|  |  |  |
| --- | --- | --- |
| **REPUBLIQUE TUNISIENNE**  **\*\*\*\*** |  |  |

Rapport de Fin d’Etudes

Préparé en vue de l’obtention du :

Diplôme National d’Ingénieur en Informatique parcours **Génie Logiciel et Système D’information**

**CONCEPTION ET DÉVELOPPEMENT D'UN SYSTÈME DE GESTION DES RESSOURCES HUMAINES ET DE POINTAGE**

**Réalisé par : ANDZOUANA DENIS BONHEUR MERVEILLE**

**Organisme d’accueil :** **NEXT CONSULT**



Encadrant pédagogique : Mme TAKOUA HAMDI

Encadrante Professionnel : MmeSLIMANI SANA

**Année universitaire : 2024-2025**

**REMERCIEMENTS**

À l’issue de ce travail, je souhaite exprimer ma profonde gratitude à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce projet.

Je tiens à adresser mes remerciements les plus sincères à **Mme TAKOUA HAMDI**, pour son encadrement bienveillant, ses conseils éclairés et sa disponibilité constante tout au long de ce projet. Son accompagnement a été d’une aide précieuse.

Je remercie également **Mme SLIMANI SANA** ainsi que toute l’équipe de **Next Consult** pour leur accueil chaleureux, leur collaboration et leur soutien, qui ont grandement facilité mon intégration au sein de l’entreprise et le bon déroulement de cette expérience professionnelle.

Ma reconnaissance s'adresse aussi à l'administration de l'entreprise **Next Consult** pour m'avoir offert cette opportunité enrichissante et pour avoir mis à ma disposition tous les moyens nécessaires.

Je n’oublie pas mes camarades de promotion, avec qui j’ai partagé des moments d’échange, de travail et de solidarité, qui ont enrichi cette belle aventure humaine et intellectuelle.

Enfin, je tiens à remercier du fond du cœur **ma famille**, pour son soutien indéfectible, ses encouragements constants et sa présence à chaque étape de mon parcours. Sans vous, rien de tout cela n’aurait été possible.

**DÉDICACES**

À tous ceux qui ont semé en moi la force de croire et d’apprendre, je dédie ce travail, reflet de mon évolution et de leur influence bienveillante :

**À mon père ANDZOUANA DENIS GEOFFROY,** Pour tous les sacrifices que tu as consentis afin que je puisse poursuivre mes études à l’étranger, pour ton soutien indéfectible, tes conseils pleins de sagesse et ta manière unique de m’apprendre à rester humble, à faire profil bas, même quand je sais. Tu es un exemple de force et de discernement.

**À ma mère MBOYO EGNAN TISHNA ROLANDE,** Pour tes prières constantes qui m’accompagnent partout où je vais, pour tes paroles pleines de sagesse et tes adages qui me guident encore aujourd’hui — *« Soso aliya kaka eloko ekokaka na mongongo naye »* — pour ton amour et ta présence à chaque étape de mon parcours.

**À mes sœurs,** Pour votre amour, vos encouragements, et cette chaleur fraternelle que vous savez transmettre même à des kilomètres. Vos mots, vos blagues, vos appels ont souvent été mon réconfort dans les moments de solitude. Merci d’être ces piliers invisibles mais essentiels.

**À mon petit frère,** Ton innocence, même loin de toi, tes rires sincères et ton affection sont pour moi une source immense de bonheur et de motivation. Tu me rappelles chaque jour pourquoi je dois continuer à me battre.

**À mes amis,** pour leur présence et leur soutien moral tout au long de ce parcours.

*À tous ceux qui ont contribué de près ou de loin à ma formation et à mon épanouissement personnel et professionnel.*

**Résumé**

Ce projet de fin d'études porte sur la conception et le développement d'un système intégré de gestion des ressources humaines et de pointage. Face aux défis actuels de digitalisation des processus RH, cette solution propose une plateforme complète permettant d'automatiser et d'optimiser la gestion du personnel, le suivi des présences, la gestion des congés, l'attribution et le suivi des tâches, ainsi que la communication interne.

Le système développé repose sur une architecture moderne utilisant le stack MERN (MongoDB, Express.js, React.js, Node.js), offrant une interface utilisateur intuitive et responsive. L'application intègre également un assistant IA pour faciliter l'analyse des performances et l'aide à la décision.

Ce rapport présente l'ensemble du processus de développement, depuis l'analyse des besoins jusqu'à l'implémentation finale, en passant par la conception et la modélisation UML. Les résultats obtenus démontrent une amélioration significative des processus RH, une réduction des tâches administratives et une meilleure gestion des ressources humaines.

**Mots-clés :** Gestion des ressources humaines, système de pointage, MERN stack, UML, assistant IA, automatisation des processus RH, développement web.

**Liste des figures**

Figure I.1 : Logo de l’entreprise Next Consult…………………………………………………

Figure I.2 : Organigramme de l'entreprise Next Consult …………………………………

Figure I.3 : Processus actuel de gestion RH…………………………………………………

Figure I.4 : Problématiques identifiées dans le système existant……………………………

Figure I.5 : Architecture générale de la solution proposée…………………………………

Figure II.1 : Architecture logicielle du système……………………………………………

Figure II.2 : Diagramme de cas d'utilisation global………………………………………..

Figure II.3 : Diagramme de cas d'utilisation - Module Authentification……………………

Figure II.4 : Diagramme de cas d'utilisation - Module Administration………………………

Figure II.5 : Diagramme de cas d'utilisation - Module Employé……………………………

Figure II.6 : Diagramme de séquence - Processus d'authentification………………………

Figure II.7 : Diagramme de classes du système………………………………………………

Figure III.1 : Architecture technique du système……………………………………………

Figure III.2 : Interface d'authentification…………………………………………………

Figure III.3 : Tableau de bord administrateur………………………………………………

Figure III.4 : Gestion des utilisateurs…………………………………………………

Figure III.5 : Gestion des départements…………………………………………………

Figure III.6 : Tableau de bord employé…………………………………………………

Figure III.7 : Interface de pointage…………………………………………………

Figure III.8 : Gestion des congés…………………………………………………

Figure III.9 : Gestion des tâches…………………………………………………

Figure III.10 : Messagerie interne…………………………………………………

Figure III.11 : Assistant IA…………………………………………………

Figure III.12 : Rapports et analyses…………………………………………………

**Liste des Tableaux**

Tableau I.1 : Comparaison des solutions existantes sur le marché…………………………

Tableau I.2 : Avantages et inconvénients du système actuel………………………………

Tableau II.1 : Acteurs du système…………………………………………………

Tableau II.2 : Besoins non fonctionnels…………………………………………………

Tableau III.1 : Environnement matériel…………………………………………………

Tableau III.2 : Environnement logiciel…………………………………………………

Tableau III.3 : Tests fonctionnels…………………………………………………

Tableau III.4 : Performance du système…………………………………………………

**Sommaire**

**Chapitre I : Présentation du contexte général du projet et étude de l'existant**

I. Introduction

II. Cadre du projet

III. Présentation de la société

IV. Étude de l'existant

IV.1. Descriptif de l'existant

IV.2. Critique de l'existant

IV.3. Solution

V. Conclusion

**Chapitre II : Analyse et spécification des besoins**

I. Introduction

II. Architecture logicielle

III. Méthodologie de conception UML

IV. Capture des besoins

IV.1. Besoins fonctionnels

IV.1.1. Modélisation du diagramme de cas d'utilisation

IV.1.2. Raffinement du cas d'utilisation "S'authentifier"

IV.1.3. Raffinement du cas d'utilisation "Administrateur"

IV.1.4. Raffinement du cas d'utilisation "Employé"

IV.2. Besoins non fonctionnels

V. Conclusion

**Chapitre III : Réalisation et implémentation**

I. Environnement de travail

I.1. Environnement matériel

I.2. Environnement logiciel

II. Interfaces de l'application

II.1. Interface d'authentification

II.2. Interface Admin

II.3. Interface utilisateur

II.4. Gestionnaire d'interface

III. Conclusion

**Conclusion Générale**

**Bibliographie**

**Annexes**

- Annexe 1 : Structure du code source

- Annexe 2 : Guide d'utilisation

- Annexe 3 : API Documentation

**Chapitre I : Présentation du contexte général du projet et étude de l'existant**

**I. Introduction**

Dans un contexte économique de plus en plus compétitif, les entreprises cherchent constamment à optimiser leurs processus internes pour gagner en efficacité et en productivité. La gestion des ressources humaines (GRH) et le suivi du temps de travail (pointage) représentent des enjeux majeurs pour toute organisation, quelle que soit sa taille ou son secteur d'activité.

Traditionnellement, ces processus étaient gérés manuellement ou à l'aide d'outils disparates, entraînant une perte de temps considérable, des erreurs humaines fréquentes et une difficulté à obtenir une vision globale et en temps réel des ressources humaines de l'entreprise. Face à ces défis, la digitalisation des processus RH s'impose comme une solution incontournable et indispensable pour le futur de demain.

Ce projet de fin d'études s'inscrit dans cette dynamique de transformation digitale et vise à concevoir et développer un système intégré de gestion des ressources humaines et de pointage. Cette solution a pour objectif de centraliser l'ensemble des données et des processus RH, d'automatiser les tâches répétitives, de faciliter la communication interne et d'offrir des outils d'analyse pour une prise de décision éclairée par le DRH et les managers de l’entreprise.

Le présent rapport détaille l'ensemble de la démarche suivie pour la réalisation de ce projet, depuis l'analyse du contexte et des besoins jusqu'à l'implémentation technique, en passant par la conception et la modélisation. Il met en lumière les choix technologiques effectués, les défis rencontrés et les solutions apportées pour aboutir à un système performant, évolutif et adapté aux besoins spécifiques de l'entreprise.

**II. Cadre du projet**

Ce projet de fin d'études s'inscrit dans le cadre de l'obtention du diplôme **Licence en Science de l'Informatique**, spécialité **Génie Logiciel et Systèmes d'Informations**. Il représente l'aboutissement de trois années de formation théorique et pratique, et constitue une opportunité de mettre en application les connaissances acquises dans un contexte professionnel réel.

Le stage, d'une durée de quatre mois, s'est déroulé au sein de l'entreprise Next Consul, spécialisée dans le développement de solutions informatiques pour les entreprises et autres. Cette immersion en milieu professionnel m’a permis de confronter les aspects théoriques de la formation aux réalités du terrain, tout en développant des compétences techniques et transversales essentielles pour mon insertion dans le monde professionnel.

Le projet confié consistait à développer un système intégré de gestion des ressources humaines et de pointage pour répondre aux besoins internes de l'entreprise, avec la perspective de commercialiser cette solution auprès de ses clients. Cette double finalité a nécessité une approche particulièrement rigoureuse en termes de qualité, de performance et d'évolutivité du système.

Les objectifs principaux assignés à ce projet étaient les suivants :

- Analyser les processus RH existants et identifier les axes d'amélioration

- Concevoir une architecture logicielle moderne, évolutive et sécurisée

- Développer une solution complète intégrant l'ensemble des fonctionnalités requises

- Assurer une expérience utilisateur optimale à travers une interface intuitive et responsive

- Mettre en place des mécanismes d'analyse et de rapport pour faciliter la prise de décision

- Documenter l'ensemble du système (code du projet) pour faciliter sa maintenance et son évolution future

La réalisation de ce projet a nécessité la mobilisation de compétences variées, tant techniques (développement web, bases de données, sécurité informatique) que méthodologiques (gestion de projet, analyse des besoins, modélisation UML) et transversales (communication, travail en équipe, autonomie).

**III. Présentation de la société**



**NEXT CONSULT** est une entreprise tunisienne spécialisée dans la transformation digitale, offrant une gamme complète de services technologiques adaptés aux besoins des entreprises modernes.

**Domaines d'expertise**

* **Développement d'applications web et mobiles** : Conception et réalisation d'applications sur mesure, compatibles avec les principales plateformes (Android, iOS, Windows Phone), en utilisant des technologies telles que HTML5, PHP et le responsive design.
* **Web marketing et SEO** : Optimisation de la visibilité en ligne grâce à des stratégies de marketing digital et de référencement naturel.
* **Conseil et audit IT** : Évaluation des infrastructures informatiques existantes, identification des vulnérabilités, et recommandations pour une migration vers des solutions cloud (privé, public ou hybride).
* **Assurance qualité et Pentesting** : Tests rigoureux pour garantir la fiabilité et la sécurité des solutions proposées.
* **Formation continue via l'Académie NEXT** : Mise à jour régulière des compétences des employés pour rester à la pointe des avancées technologiques.

**Philosophie et engagement**

NEXT CONSULT se distingue par une approche centrée sur le client, visant à fournir des solutions innovantes, fiables et de haute qualité. L'entreprise s'engage à :

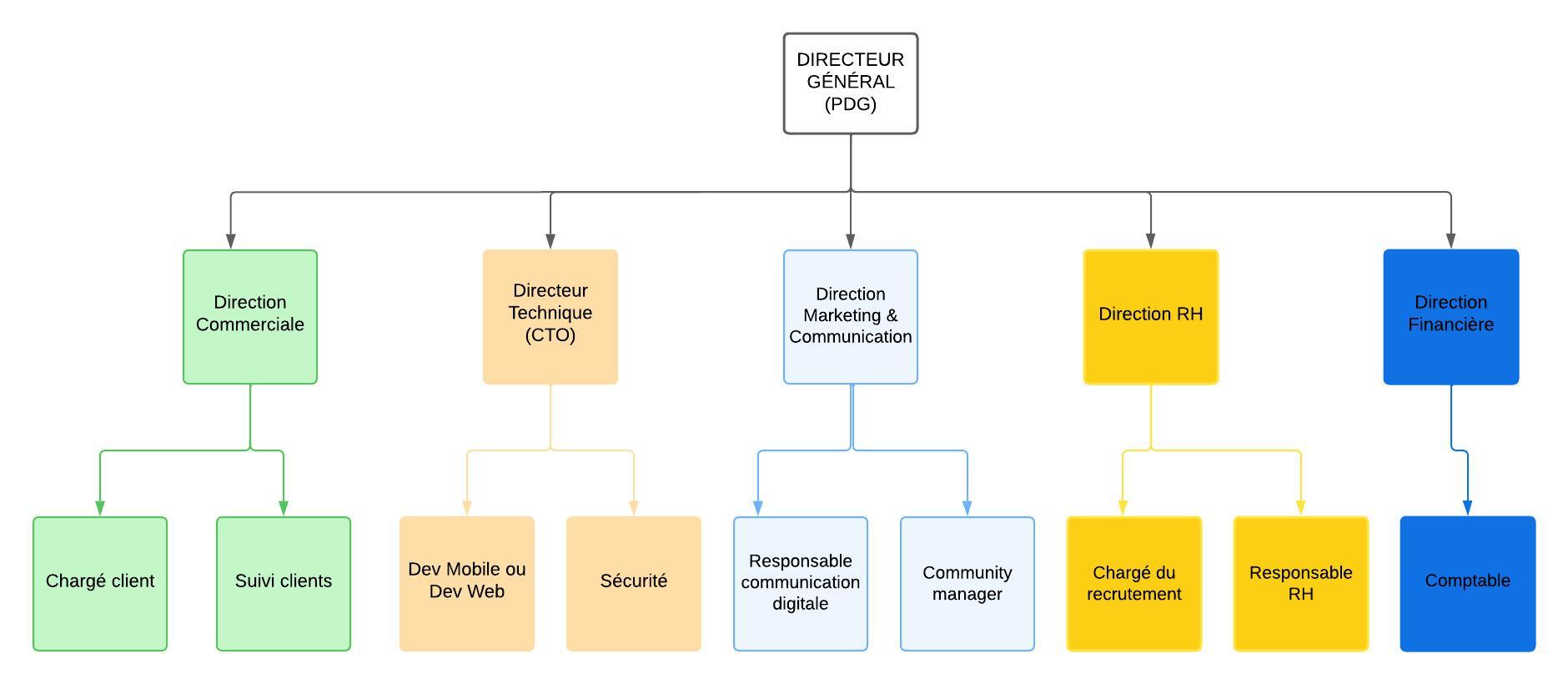
* Offrir des services personnalisés répondant aux besoins spécifiques de chaque client.
* Maintenir des normes élevées de qualité et de sécurité.
* Proposer une tarification transparente et compétitive.
* Accompagner les clients tout au long de leurs projets, de la conception à la mise en œuvre.

**Coordonnées**

* **Adresse** : El Menzah 4, Tunis, Tunisie
* **Téléphone** : +216 71 34 30 76 / +216 98 11 48 44
* **Site web** : https://www.next.tn

Sur le plan organisationnel, **NEXT CONSULT** est structurée en départements fonctionnels (commercial, RH, finance) et opérationnels (conseil, développement, qualité, support), avec une approche projet transversale permettant de constituer des équipes pluridisciplinaires adaptées aux besoins spécifiques de chaque client.

La gestion des ressources humaines représente un enjeu stratégique pour **NEXT CONSULT**, dans un secteur très concurrentiel en matière d’attraction et de fidélisation des talents. L'entreprise a donc souhaité se doter d'un système moderne et performant pour optimiser ses processus RH et offrir une expérience collaborateur de qualité.



**Figure I.2 : Organigramme de l'entreprise NEXT CONSULT**

**IV. Étude de l'existant**

**IV.1. Descriptif de l'existant**

Avant la mise en place du nouveau système, la gestion des ressources humaines et le suivi du temps de travail au sein de **NEXT CONSULT** reposaient sur une combinaison d'outils disparates et de processus partiellement manuels, reflétant l'évolution organique de l'entreprise depuis sa création.

**Gestion administrative du personnel**

La gestion administrative du personnel était assurée à l'aide d'un logiciel de paie standard, complété par des fichiers Excel pour le suivi des informations complémentaires (coordonnées, documents administratifs, historique des postes, etc.). Cette dualité des outils entraînait des saisies redondantes et des risques d'incohérence entre les différentes sources de données.

Le processus d'intégration des nouveaux collaborateurs était principalement manuel, avec l'envoi de formulaires papier à remplir et la création manuelle des différents comptes nécessaires (email, accès aux applications, etc.). Ce processus, chronophage et source d'erreurs, pouvait prendre jusqu'à trois jours pour être finalisé.

**Gestion des présences et des congés**

Le suivi des présences s'effectuait via un système de pointeuse biométrique pour les collaborateurs du siège, tandis que les équipes des agences et celles en mission chez les clients utilisaient un fichier Excel partagé pour déclarer leurs heures de travail. Cette disparité des méthodes compliquait la consolidation des données et leur exploitation pour la paie ou les analyses RH.

La gestion des demandes de congés reposait sur un circuit de validation par e-mail : le collaborateur envoyait sa demande à son responsable, qui la transmettait au service RH après validation. Le suivi des soldes de congés était effectué manuellement dans un fichier Excel, avec des mises à jour périodiques communiquées aux collaborateurs.

**Gestion des tâches et des projets**

L'entreprise utilisait un outil de gestion de projet commercial pour le suivi des projets clients, mais ne disposait pas de solution intégrée pour la gestion des tâches internes et leur attribution aux collaborateurs. Cette situation conduisait à une utilisation intensive des emails et des réunions pour la coordination des équipes, avec une visibilité limitée sur la charge de travail de chacun.

**Communication interne**

La communication interne s'appuyait principalement sur les emails, complétés par une messagerie instantanée grand public pour les échanges informels. L'absence d'outil de communication structuré et intégré aux autres systèmes de l'entreprise limitait la fluidité des échanges et la capitalisation des informations partagées.

**Rapport et analyse**

La production de tableaux de bord RH nécessitait un travail manuel important de collecte et de consolidation des données issues des différents systèmes. Ce processus, réalisé mensuellement, mobilisait une ressource à temps plein pendant plusieurs jours et ne permettait pas de disposer d'indicateurs en temps réel pour piloter l'activité.



**Figure I.3 : Processus actuel de gestion RH**

**IV.2. Critique de l'existant**

L'analyse approfondie du système existant a permis d'identifier plusieurs problématiques impactant l'efficacité des processus RH et la qualité de service offerte aux collaborateurs :

**Fragmentation des données et des outils**

La multiplicité des outils utilisés (fichiers Excel, emails, système de pointage) entraînait une fragmentation des données RH, rendant difficile l'obtention d'une vision unifiée et cohérente des informations relatives aux collaborateurs. Cette situation générait des risques d'erreurs lors des saisies multiples et compliquait la maintenance des données à jour.

**Processus manuels chronophages**

De nombreux processus RH reposaient sur des interventions manuelles (saisie de données, envoi de formulaires, validation par email), mobilisant des ressources importantes et allongeant les délais de traitement. Cette situation limitait la capacité du service RH à se concentrer sur des activités à plus forte valeur ajoutée, comme l'accompagnement des collaborateurs ou le développement des compétences.

**Manque de visibilité en temps réel**

L'absence de système centralisé et la production manuelle des tableaux de bord ne permettaient pas de disposer d'une visibilité en temps réel sur les indicateurs clés (présences, congés, charge de travail, etc.). Cette situation compliquait la prise de décision et limitait la capacité de l'entreprise à réagir rapidement aux évolutions de son environnement.

**Expérience collaborateur sous-optimale**

Les processus RH existants offraient une expérience utilisateur peu satisfaisante pour les collaborateurs, avec des démarches administratives complexes, des délais de traitement longs et un accès limité à leurs informations personnelles. Cette situation pouvait impacter négativement l'engagement et la satisfaction des équipes.

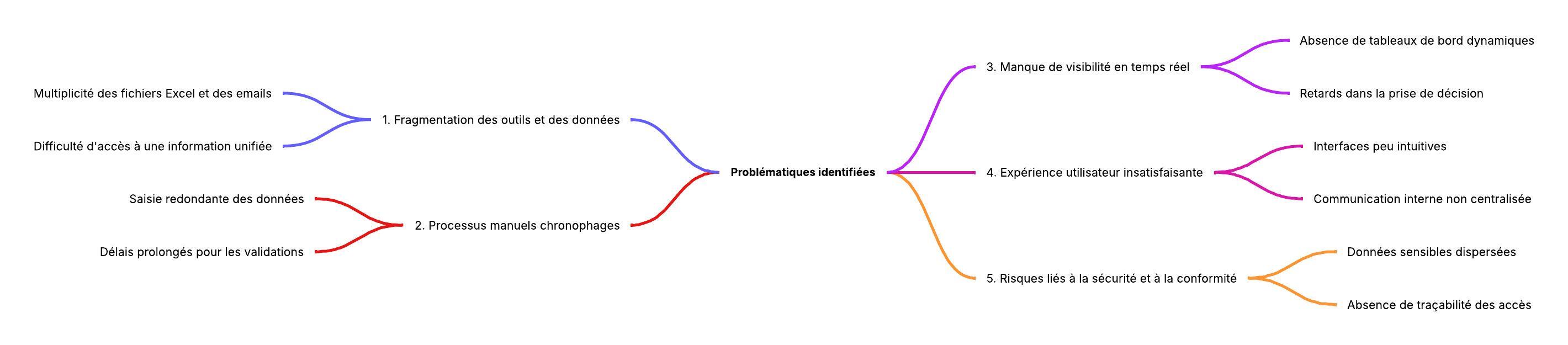
**Limites en termes d'évolutivité**

Le système existant, basé sur des outils disparates et des processus partiellement manuels, présentait des limites importantes en termes d'évolutivité face à la croissance de l'entreprise. L'augmentation du nombre de collaborateurs et la diversification des activités risquaient d'accentuer les problématiques identifiées et de générer des coûts croissants de gestion administrative.

**Sécurité et conformité des données**

La dispersion des données RH sur différents supports (logiciel, fichiers Excel, emails) soulevait des questions en termes de sécurité et de conformité, notamment au regard des réglementations sur la protection des données personnelles. L'absence de politique unifiée

de gestion des accès et de traçabilité des actions constituait un risque potentiel pour l'entreprise.



**Figure I.4 : Problématiques identifiées dans le système existant**

**Etas des lieux**

Dans le cadre de l’étude préalable au développement d’une solution de gestion des ressources humaines et de pointage, une analyse comparative des outils RH existants a été réalisée. Cette analyse a pour objectif d’identifier les fonctionnalités proposées par les principales solutions disponibles sur le marché et de relever leurs limites dans le contexte spécifique de notre entreprise.

Tableau I.1 : Comparaison des solutions existantes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Critères / Solutions | Sage RH | Zoho People | Factorial | Notre solution prévue |  |  |  |  |  |
| Accessibilité Web | Oui | Oui | Oui | Oui |  |  |  |  |  |
| Application mobile | Moyenne | Oui | Oui | Oui |  |  |  |  |  |
| Suivi des présences | Limité | Oui | Oui | Avancé |  |  |  |  |  |
| Gestion des congés | Oui | Oui | Oui | Personnalisée |  |  |  |  |  |
| Automatisation | Faible | Moyenne | Bonne | Élevée (via IA intégrée) |  |  |  |  |  |
| Tableau de bord RH | Non dynamique | Dynamique | Dynamique | Dynamique temps réel |  |  |  |  |  |
| Personnalisation | Faible | Moyenne | Moyenne | Élevée |  |  |  |  |  |
| Coût | Élevé | Abordable | Abordable | Sur mesure / Faible coût |  |  |  |  |  |
| Intégration d’IA | Non | Non | Non | Oui (assistant intelligent) |  |  |  |  |  |

**PROBLEMATIQUE IDENTIFIER**

Suite à l’étude du système RH actuellement en place, plusieurs dysfonctionnements ont été relevés. Le tableau ci-dessous résume les avantages et inconvénients du système existant :

Tableau I.2 : Avantages et inconvénients du système actuel

|  |  |
| --- | --- |
| Avantages | Inconvénients |
| Outils familiers (Excel, email) | Risques d’erreurs lors des saisies manuelles |
| Facilité de partage par email | Multiplication des fichiers, mauvaise traçabilité |
| Aucune formation spécifique requise | Manque de centralisation et redondance des informations |
| Simplicité de certaines tâches | Aucune automatisation, processus lents et rigides |
| Faible coût d'utilisation (pas d’abonnement) | Pas de sécurité renforcée ni d’accès en temps réel |

**IV.3. Solution**

Face aux problématiques identifiées dans le système existant, la solution proposée consiste en un système intégré de gestion des ressources humaines et de pointage, reposant sur une architecture moderne et évolutive. Cette solution vise à centraliser l'ensemble des données et des processus RH, à automatiser les tâches répétitives et à offrir une expérience utilisateur optimale pour l'ensemble des acteurs concernés.

**Architecture générale de la solution**

La solution s'appuie sur une architecture de service web moderne, utilisant le stack MERN (MongoDB, Express.js, React.js, Node.js) pour garantir performance, évolutivité et facilité de maintenance. Cette architecture permet de développer une application web responsive, accessible depuis différents types de terminaux (ordinateurs, tablettes, smartphones) et offrant une expérience utilisateur fluide et intuitive.

Le système est conçu selon une approche modulaire, permettant d'activer ou de désactiver certaines fonctionnalités en fonction des besoins spécifiques de l'entreprise, et facilitant l'ajout de nouveaux modules à l'avenir. Cette modularité s'accompagne d'une API robuste, ouvrant la possibilité d'intégrations avec des systèmes de l'entreprise (ERP, CRM, outils de paie, etc.).

**Fonctionnalités principales**

Le système intègre les fonctionnalités suivantes, regroupées par modules :

**1. Module d'administration et de gestion des utilisateurs**

1. Gestion des profils utilisateurs et des droits d'accès

2. Gestion de la structure organisationnelle (départements)

3. Tableau de bord administrateur avec indicateurs clés

**2. Module de gestion des présences**

1. Pointage en ligne effectuer par l’assistante (sur web ou mobile)

2. Suivi des heures de travail et des retards

3. Génération des rapports de présence

**3. Module de gestion des congés**

1. Demande et validation en ligne des congés

2. Suivi des soldes de congés en temps réel

3. Calendrier partagé des absences

**4. Module de gestion des tâches**

1. Création et attribution des tâches

2. Suivi de l'avancement et des échéances

3. Rapport sur la charge de travail

**5. Module de messagerie interne**

1. Communication individuelle

2. Partage de fichiers et de documents

3. Historique des conversations

**6. Module d'assistant IA**

1. Analyse des performances

2. Recommandations personnalisées

3. Réponses automatisées aux questions fréquentes

**7. Module de rapport et d'analyse**

1. Tableaux de bord personnalisables

2. Génération automatique de rapports

3. Outils d'analyse prédictive

**Avantages de la solution**

La solution proposée apporte les avantages suivants par rapport au système existant :

**Centralisation des données :** toutes les informations RH sont regroupées dans une base de données unique, garantissant cohérence et fiabilité.

**Automatisation des processus :** les tâches répétitives sont automatisées, réduisant les délais de traitement et les risques d'erreur.

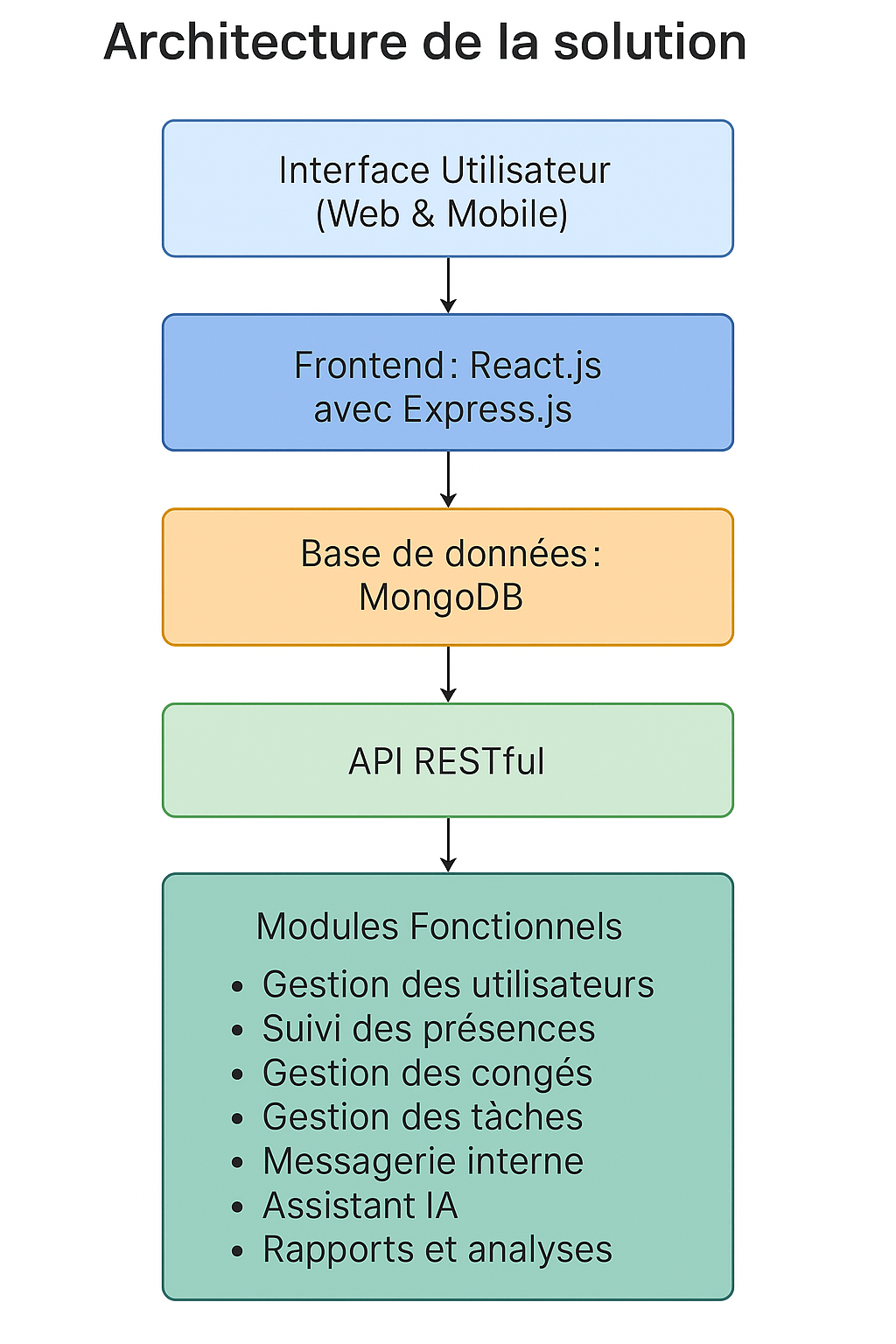
**Accessibilité et mobilité** : le système est accessible 24/7 depuis différents types de terminaux, facilitant le travail à distance.

**Visibilité en temps réel :** les tableaux de bord et les indicateurs sont mis à jour en temps réel, facilitant le pilotage de l'activité.

**Expérience utilisateur améliorée :** l'interface intuitive et les workflows simplifiés améliorent l'expérience des collaborateurs.

**Sécurité renforcée :** le système intègre des mécanismes avancés de sécurité et de protection des données personnelles.

**Évolutivité :** l'architecture modulaire permet d'adapter le système à l'évolution des besoins de l'entreprise.



**Figure I.5 : Architecture générale de la solution proposée**

**V. Conclusion**

L'étude du contexte général et de l'existant m’a permis de mettre en évidence les limites du système actuel de gestion des ressources humaines et de pointage au sein de **NEXT CONSULT**. La fragmentation des outils, la prépondérance des processus manuels et le manque de visibilité en temps réel constituent des freins à l'efficacité opérationnelle et à la qualité de service offerte aux collaborateurs.

Face à ces constats, la solution proposée vise à transformer en profondeur la gestion RH de l'entreprise, en s'appuyant sur une architecture moderne, des fonctionnalités complètes et une approche centrée sur l'expérience utilisateur. Cette solution s'inscrit pleinement dans la stratégie de transformation digitale de **NEXT CONSULT** et constitue un levier important pour l'optimisation de ses processus internes.

Le chapitre suivant détaillera l'analyse et la spécification des besoins, étape essentielle pour garantir l'adéquation de la solution développée aux attentes des différentes parties prenantes.

**Chapitre II : Analyse et spécification des besoins**

**I. Introduction**

L'analyse et la spécification des besoins constituent une étape fondamentale dans le processus de développement logiciel, permettant de définir précisément ce que le système doit faire et comment il doit le faire. Cette phase est d'autant plus critique pour un système de gestion des ressources humaines et de pointage, qui touche à des processus sensibles et implique de nombreux acteurs au sein de l'organisation.

Ce chapitre présente l'approche méthodologique adoptée pour capturer et formaliser les besoins liés au système, en s'appuyant sur les principes de l'ingénierie des besoins et les outils de modélisation UML (Unified Modeling Language). Il détaille l'architecture logicielle retenue, les besoins fonctionnels et non fonctionnels identifiés, ainsi que les modèles élaborés pour représenter les différentes vues du système.

La démarche suivie a impliqué l'ensemble des parties prenantes du projet, à travers des entretiens individuels, des ateliers de travail collectifs et des séances de validation des spécifications. Cette approche a permis de garantir la prise en compte des besoins réels des utilisateurs finaux et d'anticiper les évolutions futures du système.

Les spécifications présentées dans ce chapitre ont servi de base pour les phases ultérieures du projet, notamment la conception détaillée et l'implémentation technique. Elles constituent également un référentiel pour l'évaluation de la solution finale et la vérification de sa conformité aux attentes initiales.

**II. Architecture logicielle**

L'architecture logicielle du système de gestion des ressources humaines et de pointage a été conçue pour répondre aux exigences de modularité, d'évolutivité et de performance identifiées lors de l'analyse des besoins. Elle s'appuie sur le stack MERN (MongoDB, Express.js, React.js, Node.js), qui offre un cadre robuste et moderne pour le développement d'applications web.

**Architecture générale**

Le système adopte une architecture en trois couches distinctes :

**1. Couche présentation (Frontend) :** Développée avec React.js, cette couche est responsable de l'interface utilisateur et de l'interaction avec les utilisateurs. Elle s'appuie sur des composants réutilisables et une gestion d'état centralisée pour offrir une expérience utilisateur fluide et cohérente pour chaque utilisateur.

**2. Couche métier (Backend) :** Implémentée avec Node.js et Express.js, cette couche encapsule la logique métier de l'application et expose des API RESTful pour la communication avec le frontend. Elle intègre également des mécanismes d'authentification et d'autorisation pour sécuriser l'accès aux ressources.

**3. Couche données :** Basée sur MongoDB, une base de données NoSQL orientée documents, cette couche assure le stockage persistant des données de l'application. Le choix de MongoDB permet une grande flexibilité dans la structure des données et une excellente scalabilité horizontale.

**Patterns architecturaux**

Plusieurs patterns architecturaux ont été mis en œuvre pour structurer le développement et garantir la qualité du code :

**- MVC (Model-View-Controller) :** Ce pattern fondamental sépare la logique métier (Model), l'interface utilisateur (View) et la gestion des interactions (Controller), facilitant la maintenance et l'évolution du code.

**- Repository Pattern :** Utilisé pour abstraire l'accès aux données et centraliser la logique de persistance, ce pattern simplifie les tests unitaires et permet de changer facilement d'implémentation de stockage si nécessaire.

**- Dependency Injection :** Ce principe permet de réduire le couplage entre les composants du système et facilite les tests unitaires en permettant de remplacer les dépendances par des mocks.

**Communication entre les composants**

La communication entre le frontend et le backend s'effectue via des API RESTful, avec échange de données au format JSON. Cette approche standardisée facilite l'intégration avec d'autres systèmes et permet le développement indépendant des différentes couches de l'application.

Pour les fonctionnalités nécessitant des mises à jour en temps réel (messagerie, notifications), pour cella en a utilisé le système Socket.IO, une bibliothèque permettant une communication bidirectionnelle en temps réel entre le client et le serveur.

**Sécurité**

L'architecture intègre plusieurs mécanismes de sécurité pour protéger les données sensibles et prévenir les accès non autorisés :

**- Authentification :** Elle est basée sur JSON Web Tokens (JWT), avec support de l'authentification à deux facteurs pour les comptes sensibles.

**- Autorisation :** Système de contrôle d'accès basé sur les rôles (RBAC), permettant de définir finement les permissions de chaque utilisateur.

**- Chiffrement :** Les données sensibles sont chiffrées en base de données, et les communications entre le client et le serveur sont sécurisées via HTTPS.

**- Protection contre les attaques courantes :** Mise en place de mesures contre les attaques XSS, CSRF, injection SQL, etc.

**Déploiement et scalabilité**

L'architecture a été conçue pour faciliter le déploiement et la scalabilité du système :

**- Conteneurisation :** Utilisation de Docker pour encapsuler les différents composants de l'application dans des conteneurs isolés.

**- Orchestration :** Possibilité de déployer le système sur Kubernetes pour une gestion automatisée des conteneurs et une haute disponibilité.

**- Scalabilité horizontale :** L'architecture permet d'ajouter facilement des instances supplémentaires pour répondre à une augmentation de la charge.

**- Monitoring :** Intégration d'outils de surveillance pour suivre les performances du système et détecter les problèmes potentiels.

Cette architecture logicielle constitue le socle technique du système, offrant un cadre solide pour l'implémentation des fonctionnalités métier tout en garantissant les qualités attendues en termes de performance, de sécurité et d'évolutivité.

**III. Méthodologie de conception UML**

La conception du système de gestion des ressources humaines et de pointage s'est appuyée sur la méthodologie UML (Unified Modeling Language), standard de facto pour la modélisation des systèmes d'information. Cette approche a permis de représenter les différentes facettes du système à travers des diagrammes complémentaires, facilitant la communication entre les parties prenantes et guidant le développement.

**Approche de modélisation**

La démarche de modélisation a suivi une approche itérative et incrémentale, en cohérence avec la méthodologie Agile adoptée pour le projet. Les modèles ont été progressivement affinés au fur et à mesure de l'avancement du projet et des retours des utilisateurs, permettant une adaptation continue aux besoins émergents.

**La modélisation s'est articulée autour de trois perspectives complémentaires :**

**1. Vue fonctionnelle :** Représentation des fonctionnalités du système du point de vue des utilisateurs, à travers les diagrammes de cas d'utilisation.

**2. Vue dynamique :** Modélisation des interactions entre les acteurs et le système, ainsi que des flux de contrôle, à l'aide des diagrammes de séquence et d'activité.

**3. Vue structurelle :** Description de l'organisation interne du système en termes de classes, d'objets et de leurs relations, via les diagrammes de classes et d'objets.

**Diagrammes UML utilisés**

J’ai utilisé plusieurs types de diagrammes UML pour élaborés et pour capturer les différents aspects du système :

**- Diagrammes de cas d'utilisation :** Ces diagrammes ont permis d'identifier les acteurs du système et les fonctionnalités qu'ils peuvent utiliser. Un diagramme global a été complété par des diagrammes détaillés pour chaque module fonctionnel.

**- Diagrammes de séquence :** Utilisés pour modéliser les interactions entre les acteurs et le système dans le cadre de scénarios spécifiques, ces diagrammes ont été particulièrement utiles pour clarifier les flux de données et les responsabilités des différents composants.

**- Diagrammes d'activité :** Ces diagrammes ont servi à représenter les workflows complexes, comme le processus de validation des demandes de congés ou l'attribution des tâches, en mettant en évidence les décisions et les actions parallèles.

**- Diagrammes de classes :** Le diagramme de classes principal a défini la structure statique du système, en identifiant les entités métier, leurs attributs et leurs relations. Des diagrammes plus détaillés ont été élaborés pour certains modules spécifiques.

**- Diagrammes d'état-transition :** Utilisés pour modéliser le cycle de vie de certaines entités clés, comme les demandes de congés ou les tâches, en mettant en évidence les états possibles et les transitions entre ces états.

**Outils de modélisation**

La modélisation UML a été réalisée à l'aide d'outils spécialisés permettant la création, la modification et le partage des diagrammes :

**- PowerAMC :** utilisé principalement pour créer des **modèles de bases de données** (MCD, MLD, MPD) et des **diagrammes UML**. Il facilite la **conception des systèmes d'information**, la **génération de documentation technique**, ainsi que **l’exportation des modèles** dans différents formats.

**Traçabilité des exigences**

Une attention particulière a été portée à la traçabilité entre les exigences fonctionnelles et non fonctionnelles et les éléments de modélisation UML. Cette traçabilité a été assurée par :

- L'association explicite de chaque cas d'utilisation à une ou plusieurs exigences fonctionnelles

- L'annotation des diagrammes avec des références aux exigences correspondantes

- La documentation des choix de conception en lien avec les exigences non fonctionnelles

Cette approche a permis de vérifier la couverture complète des exigences par la conception et de faciliter l'évaluation de l'impact des changements d'exigences sur les modèles existants.

La méthodologie de conception UML adoptée a ainsi fourni un cadre structuré pour la représentation du système sous différentes perspectives, facilitant la communication entre les parties prenantes et guidant efficacement le développement.

**IV. Capture des besoins**

**IV.1. Besoins fonctionnels**

Les besoins fonctionnels définissent les fonctionnalités que le système doit offrir pour répondre aux attentes des utilisateurs. Ils ont été identifiés à travers des entretiens avec les différentes parties prenantes, l'analyse des processus existants et des ateliers de travail collaboratifs.

Ces besoins ont été organisés par modules fonctionnels, correspondant aux principales composantes du système :

**Module d'authentification et de gestion des utilisateurs**

- Permettre la création des nouveaux utilisateurs par un administrateur

- Gérer l'authentification des utilisateurs avec méthode (email/mot de passe)

- Implémenter un système de récupération de mot de passe

- Gérer les profils utilisateurs avec différents niveaux d'accès (administrateur, manager, assistante, employé)

- Permettre aux utilisateurs de compléter et modifier leur profil

- Gérer la structure organisationnelle (départements)

**Module de gestion des présences**

- Permettre aux employés d'enregistrer leur arrivée et leur départ auprès de l’assistante

- Suivre les heures de travail, les retards et les absences

- Générer des rapports de présence par département

- Exporter les données de présence pour la paie

- Visualiser les tendances de présence sur des périodes définies

**Module de gestion des congés**

- Permettre aux employés de soumettre des demandes de congés

- Implémenter un workflow de validation en hiérarchie des demandes par les managers et en suite par admin (DRH)

- Suivre les soldes de congés en temps réel

- Notifier les employés et les managers des actions requises

- Générer des rapports sur les congés

- Visualiser les conges sous forme de liste, de tableau Kanban ou de calendrier

**Module de gestion des tâches**

- Créer et attribuer des tâches aux employés

- Définir des priorités, des échéances et des dépendances entre tâches

- Suivre l'avancement des tâches et leur statut

- Permettre aux employés de mettre à jour l'état de leurs tâches

- Générer des rapports sur la charge de travail et la productivité

- Visualiser les tâches sous forme de liste, de tableau Kanban ou de calendrier

**Module de messagerie**

- Permettre la communication entre utilisateurs individuels

- Partager des fichiers et des documents

- Rechercher dans l'historique des conversations

- Recevoir des notifications pour les nouveaux messages

- Intégrer un assistant IA pour répondre aux questions fréquentes

**Module d'assistant IA**

- Analyser les performances des tâches des employés

- Fournir des recommandations personnalisées

- Répondre aux questions fréquentes des utilisateurs

**Module de rapport et d'analyse**

- Générer des tableaux de bord personnalisables

- Produire des rapports prédéfinis sur les différents aspects RH

- Permettre l'export des données dans un format

- Visualiser les indicateurs clés de performance (KPI)

- Analyser les tendances et les évolutions

- Comparer les données entre différentes périodes ou entités

**IV.1.1. Modélisation du diagramme de cas d'utilisation**

Le diagramme de cas d'utilisation global représente l'ensemble des fonctionnalités offertes par le système et les acteurs qui interagissent avec ces fonctionnalités. Il offre une vue synthétique des capacités du système et constitue le point de départ pour une analyse plus détaillée.

**Acteurs du système**

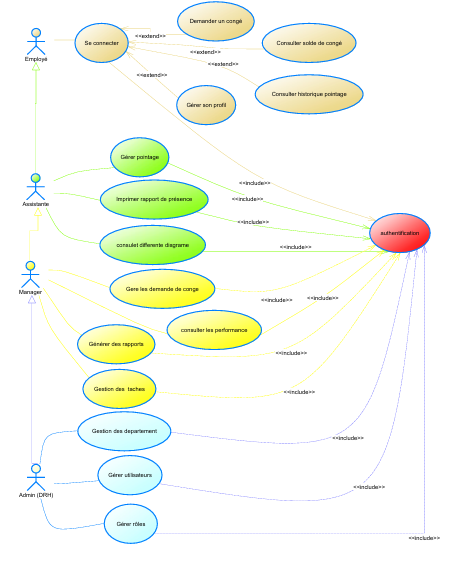
Le système interagit avec plusieurs types d'acteurs, ayant des rôles et des responsabilités distincts :

***Tableau II.1 : Acteurs du système***

|  |  |  |
| --- | --- | --- |
|  | Description | Responsabilités principales |
| **Administrateur (DRH)** | Utilisateur disposant des droits d'accès les plus élevés, responsable de la configuration et de la supervision globale du système. | - Configuration du système  - Gestion des utilisateurs et des droits d'accès  -Gestion des départements  -Supervision de l'ensemble des modules  -Génération de rapports globaux |
| **Manager** | Utilisateur responsable d'un département, disposant de droits étendus sur les membres de son département. | - Validation des demandes de congés.  -Attribution et suivi des tâches.  - Analyse des performances des tâche.  - Génération de rapports départementaux. |
| **Employé** | Utilisateur standard du système, disposant d'accès limités à ses propres données et aux fonctionnalités de base. | - Gestion de son profil  - Enregistrement de ses présences  - Demande de congés  - Suivi et mise à jour de ses tâches  - Communication avec ses collègues  - Consultation de l'assistant IA |
| **Assistante** | Utilisateur charger de faire l’enregistrement, importer ou exporte des pointages des utilisateur | -Gestion des pointages  -Suivi et mise à jour de ses tâches  -Génération de rapports globaux |
| **Système externe** | Systèmes tiers pouvant interagir avec le système via des API ou des intégrations spécifiques. | - Synchronisation des données  - Réception de rapports automatisés  - Envoi de notifications  - Authentification externe |
| **Assistant IA** | Composant intelligent du système capable d'analyser les données et de fournir des recommandations. | - Analyse des performances  - Génération de recommandations  - Réponse aux questions des utilisateurs |

**Diagramme de cas d'utilisation global**

Le diagramme de cas d'utilisation global met en évidence les principales fonctionnalités du système et leurs relations avec les acteurs identifiés. Il est structuré autour des modules fonctionnels décrits précédemment, avec des relations d'inclusion et d'extension entre certains cas d'utilisation.



Ce diagramme global est complété par des diagrammes plus détaillés pour chaque module fonctionnel, permettant d'approfondir les cas d'utilisation spécifiques et leurs interactions.

**IV.1.2. Raffinement du cas d'utilisation "S'authentifier"**

Le cas d'utilisation "S'authentifier" est fondamental pour le système, car il contrôle l'accès aux différentes fonctionnalités en fonction du profil de l'utilisateur. Son raffinement permet de détailler les différentes méthodes d'authentification supportées et les scénarios alternatifs.

**Description détaillée**

**Nom :** S'authentifier

**Acteur principal :** Utilisateur (Administrateur, Manager, Employé, Assistante)

**Acteurs secondaires :** Système d'authentification externe (Google)

**Préconditions :** L'utilisateur dispose d'un compte valide dans le système

**Postconditions :** L'utilisateur est authentifié et accède à son tableau de bord personnalisé

**Flux principal :**

1. L'utilisateur accède à la page d'authentification

2. L'utilisateur saisit son email et son mot de passe

3. Le système vérifie les informations d'identification

4. Si les informations sont valides, le système génère un token d'authentification

5. L'utilisateur est redirigé vers son tableau de bord personnalisé

**Flux alternatifs :**

**A1 : Authentification avec Google**

1. L'utilisateur clique sur le bouton "Se connecter avec Google"

2. L'utilisateur est redirigé vers la page d'authentification Google

3. Après validation par Google, l'utilisateur est redirigé vers le système

4. Le système vérifie l'identité de l'utilisateur et génère un token

**A2 : Mot de passe oublié**

1. L'utilisateur clique sur le lien "Mot de passe oublié"

2. L'utilisateur saisit son email

3. Le système envoie un lien de réinitialisation à l'adresse email

4. L'utilisateur définit un nouveau mot de passe via le lien reçu

**A3 : Première connexion**

1. Si c'est la première connexion de l'utilisateur, le système l'invite à compléter son profil

2. L'utilisateur saisit les informations complémentaires requises

3. Le système enregistre les informations et redirige l'utilisateur vers son tableau de bord

**Exceptions :**

**E1 : Informations d'identification invalides**

1. Le système affiche un message d'erreur

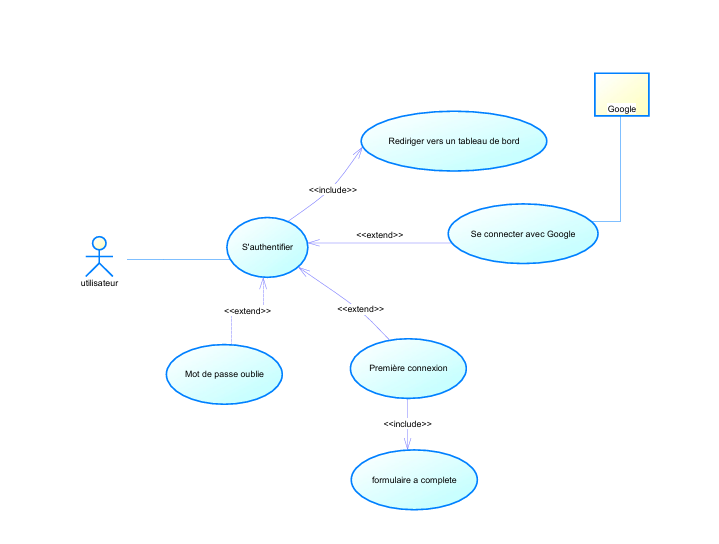
2. L'utilisateur peut réessayer ou utiliser la fonction "Mot de passe oublié"

**E2 : Compte désactivé**

1. Le système affiche un message indiquant que le compte est désactivé

2. L'utilisateur est invité à contacter l'administrateur

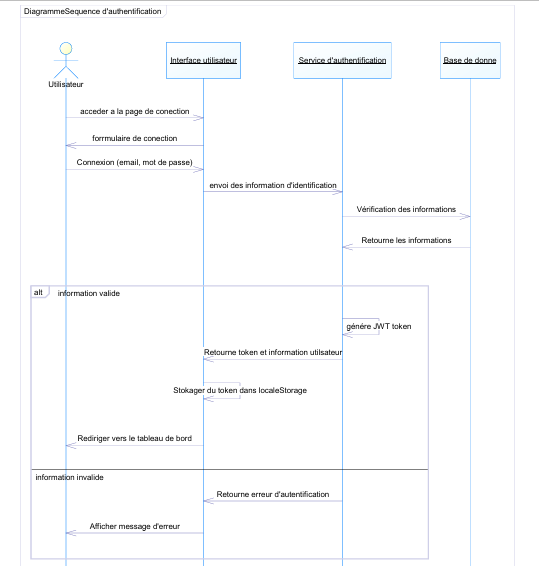
**Diagramme de cas d'utilisation authentifier**



Ce diagramme du cas d’utilisation *S’authentifier* illustre les différentes possibilités offertes à l’utilisateur pour accéder au système, incluant les scénarios alternatifs tels que l’authentification via Google, la gestion du mot de passe oublié ou encore la première connexion.

**Diagramme de séquence**

Le diagramme de séquence ci-dessous illustre les interactions entre l'utilisateur, l'interface d'authentification et le backend du système lors du processus d'authentification standard :



Ce diagramme met en évidence les différentes étapes du processus d'authentification et les interactions entre les composants du système, facilitant la compréhension du flux d'information et des responsabilités de chaque élément.

**IV.1.3. Raffinement du cas d'utilisation "Administrateur"**

Le cas d'utilisation "Administrateur" englobe l'ensemble des fonctionnalités accessibles aux utilisateurs ayant le rôle d'administrateur dans le système. Son raffinement permet de détailler les différentes actions que peut effectuer un administrateur et leurs implications.

**Description détaillée**

**Nom :** Gérer les utilisateurs (sous-cas d'utilisation de l'Administrateur)

**Acteur principal :** Administrateur

**Préconditions :** L'administrateur est authentifié dans le système

**Postconditions :** Les modifications sur les utilisateurs sont enregistrées dans le système

**Flux principal :**

1. L'administrateur accède à la section de gestion des utilisateurs

2. Le système affiche la liste des utilisateurs existants

3. L'administrateur peut créer, modifier, désactiver ou supprimer des utilisateurs

4. Pour chaque action, le système demande confirmation

5. Le système enregistre les modifications et met à jour la liste des utilisateurs

**Flux alternatifs :**

**A1 : Création d'un nouvel utilisateur**

1. L'administrateur clique sur "Ajouter un utilisateur"

2. L'administrateur saisit les informations de base (nom, email, rôle)

3. Le système génère un email d'invitation avec un lien de création de compte

4. L'utilisateur reçoit l'email et finalise la création de son compte

**A2 : Modification des droits d'accès**

1. L'administrateur sélectionne un utilisateur

2. L'administrateur modifie le rôle ou les permissions spécifiques

3. Le système applique les nouvelles permissions immédiatement

**A3 : Réinitialisation du mot de passe**

1. L'administrateur sélectionne l'option "Réinitialiser le mot de passe" pour un utilisateur

2. Le système génère un email avec un lien de réinitialisation

3. L'utilisateur définit un nouveau mot de passe via le lien reçu

**Exceptions :**

**E1 : Email déjà utilisé**

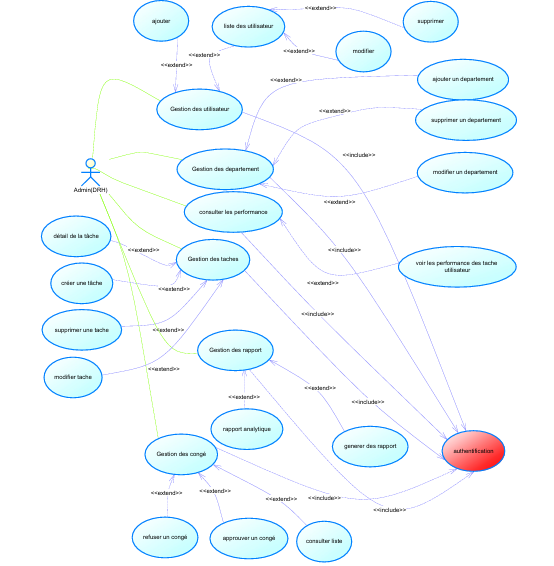
1. Lors de la création d'un utilisateur, si l'email est déjà utilisé, le système affiche un message d'erreur

**E2 : Suppression d'un administrateur unique**

1. Si l'administrateur tente de supprimer le dernier compte administrateur, le système bloque l'action et affiche un avertissement

**Diagramme de cas d'utilisation détaillé**

Le diagramme ci-dessous détaille les différentes fonctionnalités accessibles à l'administrateur, organisées par modules :



Ce diagramme met en évidence la diversité des fonctionnalités accessibles à l'administrateur et leur organisation hiérarchique, facilitant la compréhension des responsabilités associées à ce rôle.

**IV.1.4. Raffinement du cas d'utilisation "Employé"**

Le cas d'utilisation "Employé" englobe l'ensemble des fonctionnalités accessibles aux utilisateurs ayant le rôle d'employé dans le système. Son raffinement permet de détailler les différentes actions qu'un employé peut effectuer au quotidien pour gérer ses activités professionnelles.

**Description détaillée**

**Nom :** Gérer ses tâches (sous-cas d'utilisation de l'Employé)

**Acteur principal :** Employé

**Préconditions :** L'employé est authentifié dans le système

**Postconditions :** Les modifications sur les tâches sont enregistrées dans le système

**Flux principal :**

1. L'employé accède à la section de gestion des tâches

2. Le système affiche la liste des tâches assignées à l'employé

3. L'employé peut consulter les détails d'une tâche, mettre à jour son statut ou ajouter des commentaires

4. Le système enregistre les modifications et met à jour la liste des tâches

**Flux alternatifs :**

**A1 : Création d'une nouvelle tâche personnelle**

1. L'employé clique sur "Ajouter une tâche"

2. L'employé saisit les informations de la tâche (titre, description, échéance)

3. Le système enregistre la tâche et l'ajoute à la liste

**A2 : Mise à jour du statut d'une tâche**

1. L'employé sélectionne une tâche

2. L'employé modifie le statut (à faire, en cours, terminée)

3. Le système met à jour le statut et notifie le manager si nécessaire

**A3 : Ajout d'un commentaire ou d'une pièce jointe**

1. L'employé sélectionne une tâche

2. L'employé ajoute un commentaire ou télécharge une pièce jointe

3. Le système enregistre l'ajout et notifie les personnes concernées

**Exceptions :**

**E1 : Échéance dépassée**

1. Si l'échéance de la tâche est dépassée, le système affiche un indicateur visuel

2. L'employé peut toujours mettre à jour le statut mais un commentaire est requis

**Diagramme de cas d'utilisation détaillé**

Le diagramme ci-dessous détaille les différentes fonctionnalités accessibles à l'employé, organisées par modules :



Ce diagramme met en évidence la diversité des fonctionnalités accessibles à l'employé et leur organisation hiérarchique, facilitant la compréhension des possibilités offertes par le système pour la gestion quotidienne des activités professionnelles.

**Diagramme de classes du système**

Ce diagramme de classes représente un système de gestion d'entreprise avec plusieurs entités principales et leurs relations.



Ce diagramme de classes constitue une base solide pour un système de gestion d'entreprise, couvrant les principaux aspects tels que les départements, les employés, les tâches, les présences et les congés.

**IV.2. Besoins non fonctionnels**

Les besoins non fonctionnels définissent les critères de qualité et les contraintes que le système doit respecter, indépendamment des fonctionnalités spécifiques. Ils jouent un rôle crucial dans la satisfaction des utilisateurs et l'acceptation du système.

Ces besoins ont été regroupés par catégories, selon la classification standard des attributs de qualité logicielle :

***Tableau II.3 : Besoins non fonctionnels***

|  |  |
| --- | --- |
| **Catégorie** | **Besoins non fonctionnels** |
| **Performance et efficacité** | - Le système doit supporter simultanément au moins 200 utilisateurs actifs sans dégradation notable des performances.  - Le temps de réponse pour les opérations courantes ne doit pas excéder 2 secondes  - Les opérations de génération de rapports complexes doivent s'exécuter en moins de 30 secondes.  - Le système doit être optimisé pour minimiser la consommation de ressources (CPU, mémoire, bande passante).  - Les requêtes à la base de données doivent être optimisées pour garantir des temps de réponse rapides. |
| **Disponibilité et fiabilité** | - Le système doit être disponible 24/7, avec un taux de disponibilité d'au moins 99,5%  - Les opérations de maintenance planifiées doivent être réalisées en dehors des heures de travail.  - Le système doit inclure des mécanismes de reprise après incident, avec un objectif de temps de reprise (RTO) de 4 heures maximum  - Les données critiques doivent être sauvegardées quotidiennement, avec un objectif de point de reprise (RPO) de 24 heures maximum.  - Le système doit gérer gracieusement les erreurs et fournir des messages explicites aux utilisateurs. |
| **Sécurité et confidentialité** | - L'accès au système doit être sécurisé par une authentification forte, avec support de l'authentification.  - Les mots de passe doivent respecter des règles de complexité et être stockés de manière sécurisée (hachage avec sel).  - Les communications entre le client et le serveur doivent être chiffrées via HTTPS.  - Le système doit implémenter un contrôle d'accès basé sur les rôles (RBAC) pour protéger les données sensibles.  - Les actions sensibles doivent être journalisées pour permettre un audit de sécurité.  - Le système doit être protégé contre les attaques courantes (injection SQL, XSS, CSRF, etc.).  - Le traitement des données personnelles doit être conforme aux réglementations en vigueur (RGPD, etc.). |
| **Utilisabilité et expérience utilisateur** | - L'interface utilisateur doit être intuitive et facile à prendre en main, même pour des utilisateurs non techniques.  - Le système doit être responsive et s'adapter à différents types de terminaux (ordinateurs, tablettes, smartphones).  - Les fonctionnalités fréquemment utilisées doivent être accessibles en un minimum de clics.  - Le système doit fournir une aide contextuelle pour guider les utilisateurs.  - Les messages d'erreur doivent être clairs et proposer des solutions lorsque c'est possible.  - L'interface doit respecter les principes d'accessibilité (WCAG 2.1 niveau AA) pour les utilisateurs en situation de handicap. |
| **Maintenabilité et évolutivité** | - Le code source doit être modulaire, bien documenté et respecter les standards de codage définis.  - L'architecture doit permettre l'ajout de nouvelles fonctionnalités avec un impact minimal sur l'existant.  - Le système doit être conçu pour évoluer facilement en termes de nombre d'utilisateurs et de volume de données.  - Les dépendances externes doivent être minimisées et clairement documentées  - Le système doit inclure des tests automatisés (unitaires, d'intégration, fonctionnels) pour faciliter les évolutions.  - La documentation technique doit être complète et maintenue à jour. |
| **Interopérabilité et intégration** | - Le système doit exposer des API RESTful documentées pour permettre l'intégration avec d'autres applications.  - Les formats d'échange de données doivent respecter des standards ouverts (JSON, CSV, etc.).  - Le système doit pouvoir s'intégrer avec les outils de paie et de comptabilité courants.  - L'import et l'export de données doivent être possibles dans des formats standards.  - Le système doit supporter l'authentification unique (SSO) pour s'intégrer dans l'écosystème informatique de l'entreprise. |
| **Conformité légale et réglementaire** | - Le système doit être conforme aux réglementations sur la protection des données personnelles (RGPD, etc.).  - Le système doit respecter les obligations légales en matière de conservation des données RH.  - Les fonctionnalités de pointage doivent être conformes à la législation du travail.  - Le système doit permettre la génération des documents légaux obligatoires.  - Les processus de validation doivent intégrer les workflows requis par la réglementation. |

Ces besoins non fonctionnels ont guidé les choix d'architecture et de conception, et constituent des critères d'évaluation essentiels pour la validation du système final.

**V. Conclusion**

L'analyse et la spécification des besoins ont permis de définir de manière précise et structurée les attentes fonctionnelles et non fonctionnelles pour le système de gestion des ressources humaines et de pointage. Cette étape fondamentale a fourni une base solide pour les phases ultérieures du projet, en garantissant une compréhension partagée des objectifs et des contraintes entre toutes les parties prenantes.

La méthodologie UML adoptée a facilité la modélisation du système sous différentes perspectives (fonctionnelle, dynamique, structurelle), offrant une représentation visuelle claire des interactions, des processus et des structures de données. Les diagrammes élaborés constituent une documentation précieuse pour le développement et la maintenance future du système.

L'architecture logicielle définie, basée sur le stack MERN et structurée en couches distinctes, offre un cadre robuste et évolutif pour l'implémentation des fonctionnalités identifiées. Les patterns architecturaux choisis favorisent la modularité, la testabilité et la maintenabilité du code, répondant ainsi aux exigences non fonctionnelles du projet.

La capture détaillée des besoins fonctionnels, organisés par modules et raffinés à travers des cas d'utilisation spécifiques, a permis d'identifier clairement les fonctionnalités attendues pour chaque type d'utilisateur. Cette approche modulaire facilite le développement incrémental du système et permet de prioriser les fonctionnalités en fonction de leur valeur métier.

Les besoins non fonctionnels, couvrant des aspects tels que la performance, la sécurité, l'utilisabilité et la maintenabilité, ont défini les critères de qualité essentiels pour garantir l'acceptation et la pérennité du système. Ces exigences transversales ont influencé l'ensemble des décisions techniques et méthodologiques du projet.

La prochaine étape du projet, détaillée dans le chapitre suivant, consistera à traduire ces spécifications en une implémentation concrète, en s'appuyant sur les environnements techniques sélectionnés et en développant les interfaces utilisateur conformes aux besoins identifiés.