

République Algérienne Démocratique et Populaire
Ministre de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira-Bejaia

Faculté des Sciences Exactes
Département Informatique



PROJET DE FIN DE CYCLE

En vue l'obtention du diplôme de Master Informatique
Spécialité : Génie logiciel

T H È M E

Conception et réalisation d'une application web
de la gestion de présence des employés de l'entreprise
NAFTAL

Présenté par :

MAHMOUDI Kenza & MAMMERI Cylia

Le JURY :

- Présidente : Dr **TIGHIDET Soraya**
- Examineur : Dr **TAHAKOURT Zineb**
- Encadrant : Dr **SIDER Abderrahmane**

ANNÉE UNIVERSITAIRE : 2022/2023

REMERCIEMENTS

Au terme de ce parcours d'étude, il convient de prendre un moment pour exprimer nos gratitudes ceux qui nous ont soutenus et guidés tout au long de cette aventure intellectuelle.

Tout d'abord, nous tenons à exprimer nos remerciements les plus sincères à Dieu, qui est la source de sagesse et de force, il nous a donné le courage et la patience durant notre formation master et pouvoir réaliser ce travail.

À nos Chers Parents pour leurs soutiens inconditionnels et leurs encouragements constants durant cette période d'étude et leur confiance qui a été une source d'inspiration sans fin, et nous sommes infiniment reconnaissants de pouvoir compter sur nous.

*Nous tenons à remercier, notre encadrant M^r : **A .SIDER** pour son aide et son encadrement attentif, et ces conseils précieux.*

*On tient à ne pas oublier de remercier l'entreprise NAFTAL qui nous a accueilli pour réaliser notre stage en particulier **M^r M.ALBANI** notre encadrant de stage, à son encadrement attentif et ces conseils éclairés.*

Nos gratitudes aux membres de jury envers chacun d'entre vous pour le temps que vous avez consacré pour nous et vos connaissances afin d'évaluer notre travail.

Un grand remerciement à toutes les personnes qui nous ont apporté leurs soutiens, leurs encouragements et leurs aides tout au long de cette aventure.

DÉDICACE

À mon dieu Allah, qui toujours avec moi, qui me donne l'aide et le courage et la santé.

Je dédie ce mémoire

À mes parents, vous êtes les héros de ma vie. Votre bienveillance et votre sacrifice ont été les fondations sur lesquelles j'ai pu construire mes rêves. Vos encouragements et vos précieux conseils.

*À mon père « **Ikhlef** », le plus généreux homme du monde qui me donne la force et le courage tu es le meilleur père de tous les pères tu m'as soutenu moralement et physiquement tout au long de ma vie, tes conseils, ta confiance qui est la source de ma réussite. Des mots qui ne pourront jamais exprimer la profondeur du respect, et la considération et l'amour éternel je t'aime papa*

*À ma mère « **Nadia** », La lumière de mes jours, ta présence tes sacrifices ton éducation pour moi qui me fait de force pour réaliser mes objectifs. Aucune parole ne serait pointe te remercier ta juste valeur : d'amour, le respect que je porte pour toi ma chère maman je t'aime beaucoup. que dieu te protège, te garde en bonne santé tout au long de vie.*

*À l'homme de ma vie « **Achour** », tu es ma source d'inspiration quotidienne. Ton amour ton soutien et ta confiance en moi ont été des moteurs puissants dans la réalisation de mes aspirations. je me suis reconnaissante pour ton amour inconditionnel.*

*À toutes ma famille mes frères « **Malek** » et « **Ramtane** », mes sœurs, mes cousins, mes oncles paternelle et maternelle.*

*Et à toutes les personnes qui m'ont encouragée, que ce soit par de simples mots d'encouragement, des gestes bienveillants ou des encouragements sincères, votre soutien a été un cadeau précieux. Votre confiance en mes capacités et vos encouragements constants m'ont donné la force de surmonter les obstacles et de me dépasser. et à ma binôme « **Kenza** » que j'ai partagée mon travail.*

À vous tous, je dédie mes réussites je suis reconnaissante pour tout ce que vous avez fait pour moi.

Cyria .

DÉDICACE

Louange à Dieu tout puissant, qui m'a permis de voir ce jour tant attendu.

Je dédie ce mémoire

*À mon très cher père « **Ali** », Tu as toujours été pour moi un un exemple du père respectueux, honnête, de la personne méticuleuse, je tiens à honorer l'homme que tu es. Grâce à toi papa j'ai appris le sens du travail et de la responsabilité. Je voudrais te remercier pour ton amour, ton confiance, ta générosité, ta compréhension... Ton soutien fut une lumière dans tout mon parcours. Aucune dédicace ne saurait exprimer l'amour l'estime et le respect que j'ai toujours eu pour toi. Je t'aime papa et j'implore le tout-puissant pour qu'il t'accorde une bonne santé et une vie longue et heureuse.*

*À ma mère « **Dahbia** », Tu as toujours été là pour moi, tu es restée ma plus belle certitude quand tout le reste s'est effondré. tu étais là : constante et forte, inébranlable, ma consolatrice, mon réconfort, patiente, indulgente, bienveillante et compréhensive, la meilleure des amies. Tu es et ma certitude. Mon guide dans l'existence Un compagnon de vie si sage. Depuis ma naissance ma plus belle chance.*

*A mon cher frère « **Amine** » A tous les moments d'enfance passés avec toi mon frère, engage de ma profonde estime pour l'aide que tu m'as apporté. Tu m'as soutenu, réconforté et encouragé. Puissent nos liens fraternels se consolider et se pérenniser encore plus.*

*À mes petites sœurs adorables « **Samira** », « **Imane** » et « **Amina** » je vous souhaite le bonheur et la réussite dans vos vies et vos études.*

A toutes ma famille

*A ma binôme « **Cylia** », avec qui j'ai partagé de belles années d'études.*

Kenza .

Table des matières

Liste des Tableaux	vii
Table des Figures	viii
Table d'abréviation	ix
INTRODUCTION GÉNÉRALE	1
1 Présentation de l'organisme d'accueil et méthodologie de développement	3
1.1 Introduction	3
1.2 Présentation de l'entreprise NAFTAL	3
1.2.1 Définition de l'entreprise	3
1.2.2 Historique et développement de l'entreprise Naftal	3
1.2.3 Les missions de l'entreprise Naftal de Bejaïa	4
1.3 L'organisme d'accueil	4
1.3.1 L'organigramme de la branche GPL de Bejaia	4
1.3.2 Présentation du département PMC	6
1.3.3 Définition du département PMC	6
1.4 Définition de gestion de présence	7
1.5 Contexte	7
1.6 Problématique	7
1.7 L'objectif du projet	7
1.8 Définition d'une application web	8
1.9 Méthodologie du développement	8
1.9.1 Présentation de la méthode UP (Unified process)	8
1.10 Langage de modélisation	10
1.10.1 UML	10
1.11 Conclusion	11
2 Analyse des besoins et conception	12
2.1 Introduction	12
2.2 Spécification des besoins	12
2.2.1 Les besoins fonctionnels	12
2.2.2 Les besoins non fonctionnels	12
2.3 Analyse des besoins	13
2.3.1 Identification des acteurs	13
2.3.2 Identification des cas d'utilisations	13
2.3.3 Diagramme de cas d'utilisation	13
2.4 Description des cas d'utilisation	14
2.4.1 Cas d'utilisation « Inscription »	14
2.4.2 Cas d'utilisation « Authentification »	16

2.4.3	Cas d'utilisation « Liste des employés»	17
2.4.4	Cas d'utilisation « Liste des justifications»	18
2.4.5	Cas d'utilisation « Liste des jours fériés»	20
2.4.6	Cas d'utilisation « Liste des horaires»	22
2.4.7	Cas d'utilisation «Importer pointage»	24
2.4.8	Cas d'utilisation « Feuille de présence»	25
2.5	Diagramme de classe	26
2.6	Conclusion	27
3	Réalisation et implémentation de l'application	28
3.1	Introduction	28
3.2	Outils et logiciels de programmation	28
3.2.1	Visual Studio Code	28
3.2.2	NodeJS	28
3.2.3	MongoDB Compass	28
3.2.4	Postman	29
3.3	Langages et frameworks de programmation	29
3.3.1	React	29
3.3.2	Bootstrap	29
3.3.3	ExpressJS	29
3.3.4	Javascript	29
3.3.5	HTML(HyperText Markup Language)	29
3.3.6	CSS(Cascading Style Sheets)	30
3.4	Implémentation du Modèle-Vue-Contrôleur(MVC)	30
3.5	Présentation de l'application	32
3.5.1	Nom d'application	32
3.6	Quelques vues de l'application	33
3.6.1	Présentation des vues	33
3.6.2	Vue «d'accueil »	34
3.6.3	Vue «Inscription »	34
3.6.4	Vue «Menu »	35
3.6.5	Vue «Liste des employés »	35
3.6.6	Vue «Liste des horaires »	36
3.6.7	Vue «Ajouter horaire »	36
3.6.8	Vue «Modifier horaire »	37
3.6.9	Vue «Liste des jours fériés»	37
3.6.10	Vue «Ajouter jour férié»	38
3.6.11	Vue «Modifier jour férié»	38
3.6.12	Vue « Liste des justifications »	39
3.6.13	Vue « Ajouter justification »	39
3.6.14	Vue « Modifier justification »	40
3.6.15	Vue «Importer pointage »	40
3.6.16	Vue «Feuille de présence »	41
3.7	Conclusion	41
	CONCLUSION GÉNÉRALE	42
	BIBLIOGRAPHIE	43

Liste des tableaux

2.1	Cas d'utilisations	13
2.2	Description du cas d'utilisation « Inscription »	15
2.3	Description du cas d'utilisation « Authentification »	16
2.4	Description du cas d'utilisation « Liste des employés »	17
2.5	Description du cas d'utilisation « Liste des justifications »	18
2.6	Description du cas d'utilisation « Liste des jours fériés »	20
2.7	Description du cas d'utilisation « Liste des horaires »	22
2.8	Description du cas d'utilisation « Importer pointage »	24
2.9	Description du cas d'utilisation « Feuille de présence »	25

Table des figures

1.1	Ogranigramme de District GPL Bejaia [2]	5
1.2	Organigramme pour le département PMC [2]	6
1.3	Enchainements d'activités au cours du cycle de vie [4]	9
1.4	Diagrammes UML [6]	10
2.1	Cas d'utilisations du système	14
2.2	Diagramme de séquence de cas d'utilisation « Inscription »	15
2.3	Diagramme de séquence du cas d'utilisation « Authentification »	16
2.4	Diagramme de séquence du cas d'utilisation « Rechercher employé »	17
2.5	Diagramme de séquence du cas d'utilisation « Liste des justifications »	19
2.6	Diagramme de séquence du cas d'utilisation « Liste des jours fériés »	21
2.7	Diagramme de séquence du cas d'utilisation « Liste des horaires »	23
2.8	Diagramme de séquence du cas d'utilisation « Importer pointage »	24
2.9	Diagramme de séquence du cas d'utilisation « Feuille de présence »	25
2.10	Diagramme de classe	26
3.1	Modèle justification.js	30
3.2	Vue Ajouter justification	31
3.3	Contrôleur Ajouterjustification	31
3.5	Vue « Authentification »	34
3.6	Vue « Inscription »	34
3.7	Vue « Menu »	35
3.8	Vue « Liste des employés »	35
3.9	Vue « Liste des horaires »	36
3.10	Vue « Ajouter horaire»	36
3.11	Vue « Modifier horaire»	37
3.12	Vue «Liste des Jours Fériés »	37
3.13	Vue «Ajouter Jour Férié »	38
3.14	Vue «Modifier Jour Férié »	38
3.15	Vue « Liste des justifications »	39
3.16	Vue «Ajouter justification »	39
3.17	Vue «Modifier justification »	40
3.18	Vue « Importer pointage »	40
3.19	Vue « Feuille de présence »	41

Table d'abréviation

Abréviation	Désignation
Abréviation de l'entreprise	
GPL	Gaz de Pétrole Liquéfié
RGP	Responsable gestion de personnel
PMC	Personnel et moyens communs
Abréviation technique	
UP	Unified Process
UML	Unified modeling language
VSC	Visual studio code
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
BDD	Base de données
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	Javascript

INTRODUCTION GÉNÉRALE

Au fil des années, la gestion de la présence des employés a évolué. Auparavant, cela se faisait manuellement sur une feuille de présence. Puis avec l'avènement des nouvelles technologies, les entreprises ont commencé à utiliser des pointeuses pour enregistrer les heures de présence des employés.

Cependant, malgré ces avancées technologiques, la gestion de la présence reste un défi pour de nombreuses entreprises, notamment celles qui ont des employés répartis sur plusieurs sites ou qui travaillent en horaires décalés. C'est le cas de l'entreprise Naftal, qui utilise une pointeuse pour enregistrer le pointage des employés, mais qui rencontre des difficultés pour gérer les absences, les retards, les justifications, les heures supplémentaires, etc. L'entreprise ne dispose pas d'une application qui permettrait de gérer la présence des employés de manière fiable et automatique. En effet, La gestion de présence des employés est un enjeu clé pour toute entreprise soucieuse de son efficacité, de sa productivité et de la qualité de ses prestations.

Dans le cas de l'entreprise Naftal, spécialisée dans la distribution de carburant de Béjaïa, la gestion de la présence des employés revêt une importance particulière. Alors, les employés de Naftal sont les acteurs de l'entreprise auprès des clients et des partenaires. Leur professionnalisme, leur compétence et leur engagement sont des facteurs déterminants pour la qualité de service et la satisfaction des clients. De plus, dans un contexte concurrentiel, la performance et l'efficacité des employés sont des atouts clé pour la réussite de l'entreprise. Ainsi, la gestion de la présence de ses fonctionnaires doit pouvoir optimiser leurs présences sur terrain, favoriser leurs participations et leurs motivations, tout en garantissant le respect de la réglementation et des normes de sécurité en vigueur dans le secteur de l'énergie.

Dans ce contexte, ce mémoire de fin d'étude vise à concevoir et à développer une application web pour la gestion de présence des employés de l'entreprise Naftal. Cette dernière permettra de générer d'une manière fiable et automatique une feuille de présence.

Ce mémoire est organisé en trois chapitres

Dans le premier chapitre, nous avons fait une introduction sur l'entreprise et la présentation de l'organisme Naftal et définir le concept de gestion de présence des employés et d'expliquer pourquoi il est important pour l'entreprise Naftal. Ensuite on va présenter notre projet en introduisant le contexte ainsi que la problématique posée, on va présenter également les objectifs et définir les solutions existantes. Puis présenter la méthode choisie pour mener à bien notre projet.

Le deuxième chapitre, intitulé l'analyse des besoins et conception où on va présenter les différentes étapes de conception de notre application web. Nous commençons par l'analyse des besoins et la spécification des exigences. Nous présentons ensuite les différents diagrammes UML que nous utiliserons pour modéliser notre application.

Le troisième chapitre, intitulé la réalisation et implémentation de l'application où nous allons présenter les technologies et les outils utilisés. Nous présentons également les différentes fonctionnalités que nous avons développées et comment elles répondent aux besoins identifiés dans le chapitre 2.

Enfin, on termine par une conclusion générale qui résumera le travail réalisé et de montrer les limites de notre projet et les perspectives d'amélioration pour les travaux futurs. Après nous concluons notre mémoire en expliquant comment notre projet répond aux objectifs que nous avons fixés .

Chapitre 1

Présentation de l'organisme d'accueil et méthodologie de développement

1.1 Introduction

Dans ce premier chapitre, nous présentons l'organisme Naftal où se déroulait notre stage, nous définissons ainsi le contexte de notre projet dans la gestion de personnel qui comprend la problématique à résoudre pour éteindre un objectif.

Ensuite, nous présentons la méthodologie de développement que nous avons choisie pour la conception et la réalisation de cette application web qui est la méthode Unified Process (UP). Cette méthode est une approche itérative et incrémentale, qui permet de développer des systèmes logiciels de manière flexible et adaptative.

1.2 Présentation de l'entreprise NAFTAL

1.2.1 Définition de l'entreprise

Naftal est une entreprise publique algérienne spécialisée dans la vente et la commercialisation de carburants, lubrifiants et gaz naturel auprès des particuliers et des entreprises. La société est le premier distributeur de carburant en Algérie et dispose d'un vaste réseau de stations-services dans tout le pays.

Dans le cadre de l'amélioration de ses processus de gestion de personnel, Naftal dispose d'un système qui gère le pointage des employés, mais d'une façon moins fiable et semi-automatique, alors nous proposons une application web permettra de gérer les données d'entrée et de sortie des employés, de générer des rapports de présence et d'assurer un suivi efficace des absences et des retards[15].

1.2.2 Historique et développement de l'entreprise Naftal

Naftal est une entreprise algérienne est chargée de l'industrie de raffinage et de la commercialisation la distribution de produits pétroliers. Elle a été créée par le décret N° 80/101 le 06 avril 1981à la suite de la fusion de deux entreprises de l'industrie pétrolière algérienne : la SODIGAZ (Société de Distribution des Gaz de Pétrole Liquéfiés) et la SOPRATEL (Société de Production des Tél services)[16].

Depuis sa création, Naftal a connu un développement important, en particulier dans les années 1990, avec la réforme du secteur des hydrocarbures en Algérie. Elle a ainsi développé son réseau de distribution sur tout le territoire algérien, avec la création de nouvelles stations-service et de centres de stockage, le 18 avril 1998 Naftal est transformée en société par actions (SPA) au capital social de 6.65 milliards de DA, filiale à 100% de groupe Sonatrach.

En 2008, Naftal a engagé un important programme de modernisation de ses infrastructures, avec la construction de nouvelles unités de production et de stockage, la modernisation de ses stations-service et la mise en place de nouveaux systèmes de gestion.

En 2014, ouverture de la première station du gaz naturel comprimé-carburant.

Actuellement, Naftal est le principal distributeur de produits pétroliers en Algérie, avec un réseau de plus de 600 stations-service et une part de marché de plus de 70%. L'entreprise est également présente dans d'autres secteurs, tels que la distribution de GPL (gaz de pétrole liquéfié) et la production d'électricité à partir de centrales thermiques. Naftal emploie plus de 15 000 personnes et contribue significativement à l'économie du pays[1].

1.2.3 Les missions de l'entreprise Naftal de Bejaïa

Naftal a pour mission principale, la distribution et la commercialisation des produits pétroliers et dérivés sur le marché national, elle est constituée de 3 branches suivantes qui sont toutes chargées de respecter les exigences d'hygiène, de sécurité, d'approvisionnement et de la qualité [1] :

- **La branche carburants**

Cette branche est chargée de l'approvisionnement, de stockage et de distribution des carburants terre, aviation et marine.

- **La branche GPL**

Elle est chargée de l'approvisionnement en gaz Butane que ce soit la B13, B03 ou B06 ainsi que en gaz Propane que ce soit la P35 ou P11 et aussi le GPL/C (GPL Carburant) qui dispose d'un nom commercial "SIRGHAZ" qui est un mélange de 80% de Propane avec 20% de Butane.

- **La branche commercialisation**

Elle a pour mission la mise en disposition tous les produits pétroliers à travers les stations-services et par la vente aux gros consommateurs sur le territoire national.

1.3 L'organisme d'accueil

1.3.1 L'organigramme de la branche GPL de Bejaïa

La Figure 1.1 illustre l'organigramme de District GPL de Bejaïa.

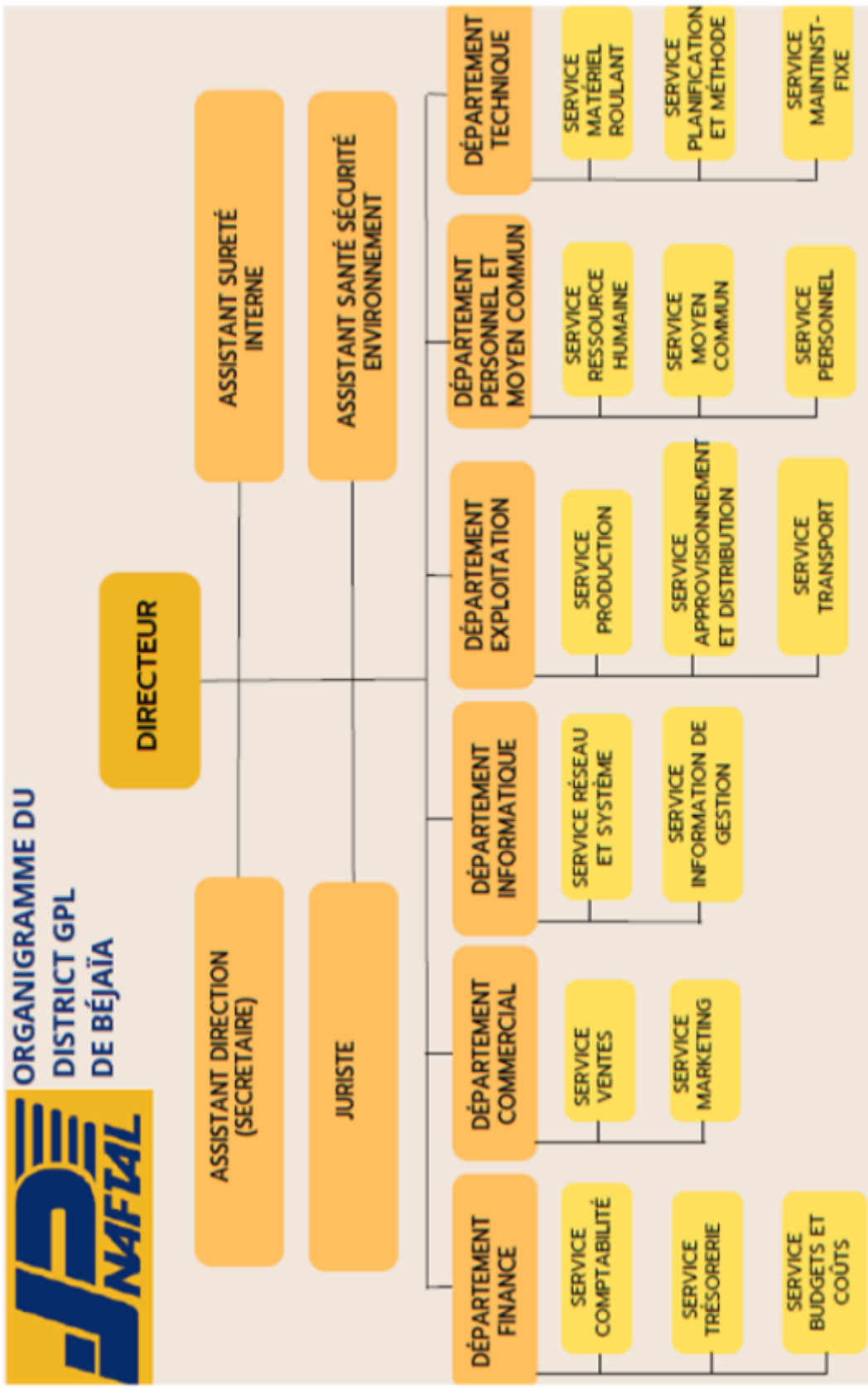


FIGURE 1.1 – Organigramme de District GPL Béjaïa [2]

1.3.2 Présentation du département PMC

Selon l'organigramme de District GPL en dessus exploite les fonctionnalités de département PMC.

Dans notre cas d'étude on s'intéresse au département personnel et moyen communs (PMC) de l'entreprise NAFTAL qu'une unité organisationnelle qui gère les ressources humaines et les moyens logistiques de l'entreprise et le service personnel.

La Figure 1.2 en dessous montre les différents services et sections de département PMC.

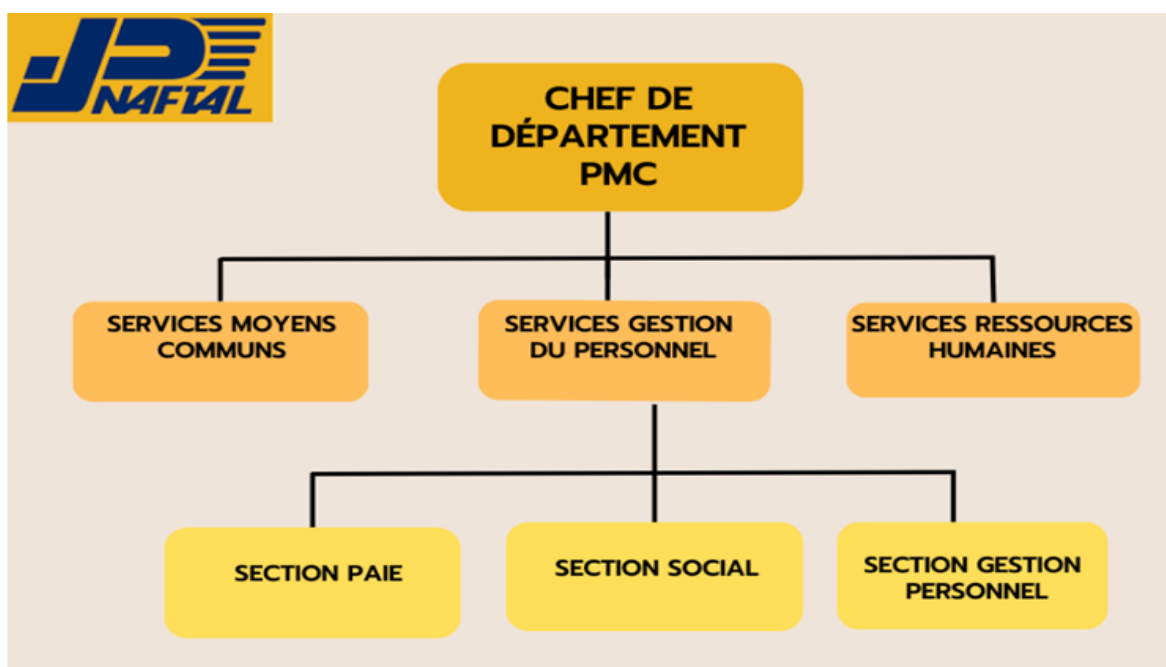


FIGURE 1.2 – Organigramme pour le département PMC [2]

1.3.3 Définition du département PMC

Le département PMC travaille en étroite collaboration avec les autres départements de l'entreprise naftal, pour assurer une utilisation efficace des ressources et une coordination optimale des activités de l'entreprise. A cet égard nous portons une attention particulière aux systèmes de suivi de leur assiduité, en s'intéressant à la gestion du temps et des présences des employés et formulerons des recommandations pour améliorer l'efficacité de cette gestion.

1.4 Définition de gestion de présence

La gestion de présence fait référence à l'ensemble des processus et des outils utilisés pour suivre et enregistrer la présence des employés dans une entreprise. Cela peut inclure des méthodes telles que la saisie manuelle des heures de travail ou l'utilisation de systèmes de pointage automatisés.

L'objectif principal de la gestion de présence est d'assurer une présence régulière et fiable des employés, afin de maintenir la productivité et d'optimiser les ressources de l'entreprise.

1.5 Contexte

Le District GPL de Bejaia dispose d'un système de pointage pour enregistrer la présence des employés de chaque département.

Lors de notre stage, nous avons constaté qu'un employé enregistre quatre pointages par jour dans divers créneaux horaires spécifiques, et qu'une feuille de présence est établie chaque fin de mois par le service personnel.

En cas d'absence d'un employé, quelle qu'en soit la raison (ordre de mission, maladie, congé, jour férié ou toute autre absence justifiée), le logiciel de gestion de présence ne marque pas l'effet de leur absence dans la feuille de présence, elle se fait manuellement par chaque responsable de département après avoir reçu les justifications.

Et comme les employés ont le potentiel d'augmenter les heures de travail comme heures supplémentaires le logiciel de gestion de présence qui est utilisé à Naftal ne les prend pas en considération. Par conséquent, cela est fait manuellement. Après avoir établi la feuille de présence complète, elle sera renvoyée au responsable de la section paie qui s'occupe du calcul de la paie mensuelle des agents.

1.6 Problématique

— Gestion manuelle des justificatifs

L'ajout manuel des justificatifs d' absences sur un support papier peut introduire une perte d'informations, une lourdeur dans l'exécution des tâches et cela aussi non sécurisé.

— Le rajout manuel des nombres d'heures supplémentaires

En effet cela peut engendrer des problèmes au niveau de département PMC et ça rentre dans la sécurité de travail d'un employé, donc l'information portée sur la feuille de présence n'est pas fiable.

— Communication interdépartementale

la communication entre les fonctionnaires de divers services, telle que l'envoi de feuilles de présence incomplètes par le responsable de la Gestion de personnel aux chefs de département, suivi par la collecte manuelle des justificatifs, entraîne des retards et une perte de temps significative dans l'établissement de la feuille de présence qui cause le retard sur les calculs de la paie.

1.7 L'objectif du projet

Face à cette problématique, nous avons choisi de réaliser un mémoire de fin d'études portant sur la conception et la mise en place d'une application web de gestion de présence des

employés pour l'entreprise Naftal. Notre objectif principal est de développer une application web qui permettra aux responsables de Naftal de suivre efficacement les heures de présence et d'absence de leurs employés, tout en automatisant les processus de présence. Cela permettra à l'entreprise de gagner en productivité et en efficacité en évitant les erreurs de saisie et de traitement des données.

Pour répondre à cette problématique, nous avons choisi d'utiliser la méthode Unified Process (UP), qui est une méthode de développement de logiciels itérative et incrémentale. La méthode UP est adaptée à notre projet car elle permet de gérer efficacement les risques, d'assurer une communication régulière avec les parties prenantes et de produire des résultats concrets tout au long du projet. C'est une application web qui facilitera la gestion de présence et satisfaction aux besoins des clients.

1.8 Définition d'une application web

Une application web est un logiciel hébergé sur un serveur distant auquel on accède via une connexion au réseau, généralement en utilisant le protocole HTTP, et qui s'exécute souvent dans un navigateur web.

1.9 Méthodologie du développement

Le développement d'une application informatique se passe par un ensemble d'étapes à suivre qui sont donné par la méthode choisi dans ce dernier, dans notre cas d'étude en focalisant à la méthode UP qui mène bien notre projet pour satisfaire le client[17].

1.9.1 Présentation de la méthode UP (Unified process)

UP (Unified Process) est une méthode de développement logiciel orientée objet et agile qui suit un processus incrémental et itératif pour développer des logiciels en équipe. Cette méthode est basé sur le langage de modélisation unifié (UML) et utilise des diagrammes UML pour modéliser les différents aspects d'un système et se concentre sur les principes de modélisation, du tests et du développement itératif.

UP est un processus personnalisable et adaptable qui peut être utilisé pour différents types de projets, allant des petits projets à une équipe unique, aux grands projets à plusieurs équipes[3].

On va suivre cette structure d'étape pour notre mémoire.

1. Dimension de UP

UP est décrit en deux dimensions les suivants :

- **Un axe horizontale** : présente le temps c'est l'aspect dynamique du processus. Sur cette axe, le processus est organisé en itérations (analyse des besoins, élaboration, construction, et transition).
- **Un axe vertical** : c'est l'aspect statique du processus. Sur cette axe, le processus est organisé en activités (expression des besoins, analyse, conception, implémentation, tests).

On a décomposé notre mémoire en trois phases selon la figure ci-dessous [6].

1. **Expression des besoins** : cette étape consiste à identifier les besoins des employés, les fonctionnalités nécessaires pour répondre à ces besoins et les contraintes à respecter pour la confidentialité des données.
2. **Analyse des besoins et Conception** : cette étape consiste à définir l'architecture de notre application en utilisant des diagrammes de cas d'utilisation, de séquences et de classe. Nous pouvons également définir les entités de données nécessaires pour stocker les informations relatives à la présence des employés.
3. **Réalisation et implémentation** : cette étape est composée en deux partie, la première partie consiste à coder l'application en utilisant les spécifications fonctionnelles et la conception détaillée. Nous pouvons également intégrer des bibliothèques de code open-source pour accélérer le développement et améliorer la qualité. La deuxième partie est consacré pour l'implémentation, qui consiste la présentation des interfaces de notre application.

La Figure 1.3 décrit les différentes activités et les phases du processus unifié

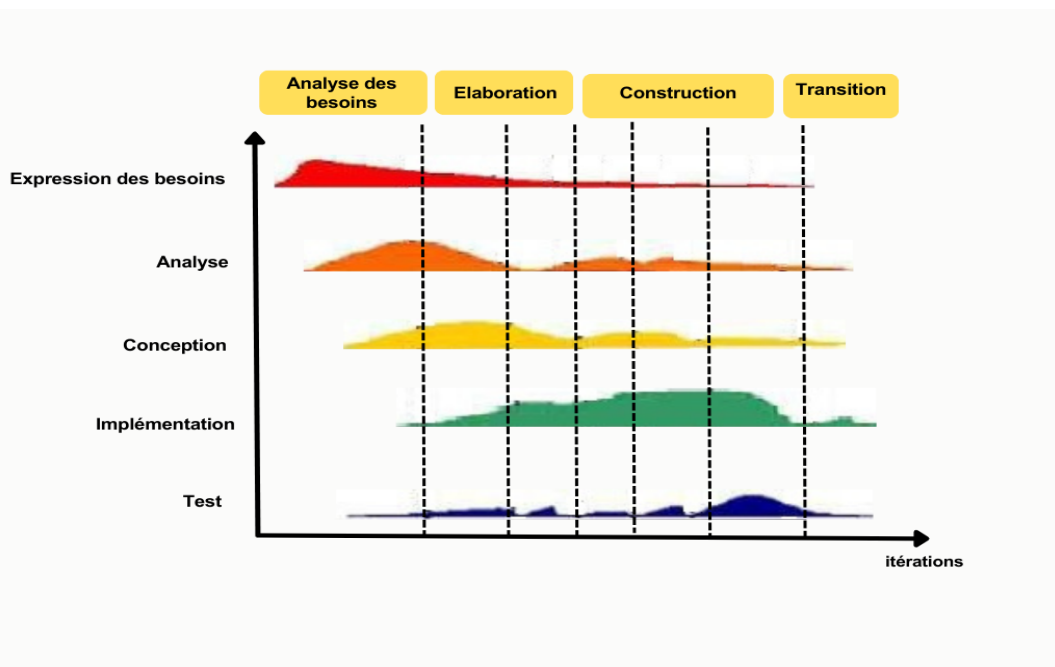


FIGURE 1.3 – Enchaînements d'activités au cours du cycle de vie [4]

1.10 Langage de modélisation

Comme le processus unifié basé sur le langage UML (Unified Modeling Language), est un outil essentiel à utilisé pour représenter les diagrammes UML.

1.10.1 UML

Langage UML ou le langage de modélisation unifié, est apparu dans les années 90 pour pouvoir palier aux problèmes que pose l'algorithmique lorsqu'il fallait représenter les classes et les interactions entre les classes[3].

Uml est basé sur l'approche objet qui requière de modéliser avant de concevoir, il est considéré en tant que standards et il est devenu une référence incontournable dans le domaine de génie logiciel.

La Figure 2.4 illustre les différents diagrammes UML.

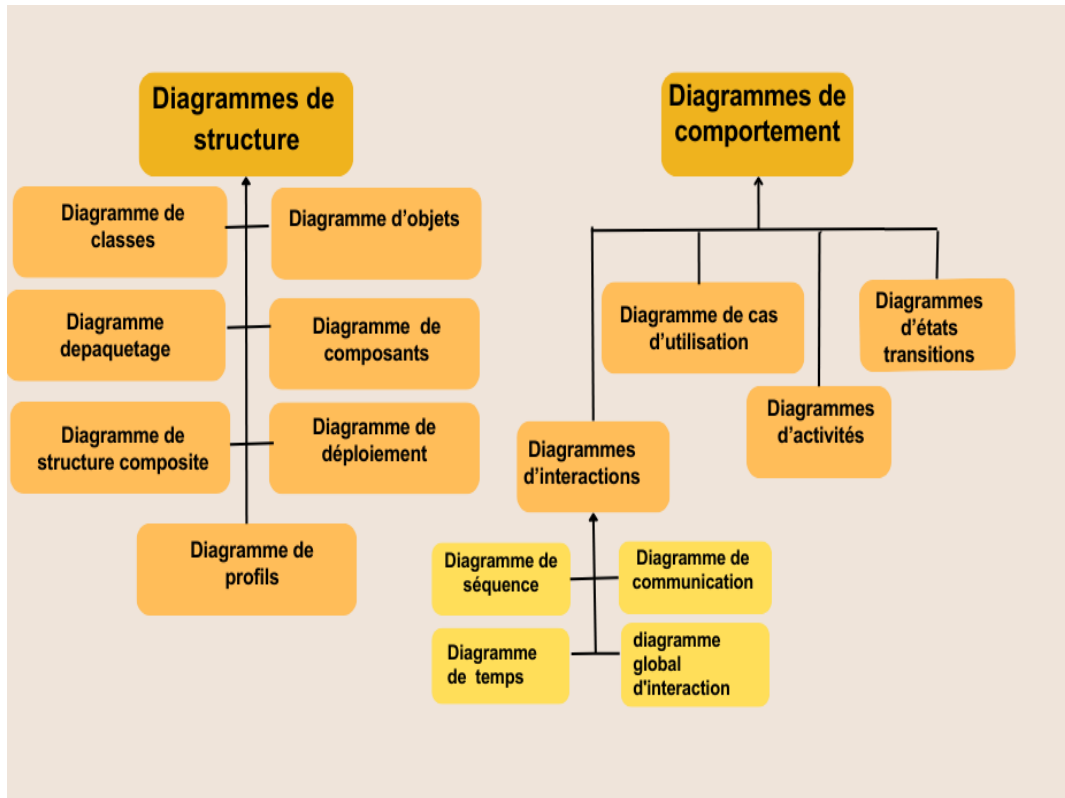


FIGURE 1.4 – Diagrammes UML [6]

1.11 Conclusion

Nous avons partagé notre travail en deux parties dans ce chapitre.

Dans la première partie nous avons commencé par la présentation de l'organigramme de l'entreprise. Ensuite, on a montré quelques problématiques liées à la gestion de présence dans la société NAFTAL. Enfin, on a indiqué le point important à atteindre dans notre développement comme définition des objectifs.

Dans la deuxième partie on a exposé la méthode UP qui mène bien notre mémoire. Enfin on a défini le langage UML et ses différents diagrammes. Dans le second chapitre nous présenterons la partie conception et l'analyse des besoins.

Chapitre 2

Analyse des besoins et conception

2.1 Introduction

Dans ce chapitre, nous décrirons l'analyse des besoins et conception de l'application web de gestion de présence des employés. Cette phase est cruciale car elle permet de définir les fonctionnalités et les exigences techniques de notre application.

2.2 Spécification des besoins

2.2.1 Les besoins fonctionnels

Les besoins fonctionnels expriment un ensemble d'action que doit effectuer le système en réponse à une demande. Dans cette phase, nous effectuons une étude détaillée des besoins de l'entreprise Naftal., nous identifions les exigences fonctionnelles et techniques du système.

- **Les exigences fonctionnelles** : les exigences fonctionnelles comprennent la capacité de saisir les données d'entrée et de sortie des employés.
- **Les exigences techniques** : les exigences techniques comprennent l'utilisation de technologies web modernes telles que NodeJS , ExpressJS, ReactJS et MongoDB.

2.2.2 Les besoins non fonctionnels

Les besoins non fonctionnels permettent d'améliorer la qualité de service de notre application. Ils représentent les exigences implicites offertes par le système, la liste suivante représente les besoins non fonctionnels de notre application

- **Sécurité** : garantir la sécurité des données des employés en utilisant des mesures de sécurité telles que l'authentification.
- **La disponibilité** : l'application doit être disponible pour les utilisateurs à tout moment, sans interruption, et doit être capable de gérer une charge de travail élevée.
- **La convivialité** : l'application doit être facile à utiliser et à naviguer pour les utilisateurs, avec une interface utilisateur claire et intuitive.
- **La performance** : l'application doit être rapide et réactive, avec des temps de réponse courts pour les requêtes utilisateur.
- **La compatibilité** : l'application doit être compatible avec différents navigateurs web et appareils.

- **La conformité** : l'application doit respecter les règles et réglementations en matière de protection des données personnelles.
- **La fiabilité** : l'application doit être fiable et stable, avec une faible probabilité de défaillance ou d'erreur.

2.3 Analyse des besoins

L'analyse des besoins vise à identifier les exigences ainsi que les acteurs de l'application et les tâches associées à chacun.

2.3.1 Identification des acteurs

Un acteur est un rôle joué par une entité externe, qui interagit avec le système pour accomplir un objectif particulier. Les acteurs sont représentés sous forme de blocs sur le diagramme, situés à l'extérieur du système, et sont connectés aux cas d'utilisation par des flèches.

Dans notre application on a un seul acteur principal qui est le responsable de gestion personnel (RGP), et un assistant qui fait les mêmes fonctionnalités comme le RGP.

RGP : est chargé de superviser le processus du suivi de présence des employés dans l'entreprise Naftal. Cela implique de s'assurer que les employés enregistrent leurs présence de manière précise.

2.3.2 Identification des cas d'utilisations

Un cas d'utilisation décrit une suite de séquences d'actions qu'un système exécute pour atteindre l'objectif, les cas d'utilisation fournissent une structure permettant de représenter des exigences fonctionnelles dans le contexte des processus système et métier. Le Tableau ci-dessous montre les différents cas d'utilisations et l'acteur de l'application.

N° cas	Cas d'utilisation	Acteur
1	Inscription	RGP
2	Authentification	RGP
3	Liste des employés	RGP
4	Liste des justifications	RGP
5	Liste des Jours fériés	RGP
6	Liste des horaires	RGP
7	Imputer pointage	RGP
8	Feuille de présence	RGP

TABLEAU 2.1 – Cas d'utilisations

2.3.3 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un outil de modélisation utilisé pour représenter les interactions entre les acteurs et le système. Il montre comment les utilisateurs interagissent avec le système pour atteindre un objectif ou une fonctionnalité.

La Figure 2.1 présente le diagramme de cas d'utilisations de notre application.

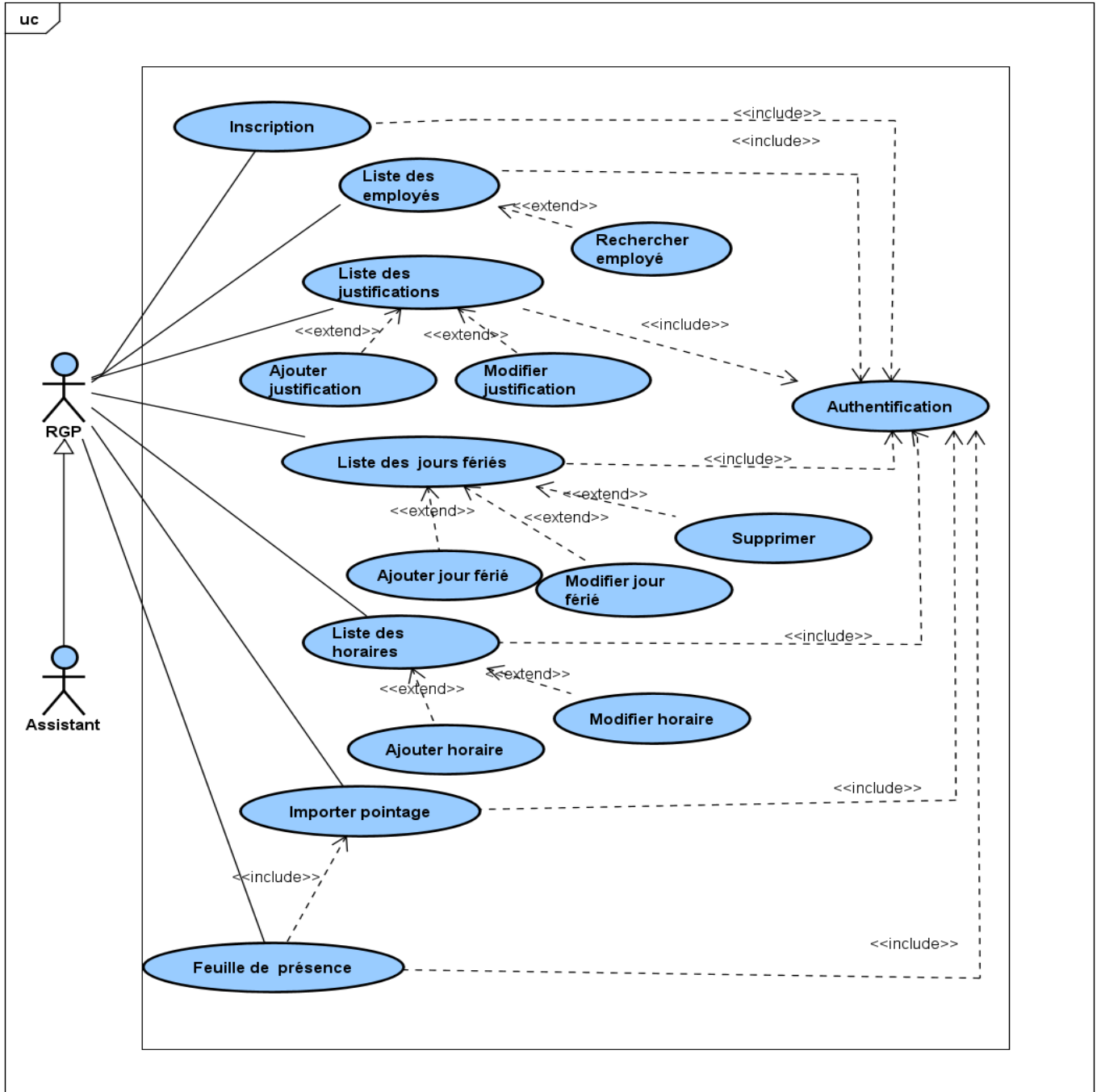


FIGURE 2.1 – Cas d'utilisations du système

2.4 Description des cas d'utilisation

2.4.1 Cas d'utilisation « Inscription »

Acteur(s)	RGP
Objectifs	- Permettre au RGP de créer un compte dans l'application.
Pré-condition(s)	- accès à l'application.
Post-condition(s)	- Le compte est créé et il peut s'identifier dans le système.
Scénario nominal	-Le RGP accède à la page d'inscription de l'application. - Le RGP entre ses informations personnelles : nom d'utilisateur, e-mail, mot de passe. -Le RGP soumet le formulaire d'inscription. - Le système vérifie que toutes les informations sont valides et uniques. - Le système crée un compte pour l'utilisateur avec les informations fournies. - Le système affiche un message de confirmation de création de compte.
Scénario alternatif	- Si l'un des champs du formulaire d'inscription n'est pas valide, le système affiche un message d'erreur correspondant. - Si l'adresse e-mail ou le nom d'utilisateur fourni par l'utilisateur est déjà utilisé par un autre compte, le système affiche un message d'erreur correspondant et demande à l'utilisateur de fournir des informations différentes.

TABLEAU 2.2 – Description du cas d'utilisation « Inscription »

Diagramme de séquence

La Figure 2.2 présente le diagramme de séquence du cas d'utilisation « Inscription »

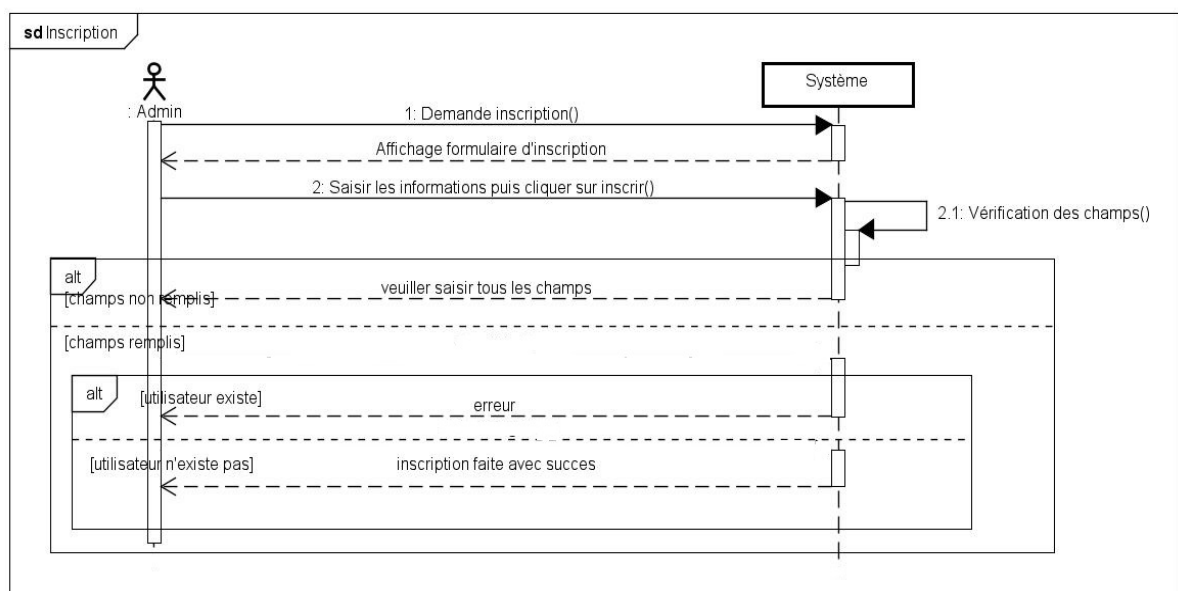


FIGURE 2.2 – Diagramme de séquence de cas d'utilisation « Inscription »

2.4.2 Cas d'utilisation « Authentification »

Acteur(s)	RGP
Objectifs	- Accéder à l'application pour que les acteurs puissent se servir des fonctionnalités de l'application. - Garantir la confidentialité.
Pré-condition(s)	-Le RGP doit être enregistré dans le système.
Post-condition(s)	- Si le RGP est authentifié avec succès, il peut accéder aux fonctionnalités de l'application. - Si le RGP n'est pas authentifié, il ne peut pas accéder aux fonctionnalités de l'application.
Scénario nominal	- Le RGP accède à la page d'authentification de l'application. - L'application affiche les champs d'identification, tels que le nom d'utilisateur et le mot de passe. - Le RGP entre ses informations d'identification . - La vérification des informations d'identification de l'utilisateur avec ceux stockés dans la base de données d'utilisateurs de l'application. - Si les informations d'identification sont valides, l'application authentifie l'utilisateur et lui accorde l'accès aux fonctionnalités restreintes. - Si les informations d'identification ne sont pas valides, l'application rejette l'authentification.
Scénario alternatif	- Si le système ne parvient pas à vérifier les informations d'identification de l'utilisateur, il affiche un message d'erreur indiquant que l'authentification a échoué.

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

Diagramme de séquence

La Figure 2.3 présente le diagramme de séquence du cas d'utilisation « Authentification » :

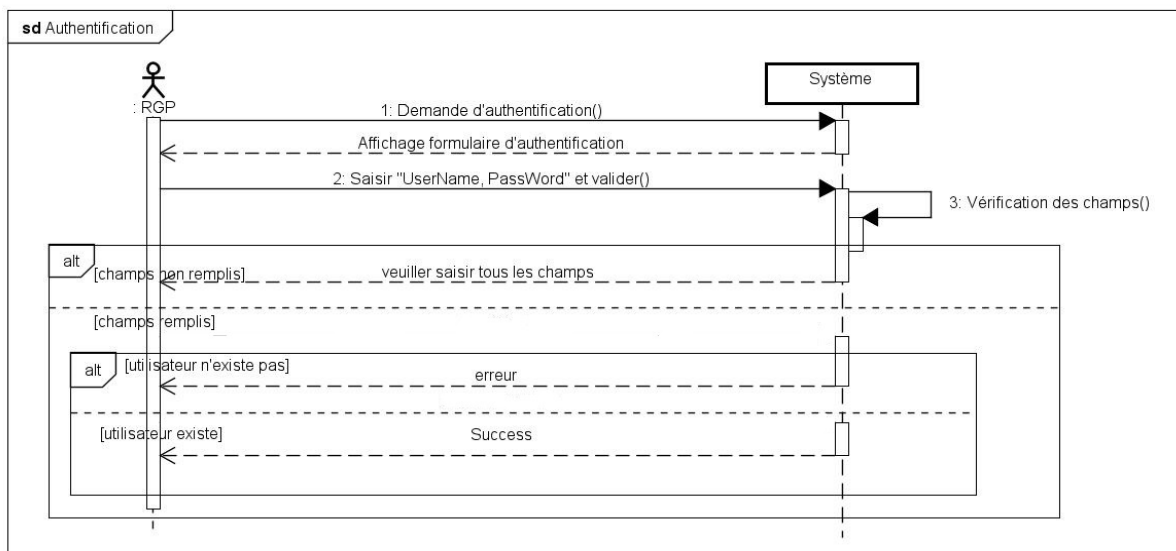


FIGURE 2.3 – Diagramme de séquence du cas d'utilisation « Authentification »

2.4.3 Cas d'utilisation « Liste des employés »

Acteur principal	RGP
Objectif	Récupérer la liste des employés à partir de la base de données
Pré-conditions	- Le RGP doit avoir les droits d'accès appropriés à la base de données. - La base de données doit être disponible et accessible.
Post-conditions	- Le RGP dispose de la liste complète des employés récupérée depuis la base de données. - Les informations des employés sont correctement extraites et présentées.
Scénario nominal	-Le RGP se connecte à l'application ou au système qui permet d'accéder à la base de données. - Le RGP sélectionne l'option "Liste des employés" dans l'interface de l'application. - Le système vérifie les droits d'accès du RGP à la base de données. - Le système récupère les données des employés depuis la base de données. - Le RGP peut effectuer des actions supplémentaires sur la liste des employés, telles que la recherche, le tri
Scénario alternatif	- Si la base de données n'est pas disponible, le système affiche un message d'erreur indiquant l'indisponibilité de la base de données.

TABLEAU 2.4 – Description du cas d'utilisation « Liste des employés »

Diagramme de séquence

La Figure 2.4 présente le diagramme de séquence du cas d'utilisation « Rechercher employé » :

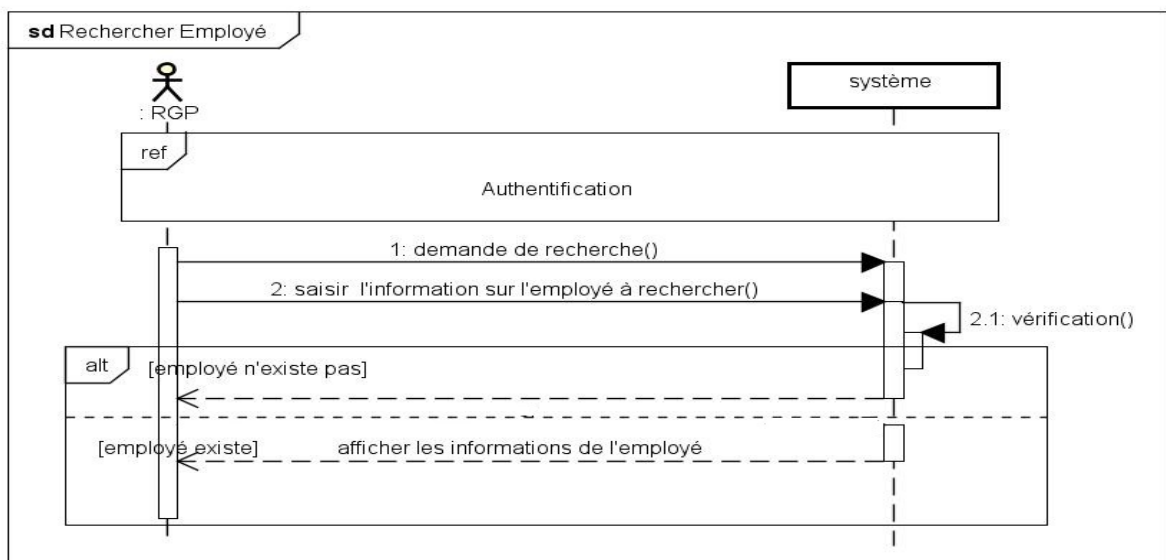


FIGURE 2.4 – Diagramme de séquence du cas d'utilisation « Rechercher employé »

2.4.4 Cas d'utilisation « Liste des justifications »

Acteur(s)	RGP
Objectifs	- Permettre à un responsable d'ajouter une justification soumise par un employé pour expliquer son absence ou son retard au travail.
Pré-condition(s)	Le responsable a les droits d'accès nécessaires pour ajouter une justification.
Post-condition(s)	- Si l'employé a justifié son absence ou son retard, la justification sera enregistrée dans les justificatifs.
Scénario nominal	<ul style="list-style-type: none"> - L'employé soumet une justification pour expliquer son absence ou son retard au travail. - Le responsable accède à la liste des justifications sou- mises et sélectionne celle qu'il souhaite gérer. - Le système affiche la justification et permet au respon- sable de l'ajouter et de la modifier. - Si la justification est ajoutée, elle sera enregistrée dans la liste des justificatifs. - Si la justification est modifiée, la liste des justificatifs sera modifier.
Scénario alternatif	- Si le responsable n'est pas convaincu par la justification fournie, il peut demander des informations complémen- taires à l'employé pour étayer sa justification.

TABLEAU 2.5 – Description du cas d'utilisation « Liste des justifications »

Diagramme de séquence

La Figure 2.5 présente le diagramme de séquence du cas d'utilisation « Liste des justifica- tions »

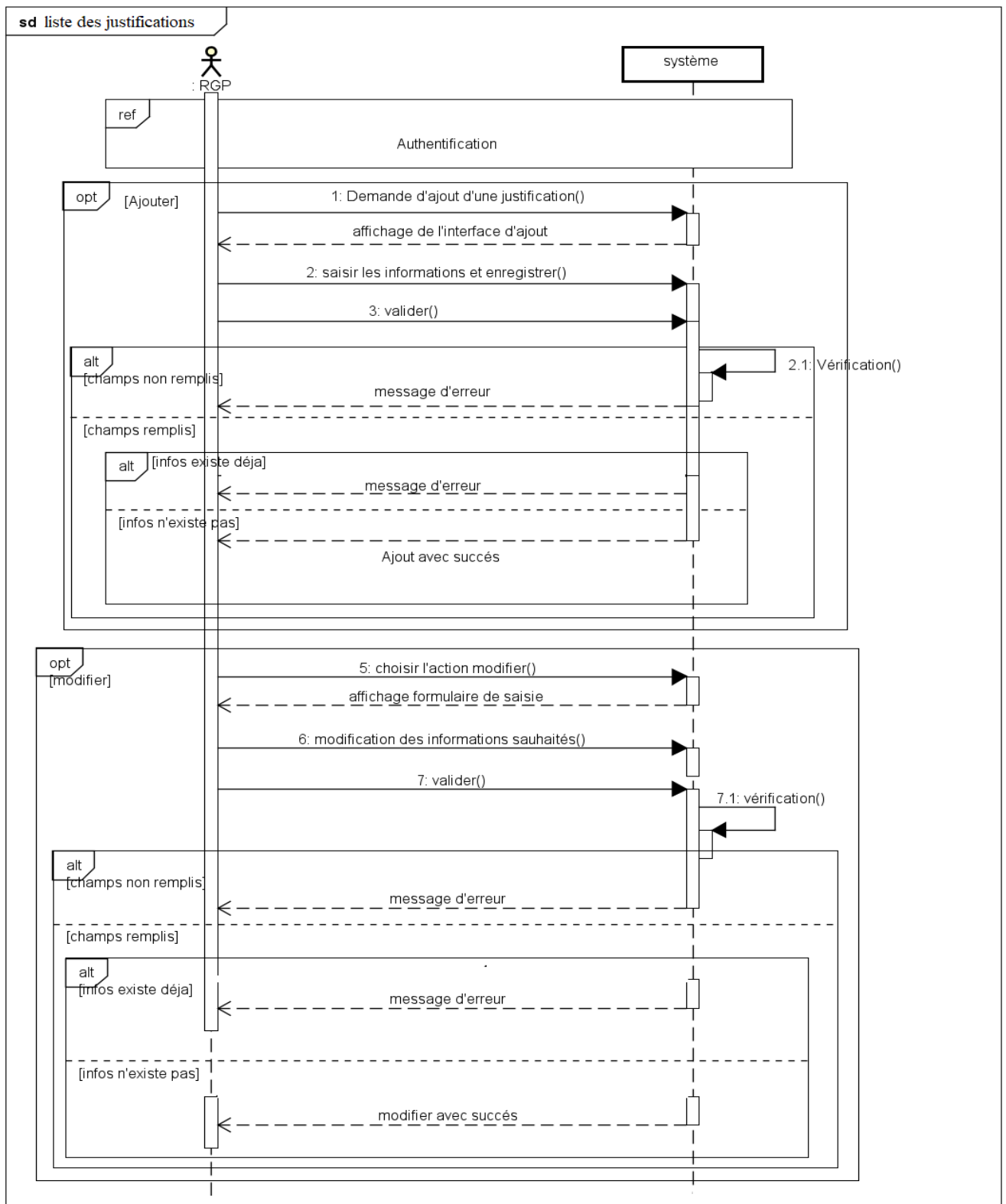


FIGURE 2.5 – Diagramme de séquence du cas d'utilisation «Liste des justifications »

2.4.5 Cas d'utilisation « Liste des jours fériés»

Acteur(s)	RGP
Objectifs	- Permettre à un responsable de gérer les jours fériés de l'entreprise et de s'assurer que les employés ne sont pas programmés pour travailler ces jours-là.
Pré-condition(s)	- Le responsable a les droits d'accès nécessaires pour gérer les jours fériés. - La liste des jours fériés doit être disponible dans le système.
Post-condition(s)	- Les employés ne sont pas programmés pour travailler les jours fériés définis dans le calendrier.
Scénario nominal	1. Le responsable accède à la liste des jours fériés dans le système. 2. Le système affiche la liste des jours fériés programmés pour l'année en cours. 3. Le responsable peut ajouter ou modifier des jours fériés si nécessaire. 4. Si le responsable ajoute ou modifie un jour férié, le système met à jour la liste.
Scénario alternatif	- Si le responsable souhaite ajouter un nouveau jour férié, le système vérifie que le jour sélectionné n'est pas déjà un jour de travail pour les employés.

TABLEAU 2.6 – Description du cas d'utilisation « Liste des jours fériés »

Diagramme de séquence

La Figure 2.6 présente le diagramme de séquence du cas d'utilisation « Liste des jours fériés»

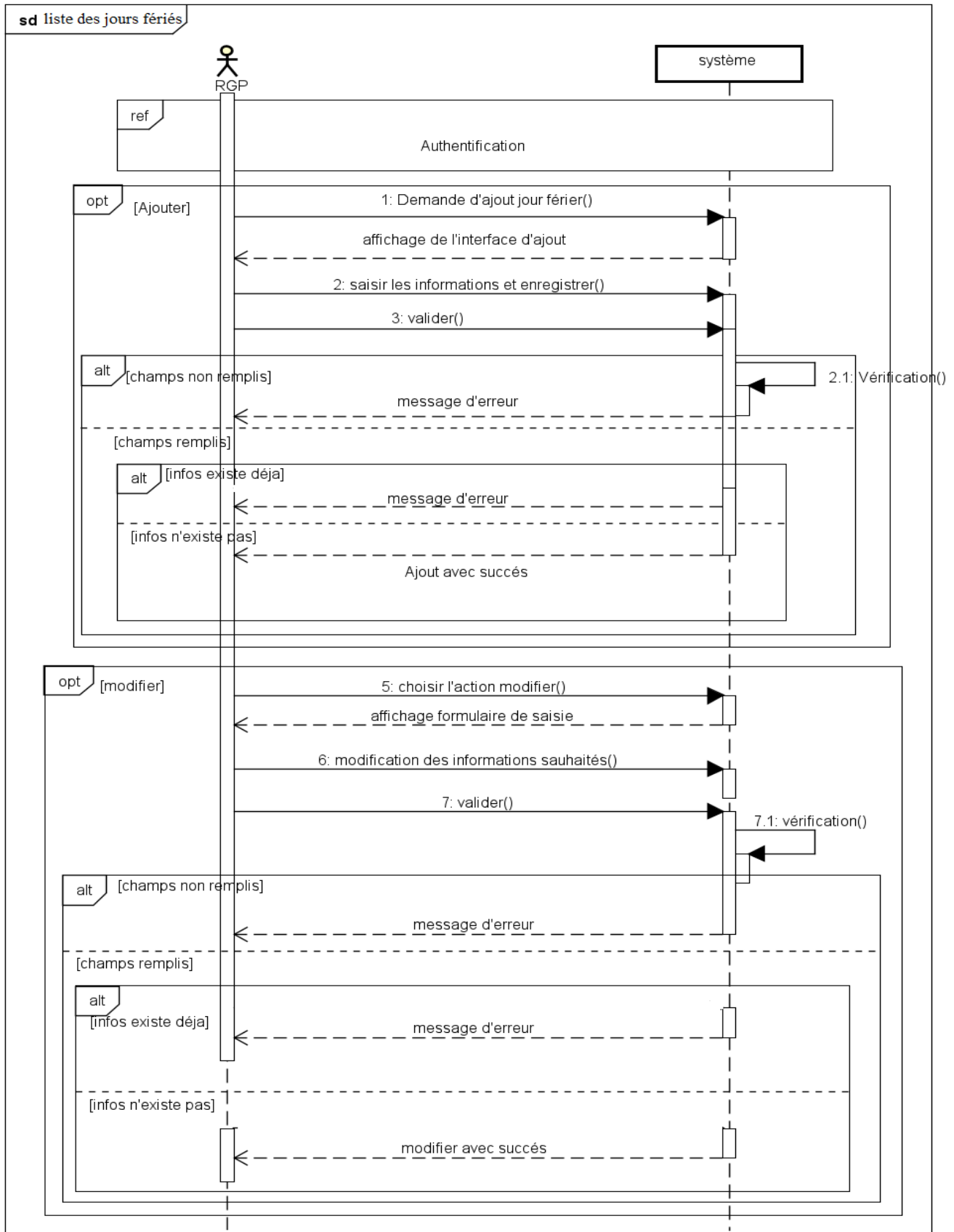


FIGURE 2.6 – Diagramme de séquence du cas d'utilisation «Liste des jours fériés »

2.4.6 Cas d'utilisation « Liste des horaires »

Acteur(s)	RGP
Objectifs	- Permettre au Responsable de la gestion du projet (RGP) de créer des horaires pour les employés dans le système.
Pré-condition(s)	- Les informations des employés doivent être enregistrées dans le système. - Le RGP doit disposer des autorisations nécessaires pour créer des horaires.
Post-condition(s)	- Les horaires sont créés avec succès pour les employés. - Les horaires sont enregistrés dans le système et disponibles pour consultation.
Scénario nominal	- Le RGP accède à l'interface de gestion des horaires du système. - Le RGP sélectionne la formulaire de création d'un nouvel horaire. - Le RGP définit les détails de l'horaire, tels que les heures de début et de fin. - Le RGP enregistre l'horaire dans le système. - Le système affiche une notification indiquant que l'horaire a été créé avec succès.
Scénario alternatif	- Le RGP peut choisir de modifier l'horaire conflictuel pour résoudre le conflit

TABLEAU 2.7 – Description du cas d'utilisation « Liste des horaires »

Diagramme de séquence

La Figure 2.7 présente le diagramme de séquence du cas d'utilisation « Liste des horaires »

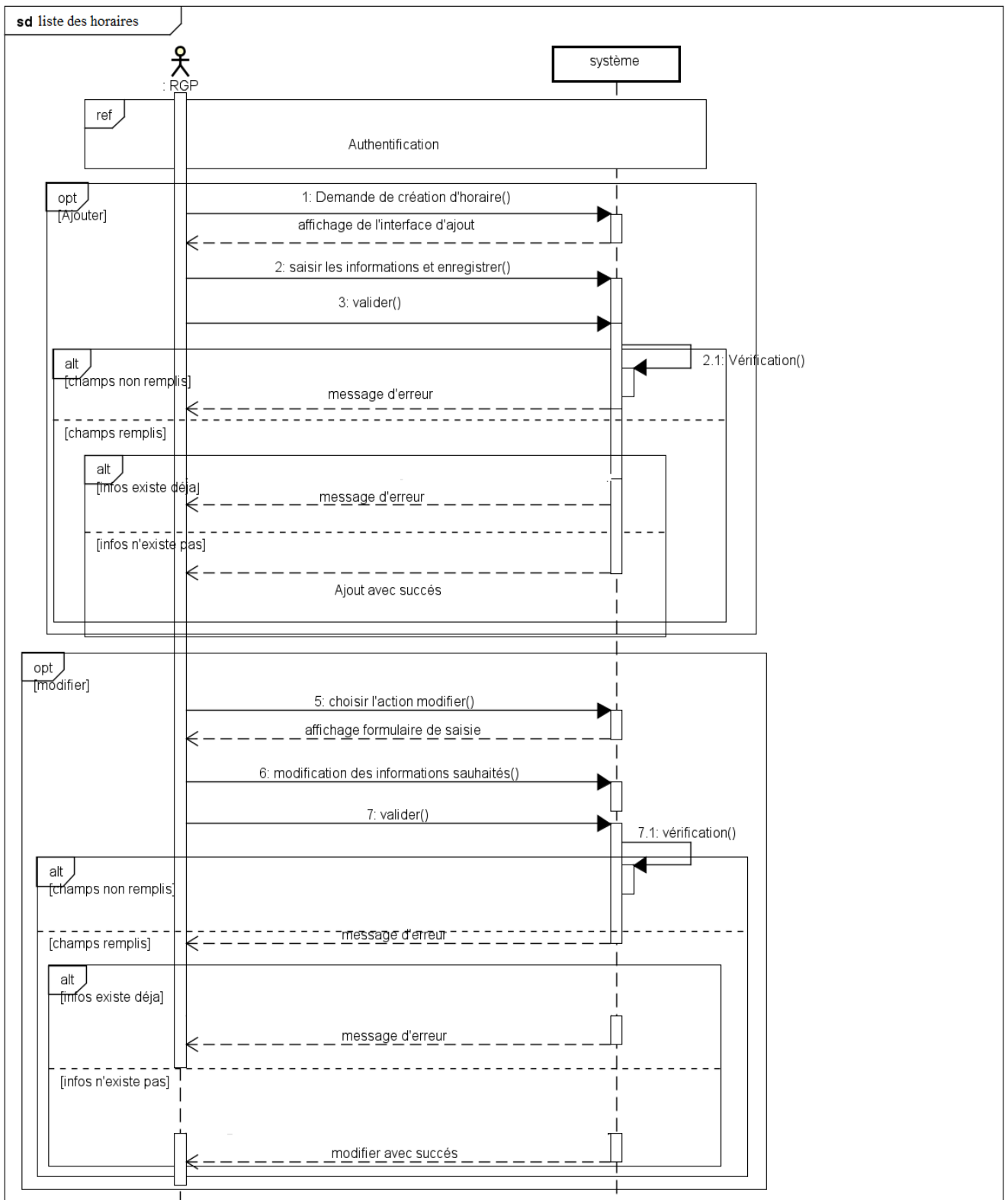


FIGURE 2.7 – Diagramme de séquence du cas d'utilisation «Liste des horaires »

2.4.7 Cas d'utilisation «Importer pointage»

Acteur(s)	RGP
Objectif	Permettre au RGP d'importer des données à partir d'une base de donnée Access externe dans l'application.
Pré-conditions	- L'utilisateur doit être connecté à l'application - Le fichier de base de données Access qui doivent être disponibles.
Post-conditions	- Les données de fichier de bdd Access sont importées avec succès dans la bdd de l'application. - L'utilisateur peut accéder aux données via l'application.
Scénario nominal	L'utilisateur sélectionne l'option importer BDD dans le menu de l'application. - L'utilisateur clic sur le bouton "importer pointage" le système affiche le fichier à importer Le système traite et l'importer le fichier Access. L'utilisateur peut maintenant accéder aux données importées dans l'application.
Scénario alternatif	- Données non valides : Si le fichier de base de données Access contient des données non valides ou manquantes, le système peut générer un rapport d'erreur indiquant les problèmes rencontrés.

TABLEAU 2.8 – Description du cas d'utilisation «Importer pointage»

Diagramme de séquence

La Figure 2.8 présente le diagramme de séquence du cas d'utilisation « Importer pointage »

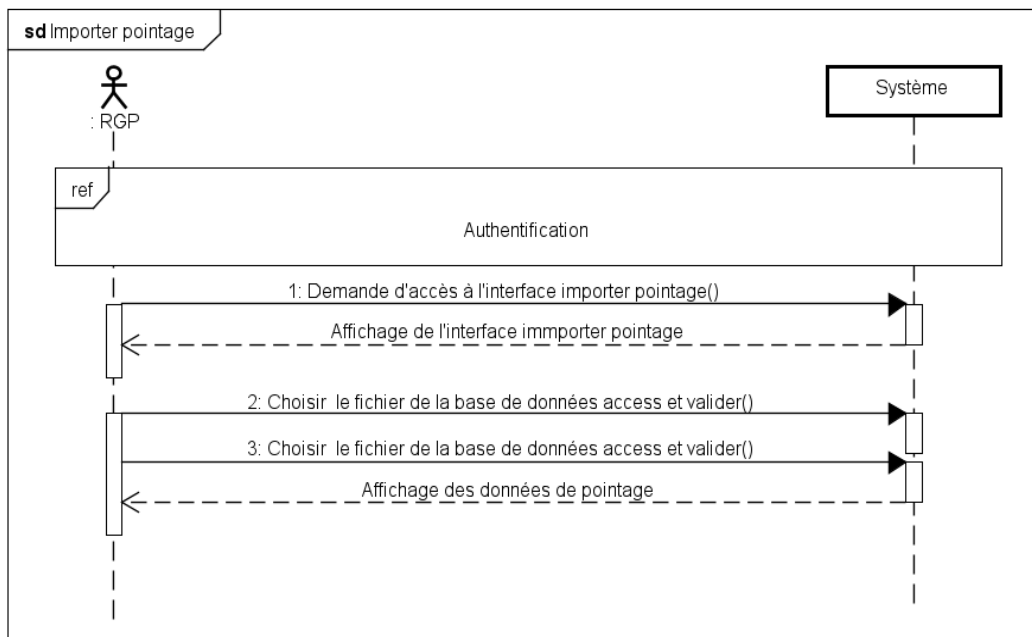


FIGURE 2.8 – Diagramme de séquence du cas d'utilisation «Importer pointage »

2.4.8 Cas d'utilisation « Feuille de présence »

Acteur(s)	RGP
Objectif	Permettre au RGP de gérer la feuille de présence des employés dans l'application
Pré-conditions	- Les informations des employés doivent être enregistrées dans l'application - Le RGP doit disposer des autorisations nécessaires pour gérer la feuille de présence
Post-conditions	- La feuille de présence d'un employé est établie et prête à imprimer.
Scénario nominal	Le RGP accède à l'interface, de la feuille de présence Le système affiche la feuille de présence Le RGP filtre par employé et spécifie le mois pour l'établissement de sa feuille de présence Le RGP imprime la feuille de présence souhaité. Le système affiche une confirmation de la mise à jour de la feuille de présence

TABLEAU 2.9 – Description du cas d'utilisation « Feuille de présence »

Diagramme de séquence

La Figure 2.9 présente le diagramme de séquence du cas d'utilisation « Feuille de présence »

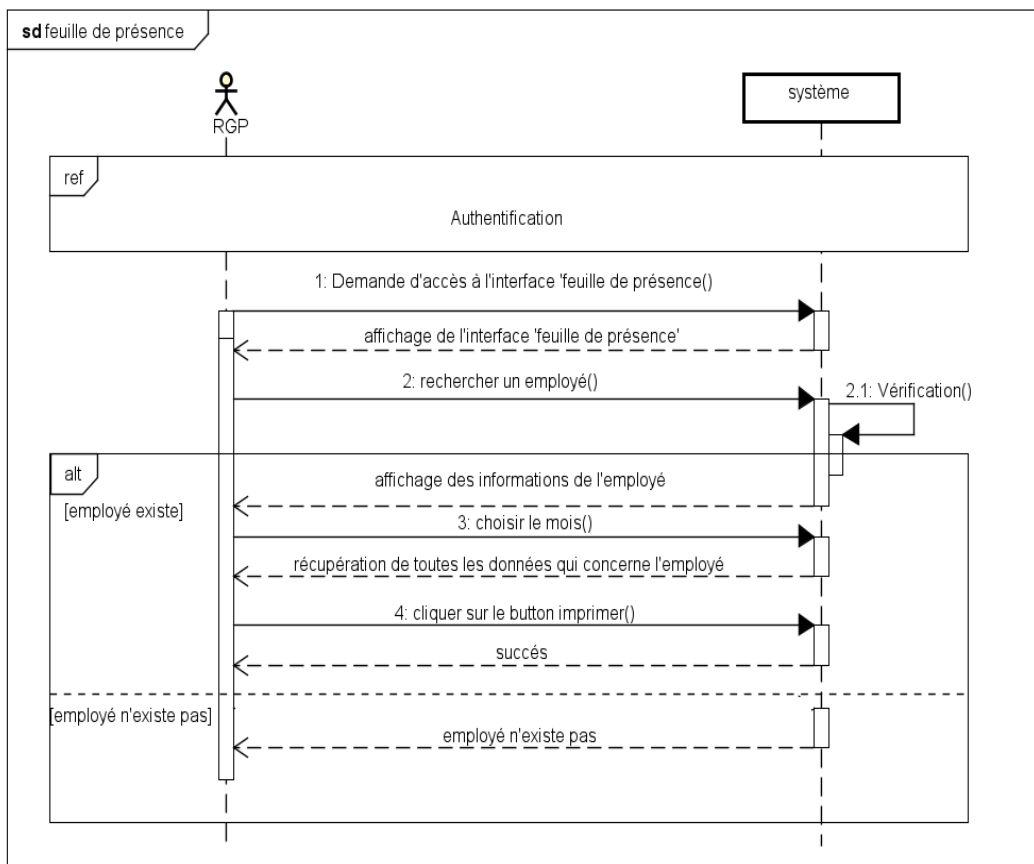


FIGURE 2.9 – Diagramme de séquence du cas d'utilisation « Feuille de présence »

2.5 Diagramme de classe

Un diagramme de classe est un outil de modélisation utilisé pour représenter la structure et les relations entre les classes dans un système. Il montre les attributs et les méthodes de chaque classe, ainsi que les relations entre les classes, telles que l'héritage, l'association et la dépendance. La Figure 2.10 présente le diagramme de classe de notre application :

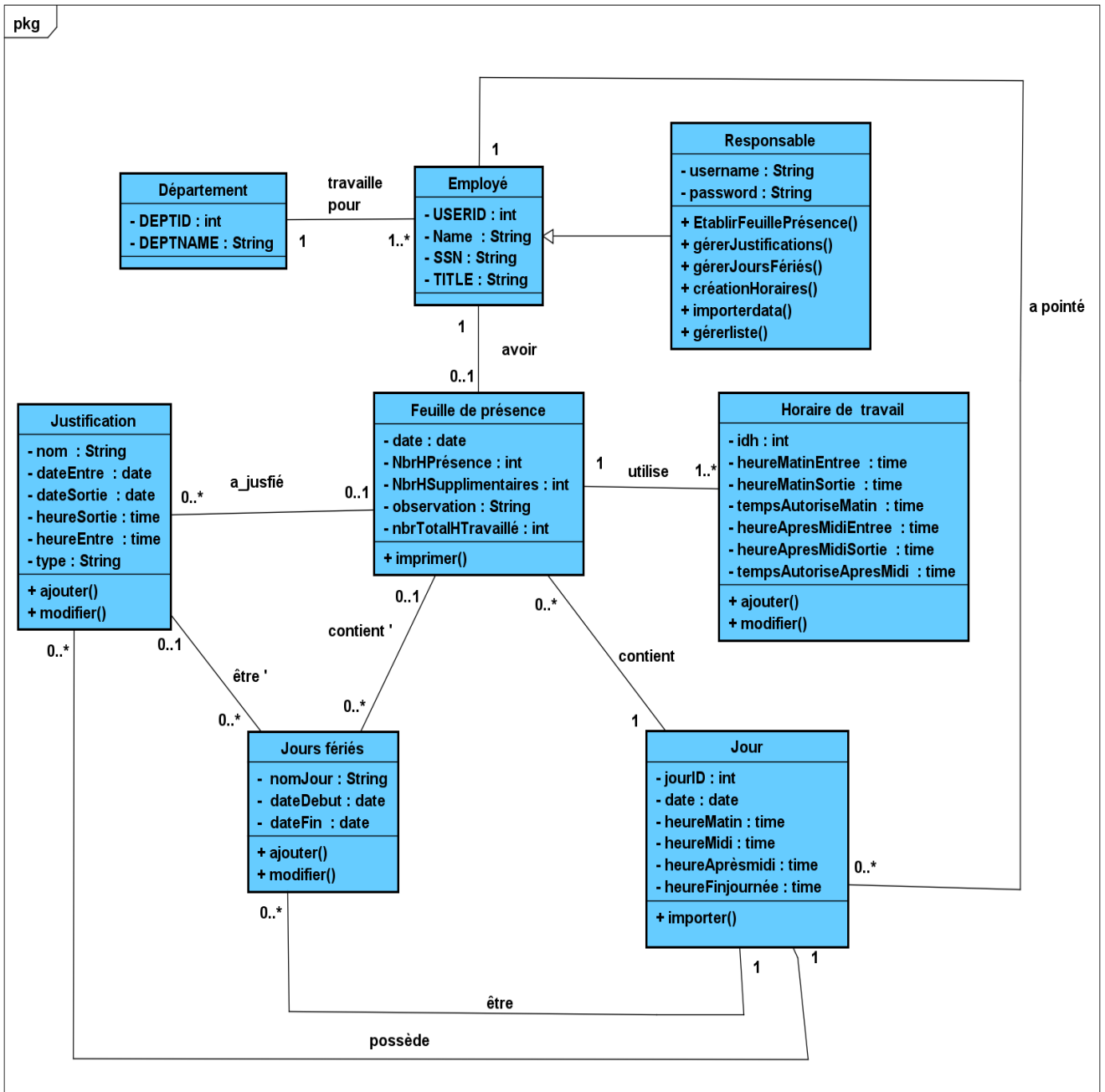


FIGURE 2.10 – Diagramme de classe

2.6 Conclusion

La conception elle joue un rôle essentiel dans la réalisation d'un projet de l'application web pour gestion de présence des employés pour Naftal. Nous avons identifié les exigences fonctionnelles et techniques du système, ainsi que les contraintes liées à l'environnement de déploiement. Nous avons également utilisé des diagrammes et des scénarios pour représenter et tester les différentes parties de l'application. Cette phase est cruciale pour garantir le succès de la mise en œuvre de l'application et pour répondre aux besoins de l'entreprise Naftal.

Chapitre 3

Réalisation et implémentation de l'application

3.1 Introduction

Dans ce chapitre, nous présentons les différentes étapes de développement de l'application de gestion de présence des employés. Nous décrivons également les outils, les frameworks utilisés, l'architecture de l'application et les fonctionnalités à implémentées.

3.2 Outils et logiciels de programmation

Pour le développement de notre application, nous avons utilisés les outils et logiciels suivants :

3.2.1 Visual Studio Code

C'est un éditeur de code source gratuit et open-source qui est développé par Microsoft. C'est un outil léger et puissant pour le développement logiciel. Prenant en charge de nombreux langages de programmation et offre des fonctionnalités telles que la coloration syntaxique, l'indentation automatique, la complétion de code, la détection d'erreurs, et propose de nombreux extensions pour personnaliser l'environnement de développement [8].

3.2.2 NodeJS

NodeJS est une plateforme de développement back-end basée sur le moteur JavaScript, adaptée pour créer des serveurs web, des API REST. C'est open source et compatible avec Windows, Mac et Linux[9].

3.2.3 MongoDB Compass

MongoDB Compass est un outil graphique pour explorer, surveiller et gérer les données. Il fournit une interface utilisateur intuitive pour se connecter à une instance MongoDB, naviguer à travers les collections de données, visualiser et analyser les données, exécuter des requêtes, créer des index et gérer la configuration du cluster MongoDB[10].

3.2.4 Postman

Postman est un outil de collaboration et de développement d'API qui permet aux développeurs de tester, de documenter et de déboguer les API. Il fournit une interface utilisateur graphique conviviale pour envoyer des requêtes HTTP et des réponses, ainsi que pour visualiser et analyser les résultats. Postman prend en charge un large éventail de formats de données, tels que JSON et XML, et offre des fonctionnalités avancées telles que la gestion des cookies et des autorisations d'authentification. Il est disponible sous forme d'application de bureau et d'extension de navigateur[11].

3.3 Langages et frameworks de programmation

Les langages de programmation et frameworks suivantes ont été utilisées pour développer notre l'application :

3.3.1 React

React est une bibliothèque JavaScript open-source développée par Facebook, comme un framework front-end qui permet de créer des interfaces utilisateur (UI) interactives pour les applications web. Elle permet de concevoir des composants d'interface utilisateur réutilisables qui gèrent leur propre état, puis de les composer pour former des interfaces utilisateur complexes[6].

3.3.2 Bootstrap

Bootstrap est un framework open-source pour la création de sites et d'applications web. Il fournit un ensemble de classes CSS, de composants JavaScript et d'outils de développement pour faciliter la création d'interfaces utilisateur responsives et attractives[12].

3.3.3 ExpressJS

ExpressJS est comme framework back-end open-source pour le développement d'applications web côté serveur en JavaScript. Il est basé sur NodeJS et fournit des fonctionnalités pour la gestion des requêtes HTTP, des routes, des modèles de données et de nombreux autres aspects liés au développement d'applications web[13].

3.3.4 Javascript

Javascript est un langage de type script, orientée objet utilisé pour créer des applications web interactives et dynamiques, ce qui signifie d'être un langage de programmation côté client dans un navigateur web et utilisé dans côté serveur avec des environnements comme NodeJS[14].

3.3.5 HTML(HyperText Markup Language)

est un langage de description de format de document qui se présente sous la forme d'un langage de balisage pour définir la structure et l'organisation des éléments d'une page web[18].

3.3.6 CSS(Cascading Style Sheets)

CSS est un langage de feuille de style utilisé pour définir la présentation et la mise en forme des éléments d'une page web écrite en HTML[19].

3.4 Implémentation du Modèle-Vue-Contrôleur(MVC)

Le modèle MVC (Modèle-Vue-Contrôleur) est un modèle architectural qui permet de séparer les différents composants de l'application en trois parties : le modèle qui gère la logique métier, la vue qui s'occupe de l'affichage et le contrôleur qui gère la communication entre les deux.

La structuration de l'application gestion des présences des employés avec le modèle MVC est faite de la façon suivante :

— Modèle

Le modèle représente la couche de données de l'application et est responsable de la gestion de la base de données et de la manipulation des données. Pour notre application, le modèle correspond à une base de données MongoDB qui stocke les données de l'application et les opérations CRUD ((Create, Read, Update, Delete) qui y sont liées. L'exemple ci-dessous illustre un exemple d'application du modèle MVC ajouter justification dans la vue gestion des justifications

```
const mongoose = require('mongoose')
const justificationSchema = new mongoose.Schema({
  nom: {
    type: String,
    required: true
  },
  dateSortie: {
    type: String,
    required: true
  },
  dateEntree: {
    type: String,
    required: true
  },
  heureSortie : {
    type: String,
    required: true
  },
  heureEntree: {
    type: String,
    required: true
  },
  type: {
    type: String,
    required: true
  }
})

const justification= new mongoose.model('justification',justificationSchema);

module.exports = justification;
```

FIGURE 3.1 – Modèle justification.js

— Vue

La vue représente la couche de présentation de l'application. Elle est responsable de l'affichage des données à l'utilisateur final. Pour notre application, la vue serait représentée par l'interface utilisateur (UI) qui afficherait le menu de notre application ainsi ces différents interfaces en utilisant reactjs.

La vue ajouter justification présente une interface qui permet d'afficher les données du modèle justification avec un format compréhensible et visible.

La Figure 3.2 présente la vue Ajouter justification



Ajouter justification

Nom: Mourad Albani

Date de sortie: 01/02/2022

Date d'entrer: 02/02/2023

Heure de sortie: --:--

heure d'entrer: --:--

Type de justification: congé de récupération

Enregistrer

FIGURE 3.2 – Vue Ajouter justification

— Contrôleur

Le contrôleur représente la couche de contrôle de l'application. Il est responsable de la gestion des actions de l'utilisateur et de la coordination entre la vue et le modèle. Pour notre application, le contrôleur serait représenté par l'application Express.js qui traiterai les requêtes http, appellerait les fonctions appropriées du modèle et rendrait les vues appropriées à l'utilisateur.

Exemple : Il agit comme un intermédiaire entre le modèle justification et la vue ajouter justification. Il gère les interactions entre ces deux composants et il permet d'ajouter une justification .

La Figure ci-dessous montre un modèle du contrôleur.

```
const Ajouterjustification= async (req,res) => {  
  
  const data = new justification(req.body)  
  const resultat = await data.save()  
  
  if (!resultat) {  
    res.json({  
      status: "Echec",  
      message: "erreur"  
    })  
  }  
  else {  
    res.json({  
      status: "succès",  
      message: "justification insérer avec succès",  
    })  
  }  
}  
module.exports = Ajouterjustification
```

FIGURE 3.3 – Contrôleur Ajouterjustification

- Création des routes : nous avons créé des routes pour connecter les contrôleurs et les vues. Les routes sont utilisées pour gérer les demandes de l'utilisateur et pour renvoyer les réponses appropriées.

En résumé, le modèle MVC pour notre l'application de gestion de présence des employés est composé de la base de données MongoDB comme modèle, l'interface utilisateur pour la vue et Express.js pour le contrôleur.

3.5 Présentation de l'application

3.5.1 Nom d'application

L'application est nommée « Présence_Management » c'est une application de gestion de présence des employés qu'est un outil essentiel pour les entreprises qui souhaitent gérer efficacement la présence de leurs employés. L'application offre une interface conviviale pour le responsable de RGP d'accéder aux différentes fonctionnalités, notamment la Listes des employés, la gestion des Justifications, la gestion des jours fériés, la création des horaires, importation de pointage ainsi que la génération de feuille de présence.

Afin de présenter les interfaces que nous avons fait pour cette application nous présentons tout d'abord quelques étapes illustré ci-dessous pour comprendre l'implémentation de l'application qui est basée sur :

- **Front-end**

la partie visible de l'application, celle qui interagit avec l'utilisateur a été développé en utilisant la bibliothèque React et le framework Bootstrap. Nous avons utilisé React pour la gestion de l'état de l'application et la construction de l'interface utilisateur, et Bootstrap pour le design et la mise en page de l'interface.

- **Back-end**

la partie invisible de l'application, celle qui gère la logique métier et la communication avec la base de données a été développé en utilisant le framework Express et l'environnement Node.js. Nous avons créé une API REST pour permettre la communication entre le front-end et la base de données.

- **API**

une interface de programmation d'application, qui permet à deux applications de communiquer entre elles, Nous avons souligné cette importance afin de faire et la communication entre le front-end et le back-end à travers une API REST pour faciliter le développement de l'application.

3.6 Quelques vues de l'application

Notre application comporte un seul type d'utilisateur qui est le responsable gestion de présence qui a accès à toutes les vues de l'application, nous allons présenter un ensemble de vues de notre application en commençons par :

Logo de l'application

Le logo est un symbole graphique qui permet d'identifier une entreprise dans notre projet on parle sur Naftal, nous avons fait le logo de qui référencé a l'entreprise Naftal qui est une image associée a celui-ci :



(a) Logo sur le front blanc



(b) Logo sur le front noir

3.6.1 Présentation des vues

- Authentification
- Inscription
- Menu
- Liste des employés
- Liste des horaires
- Ajouter horaire
- Modifier horaire
- Liste des jours fériés
- Ajouter jour férié
- Modifier jour férié
- Liste des justifications
- Ajouter justification
- Modifier justification
- Importer pointage
- Feuille de présence

3.6.2 Vue «d'accueil »

La page d'accueil de notre application c'est la vue authentification où le RGP accède à son application via une connexion en saisissant son nom d'utilisateur et son mot de passe.

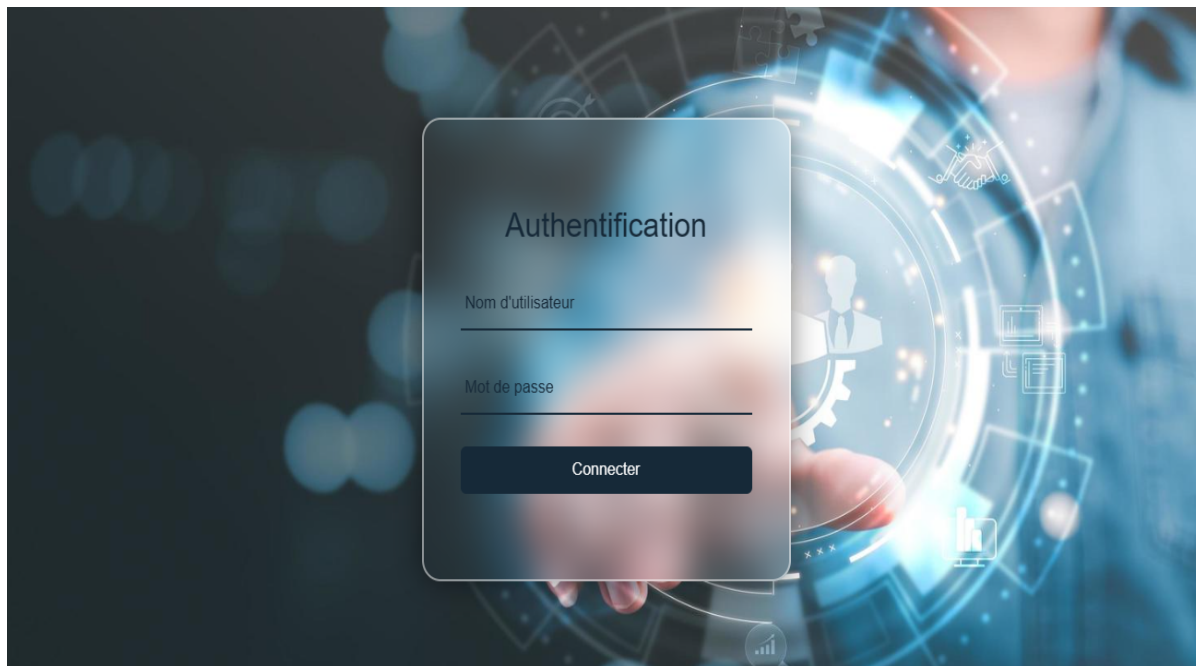


FIGURE 3.5 – Vue « Authentification »

3.6.3 Vue «Inscription »

Le RGP inscrit pour accéder à application.



FIGURE 3.6 – Vue « Inscription »

3.6.4 Vue «Menu »

Le Menu de l'application illustre l'ensemble des vues interactive qui permet au RGP d'accéder aux différentes fonctionnalités et sections de l'application.

The screenshot shows a sidebar menu on the left with the following items: Liste des employés, Liste des horaires, Liste des jours fériés, Liste des justifications, Importer pointage, Feuille de présence, Inscription, and Déconnexion. The main content area is titled 'Rechercher par département' and features a dropdown menu with 'Choose...' selected and a search input field labeled 'search Name...'. Below this is a table with the following data:

USERID	SSN	NAME	TITLE	DEFAULTDEPTID
233	14259N	OUEZGDOUH TATA	CADRE D'ETUDES N2	21
234	29374R	HAMLAT HASSIBA	CHEF SCE SYS/RES	21
235	43156K	CHEURFA ABDELH...	ANALYSTE N2	21
236	22012N	ALBANI MOURAD	CHEF DPT INF	14
237	04369R	RABHI NABILA	CHEF SCE PRODU...	20
238	21577B	Ouadah Amel	CADRE TECHNIQU...	20
239	15781U	MOHAMADI RACHID	CHEF DP EXPLOITA...	14
240	15631P	BOUDEHOUCHE HA...	CHARGE D'ETUDES	20
241	14255E	BENMOUHOU B ME...	CHEF SCE TRANSP...	20
242	73289S	BELKACEMI YASMINA	CHARGE D'ETUDES	20

FIGURE 3.7 – Vue « Menu »

3.6.5 Vue «Liste des employés »

Le RGP accède à la page listes employés et filtre les employés selon leurs départements et la recherche selon le nom d'un employé.

The screenshot shows the 'Liste des employés' view. It features a sidebar menu on the left with the following items: Liste des employés, Liste des horaires, Liste des jours fériés, Liste des justifications, Importer pointage, Feuille de présence, Inscription, and Déconnexion. The main content area is titled 'Rechercher par département' and features a dropdown menu with 'Choose...' selected and a search input field labeled 'search Name...'. Below this is a table with the following data:

USERID	SSN	NAME	TITLE	DEFAULTDEPTID
472	13248P	BOUDRAA HAKIMA	CHEF SCE GES PERS	17
531	42896V	CHIBANE TAYEB	CHEF SCE MRO	19
737	73162K	MAALIOU YASMINA	CONTROLEUR GES...	15

At the bottom of the table, there is a pagination control: 'Rows per page: 10' (with a dropdown arrow), '71-73 of 73', and navigation arrows (left, right, first, last).

FIGURE 3.8 – Vue « Liste des employés »

3.6.6 Vue «Liste des horaires »

Le RGP de l'entreprise peut ajouter où modifier des horaires.

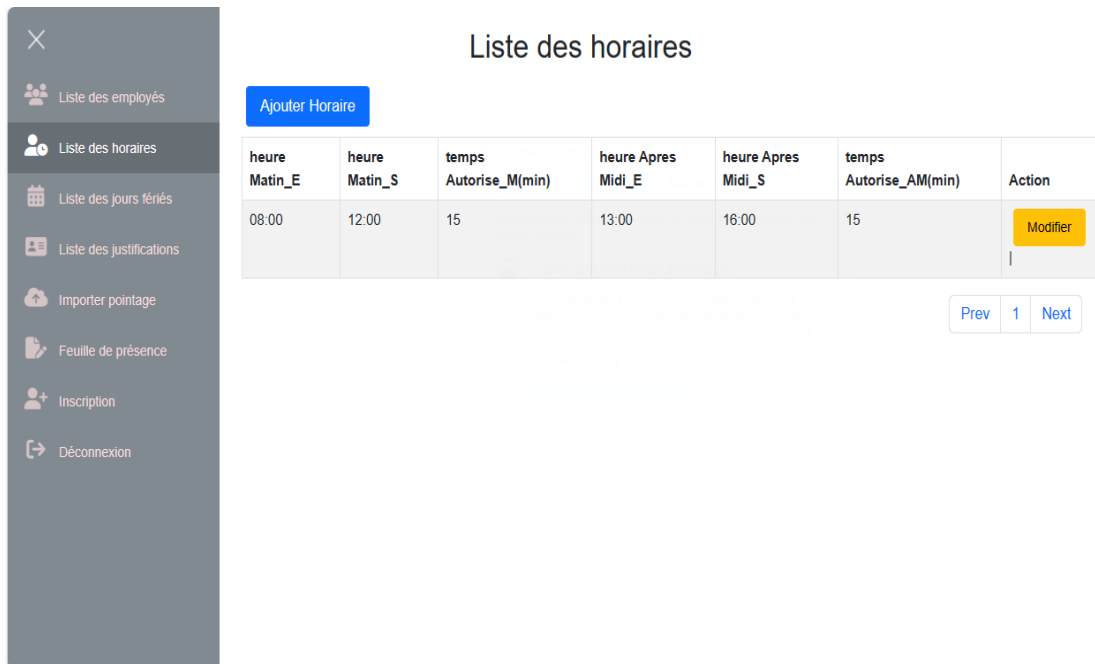


FIGURE 3.9 – Vue « Liste des horaires »

3.6.7 Vue «Ajouter horaire »

La figure ci-dessous illustre la vue ajouter horaire.

Ajouter horaire

heureMatinEntree: 08:00

heureMatinSortie: 12:00

tempsAutoriseMatin: 15

heureApresMidiEntree: 13:00

heureApresMidiSortie: 16:00

tempsAutoriseApresMidi: 15

Enregistrer

FIGURE 3.10 – Vue « Ajouter horaire »

3.6.8 Vue «Modifier horaire »

La figure ci-dessous illustre la vue modifier un horaire.

Modifier horaire

heureMatinEntree
08:00

heureMatinSortie
12:00

tempsAutoriseMatin

heureApresMidiEntree
13:00

heureApresMidiSortie
16:00

tempsAutoriseApresMidi
15

Enregistrer

FIGURE 3.11 – Vue « Modifier horaire »

3.6.9 Vue «Liste des jours fériés»

Le RGP de l'entreprise peut ajouter ou modifier un jour férié, et de consulter la liste des jours fériés .

Liste des jours fériés

Ajouter Jour Férié

date debut	date fin	nom jour	Action
2022-01-01	2022-01-01	Nouvel l'an	Modifier
2023-11-01	2023-11-01	Fête de la révolution	Modifier
2023-07-05	2023-07-05	Fête de l'indépendance de l'Algérie	Modifier
2023-05-01	2023-05-01	Fête des travailleurs	Modifier
2023-01-12	2023-01-12	Yennayer	Modifier

Prev 1 Next

FIGURE 3.12 – Vue «Liste des Jours Fériés »

3.6.10 Vue «Ajouter jour férié»

La figure ci-dessous illustre l'ajout d'un jour férié .

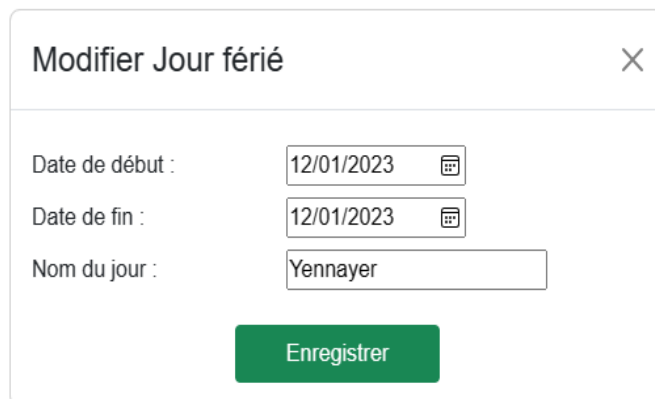


The screenshot shows a modal window titled "Ajouter jour férié" with a close button (X) in the top right corner. The form contains three input fields: "Date de début :" with the value "01/11/2023" and a calendar icon; "Date de fin :" with the value "01/11/2023" and a calendar icon; and "Nom du jour :" with the value "Fête de révolution". Below the fields is a green button labeled "Enregistrer".

FIGURE 3.13 – Vue «Ajouter Jour Férié »

3.6.11 Vue «Modifier jour férié»

La figure ci-dessous illustre la modification d'un jour férié .

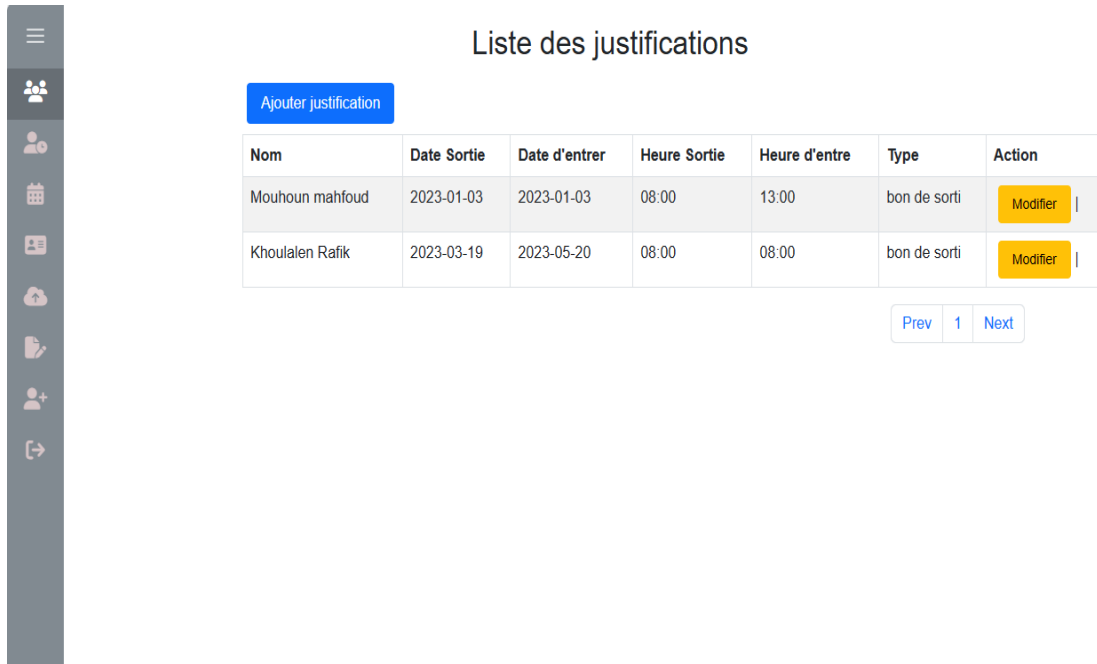


The screenshot shows a modal window titled "Modifier Jour férié" with a close button (X) in the top right corner. The form contains three input fields: "Date de début :" with the value "12/01/2023" and a calendar icon; "Date de fin :" with the value "12/01/2023" and a calendar icon; and "Nom du jour :" with the value "Yennayer". Below the fields is a green button labeled "Enregistrer".

FIGURE 3.14 – Vue «Modifier Jour Férié »

3.6.12 Vue « Liste des justifications »

Le RGP de l'entreprise peut ajouter ou modifier une justification aux employés pour justifier leurs absences quelque soit leur type et consulter la liste des justifications .



Nom	Date Sortie	Date d'entrer	Heure Sortie	Heure d'entre	Type	Action
Mouhoun mahfoud	2023-01-03	2023-01-03	08:00	13:00	bon de sorti	Modifier
Khoulaien Rafik	2023-03-19	2023-05-20	08:00	08:00	bon de sorti	Modifier

FIGURE 3.15 – Vue « Liste des justifications »

3.6.13 Vue « Ajouter justification »

La figure ci-dessous illustre l'ajout d'une justification.



Ajouter justification [X]

Nom: Mourad Albani

Date de sortie: 01/02/2022

Date d'entrer: 02/02/2023

Heure de sortie: --:--

heure d'entrer: --:--

Type de justification : congé de récupération

Enregistrer

FIGURE 3.16 – Vue «Ajouter justification »

3.6.14 Vue « Modifier justification »

La figure ci-dessous illustre la modification d'une justification.

Modifier justification

Nom: Mouhoun mahfoud

Date de sortie: 03/01/2023

Date d'entrer: 03/01/2023

Heure de sortie: 08:00

heure d'entrer: 13:00

Type de justification: Bon de sorti

Modifier

FIGURE 3.17 – Vue «Modifier justification »

3.6.15 Vue «Importer pointage »

Le RGP de l'entreprise importe via l'application les données de pointage.

Importer pointage

search-date...

USER	CHECKTIME	VERIFYCODE
288	17/01/2023 07:48:59	15
288	17/01/2023 11:47:07	15
288	17/01/2023 16:16:13	15
288	18/01/2023 12:14:19	15
288	18/01/2023 13:06:47	15
288	19/01/2023 08:10:49	15
288	19/01/2023 11:57:39	15
288	01/01/2023 07:53:10	15
288	01/01/2023 11:46:42	15
288	01/01/2023 12:45:59	15

FIGURE 3.18 – Vue « Importer pointage »

3.6.16 Vue «Feuille de présence »

Sélectionner une date : 01/01/2023

Imprimer

Rechercher un employé... rechercher

Date	Jour	Présence	Heures Présences	Retard	Heures supplémentaire	Indiments	Observations
01/01	dim.						Nouvel An
02/01	lun.						
03/01	mar.						
04/01	mer.						
05/01	jeu.						
06/01	ven.						

FIGURE 3.19 – Vue « Feuille de présence »

3.7 Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes de développement et d'implémentation de notre application gestion de présence des employés. Nous avons également décrit les outils et les technologies que nous avons utilisés pour réaliser cette application. L'un des défis majeurs que nous avons rencontrés au cours de ce processus était le temps nécessaire pour importer les données d'une base de données Access dans MongoDB. Ce processus s'est avéré plus complexe et prendre du temps que prévu, à cause de la structure de données différente entre les deux systèmes. De plus, l'importation des données de pointage vers la feuille de présence a posé des défis en termes de traitement et de mise en forme des données pour assurer leur intégrité et leur précision. Malgré ces obstacles, notre approche basée sur l'architecture MVC s'avère être une solution adaptée pour garantir une séparation claire des différents composants du système et permet de surmonter avec succès ces difficultés. Enfin nous présentés quelques interface de notre application.

CONCLUSION GÉNÉRALE

Dans notre projet, nous avons souligné les principales réalisations majeures de notre application de gestion de présence des employés, notamment l'intégration des jours fériés, la migration des données vers MongoDB, l'automatisation de l'importation de pointage et le calcul automatique des heures de travail. Ces réalisations ont été justifiées par leur capacité à améliorer l'efficacité et la précision de la gestion de la présence des employés.

Pour la mise en oeuvre de notre application, nous avons choisi d'adopter la méthode UP pour notre projet, car c'est une approche itérative et incrémentale, elle nous permet de rester flexible vers l'achèvement de notre projet. Ainsi l'utilisation de diagramme UML tels que le diagramme de cas d'utilisation, le diagramme de classes et les diagrammes de séquence étaient essentiels pour visualiser l'architecture de l'application et les interactions entre les composants. Cela nous a aidé à concevoir un système bien structuré. Nous avons utilisé le Visual Studio Code comme un environnement de développement et les technologie React et Bootstrap pour développer l'interface utilisateur de notre application et Expressjs pour notre serveur back-end et nous avons utilisé une base de données mongodb pour stocker et gérer les données. Adoption de l'architecture MVC pour fournir une séparation claire des tâches au sein de l'application. Cette approche nous a permis de diviser l'application en trois composants interconnectés : un modèle pour la gestion des données, une vue pour l'interface utilisateur et un contrôleur pour la logique de traitement des requêtes.

Nous avons également identifié des perspectives d'amélioration pour notre application, notamment l'ajout de fonctionnalités de calcul des heures supplémentaires et de prendre en considération retards et la génération automatique de feuilles de présence mensuelles pour chaque employé et chaque département. Ces améliorations futures visent à répondre aux besoins évolutifs de notre entreprise et à renforcer encore notre capacité à gérer efficacement la présence des employés.

En résumé, cette combinaison d'outils et de méthode, nous avons pu créer une application de gestion de présence des employés qui représente une étape importante et évolutive et efficace. Nous sommes déterminés à continuer d'optimiser cette application pour répondre aux besoins de notre entreprise et à s'adapter aux évolutions futures.

BIBLIOGRAPHIE

- [1] Historique de NAFTAL.doc, récupéré le 16/02/2023.
- [2] Document interne de l'entreprise, récupéré le 6/03/2023.
- [3] J.GABAY, D.GABAY (2008), UML 2 Analyse et Conception 1ère édition, DUNOD
- [4] <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/images/pic4.jpg>
- [5] <https://upload.wikimedia.org/wikipedia/commons/thumb/9/9a/DiagrammesUML.svg/440px-DiagrammesUML.svg.png>
- [6], <https://fr.wikipedia.org/wiki/React>
- [7] https://fr.wikipedia.org/wiki/Processus_unifi, consulté le 16,04,2023
- [8] <https://www.techno-science.net/glossaire-definition/Visual-Studio.html>
- [9] <https://www.pappleweb.com/index/definition-de-nodejs/>, consulté le 01/06/2023
- [10] <https://hevodata.com/learn/mongodb-compass/>, consulté le 01/06/2023
- [11] <https://www.postman.com/product/what-is-postman/>, consulté le 01/06/2023
- [12] [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework)), consulté le 25/05/2023
- [13] <https://fr.wikipedia.org/wiki/Express.js>, consulté le 16/03/2023
- [14] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/>,
- [15] <https://fr.wikipedia.org/wiki/Naftal>, consulté le 15/02/2023
- [16] <https://www.naftal.dz/fr/index.php/a-propos-de-naftal/historique>, consulté le 17/02/2023
- [17] https://dial.uclouvain.be/downloader/downloader.php?pid=boreal:18216&datastream=PDF_01, consulté le 02/04/2023
- [18] <http://www.eprojet.fr/cours/pdf/html-css-support.pdf> consulté le 05/09/2023
- [19] , <http://tvaira.free.fr/web/cours-css.pdf> consulté le 05/09/2023

RÉSUMÉ

Ce mémoire a été réalisé en vue de l'obtention de diplôme de fin d'étude en informatique option génie logiciel. Ce mémoire présente la réalisation d'une application web dédiée à la gestion des présences des employés pour l'entreprise Naftal. L'objectif principal de cette application est de simplifier et d'automatiser le suivi des présences, offrant ainsi à l'entreprise une solution efficace et précise.

Pour mettre en œuvre notre solution, nous avons utilisé la méthode UP pour la conception selon ces différentes phases du processus UP, qui se base sur UML comme langage de modélisation. La mise en œuvre, de notre application est réalisée en utilisant le framework react pour le front-end et Express.js pour le back-end offrant une gestion de requêtes et de données stockées dans la base mongodb. L'application développée permet à garantir et atteindre l'objectif de l'entreprise.

Mots clés : RGP, Feuille de présence, Importation pointage, UP, VSC, REACT, EXPRESSJS

ABSTRACT

This dissertation was produced for the purpose of obtaining the end-of-study diploma in computer science, software engineering option. This thesis presents the realization of a web application dedicated to the management of employee attendance for the company Naftal. The main objective of this application is to simplify and automate attendance tracking, thus providing the company with an efficient and accurate solution.

To implement our solution, we used the UP method for the design according to these different phases of the UP process, which is based on UML as a modeling language. The implementation of our application is carried out using the react framework for the front-end and Express.js for the back-end offering management of queries and data stored in the mongodb. The developed application makes it possible to guarantee and achieve the objective of the company.

Keys- Words : RGP, Timesheet, Score import, VSC, UP, REACT, EXPRESSJS