }, raw: true })
    .then(user => {
      if (!user) {
        return res.status(401).json({ message: 'Utilisateur non trouvé' });
      }

      return bcrypt.compare(mdp, user.mdp_user).then(motdepasse => {
        if (!motdepasse) {
          return res.status(401).json({ message: 'Mot de passe incorrect' });
        }

        const token = jwt.sign(
          {
            id_user: user.id_user,
            nom_user: user.nom_user,
            email: user.email,
            role: user.role,
          },
          process.env.JWT_SECRET, { expiresIn: process.env.JWT_EXPIRES_IN }
        );
        return res.status(200).json({
          message: 'Login réussi',
          access_token: token,
        });
      });
    })
    .catch(error => {
      res.status(500).json({ message: 'Erreur serveur', error: error.message });
    });
});
```

Figure 8-2 Route /login pour connecter à l'application

Ce code configure un serveur Express pour une API RESTful, permettant la gestion des employés, des absences et des demandes au sein d'une application, tout en assurant la connexion à la base de données et le routage des requêtes.

```
require('dotenv').config();
// Importation des modules
const express = require('express');
const cors = require('cors');
// Import de la connexion bd
const db = require('./db');
const Employe = require('./models/employe'); // Importer le modèle Employe
const Absence = require('./models/absence');//Importation modèle Absence
const Demande = require('./models/demande');//Importation modèle demande
const Utilisateur = require('./models/utilisateur');//Importation modèle
//initialisation de l'api
const app = express();
const app = express();
app.use(cors());
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
//Importation des routes
const employe_router = require('./routes/employes');//importation route employe
const demande_router = require('./routes/demandes');
const absence_router = require('./routes/absences');
const utilisateur_router = require('./routes/utilisateurs');
const dashboard_router = require('./routes/dashboard');

//Mise en place du routage
app.use('/employes', employe_router); //les employés
app.use('/demandes', demande_router);
app.use('/absences', absence_router);
app.use('/utile', utilisateur_router);
app.use('/dashboard', dashboard_router);
//Demarrer serveur avec test bd
(async () => {
  try {
    await db.authenticate();
    console.log('Database connection ok');
    await db.sync({ alter: true });
    console.log('Database synchronized and updated');

    app.listen(process.env.server_port, () => {
      console.log(`Server running on port ${process.env.server_port}`);
    });
  } catch (err) {
    console.log('Database error', err);
  }
})();

```

Figure 8-3 Un serveur Express pour une API RESTful

Chapitre 9 : Présentation de l'application développée

Après avoir terminé la présentation des outils de développement et la mise en œuvre de l'application, il est maintenant temps de montrer le résultat final.

9.1. Fonctionnalités de l'application

Cette application de gestion des absences des employés de Miezaka offre de nombreuses fonctionnalités qui améliorent la gestion des congés au sein de l'entreprise. Parmi ces fonctionnalités, on trouve :

- Faire une demande de congé directement depuis l'application.
- Calcul automatique des soldes de congés restants sans avoir à effectuer de calculs manuels.
- Affichage des demandes d'absence et des soldes pour chaque employé.
- Impression des demandes d'absence.
- Affichage de la liste des employés et de leurs soldes de congés respectifs.
- Ajout de nouveaux employés dans le système.
- Consultation des statistiques d'absence et de congés.

9.2. Présentation de quelques pages de l'application

9.2.1. Page d'authentification

Pour assurer la sécurité de notre application et protéger les données manipulées, il est nécessaire de mettre en place un système d'authentification. Chaque utilisateur doit posséder un compte pour accéder aux différentes fonctionnalités. En cas de connexion réussie, l'utilisateur sera redirigé vers la page d'accueil. Dans le cas contraire, un message d'erreur clair et explicite sera affiché, indiquant les raisons de l'échec de la tentative de connexion.

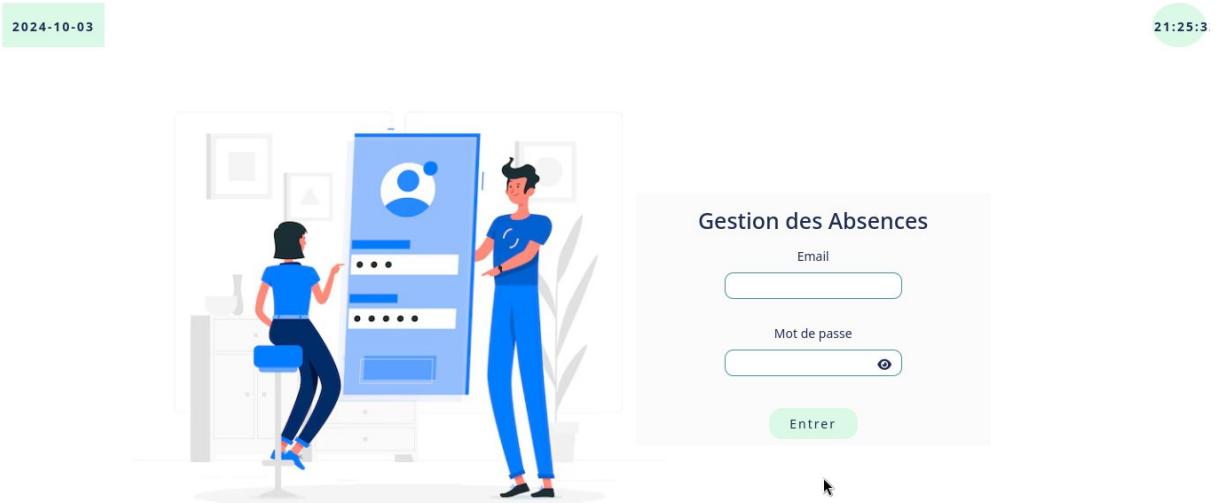


Figure 9-1Page d'authentification

9.2.2. Page d'authentification

Après avoir réussi l'authentification, l'utilisateur sera redirigé vers la page d'accueil de l'application. Cette page servira principalement à afficher le menu, permettant à l'utilisateur d'accéder facilement aux différentes fonctionnalités. Ces fonctionnalités seront disponibles dès la première page, offrant ainsi une navigation simple et intuitive.

Figure 9-2Page d'accueil

9.3.2. Page à propos de l'employé

Cette page permet d'afficher un tableau listant tous les employés, avec la possibilité de modifier ou supprimer un employé si nécessaire. Des options de recherche facilitent la localisation d'un employé spécifique, et il est également possible d'ajouter de nouveaux employés. En outre, la liste des employés peut être exportée en format Excel ou CSV, selon les besoins.

The screenshot shows a user interface for managing employees. On the left is a sidebar with icons for Accueil, Dashboard, Employé, Demande, and a Dropdown menu. The main area has a header with the email 'antonio@gmail.com' and a role indicator 'ADMINISTRATEUR'. Below this is a search bar labeled 'Recherche' and buttons for 'CSV' and 'Excel' exports. A green button labeled 'Nouvelle Employé' with a plus sign is visible. The central part of the screen displays a table of employees with the following data:

Id	Nom	Prenom	Sexe	Motif	Solde d'absence	Plafonnement	PlafBool	Action
3	FIANKINANTSOA	Maminiaina Nancia	F	Stagiaire	25	10	true	
2	RAZAFIMAPIANDRA	Françoise	F	Caissière	23	20	true	
1	RAKOTONIAINA	Jean De Dieu	M	Comptable	23	15	true	

At the bottom of the table area are navigation buttons for the first, last, and previous/next pages. A footer at the bottom right contains a small circular icon with the number 97.

Figure 9-3 Page à propos de l'employé

9.2.4. Page de demande absence

Cette page permet d'afficher et de gérer les demandes d'absence des employés. Un tableau présente la liste des employés ayant soumis des demandes. L'utilisateur peut rechercher des employés ayant fait une demande d'absence sur une période spécifique en sélectionnant une plage de dates. Une fois les résultats affichés, il est possible d'imprimer les demandes directement depuis la page. L'exportation des résultats sous format Excel ou CSV est également disponible pour répondre aux besoins spécifiques.

The screenshot shows a web application interface for managing absence requests. On the left, there is a sidebar with icons for Accueil, Dashboard, Employe, Demande, and Deconnexion. The main content area has a header with the email 'antonio@gmail.com' and a 'ADMINISTRATEUR' role indicator. Below the header is a search bar and a table listing absence requests. The table columns include Nom et Prenom, N°Matricole, Fonction, Nombre de jours, Congé restant, Date de départ, Date retour, Motif, Date Fin, and Action. The table lists five entries, each with a 'Details' (yellow) and 'Edit' (blue) button. At the top right of the main content area is a green button labeled '+ Nouvelle Demande'. At the bottom of the main content area are navigation buttons (< 1 2 >).

Figure 9-4 Page a propos de demande d'absence

9.2.5. Demande absence

Un formulaire est disponible pour soumettre une nouvelle demande d'absence. L'utilisateur peut renseigner les informations requises, telles que le nom de l'employé, le type de congé, et le motif. Après soumission, la demande est automatiquement enregistrée et peut être imprimée ou exportée selon les besoins.

The screenshot shows a modal window titled 'Créer Nouvelle Demande' (Create New Request). The modal contains fields for 'Employé' (selected: RAZAFIMAPIANDRA, Françoise), 'Type d'absence' (dropdown placeholder: Sélectionnez un type d'absence), 'Date de début' (date input: jj/mm/aaaa), 'Date retour' (date input: jj/mm/aaaa), 'Motif' (text input), and 'Jours d'absence' (text input). At the bottom of the modal is a blue 'Ajouter' (Add) button. The background of the modal is dark, and the overall interface is consistent with the previous screenshot.

Figure 9-5 Formulaire d'ajout demande absence

Après avoir cliqué sur le bouton "Imprimer", un document récapitulatif de la demande d'absence s'affiche, prêt à être imprimé.

SOCIÉTÉ MIEZAKA EURL

Fianarantsoa, 02/09/2024

Téléphone : 75 516 55

Fax : 75 52125

Fianarantsoa - 301

DEMANDE D'AUTORISATION D'ABSENCE

Nom(s) et prénom(s) : RAZAFIMAPIANDRA Françoise

N° Matricule : E006

Fonction : Caissière

Nombre de jours : 2

Congé restant : 23

Date de départ : 08-11-2024

Date de retour : 11-11-2024

Motif : maladie

L'intéressé

Collègue (Sign+Prénom)

La Direction

SOCIÉTÉ MIEZAKA EURL

Fianarantsoa, 02/09/2024

Téléphone : 75 516 55

Fax : 75 52125

Fianarantsoa - 301

DEMANDE D'AUTORISATION D'ABSENCE

Nom(s) et prénom(s) : RAZAFIMAPIANDRA Françoise

N° Matricule : E006

Fonction : Caissière

Nombre de jours : 2

Congé restant : 23

Date de départ : 08-11-2024

Date de retour : 11-11-2024

Motif : maladie

L'intéressé

Collègue (Sign+Prénom)

La Direction

Figure 9-6 Le modèle de fiche d'absence

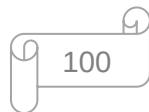
CONCLUSION GENERALE

L'application de gestion des absences de Miezaka est une avancée importante pour organiser l'absence des employés. Conçue avec Vue.js pour le frontend et Node.js pour le backend, elle est facile à utiliser et gère bien les données.

Cette solution a plusieurs objectifs : simplifier la demande d'absence, respecter les règles de gestion des congés, et mettre à jour automatiquement les soldes. Les responsables RH peuvent maintenant traiter rapidement les demandes et vérifier les soldes, ce qui fait gagner beaucoup de temps par rapport aux anciennes méthodes. Cependant, le développement a rencontré des défis, notamment pour connecter la base de données MySQL au serveur Ubuntu, ce qui a nécessité quelques ajustements techniques. Malgré ces difficultés, les retours des utilisateurs sont positifs, mettant en avant la simplicité et l'efficacité de l'application.

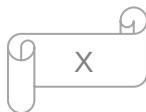
Pour l'avenir, l'application peut évoluer. Il serait utile d'ajouter des fonctions comme la gestion des absences de dernière minute et l'envoi de notifications automatiques aux employés. Une application mobile pourrait également permettre aux employés de faire des demandes de congé où qu'ils soient.

En résumé, l'application de gestion des absences de Miezaka a amélioré la gestion des congés et offre de bonnes bases pour des améliorations futures, assurant ainsi sa valeur pour l'entreprise et ses employés.



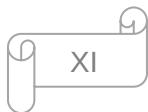
BIBLIOGRAPHIE

- [Bakari M., 2024] Bakari Maecha, 2024. Cours de conduite de projet informatique.
Université de Fianarantsoa.
- [Tsiorinantaina R., 2023] Tsiorinantaina RAKOTOARISON, 2023, Support de cours Javascript, Université de Fianarantsoa.
- [Rakotonirainy H., 2024] Rakotonirainy Hasina, 2024. Cours de Unified Modelign Language.
- Université Fianarantsoa.
- [ARISETRA E.R, 2023] ARISETRA Eddy R., 2023, Support de cours Mysql, Université de Fianarantsoa.
- Chowdhury, A., 2018. Guide de l'administrateur MySQL 8. Packt Publishing.
- [Gabarro, J., 2019.] Node.js et Express : Démarrer avec Node.js et Express.
- Gupta, S., 2020. Apprendre le développement Node.js. Packt Publishing.
- Moccia, V., 2018. Maîtriser Vue.js. Packt Publishing.
- Holmes, M., 2017. Le livre de recettes Vue.js 2. Packt Publishing.



WEBOGRAPHIE

- Node.js Documentation - <https://nodejs.org/en/docs/>
- Vue.js Guide - <https://vuejs.org/guide/introduction.html>
- Express.js Documentation - <https://expressjs.com/>
- MySQL Documentation - <https://dev.mysql.com/doc/>
- Sequelize Documentation (ORM pour Node.js) - <https://sequelize.org/>
- Docker Documentation (si tu utilises Docker) - <https://docs.docker.com/>
- MDN Web Docs - <https://developer.mozilla.org/fr/>
- W3Schools Node.js Tutorial - <https://www.w3schools.com/nodejs/>
- W3Schools Vue.js Tutorial - <https://www.w3schools.com/vue/>
- Vue Router Documentation - <https://router.vuejs.org/>
- Axios Documentation (pour les requêtes HTTP) - <https://axios-http.com/docs/intro>
- Webpack Documentation (pour la configuration du projet Vue.js) - <https://webpack.js.org/>
- Postman Documentation (pour tester les APIs) - <https://learning.postman.com/>



ANNEXE

Cette annexe présente le processus d'installation de Node.js et du framework Express, qui sont essentiels pour le développement de l'application de gestion des absences. Node.js permet d'exécuter du JavaScript côté serveur, tandis qu'Express simplifie la création d'applications web avec Node.js.

Avant de procéder à l'installation, assurez-vous d'avoir les éléments suivants :

- **Système d'exploitation** : Windows, macOS ou Linux.
- **Accès Internet** : Nécessaire pour télécharger Node.js et les dépendances.
- **Éditeur de code** : Un éditeur tel que Visual Studio Code, Sublime Text, ou tout autre de votre choix.

Installation de Node.js

- ◆ Accédez au site officiel de Node.js : [Node.js](#).
- ◆ Choisissez la version recommandée pour la plupart des utilisateurs (LTS) et téléchargez le fichier d'installation.
- ◆ Exécutez le fichier téléchargé et suivez les instructions à l'écran pour installer Node.js.
- ◆ Cochez l'option pour ajouter Node.js au chemin d'accès (PATH) lors de l'installation.
- ◆ Ouvrez le terminal ou l'invite de commande.
- ◆ Tapez la commande suivante pour vérifier que Node.js est correctement installé : node -v

Initialiser un projet Node.js :

- ◆ Créez un répertoire pour votre projet :mkdir gestion-absences,cd gestion-absences
- ◆ Initialisez un nouveau projet Node.js avec la commande suivante :npm init -y
- ◆ Cela créera un fichier package.json dans votre répertoire.

Installer Express :

- ◆ Exécutez la commande suivante pour installer Express :npm install express



RESUMÉ

Dans ce présent mémoire, nous avons conçu et réalisé un outil destiné à la « conception et réalisation d'une application de gestion des absences pour la société Miezaka ». Pour mener à bien ce projet, nous avons utilisé le framework Node.js avec une architecture RESTful, Vue.js pour le front-end, un SGBD MySQL, des conteneurs Docker, ainsi que l'outil de modélisation UML.

L'application permet de gérer efficacement les demandes de congés des employés, de mettre à jour automatiquement les soldes de congés, de gérer différents types de congés (annuels, spéciaux), et de suivre les statistiques d'absences via un tableau de bord. De plus, elle permet aux responsables RH de consulter les demandes d'absences selon des règles spécifiques.

Ce projet m'a permis de me familiariser avec le développement d'applications web, d'acquérir des compétences dans la gestion de projet informatique, ainsi que de renforcer mes connaissances pratiques en analyse des besoins, conception, et déploiement d'applications web. Comme tout projet, cette application est destinée à évoluer avec le temps et les besoins de l'entreprise.

Mots Clés : gestion d'absence, application web, Node.js, Vue.js, MySQL.

ABSTRACT

In this thesis, we designed and developed a tool for the "design and implementation of an absence management application for the Miezaka company." To successfully carry out this project, we utilized the Node.js framework with a RESTful architecture, Vue.js for the front-end, a MySQL database management system, Docker containers, and UML modeling tools.

The application efficiently manages employees' leave requests, automatically updates leave balances, manages different types of leave (annual, special), and tracks absence statistics through a dashboard. Furthermore, it allows HR managers to review absence requests according to specific rules.

This project enabled me to become familiar with web application development, acquire skills in IT project management, and strengthen my practical knowledge in requirements analysis, design, and deployment of web applications. Like any project, this application is intended to evolve over time with the needs of the company.

Keywords: absence management, web application, Node.js, Vue.js, MySQL.