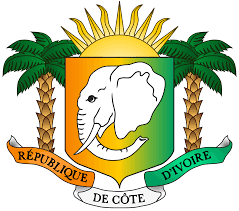
REPUBLIQUE DE COTE D IVORE



Union – Discipline -Travail



 ****

*Université internationale de*

*Yamoussoukro*

*Département des Sciences et technologie*

**RAPPORT DE STAGE**

LICENCE EN INFORMATIQUE OPTION GENIE LOGICIEL(IGL)

**CONCEPTION ET REALISATION D’UN SYSTEME WEB POUR L’OPTIMISATION DES PROCESSURS RH A l’INP-HB**

Période de stage : Du 22 août au 22 novembre

**Réalisé par** : Ayemou Bahé Marie Laure Thérèse

***Encadreur pédagogique : Maitre de stage***

Dr. Kimou ProsperM. N’Guessan Théophile

Enseignant chercheur à L’INP-HB Ingénieur Informaticien à L’INP-HB

**Année Académique 2023 – 2024**

# **DEDICACE**

Je dédie ce travail à ma famille, dont le soutien m’a toujours accompagné tout au long de mon parcours. En particulier, je rends hommage à mon père, dont la mémoire et les valeurs continueront de m'inspirer.

Je souhaite également exprimer ma gratitude envers mes professeurs, dont l’engagement et les conseils ont été précieux pour mon développement personnel et académique.

Merci à tous ceux qui ont contribué à ce cheminement.

# **REMERCIEMENT**

Nous tenons à exprimer notre profonde gratitude envers toutes les personnes qui ont contribuées, de manière directe ou indirecte, à la réussite de notre stage ainsi qu'à l'élaboration de ce rapport.

**A la structure d’accueil**

Je tiens tout d'abord à exprimer ma sincère gratitude à l'ensemble de l'équipe de la sous-direction de la digitalisation et de la sécurité informatique pour leur accueil chaleureux et leur soutien durant mon stage. Leur disponibilité et leur bienveillance ont été des atouts précieux pour ma découverte du milieu professionnel.

**À notre tuteur d'entreprise**

Nous souhaitons également exprimer notre reconnaissance à notre tuteur d'entreprise, M. N’GUESSAN THEOPHILE, dont l'accompagnement rigoureux et la guidance éclairée ont grandement facilité notre immersion et l'exécution de nos fonctions. Ses précieux conseils et son engagement à notre égard ont joué un rôle crucial dans notre intégration professionnelle et m’ont permis d’apprendre. Je lui en suis véritablement reconnaissante.

**À notre Encadreur Pédagogique**  
Nous adressons nos plus sincères remerciements à Dr Kimou Prosper, Enseignant-chercheur à l’Institut National Polytechnique Félix Houphouët-Boigny. Sa disponibilité, ses conseils éclairés et Son accompagnement durant le processus rédactionnel de ce rapport a été d'une aide inestimable, contribuant à la structuration et à la clarté de notre travail.

**SOMMAIRE**

[**DEDICACE** I](#_Toc183457744)

[**REMERCIEMENT** II](#_Toc183457745)

[**SOMMAIRE** III](#_Toc183457746)

[**AVANT-PROPOS** V](#_Toc183457747)

[**RESUME** VII](#_Toc183457748)

[**LISTE DES FIGURES** VIII](#_Toc183457749)

[**LISTE DES TABLEAUX** IX](#_Toc183457750)

[**LISTE DES ABREVIATIONS** X](#_Toc183457751)

[**INTRODUCTION GENERALE** 1](#_Toc183457752)

[Partie 1 : cadre et contexte du projet 3](#_Toc183457753)

[CHAPITRE 1 : PRESENTATION DE LA STRUCTURE D’ACCUEIL (INP-HB) 4](#_Toc183457754)

[INTRODUCTION : 4](#_Toc183457755)

[I. HISTORIQUE ET PRESENTATION DE L’INP-HB 4](#_Toc183457756)

[II. VISION 5](#_Toc183457757)

[III. MISSION 6](#_Toc183457758)

[IV. VALEUR 6](#_Toc183457759)

[V. ORGANISATION DE L’INP-HB 7](#_Toc183457760)

[CONCLUSION 8](#_Toc183457761)

[CHAPITRE 2 : DESCRIPTION DU PROJET 9](#_Toc183457762)

[INTRODUCTION 9](#_Toc183457763)

[**I.** **CONTEXTE DU PROJET** 9](#_Toc183457764)

[**II.** **OBJECTIFS DU PROJET** 9](#_Toc183457765)

[**III.** **CAHIER DES CHARGES** 10](#_Toc183457766)

[IV. Étude de l’existant 12](#_Toc183457767)

[V. Solution proposée 15](#_Toc183457768)

[Conclusion 15](#_Toc183457769)

[CHAPITRE 3 : ETUDE COMPARATIVE 17](#_Toc183457770)

[INTRODUCTION 17](#_Toc183457771)

[I. PRESENTATION DE QUELQUES METHODES D’ANALYSE 17](#_Toc183457772)

[II. COMPARAISON DE PU/UML ET MERISE 24](#_Toc183457773)

[III. CHOIX DE LA METHODE D’ANALYSE 25](#_Toc183457774)

[CONCLUSION 26](#_Toc183457775)

[CHAPITRE 4 : MODELISATION DE NOTRE APPLICATION 27](#_Toc183457776)

[INTRODUCTION 27](#_Toc183457777)

[Conclusion 61](#_Toc183457778)

[PARTIE 3 : Etude technique et réalisation 62](#_Toc183457779)

[CHAPITRE 5 : ETUDE TECHNIQUE 63](#_Toc183457780)

[INTRODUCTION 63](#_Toc183457781)

[**I.** **DESCRIPTION DE L’ENVIRONNEMENT DE TRAVAIL** 63](#_Toc183457782)

[**II.** **ANALYSE DES BESOINS TECHNIQUES** 63](#_Toc183457783)

[**III.** **CHOIX DES DIFFERENTS OUTILS, LANGAGE DE PROGRAMMATION ET FRAMEWORK** 64](#_Toc183457784)

[IV. Architecture de l'application 73](#_Toc183457785)

[V. Mise en œuvre des composants 74](#_Toc183457786)

[Conclusion 75](#_Toc183457787)

[CHAPITRE 6 : REALISATION DE L’APPLICATION 76](#_Toc183457788)

[INTRODUCTION 76](#_Toc183457789)

[III. PRESENTATION DE L'APPLICATION 77](#_Toc183457790)

[Conclusion 81](#_Toc183457791)

[**CONCLUSION** 82](#_Toc183457792)

[**BIBLIOGRAPHIE** 83](#_Toc183457793)

[**WEBOGRAPHIE** 84](#_Toc183457794)

# **AVANT-PROPOS**

Pourquoi ce projet ?

Établie en 2019 dans la capitale politique et administrative de la Côte d'Ivoire, l’Université Internationale de Yamoussoukro (UIYA) s’affirme comme un pôle d'excellence en matière d'enseignement supérieur pluridisciplinaire. Cet établissement offre un cadre propice à l'épanouissement intellectuel, intégrant les évolutions les plus récentes dans les domaines du management et des technologies de l'information.

L'UIYA propose un éventail de formations dans plusieurs disciplines, notamment :

* **Sciences Mathématiques et Informatiques**
  + Informatique, Option Génie Logiciel
  + Big Data /Analyste de données IA
  + Sécurité Informatique
* **Sciences Économiques et de Gestion**
  + Audit et Contrôle de Gestion
  + Management des Projets
  + Economie
  + Gestion
  + Finance Comptabilité
  + Finance Banque et Assurances
* **Sciences de l’Information, de la Communication et des Ressources Humaines**
  + Communication en Entreprise
  + Journalisme Télévisé
  + Ressources Humaines et Communication (RHCOM)
* **Sciences Juridiques**
  + Droit Public
  + Droit Privé
* **Sciences Linguistiques**
  + Anglais

L’UIYA s’est assignée pour mission de former des managers compétents, capables de concevoir et de mettre en œuvre des stratégies d’entreprise tant au niveau national qu'international, conformément au système d’enseignement basé sur le modèle LICENCE-MASTER-DOCTORAT (LMD).

Après trois années d'études approfondies dans le domaine de l'informatique, option Génie Logiciel, à l'Université Internationale de Yamoussoukro (UIYA), j'ai eu l'opportunité de mettre en pratique les connaissances acquises en concevant une application web dédiée à la gestion des ressources humaines. Ce projet s'inscrit dans le cadre de l'application concrète des compétences théoriques développées au cours de ma formation.

Dans cette optique, j'ai formulé une demande de stage auprès de la Direction des Ressources Humaines de l'Institut National Polytechnique Félix Houphouët-Boigny (INPHB), où j'ai eu l'honneur d'intégrer le Département des Systèmes d'Information (DSI) du 22 août au 22 novembre.

Le présent document constitue mon rapport de stage, un élément essentiel pour l'obtention de ma licence en Sciences Mathématiques et Informatiques, spécialisation Informatique, option Génie Logiciel. Ce rapport retrace la conception et la réalisation d'une application web visant à optimiser les processus RH au sein de l'INP-HB, en mettant l'accent sur les différentes étapes du projet, de l'analyse des besoins à la mise en production de l'outil.

# **RESUME**

Reçu au sein du département Informatique de l'Institut National Polytechnique Félix Houphouët-Boigny (INP-HB), nous avons réalisé notre stage de fin d'études dans le cadre du projet **“Conception et réalisation d’un système web pour l’optimisation des processus RH”**. Ce projet visait à moderniser et automatiser les processus administratifs liés à la gestion des ressources humaines au sein de l'établissement, afin de rendre la gestion du personnel plus efficace, sécurisée et accessible.

Le projet a abouti à la conception de **drhConnect**, une application web permettant de centraliser et de gérer les informations administratives des agents, telles que l'enregistrement des employés, la gestion des demandes de permissions et la consultation des données RH. L'objectif était de réduire la paperasse et d'optimiser le temps de gestion des ressources humaines tout en garantissant la sécurité des données.

Pour la réalisation de l'application, nous avons utilisé une architecture moderne, combinant **React** pour l’interface utilisateur, **Laravel** comme Framework backend, et **Inertia.js** pour assurer une interaction fluide entre le front-end et le back-end sans avoir à recourir à une API complète. La base de données a été gérée avec **MySQL**, et le serveur local a été configuré à l'aide de **WAMP** pour une gestion efficace des environnements de développement.

La modélisation du système a été réalisée à l’aide de **diagrammes UML**, permettant de formaliser les exigences fonctionnelles et non fonctionnelles de l'application et de structurer clairement les processus à implémenter.

Bien que le projet n'ait pas été entièrement achevé, l'application **drhConnect** a permis de répondre aux objectifs fixés : elle simplifie et sécurise la gestion des ressources humaines à l’INP-HB, en offrant un système efficace et facile à utiliser pour les administrateurs et les employés.

**Mots clés** : gestion des ressources humaines, application web, React, Laravel, Inertia.js, MySQL, WAMP, modélisation UML, optimisation administrative, INP-HB, application monopage.

# **LISTE DES FIGURES**

[Figure 1 : organigramme de la sous-direction de la digitalisation et sécurité informatique 7](#_Toc183459498)

[Figure 2 : courbe du soleil 19](#_Toc183459499)

[Figure 3 : Déroulement du processus Unifié 20](#_Toc183459500)

[Figure 4 : Evolution d 'UML à travers les versions 21](#_Toc183459501)

[Figure 5 : L’architecture bidirectionnelle 23](#_Toc183459502)

[Figure 6:Diagramme de cas d’utilisation global 29](#_Toc183459503)

[Figure 7:Diagramme de cas d’utilisation gestion compte utilisateur 30](#_Toc183459504)

[Figure 8:Diagramme de cas d’utilisation gérer Absence 30](#_Toc183459505)

[Figure 9:Diagramme de cas d’utilisation gestion du personnel 31](#_Toc183459506)

[Figure 10:Diagramme de cas d’utilisation droit d’accès 32](#_Toc183459507)

[Figure 11:Diagramme de cas d’utilisation suivi affectation 33](#_Toc183459508)

[Figure 12:Description textuelle du cas demander permission 48](#_Toc183459509)

[Figure 13:Description textuelle du cas valider permission 50](#_Toc183459510)

[Figure 14:Diagramme de séquence du cas S’authentifier 51](#_Toc183459511)

[Figure 15:Diagramme de séquence du cas Modifier mot de passe 52](#_Toc183459512)

[Figure 16:Diagramme de séquence du cas Modifier photo de profil 53](#_Toc183459513)

[Figure 17:Diagramme de séquence du cas se déconnecter 53](#_Toc183459514)

[Figure 18:Diagramme de séquence du cas Editer document 54](#_Toc183459515)

[Figure 19:Diagramme de séquence du cas Consulter la liste des agents affectés 55](#_Toc183459516)

[Figure 20:Diagramme de séquence du cas Afficher détail agent 56](#_Toc183459517)

[Figure 21: Diagramme de séquence du cas affecter un agent 57](#_Toc183459518)

[Figure 22:Diagramme de séquence du cas Ajouter un agent 58](#_Toc183459519)

[Figure 23:Diagramme de séquence du cas bloquer compte d un agent 59](#_Toc183459520)

[Figure 24:Diagramme de séquence du cas Modifier un agent 60](#_Toc183459521)

[Figure 25:Diagramme de classe 61](#_Toc183459522)

[Figure 26 : Logo MySQL 64](#_Toc183459523)

[**Figure 27 : Logo Wamp** 65](#_Toc183459524)

[Figure 28 : Logo VS code 66](#_Toc183459525)

[Figure 29 : Logo GitHub 67](#_Toc183459526)

[Figure 30 : Logo Laravel 67](#_Toc183459527)

[Figure 31 : Logo Tailwind Css 68](#_Toc183459528)

[Figure 32 : Logo Inertia 69](#_Toc183459529)

[Figure 33 :Logo React 72](#_Toc183459530)

[Figure 34 : Logo PhpSpreadSheat 73](#_Toc183459531)

[Figure 35:Page de connexion 78](#_Toc183459532)

# **LISTE DES TABLEAUX**

[Tableau 1 : Comparaison entre Merise et PU/UML 25](#_Toc183457851)

[Tableau 2: Identification des acteurs 28](#_Toc183457852)

[Tableau 3: Identification des cas d’utilisation 28](#_Toc183457853)

[Tableau 4 : Description textuelle du cas s ' authentifier 34](#_Toc183457854)

[Tableau 5:Description textuelle du cas Modifier mot de passe 35](#_Toc183457855)

[Tableau 6:Description textuelle du cas Modifier photo de profil 36](#_Toc183457856)

[Tableau 7:Description textuelle du cas Se déconnecter 36](#_Toc183457857)

[Tableau 8:Description textuelle du cas Editer document 38](#_Toc183457858)

[Tableau 9:Description textuelle du cas Consulter la liste des agents affectés 39](#_Toc183457859)

[Tableau 10: Description textuelle du cas Afficher détail agent 40](#_Toc183457860)

[Tableau 11:Description textuelle du cas affecter un agent 41](#_Toc183457861)

[Tableau 12:Description textuelle du cas Ajouter un agent 42](#_Toc183457862)

[Tableau 13: Description textuelle du cas bloquer compte d’un agent 43](#_Toc183457863)

[Tableau 14: Description textuelle du cas modifier agent 44](#_Toc183457864)

[Tableau 15: Description textuelle du cas gérer rôle 45](#_Toc183457865)

[Tableau 16:Description textuelle du cas gérer permission 46](#_Toc183457866)

# **LISTE DES ABREVIATIONS**

* **INP-HB** : Institut National Polytechnique Félix Houphouët-Boigny
* **GRH** : Gestion des Ressources Humaine
* **UI** : Interface Utilisateur
* **API** : Interface de Programmation d'Application
* **CRUD** : Create, Read, Update, Delete (Opérations de gestion des données)
* **UML** : Unified Modeling Language
* **PHP** : Hypertext Preprocessor
* **JS** : JavaScript (langage de programmation utilisé pour l'interface utilisateur)
* **SQL** : Structured Query Language (langage de requête pour les bases de données)
* **WAMP** : Windows, Apache, MySQL, PHP (Environnement de développement web local)
* **MVC** : Modèle-Vue-Contrôleur (Architecture de développement logiciel)
* **SPA** : Single Page Application

# **INTRODUCTION GENERALE**

La technologie progresse rapidement et transforme profondément tous les secteurs d'activité. L'informatique occupe aujourd'hui une place centrale dans le développement des institutions et des entreprises. Dans un environnement en constante évolution, les systèmes informatisés sont devenus essentiels pour une gestion plus rapide, plus sûre et plus fiable des informations. Cette évolution technologique offre une valeur ajoutée considérable, permettant aux organisations de mieux répondre aux défis contemporains. L'innovation, notamment à travers des outils numériques, facilite l'automatisation des tâches, réduit les erreurs humaines et améliore la transparence des processus, offrant ainsi un avantage concurrentiel décisif dans un monde de plus en plus globalisé et numérisé.

Avant l’informatique, la gestion des données était manuelle, souvent sur papier, engendrant des problèmes tels que la perte de temps, la dégradation des documents, des erreurs humaines et une gestion complexe des archives. Cette méthode traditionnelle, notamment dans les ressources humaines, a montré ses limites à cause de l'inefficacité et de l’accessibilité restreinte aux informations essentielles.

La gestion actuelle des ressources humaines à l'INP-HB repose sur des pratiques manuelles et obsolètes, telles que l'enregistrement des employés via des fichiers Excel et la gestion des demandes de permissions sur papier. Ces méthodes engendrent des retards dans le traitement des demandes, augmentent la vulnérabilité aux erreurs humaines, et limitent l'accès aux informations essentielles des employés. Face à ces défis, l'INP-HB se doit de trouver une solution numérique capable de centraliser les données, de digitaliser les processus administratifs et de garantir la sécurité des informations.

C’est dans ce contexte que la problématique de notre projet émerge : Comment une solution numérique pourrait-elle améliorer l'efficacité de la gestion des ressources humaines à l'INP-HB ?

C’est dans cette optique que notre projet prend forme : **CONCEPTION ET REALISATION D’UN SYSTEME WEB POUR L’OPTIMISATION DES PROCESSURS RH A l INP-HB**. L’objectif de **drhConnect** est de centraliser et de numériser les informations liées aux employés afin de rendre leur gestion plus fluide, plus sécurisée et plus transparente. À travers ce projet, nous avons cherché à offrir une solution moderne pour la gestion des ressources humaines, qui permet à l’INP-HB de mieux gérer son capital humain tout en optimisant ses processus administratifs.

Notre rapport de stage est structuré en trois parties principales :

La première partie présente le cadre et le contexte du projet, en exposant les enjeux, objectifs et motivations de la création de l’application **drhConnect**, tout en situant le projet dans son contexte institutionnel et en identifiant la problématique qu'il résout.

La deuxième partie porte sur l’étude conceptuelle, détaillant la conception fonctionnelle de l’application, les exigences du système, ainsi que la modélisation à l’aide de diagrammes UML, illustrant comment les besoins ont été traduits en fonctionnalités.

La dernière partie traite de l’étude technique et de la réalisation du projet, en expliquant les choix technologiques, les étapes de développement et les tests effectués pour valider le bon fonctionnement du système. Des **captures d'écran** de l’application sont également fournies pour illustrer l’interface utilisateur et certaines fonctionnalités clés. Elles aident à mieux comprendre les résultats de l’implémentation et la manière dont les utilisateurs interagiront avec l'application.

Enfin, notre travail est conclu par une conclusion générale et quelques perspectives ouvertes qui peuvent être envisagées comme suite à notre projet

PARTIE 1 : CADRE ET CONTEXTE DU PROJET

CHAPITRE 1 : PRESENTATION DE LA STRUCTURE D’ACCUEIL (INP-HB)

INTRODUCTION :

Avant d’entrer dans le vif du sujet, ce chapitre présente la structure d’accueil INPHB, son histoire, ses vision, missions et valeurs, de même que d’autres informations pertinentes sur ladite structure.

1. HISTORIQUE ET PRESENTATION DE L’INP-HB

Créé par décret 96-678 du 04 septembre 1996, l'Institut national polytechnique Félix HOUPHOUËT-BOIGNY (INP-HB) est le résultat de la fusion de l'Ecole nationale supérieure d'agronomie (ENSA), de l'Ecole nationale supérieure des travaux publics (ENSTP), de l'Institut agricole de Bouaké (IAB) et de l'Institut national supérieur de l'enseignement technique (INSET). Quatre institutions que l'on appelle généralement les Grandes Écoles de Yamoussoukro.

Des milliers de Managers, ingénieurs et techniciens supérieurs dans les secteurs du Commerce, des Tavaux Publics, de l'Agronomie, de l'Industrie, des Mines et de la Géologie ont été formés par l'institut, qui est reconnu en Afrique et à travers le monde. Outre ses cours fondamentaux, l'INP-HB a également mis en place divers programmes.

À ce jour, l’INP-HB dispose de 3008 chambres, accueille plus de 6000 élèves et compte plus de 520 enseignants, ainsi que onze (11) écoles réparties sur trois sites, à savoir :

* INP-Sud
* INP-Nord
* INP-Centre

Le site d'INP-Sud comprend plusieurs Écoles, Directions et Services, soit :

-La Direction Générale Adjointe Chargée de la Pédagogie et de la Recherche (DGA - PR) ;

-L'Ecole Supérieure des Travaux Publics (ESTP) ;

-L'Ecole Supérieure des Mines et Géologie (ESMG) ;

-L'Ecole Supérieure du Pétrole et de l'Energie (ESPE) ;

-La Direction des Ressources Humaines ;

-Le Centre de Promotion et d'Expertise (CPEX) ;

-Le Centre Régional de Formation Supérieure en Métrologie (CREFSEM) ;

-Le Centre d'Excellence Africain en Mines et Environnement Minier (CEA - MEM) .

**Par ailleurs**, le site d'INP-Centre, bâti en 1983, abrite aujourd’hui :

-La Direction Générale (DG) ;

-L'Ecole Supérieure d'Industrie (ESI) ;

-L'Ecole Supérieure de Commerce et d'Administration d'Entreprise (ESCAE) ;

-L'International Data Science Institute (IDSI) ;

-L'usine école du Centre .

De plus, le site d'INP-Nord, construit en 1983, abrite aujourd'hui un ensemble d'institutions et de services, parmi lesquels figurent :

-Le Centre de Formation Préparatoire au Diplôme d'Expertise Comptable de l'UEMOA ;

-La Direction Générale Adjointe Chargée de la Coopération Internationale et du Développement du Technopôle (DGA-CIT) ;

-Le Centre d'Excellence Africain pour la Valorisation des Déchets en Produits à Haute Valeur Ajoutée (CEA-VALOPRO).

En outre, trois écoles sont également représentées :

-L'école supérieure d'agronomie (ESA) ;

-L’école de formation continue et de perfectionnement des cadres (EFCPC) ;

-L’école doctorale .

1. VISION

Le but de l'INP-HB est de devenir le leader dans la formation des Ingénieurs et des Techniciens Supérieurs en Afrique et de faire connaître la qualité de son système dans les pays de la sous-région.   
Cela nécessite :

-La préservation d'un cadre de travail sain, favorable à des études supérieures de qualité supérieure ;

-La promotion de la culture de l'excellence depuis les concours jusqu'à l'obtention des diplômes ;

-La création de collaborations stratégiques en Afrique et à l'étranger avec des universités de renom et des sociétés savantes à travers le monde ;

-La mise en œuvre de partenariats avec les entreprises aussi bien pour les stages de production que pour l’expertise ;

-L’excellence de la recherche appliquée aux besoins de l’industrie afin de faire de l’INP-HB, le hub de l’innovation en Afrique ;

-Le suivi des étudiants dans les prestigieuses écoles à l’International tout en mobilisant le réseau et le soutien des alumni de l’INP-HB.

1. MISSION

En offrant des formations, des recherches, des conseils et des expertises dans les domaines primaire, secondaire et tertiaire, l'Institut s'efforce de satisfaire les besoins des entreprises et d'améliorer les conditions de vie des habitants.   
L'Institut se positionne peu à peu comme un acteur essentiel de l'intégration régionale et africaine. Ses tâches sont résumées en trois éléments :

* La formation qui comprend des formations initiales et continues qui permettent de former des techniciens supérieurs, des ingénieurs des techniques et des ingénieurs de conception dans les secteurs de l'industrie, du commerce, de l'administration, du génie civil, des mines et de la géologie.
* L’innovation : La recherche et le développement
* La Transformation : Apport d'expertise, d'assistance et de production pour les entreprises et les administrations.

1. VALEUR

Pour illustrer l'engagement de l'INP-HB envers sa mission d'excellence et de développement sociétal, voici les valeurs fondamentales qui guident ses actions et contribuent à son rayonnement dans la sous-région :

-L’excellence de la formation et de la recherche ;

-L’innovation au service de la population ;

-Le professionnalisme de l’administration ;

-Une politique d’ouverture et de Partenariat institutionnel ;

- Le respect de l’environnement.

1. ORGANISATION DE L’INP-HB
2. **Organisation**

La direction Générale de l INP-HB comprend :

-Le directeur Général ordonnateur du budget ;

-Le directeur Générale adjoint chargé de la planification, de la programmation et de la coordination des enseignements et des activités de recherche ;

-Le secrétaire General charge de la coordination des services administratifs et techniques ;

-Le directeur des Ressources Humaines ;

-Le directeur des finances ;

Toutes ces directions disposent de sous-direction et de service.

1. **La sous-direction d’accueil : la sous-direction de la digitalisation et de la sécurité informatique**

Figure 1 : organigramme de la sous-direction de la digitalisation et sécurité informatique

Ce service est chargé de :

-Développer les applications liées aux activités de l’INP-HB ;

-Traiter les données à caractère personnel des étudiants pour renseigner la base de données ;

-Gérer les différents concours au sein de l’INP-HB ;

-Gérer l’application du restau ;

-Editer et imprimer le badge des étudiants.

CONCLUSION

Avant de donner une vision claire du contexte général du projet ainsi introduire le travail effectué, ce chapitre a été consacré à la présentation du cadre et de

L’environnement de l’organisme dans lequel nous avons effectué notre stage, son

Historique depuis sa création, ainsi que sa vision, ses missions et valeurs

CHAPITRE 2 : DESCRIPTION DU PROJET

INTRODUCTION

Dans ce chapitre, nous présenterons notre projet, en suivant un fil conducteur allant de la problématisation à la solution que nous proposons. Ce chapitre a pour objectif d'explorer en profondeur notre projet, en examinant la problématique rencontrée ainsi que les solutions envisagées pour y répondre.

1. **CONTEXTE DU PROJET**

Ce projet, qui fait partie intégrante de mon cursus en informatique option génie logiciel, est réalisé dans le cadre de mon stage à l'INP-HB. Le thème principal de ce stage est la **"Conception et réalisation d’un système web pour l’optimisation des processus RH à l’INP-HB"**. L'objectif est de développer une application web, nommée **drhConnect**, visant à optimiser la gestion des ressources humaines au sein de l'institution. En numérisant les pratiques RH, nous cherchons à améliorer l'efficacité opérationnelle et à renforcer la satisfaction des employés grâce à une gestion plus fluide et automatisée des données et des processus RH.

1. **OBJECTIFS DU PROJET**
2. **Objectif générale**

L’objectif général de ce projet est de **concevoir et réaliser un système web destiné à l'optimisation des processus de gestion des ressources humaines à l’INP-HB,** permettant une gestion centralisée, efficace et automatisée des informations liées au personnel.

1. **Objectif spécifique**

Pour atteindre l'objectif général, les objectifs spécifiques sont les suivants :

* **Centralisation et gestion des données RH :**

Ce module permet de  
- centraliser toutes les informations des employés ;

- gérer leurs affectations, absences ;

- créer des fiches signalétiques détaillées pour chaque agent.

* **Digitalisation et sécurité des processus :**

Ce volet facilite :  
-Facilite la gestion des demandes de permissions ;

-l’édition de documents administratifs et renforce la sécurité des comptes utilisateurs.

* **Gestion multi-utilisateurs et rôles :**  
  Ce système offre un modèle de gestion des utilisateurs avec différents rôles et permissions, permettant aux responsables RH et aux administrateurs de contrôler l’accès et les fonctionnalités en fonction des responsabilités de chacun.

1. **CAHIER DES CHARGES**

Le cahier des charges est un document fondamental qui définit les exigences fonctionnelles et techniques que l'application « **drhConnect** » doit respecter. Ce document servira de référence tout au long du développement du projet, garantissant ainsi que les objectifs sont atteints et que les besoins des utilisateurs sont satisfaits.

1. **Exigences fonctionnelles**

Les exigences fonctionnelles définissent les fonctionnalités spécifiques que l'application « drhConnect » doit offrir pour répondre aux besoins des utilisateurs. Ces exigences précisent les comportements attendus du système, tant en termes d’entrée que de sortie.

**Gestion des utilisateurs**  
L'application doit permettre aux différents types d'utilisateurs de s'authentifier et d’accéder à leurs interfaces respectives. Deux méthodes d'inscription des employés seront proposées : un enregistrement complet directement sur l'application ou une inscription simplifiée via un lien envoyé par SMS. En matière de sécurité, l’application met en place un système d'authentification par e-mail et mot de passe, avec la possibilité pour l'utilisateur de modifier son mot de passe dès la première connexion. Par ailleurs, il doit être possible d’imprimer la fiche signalétique d'un agent, qui contient des informations personnelles, professionnelles et son historique d'affectations et d'absences. Le système doit également permettre de gérer les comptes utilisateurs avec des rôles et droits d’accès bien définis.

**Gestion des affectations des agents**  
Les responsables RH pourront affecter un agent à un poste ou un lieu de travail spécifique. Si le lieu de travail ou le poste change, l’application devra mettre à jour automatiquement l’affectation. L’historique des affectations des agents doit être consultable à tout moment, permettant de suivre les différents postes et lieux de travail occupés au fil du temps.

**Création et gestion de documents RH**  
L'application devra permettre la création et la gestion des documents nécessaires à la gestion des ressources humaines, tels que des attestations de travail, des contrats, des certificats de présence et des lettres d'affectation etc.

**Consultation des données des employés**  
Les utilisateurs habilités doivent pouvoir accéder à une liste complète des employés avec des informations essentielles telles que la date de prise de service, le grade et la situation matrimoniale etc. L'accès aux données sera sécurisé et limité selon le rôle de l'utilisateur, assurant ainsi la confidentialité des informations sensibles.

**Gestion des demandes de permissions**  
Le système doit permettre la soumission, la gestion et le suivi des demandes de permissions, telles que les congés payés, les congés maladie et les absences justifiées. Les responsables des ressources humaines auront la possibilité de consulter les absences passées et à venir des agents. Des notifications automatiques seront envoyées aux utilisateurs et gestionnaires lors de la soumission et de l’approbation des demandes.

**Exportation des données**  
L'application offrira la possibilité d'exporter les données au format Excel et PDF pour faciliter la génération de rapports et l’analyse des informations.

**Interface de gestion dynamique**  
Une fonctionnalité de recherche dynamique permettra aux utilisateurs de filtrer rapidement les agents, les affectations ou les absences. De plus, un tableau de bord avec des statistiques clés, sera disponible pour une gestion simplifiée.

1. **Exigence Techniques**

Les exigences techniques couvrent les spécifications concernant la performance, l’architecture et les technologies utilisées dans le développement de l’application.

**Technologies utilisées**  
L'application sera développée en utilisant React pour le frontend et Laravel pour le backend, intégrés via Inertia pour offrir une expérience utilisateur fluide. La gestion des données sera assurée par une base de données relationnelle MySQL.

**Compatibilité**  
L'application devra être compatible avec les principaux navigateurs web modernes tels que Chrome, Firefox, Safari et Edge. De plus, l’interface devra être responsive, garantissant une expérience optimale sur ordinateurs, tablettes et smartphones.

**Performances**  
L'application devra répondre de manière rapide et efficace, avec un temps de chargement maximal de 3 secondes pour toutes les fonctionnalités. Elle devra aussi être scalable, c’est-à-dire capable de gérer un grand nombre d'utilisateurs simultanés sans dégradation des performances.

**Documentation**  
Un manuel utilisateur sera fourni afin de guider les employés dans l’utilisation de l’application et de ses fonctionnalités.

1. **Étude de l’existant**

L'objectif de cette section est d'analyser l’état actuel des pratiques de gestion des ressources humaines à l'INP-HB et d'examiner les systèmes et outils utilisés dans ce domaine. Cette analyse permet de mieux comprendre les limitations des solutions en place et de justifier la nécessité du projet **drhConnect**, un système moderne et intégré. En identifiant les faiblesses des processus existants, nous pourrons démontrer comment l’application proposée pourra répondre aux besoins spécifiques de l'INP-HB en matière de gestion des ressources humaines. L'étude de l'existant constitue une étape fondamentale dans la démarche de toute solution informatique, car elle permet d'identifier les lacunes des pratiques en place et de proposer des améliorations concrètes.

1. Description de l’existant

Actuellement, la gestion des ressources humaines à l'INP-HB repose largement sur des méthodes traditionnelles et manuelles. Ces pratiques présentent plusieurs limites importantes qui affectent l'efficacité de la gestion des employés et la fluidité des processus administratifs.

* **Enregistrement des employés dans des fichiers Excel** :

À ce jour, la majorité des informations relatives aux employés sont enregistrées dans des fichiers Excel. Bien que cette méthode soit largement utilisée pour son côté pratique et son faible coût, elle présente plusieurs inconvénients :

**-Risque d’erreurs humaines** : La saisie manuelle des informations est sujette à des erreurs de typographie ou de calcul, ce qui peut entraîner des incohérences dans les données. ;

**-Mise à jour difficile** : L'actualisation des données dans des fichiers Excel devient complexe, surtout lorsque plusieurs personnes doivent y accéder et apporter des modifications. Cela peut créer des conflits de version et des informations obsolètes ;

**-Accès limité** : L'accès aux fichiers Excel est souvent restreint et nécessite des autorisations spécifiques. En outre, les fichiers sont souvent stockés localement ou sur des serveurs peu sécurisés, ce qui compromet la confidentialité des données.

* **Gestion des demandes administratives sur papier** :

Les demandes de congé, d'absence ou d'autres types de permissions sont encore soumises sur des formulaires papier. Cela entraîne plusieurs difficultés :

**-Retards et pertes de documents** : Le traitement des demandes papier prend beaucoup de temps et peut entraîner des retards dans le traitement des demandes des employés. Par ailleurs, il existe un risque de **perte de documents** en raison d'un mauvais archivage.

**-Manque de traçabilité** : Le suivi des demandes, une fois qu'elles ont été soumises, est souvent manuel. Il est difficile de savoir à tout moment où en est une demande (en attente, approuvée, rejetée) ;

**-Difficulté d’accès à l’information** : L'information sur les absences ou les permissions est souvent dispersée entre différents supports, rendant leur gestion complexe et peu efficace.

* **Difficultés d’analyse** :

Les données relatives aux employés sont souvent éparpillées dans des fichiers Excel, des formulaires papier ou des systèmes disparates. Cela complique l’analyse des données, rendant les **rapports et les bilans** de gestion difficiles à produire, voire inaccessibles. L'absence d'un système centralisé signifie qu'il est difficile d’obtenir une vue d'ensemble sur la gestion des ressources humaines en temps réel.

1. Critique ou limite de l’existant

L'analyse des méthodes et des outils actuels de gestion des ressources humaines à l'INP-HB révèle plusieurs limites importantes qui justifient la mise en place d'un système intégré et moderne, tel que **drhConnect**.

**Manque de centralisation et d'intégration**

L'un des problèmes majeurs des solutions existantes est l'absence de centralisation des données. Les informations relatives aux employés sont dispersées entre différents formats et supports, tels que les fichiers Excel, les documents papier et les systèmes isolés. Cette dispersion des données complique leur gestion et empêche une vision globale et cohérente de l'état des ressources humaines de l'institution. De plus, l’absence d’intégration entre les différents systèmes rend la gestion des informations fragmentée et non fluide, augmentant ainsi le risque d'erreurs et de pertes de données.

**Problèmes de sécurité et de confidentialité**

Le stockage des informations sensibles des employés sur des fichiers Excel non sécurisés ou des documents papier crée de sérieux risques en matière de sécurité et de confidentialité. En l'absence de contrôle d'accès et de sauvegardes sécurisées, les données des employés peuvent être exposées à des risques de vol, de perte ou de manipulation malveillante. De plus, l’accès limité aux informations selon les autorisations complique la gestion centralisée et la collaboration au sein des équipes RH.

**Inefficacité des processus administratifs**

Les processus administratifs basés sur des formulaires papier et des fichiers Excel sont lents, sujet aux erreurs humaines, et difficiles à suivre. Le manque d'automatisation entraîne des délais dans le traitement des demandes de congé et des absences, tandis que le manque de traçabilité rend difficile le suivi de l'état d'avancement des demandes. En conséquence, les employés et les responsables RH perdent du temps dans des tâches administratives répétitives et peu productives.

**- Difficulté de gestion en temps réel et de prise de décision**

En raison de la dispersion des données et du manque de centralisation, il est difficile pour les responsables RH d’avoir une vue d'ensemble actualisée de la gestion des ressources humaines. Les processus manuels d'analyse rendent la production de rapports et la prise de décision lente et réactive plutôt que proactive. Dans un environnement en constante évolution, cette lenteur peut nuire à la réactivité de l'institution.

1. **SOLUTION PROPOSEE**

Pour répondre à la problématique identifiée, nous proposons de développer l'application « **drhConnect** », qui centralisera les données des employés et numérisera les processus administratifs.

Conclusion

Ce chapitre a structuré la présentation de notre projet, de la problématique à la solution proposée. Nous avons mis en évidence les lacunes des méthodes actuelles et justifié l'importance d'une application moderne. Les avantages de « **drhConnect** » démontrent son potentiel à transformer la gestion des ressources humaines à l'INP, en répondant efficacement aux besoins des utilisateurs.

PARTIE 2 : ETUDE CONCEPTUELLE

CHAPITRE 3 : ETUDE COMPARATIVE

INTRODUCTION

Dans le cadre de la conception de notre application, il est essentiel d'utiliser des méthodes d'analyse robustes qui permettent de structurer et d'orienter le développement. Les méthodes d'analyse de systèmes sont des outils stratégiques qui aident à visualiser et à comprendre les besoins des utilisateurs, ainsi que les spécificités des systèmes à développer, tout en réduisant les erreurs durant le processus de développement. Dans ce chapitre, nous allons examiner différentes méthodes d’analyse, notamment la méthode Merise et PU/UML (Unified Modeling Langage), dont UML, un langage de modélisation largement utilisé dans la conception logicielle. Une étude comparative des deux approches suivies du choix de la méthode la plus adaptée à notre projet nous permettra d'appuyer la décision de conception et d'implémentation de notre application. Ce chapitre se clôturera sur une réflexion sur le choix de la méthode d’analyse que nous avons privilégiée et sur les raisons qui la justifient.

1. PRESENTATION DE QUELQUES METHODES D’ANALYSE

Il est fondamental de distinguer une méthode d'analyse d'une notation ou d'un langage de modélisation. Une méthode d'analyse, comme Merise, est un ensemble de principes et d’étapes qui guide le processus d’analyse et de conception des systèmes, souvent avec une focale particulière sur les exigences métier et les processus. À l'inverse, UML est un langage de modélisation qui fournit des notations graphiques permettant de représenter visuellement les différents aspects d’un système, facilitant ainsi la communication entre les développeurs et les parties prenantes.

1. **Présentation de la méthode merise**

* **Origine de merise**

La méthode Merise, développée dans les années 1970 par Hubert Tardieu et Jean-Louis Lemoigne, est une méthodologie d'informatisation qui a été conçue pour répondre aux besoins des systèmes d'information des entreprises. Créée initialement au sein du ministère de l'Industrie et du Centre technique d'informatique (CTI), Merise a été enrichie par des travaux académiques menés à l’Université d’Aix-Marseille et à l'INRIA.

Formalisée en 1978, la méthode Merise a été légitimée en 1983 par la norme AFNOR Z67-101. Elle repose sur une approche systémique, permettant de visualiser l’organisation dans son ensemble et d’intégrer les différentes facettes d'un système d'information. Cela permet de concilier les enjeux stratégiques et opérationnels d’un projet informatique.

Au fil des années, Merise a évolué pour s’adapter aux nouvelles technologies, notamment celles liées à l’objet et aux architectures client-serveur. Bien que des adaptations aient été faites, les principes de base de Merise restent une référence en matière de gestion de systèmes d'information, témoignage de la robustesse de la méthode.

* **Concepts de merise**
* **Cycle**

Dans le cadre du développement d'un système d'information, l'approche MERISE adopte une perspective multidimensionnelle d'analyse de l'organisation étudiée, ce qui se traduit par l'identification de plusieurs angles d'observation, appelés cycles. Cette méthodologie se compose de trois cycles fondamentaux :

**-Le cycle d'abstraction** : Ce premier cycle vise à modéliser l'organisation de manière conceptuelle, en se concentrant sur les besoins et les objectifs informationnels. Il s'agit d'identifier les principaux éléments constitutifs du système sans tenir compte des contraintes techniques. Ce cycle permet de développer une vision globale et intuitive des processus métier, des flux de données et des interactions entre les entités. Les principaux outils utilisés dans ce cycle sont :

-Modèle Conceptuel de Données (MCD) ;

-Modèle Conceptuel des Traitements (MCT).

-**Le cycle de vie ou (de développement**) : Ce cycle se concentre sur l'ensemble des étapes nécessaires à la mise en œuvre d'un système d'information, de la conception à sa réalisation, en passant par les phases de développement et de déploiement. Il englobe la planification, le développement, les tests et la maintenance, assurant ainsi un suivi structuré tout au long du cycle de vie du projet. Les principaux outils sont :

- Modèle Logique de Données (MLD) ;

- Modèle Logique des Traitements (MLT).

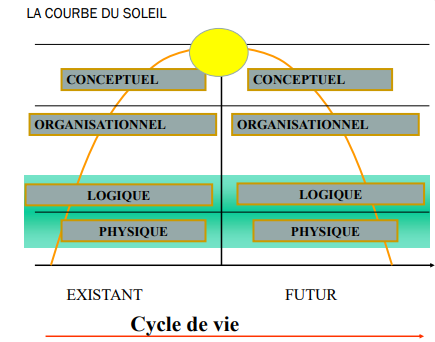


Figure 2 : courbe du soleil

**-Le cycle de décision** : Ce dernier cycle se focalise sur le cadre décisionnel au sein de l'organisation. Il implique l'analyse des processus décisionnels, des acteurs concernés et des informations requises pour une prise de décision éclairée. Ce cycle permet de s'assurer que le système d'information mis en place répond adéquatement aux besoins décisionnels des utilisateurs.

* **Démarche**

Elle s’appuie sur l’imbrication des trois cycles et sur les outils de modélisation préconisés. La démarche d’informatisation est décomposée en étapes. Merise distingue : le schéma directeur, l’étude préalable, l’étude détaillée, la réalisation, la mise en œuvre, la maintenance.

1. **Présentation de la méthode PU/UML**

* **Historique du processus unifié et définition**

Le processus unifié (PU) se définit comme un cadre de développement logiciel itératif, fondé sur l'architecture, piloté par des cas d'utilisation et visant à réduire les risques associés au projet. Ce modèle de processus est suffisamment flexible pour être adapté à une large gamme de systèmes logiciels, indépendamment du domaine d'application, des types d'entreprises, des niveaux de compétences et des tailles d'organisation

* **Les caractéristiques du processus unifié**

Le Processus Unifié se distingue par plusieurs caractéristiques essentielles :

* **Itératif et Incrémental**

Le projet est segmenté en itérations courtes, permettant ainsi un suivi efficace de la progression globale. À la fin de chaque itération, une partie exécutable du système final est livrée de manière incrémentale, par l'ajout de nouvelles fonctionnalités.

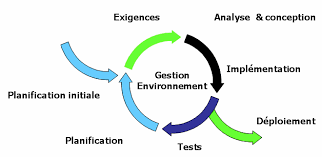


Figure 3 : Déroulement du processus Unifié

* **Centré sur l'Architecture**

Tout système complexe doit être décomposé en modules, facilitant ainsi sa maintenance et son évolution. L'architecture du système qu'elle soit fonctionnelle, logique ou matérielle doit être modélisée en utilisant le langage UML (Unified Modeling Language) plutôt que d'être simplement documentée sous forme textuelle.

Le langage de modélisation unifié (UML) a été développé au début des années 1990 grâce aux contributions de plusieurs ingénieurs en logiciel, notamment

* Grady Booch ;
* Ivar Jacobson ;
* James Rumbaugh.

En octobre 1994, Rational Software Corporation a réuni ces trois méthodes que sont Booch, OMT et OOSE pour créer un langage cohérent, aboutissant ainsi à l'émergence d'UML.

La première version officielle, UML 1.1, a été standardisée par l'Object Management Group (OMG) le 17 novembre 1997, à la suite d'une collaboration impliquant de nombreuses entreprises, dont :

* Hewlett-Packard ;
* IBM, Microsoft ;
* Oracle et Rational Software Corporation.

Depuis sa première version, plusieurs mises à jour ont été publiées. UML 1.3 a été adopté en juin 1999, et UML 1.4 est sorti en novembre 2001, incluant des améliorations basées sur les retours d'expérience des utilisateurs. La version UML 2.0, qui a introduit des changements significatifs dans la modélisation, a été standardisée en mars 2005, et des versions ultérieures, comme UML 2.5, ont continué à affiner le langage.

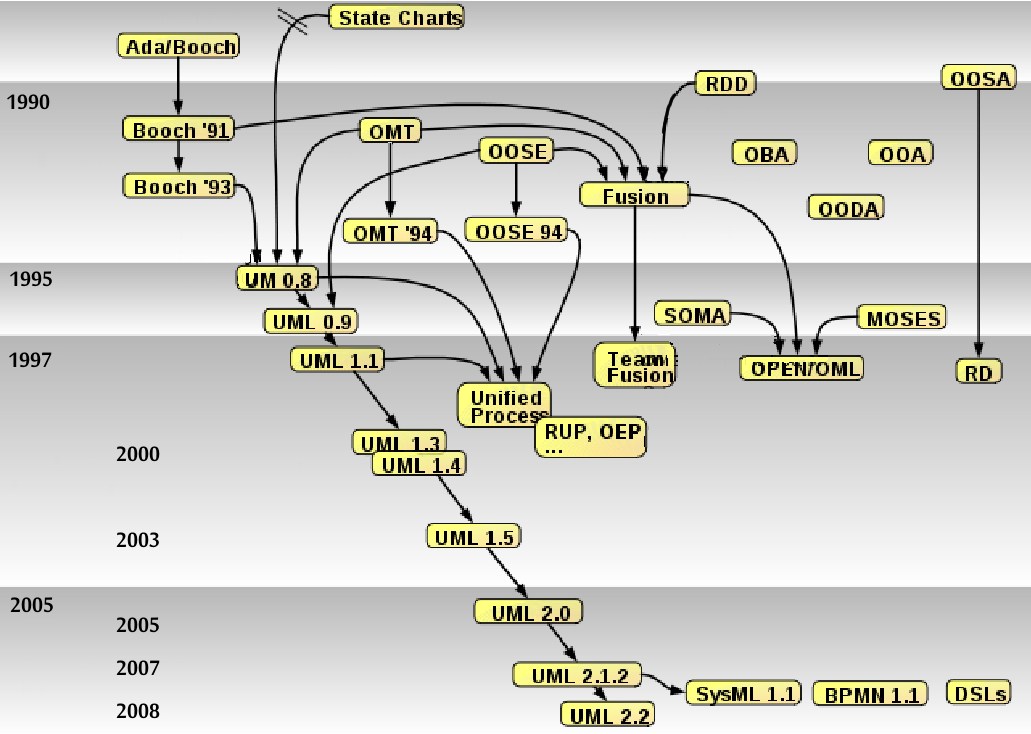


Figure 4 : Evolution d 'UML à travers les versions

L'utilisation d'UML permet non seulement d'obtenir une représentation visuelle claire et précise des composants et de leurs interactions, mais aussi de favoriser la communication entre les différents acteurs du projet. En effet, UML offre une variété de diagrammes, tels que les diagrammes de cas d'utilisation, de séquence ou de classes, qui aident à capturer les exigences fonctionnelles et techniques d'une manière standardisée et compréhensible. Ces diagrammes fournissent une approche systématique pour articuler les besoins des utilisateurs, le flux d'interactions et la structure des données.

En intégrant UML dans la phase de conception architecturale, nous contribuons à une meilleure compréhension du système, nous renforçons la traçabilité des exigences et nous améliorons la collaboration au sein des équipes de développement. La modélisation avec UML favorise également un dialogue constructif entre les développeurs, les analystes et les parties prenantes, permettant ainsi de mieux cerner les attentes et les contraintes du projet.

Cette norme robuste, tout en étant versatile, permet d'adapter l'architecture d'un système aux évolutions technologiques et aux besoins des utilisateurs. En intégrant UML dans le processus de modélisation, nous posons les bases d'une architecture solide et adaptable, capable de répondre aux défis futurs liés à l'évolution rapide du monde numérique. Par conséquent, promouvoir l'adoption d'UML ne constitue pas seulement un choix technique, mais un investissement stratégique dans la qualité et la pérennité du système.

* **Guidé par les Cas d'Utilisation d'UML**

L'objectif principal de tout système informatique est de répondre aux besoins des utilisateurs. Par conséquent, le processus de développement est axé sur l'utilisateur. Les cas d'utilisation, qui illustrent ces besoins, servent non seulement à détecter mais aussi à décrire les exigences fonctionnelles. L'ensemble de ces cas forme un modèle qui définit les fonctionnalités complètes du système.

* **Piloté par les Risques**

L'identification précoce des principaux risques associés au projet est cruciale. Les risques doivent être atténués rapidement, et les actions entreprises à cet égard déterminent l'ordre des itérations à suivre.

* **Cycle de vie du processus unifié (PU)**

Le processus unifié (UP) a pour objectif de maîtriser la complexité des projets informatiques tout en réduisant les risques. Il s'agit d'un cadre méthodologique constitué de principes génériques, qui sont adaptés en fonction des spécificités de chaque projet.

**L’architecture bidirectionnelle** :  
Le processus unifié se structure autour de deux axes principaux (illustré par la figure I.2) :

* **L'axe vertical** : Cet axe représente les principales séquences d'activités, regroupées selon leur nature. Il illustre l'aspect statique du processus, qui se traduit par des composants, des processus, des activités, des enchaînements, des artefacts et des rôles de travail.
* **L'axe horizontal** : Cet axe représente la dimension temporelle du cycle de vie du processus. Il met en lumière l'aspect dynamique du processus, exprimé par des cycles, des phases, des itérations et des jalons.

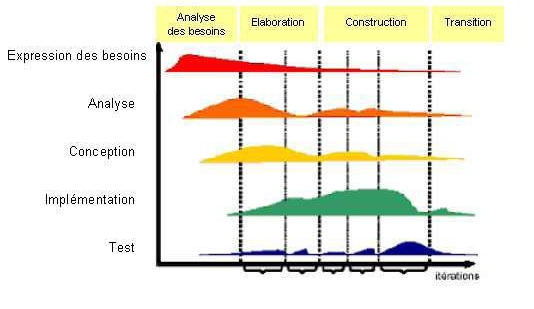


Figure 5 : L’architecture bidirectionnelle

Afin de gérer efficacement ce cycle, les développeurs doivent disposer de diverses représentations du produit logiciel, à savoir :

* Un **modèle de cas d'utilisation** : qui décrit les besoins et interactions avec le système.
* Un **modèle d’analyse** : qui détaille les cas d’utilisation en identifiant les différents acteurs et leurs interactions.
* Un **modèle de conception** : qui définit la structure statique du système sous forme de sous-systèmes, de classes et d’interfaces.
* Un **modèle d’implémentation** : qui intègre les composants logiciels nécessaires à la réalisation du système.
* Un **modèle de déploiement** : qui décrit les nœuds physiques (ordinateurs, serveurs, etc.) impliqués dans le déploiement du système.
* Un **modèle de test** : qui spécifie les cas de test permettant de valider le bon fonctionnement des cas d’utilisation.
* Une **représentation de l’architecture** : qui permet de visualiser l'organisation globale du système, ses composants et leur interaction.

1. COMPARAISON DE PU/UML ET MERISE

Afin de pouvoir choisir une méthode d’analyse parmi les deux décrites plus haut, nous allons procéder à une étude comparative des deux méthodes au biais du tableau suivant :

|  |  |  |
| --- | --- | --- |
| **Critères** | **Merise** | **Processus Unifié (PU) avec UML** |
| **Origine et Contexte** | Méthode d’analyse et de conception développée dans les années 1970 en France pour les systèmes d'information. | Méthode développée dans les années 1990 par Rational Software, centrée sur l'itération et l'incrémentation dans le cadre du développement logiciel. |
| **Modélisation** | Utilise principalement des modèles de données(entité-association) et des modèles de traitements. Les modèles sont séparés selon les différents niveaux d'abstraction (conceptuel, logique, physique). | Utilise UML pour modéliser tous les aspects du système (cas d'utilisation, diagrammes de classes, diagrammes d'activités, etc.), avec des modèles affinés à chaque itération du projet. |
| **Niveaux d’Abstraction** | Plusieurs niveaux d'abstraction : conceptuel, logique, physique. Chaque niveau est décomposé et les transitions entre eux sont clairement définies. | Un seul niveau d’abstraction global, mais avec des types de modèles variés pour décrire différents aspects du système (structure, comportement, interaction). L'affinement des modèles se fait à travers les itérations |
| **Approche de Développement** | Approche linéaire et séquentielle : les phases sont clairement définies (étude préliminaire, conception, réalisation). Chaque phase doit être complétée avant de passer à la suivante. | Un seul niveau d’abstraction global, mais avec des types de modèles variés pour décrire différents aspects du système (structure, comportement, interaction). L'affinement des modèles se fait à travers les itérations. |
| **Gestion des Données** | La méthode Merise met un accent particulier sur la modélisation des données et leur structure à travers le modèle entité-association (MERISE). | Les données sont représentées par des objets et sont encapsulées avec leurs méthodes (principe de l'orienté objet), mettant l’accent sur les relations entre les objets et leur interaction. |
| **Prise en Compte de l’Implémentation** | L’implication des utilisateurs se limite généralement aux phases de recueil des besoins et de validation des modèles. | Le PU inclut des modèles d’implémentation et de déploiement, garantissant une meilleure cohérence entre la phase de conception et celle de l’implémentation. |
| **Gestion des Opérations** | Les opérations (traitements) sont modélisées séparément des données, souvent sous forme de diagrammes de flux de données (DFD). | Les opérations (méthodes) sont intégrées aux objets via des diagrammes de classes, ce qui permet une gestion cohérente et encapsulée des traitements. |

Tableau 1 : Comparaison entre Merise et PU/UML

1. CHOIX DE LA METHODE D’ANALYSE

Après une analyse approfondie des méthodes Merise et PU/UML, nous avons opté pour PU/UML comme méthode d’analyse privilégiée pour notre projet. Cette décision repose sur plusieurs arguments :

* **Approche itérative et flexible** : PU/UML, étant une méthode itérative et incrémentale, offre une grande flexibilité, ce qui est essentiel pour un projet dont les exigences peuvent évoluer au fur et à mesure du développement. Cela permet d'ajuster le système en réponse aux retours des utilisateurs et aux changements de priorités.
* **Adaptation à des environnements agiles** : L’approche du Processus Unifié, combinée avec UML, est particulièrement bien adaptée aux environnements de développement agile, où les cycles de feedback courts et les itérations rapides sont au cœur du processus. Cette méthode permet de livrer des versions du produit à chaque itération, ce qui favorise une amélioration continue du système.
* **Modélisation complète et cohérente** : L’utilisation des diagrammes UML permet de modéliser tous les aspects du système, de la structure à son comportement, en passant par ses interactions. Cette approche globale facilite la compréhension du système par l'ensemble des parties prenantes, de la conception à l'implémentation.
* **Communication et collaboration renforcées** : PU/UML, grâce à sa standardisation, permet une meilleure communication entre les équipes multidisciplinaires. En offrant un langage commun pour la modélisation, UML facilite l'échange d'informations et la collaboration entre les développeurs, analystes, designers et autres parties prenantes du projet.
* **Cohérence avec les pratiques modernes** : Enfin, PU/UML s’aligne avec les pratiques modernes de développement logiciel, telles que la gestion de projet agile et les approches orientées objet. Ce choix nous assure que notre projet bénéficie des meilleures pratiques actuelles pour garantir sa réussite.

CONCLUSION

Après avoir comparé les méthodes Merise et PU/UML, nous avons choisi **PU/UML** comme méthode d’analyse et de conception pour notre projet. En privilégiant **PU/UML**, nous avons opté pour une méthode plus adaptée aux environnements de développement dynamique, favorisant la collaboration et la réactivité tout au long du projet.

CHAPITRE 4 : MODELISATION DE NOTRE APPLICATION

INTRODUCTION

Ce chapitre présente les étapes de conception et de modélisation du système en utilisant UML (Unified Modeling Language). Nous explorerons ainsi les diagrammes de cas d’utilisation, de séquence et de classe qui ont permis de structurer et de formaliser les exigences fonctionnelles du système avant sa mise en œuvre

1. DELIMITATION DU SYSTEME

Dans cette section, nous allons définir les limites du système que nous avons conçu et réalisé, en identifiant les principaux acteurs, les cas d'utilisation associés, et en élaborant un diagramme de cas d’utilisation pour clarifier les interactions entre les utilisateurs et le système. Cette étape est essentielle pour bien cerner les fonctionnalités du système et en définir les contours, afin de garantir que le développement se concentre uniquement sur les besoins identifiés.

1. **Identification des acteurs**

Les **acteurs** représentent les utilisateurs ou systèmes qui interagissent avec le système dans le cadre des différents processus métier. Dans le cadre de notre étude nous avons identifié trois (3) acteurs : Autre Agent, agent Rh, administrateur. Nous considérons ici l’acteur Autre Agent comme un acteur générique encapsulant les rôles partagés par tous les autres acteurs

Le tableau ci-dessous fourni les explications plus détaillées sur les acteurs que nous avons identifiées (appelé le tableau identification des acteurs)

|  |  |
| --- | --- |
| Acteurs | Rôles |
| **Autre Agent** | C’est un utilisateur << classique>> du système. Il possède le privilège partagé par les autres acteurs. |
| **Agent RH** | Responsable de la gestion des ressources humaines, le suivi des affectations, et la mise à jour des informations des employés. Il a un accès plus étendu que l'Autre agent, notamment aux données de tous les Agents. |
| **Administrateur** | Il est responsable de l'administration des rôles et permissions. Il peut effectuer toutes les actions possibles dans le système, y compris celles des autres acteurs. |

Tableau 2: Identification des acteurs

1. **Identification des cas d’utilisation**

Les cas d'utilisation représentent les scénarios qui décrivent les interactions entre les acteurs et le système. Ces cas sont essentiels pour formaliser les besoins fonctionnels du système et s’assurer que toutes les fonctionnalités nécessaires seront prises en compte.

Le tableau ci-dessous répertorie les différentes façons dont les acteurs précédemment identifiés utilisent le système :

|  |  |
| --- | --- |
| Acteur | Cas d’utilisation |
| Autre Agent | -S’authentifier  -Modifier le mot de passe  -Modifier la photo de profil  -Se déconnecter  -Gérer les absences |
| Agent RH | -Editer des documents  -Suivi du personnel  -Suivi des affectations |
| Administrateur | **-Gestion des rôles et permissions** : Permet à l'administrateur de gérer les rôles des utilisateurs, d'attribuer des permissions et de définir des accès spécifiques. |

Tableau 3: Identification des cas d’utilisation

1. **Elaboration du diagramme de cas d’utilisation**

Le diagramme de cas d’utilisation permet de représenter graphiquement les interactions entre les utilisateurs du système et les fonctionnalités proposées. Il décrit les différents scénarios d'utilisation du système, du point de vue des acteurs mais ne montre pas comment le système fonctionnent en interne.

La figure ci-après présente le **diagramme de cas d'utilisation global**, qui englobe plusieurs cas d'utilisation spécifiques, organisés en sous-packages. Chaque sous-package, tel que **Gestion compte utilisateur,** **Gestion Absences, Suivi des affectations, gestion du personnel** est détaillé dans des diagrammes séparés pour illustrer les interactions précises entre les acteurs et le système.

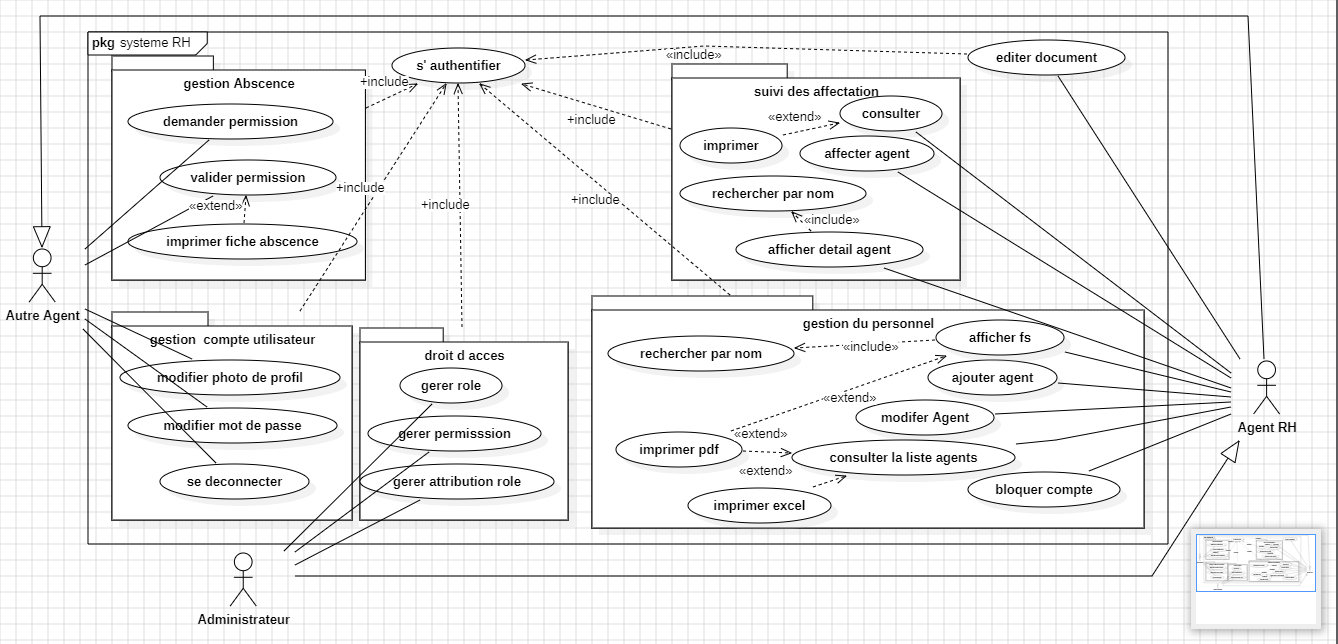


Figure 6:Diagramme de cas d’utilisation global

Dans ce diagramme global, **l'authentification** est incluse dans tous les cas d'utilisation nécessitant un accès sécurisé, tels qu’Éditer **document**, **Gérer les comptes utilisateur**, **gestion du personnel** ou **droit d’accès.** Cette relation **include** assure que ces actions ne peuvent être effectuées que par un utilisateur authentifié.

* **Gestion compte utilisateur**

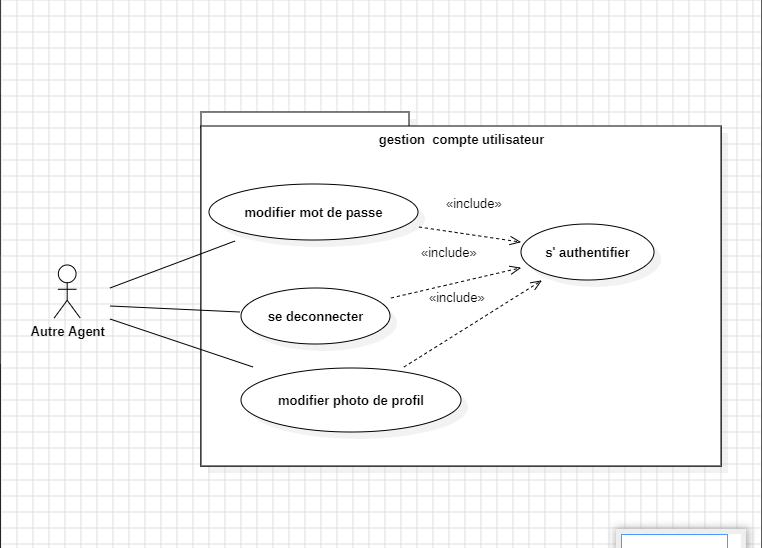


Figure 7:Diagramme de cas d’utilisation gestion compte utilisateur

* **Gérer les absences** : Le diagramme suivant montre le cas d’utilisation détaillé où un Autre Agent soumet une demande de permission, et un agent RH peut valider cette demande.

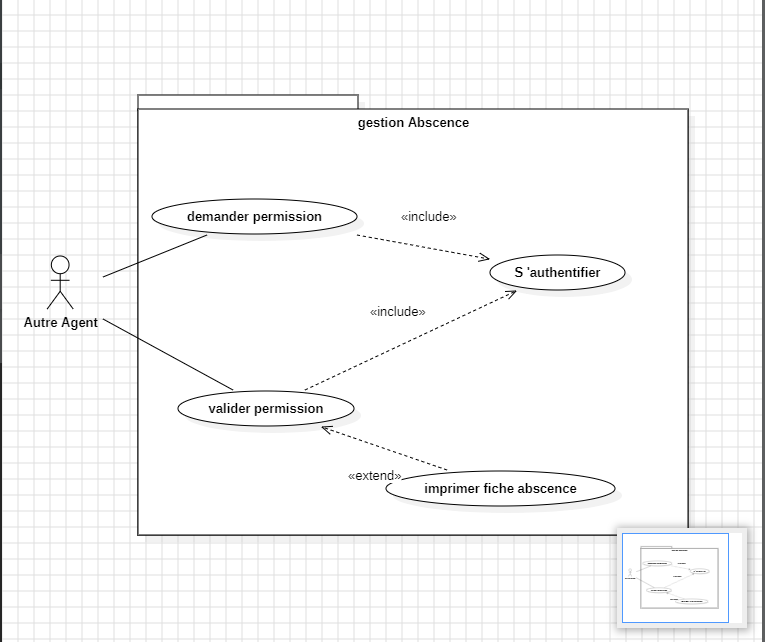


Figure 8:Diagramme de cas d’utilisation gérer Absence

* **Gestion du personnel** : Le diagramme suivant illustre les actions de l'agent RH pour ajouter, modifier et consulter les informations du personnel.

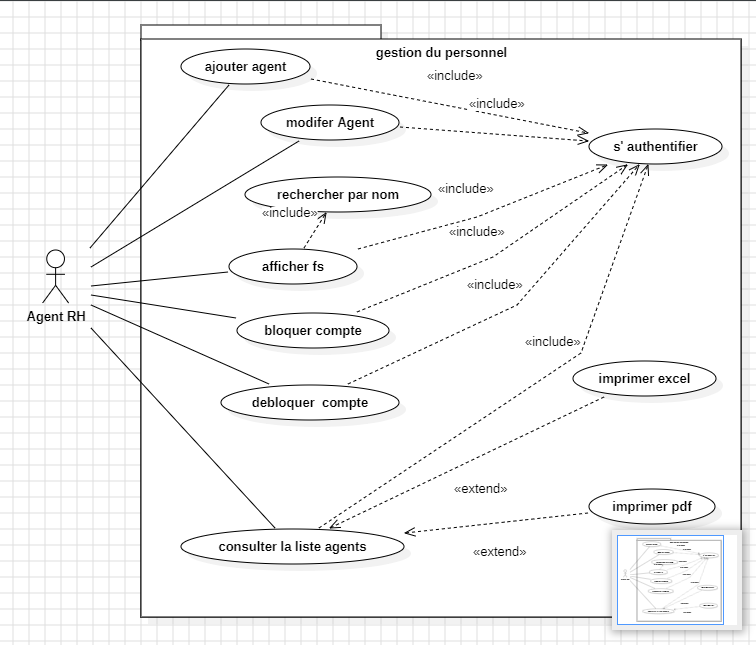


Figure 9:Diagramme de cas d’utilisation gestion du personnel

* **Droit d’accès**

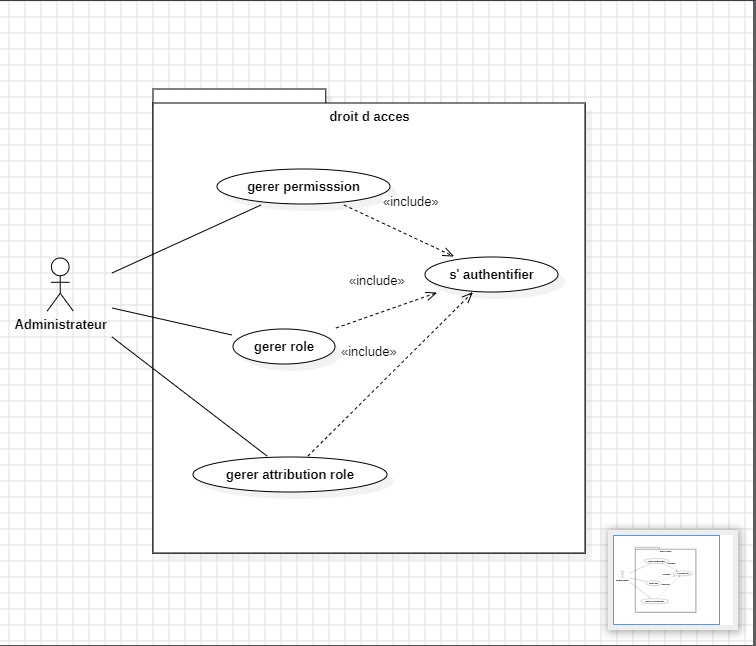


Figure 10:Diagramme de cas d’utilisation droit d’accès

* **Suivi affectation**

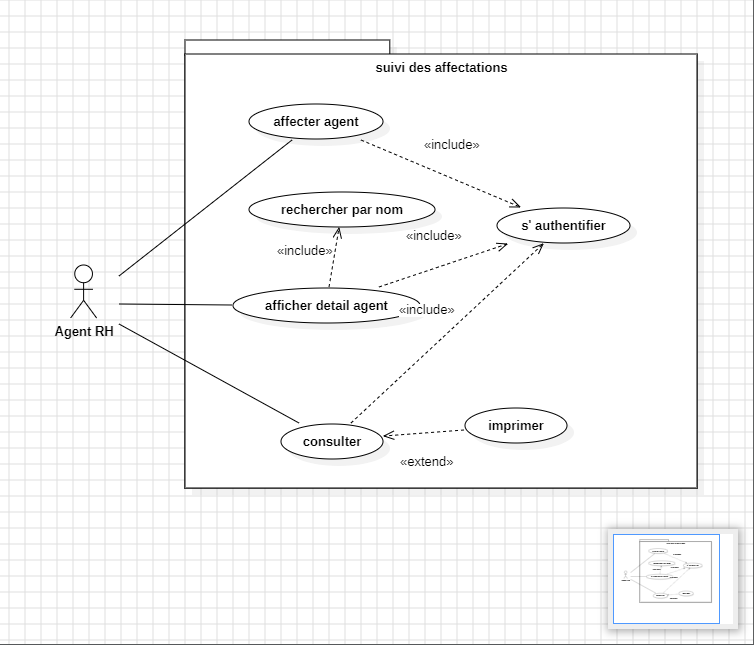


Figure 11:Diagramme de cas d’utilisation suivi affectation

1. ANALYE
2. **Description textuelle**

En complément des diagrammes de cas d'utilisation, chaque fonctionnalité du système a été détaillée à travers des descriptions textuelles. Ces descriptions visent à fournir une compréhension approfondie des actions et interactions entre les acteurs et le système.

Cas 1 : S’authentifier

|  |  |
| --- | --- |
| **Titre** | S’authentifier |
| **Auteur principal** | Autre Agent |
| **Résumé** | L'objectif est de permettre à l'utilisateur de se connecter au système en fournissant ses identifiants valides. Le système vérifie les identifiants et accorde ou refuse l'accès. |
| **Préconditions** | - L’Autre Agent doit avoir un compte déjà créé.  - L’Autre Agent doit avoir accès à Internet*.* |
| **Scénario principal** | **1. L’Autre** Agent **accède à la page de connexion  2. L’Autre** Agent **saisit ses identifiants  3. Le système vérifie les identifiants**  4. l’utilisateur est redirigé vers la page d'accueil. |
| **Exceptions** | - **E1** : **Identifiants incorrects** : le système affiche un message d’erreur et reste sur la page de de connexion |

Tableau 4 : Description textuelle du cas s ' authentifier

Cas 2 : Modifier mot de passe

|  |  |
| --- | --- |
| **Titre** | Modifier mot de passe |
| **Auteur principal** | Autre Agent |
| **Résumé** | L’Autre Agent modifie son mot de passe pour renforcer la sécurité de son compte. |
| **Précondition** | - L’Autre Agent doit être authentifié |
| **Scénario principal** | 1. L’Autre Agent clique sur sa photo de profil.  2. Un menu déroulant apparaît, et l’Autre Agent clique sur **"Voir profil"**.  3. Le système redirige l’Autre Agent vers la page de gestion de profil.  4. L'employé clique sur **"Modifier mot de passe"**.  5. L ’Autre Agent entre son mot de passe actuel, son nouveau mot de passe et confirme le nouveau mot de passe.  6. Le système vérifie la validité des informations.  7. le mot de passe est modifié et il redirigé vers page de connexion. |
| **Postconditions** | - Si la modification est réussie, le mot de passe est mis à jour et l'employé est redirigé au formulaire de connexion  - Si l'échec se produit, l'employé reste sur la page gestion profil. |
| **Exceptions** | **-E1 : Mot de passe actuel incorrect.  -E2 : Le nouveau mot de passe ne correspond pas à la confirmation.** |

Tableau 5:Description textuelle du cas Modifier mot de passe

Cas 3 : Modifier photo de profil

|  |  |
| --- | --- |
| **Titre** | Modifier photo de profil |
| **Acteur principal** | Autre Agent |
| **Résumé** | Permet à l’Autre Agent de modifier sa photo de profil pour personnaliser son compte. |
| **Précondition** | Être authentifié |
| **Scenario nominal** | 1. L’Autre Agent clique sur sa photo de profil.  2. Un menu déroulant apparaît, et l'employé  Clique sure **« Voir profil ».**  3. Le système redirige l’Autre Agent vers la page de gestion de profil.  4. L’Autre Agent clique sur **"Modifier photo de profil".**  5. L'employé choisit une nouvelle photo et clique sur **Envoyer**  6. Le système vérifie que la photo est envoyée dans le format spécifié.  7. La photo de profil est modifiée |
| **Postcondition** | La photo de profil est mise à jour et un message de succès est affiché. |
| **Scenario d’exception** | -E1 : photo au mauvais format.  -**E2** : Problème de connexion pendant le téléchargement. |

Tableau 6:Description textuelle du cas Modifier photo de profil

Cas 4 : Se déconnecter

|  |  |
| --- | --- |
| **Titre** | Se déconnecter |
| **Acteur principal** | Autre Agent |
| **Résumé** | Permet à l’Autre Agent de se déconnecter du système une fois qu'il a terminé ses tâches. |
| **Préconditions** | - L’Autre Agent doit être authentifié. |
| **Scenario nominal** | 1. L’Autre Agent clique sur le bouton **"Se déconnecter"**.  2. Le système termine la session et redirige l'employé vers la page de connexion. |
| **Postcondition** | L’Autre Agent est déconnecté et ne peut plus accéder aux fonctionnalités réservées aux utilisateurs authentifiés tant qu’il ne se reconnecte pas nouveau. |

Tableau 7:Description textuelle du cas Se déconnecter

Cas 5 : Editer document

|  |  |
| --- | --- |
| **Titre** | Editer document |
| **Acteur principal** | Agent RH |
| **Préconditions** | -L'agent RH doit être authentifié.  -L'agent RH doit avoir accès à l'onglet "**Éditer documents"**.  - L'agent RH doit avoir les autorisations nécessaires pour générer ces documents. |
| **Scenario principal** | 1. L'agent RH se connecte et accède à l'onglet **"Éditer document"**.  2. L'agent RH choisit le type de document à éditer (attestation de travail, attestation de prise de service, etc.).  3. Le système affiche une liste des employés disponibles.  4. L'agent RH sélectionne un employé parmi la liste.  5.L'agent RH clique sur **"Imprimer"** pour générer le document qu’il aura choisi.  6. Le système génère le document au format PDF avec les informations existantes de l'employé.  7. L'agent RH peut télécharger ou imprimer le document PDF. |
| **Postcondition** | Le document choisi est généré et imprimé ou téléchargé en format PDF avec les informations de l'employé. |
| **Scenario alternative** | |  | | --- | | -A1 : Employé non sélectionné : Si l'agent RH oublie de sélectionner un employé, un message d'erreur apparaît pour l'inviter à choisir un employé dans la liste. |  |  |  | | --- | --- | |  | - A2 : Type de document non sélectionné : Si l'agent RH ne sélectionne pas de type de document à éditer, un message d'erreur apparaît pour l'inviter à choisir un document. |  |  |  | | --- | --- | |  | - A3 : Erreur de connexion à la base de données : Si une erreur de connexion à la base de données se produit lors de la récupération des informations de l'employé, le système affiche un message d'erreur indiquant qu'une erreur s'est produite et invite l'agent RH à réessayer. | |
| **Exceptions** | - **E1 : Aucune donnée pour l'employé** : Si l'employé sélectionné ne dispose pas des données nécessaires pour générer le document, un message d'erreur est affiché.  - **E2 : Erreur de génération PDF** : Si le document ne peut pas être généré en raison d'une erreur technique (ex. problème de format, données manquantes), le système affiche un message d'erreur. |

Tableau 8:Description textuelle du cas Editer document

Cas 6 : Consulter la liste des agents affectés

|  |  |
| --- | --- |
| **Titre** | Consulter la liste des agents affectés |
| **Acteur principal** | Agent Rh |
| **Résumé** | Permet à l'Agent RH de consulter la liste  des agents affectés au sein de l’organisation |
| **Précondition** | - être authentifié.  - L'Agent RH doit avoir accès à l'onglet **"Affectations"**. |
| **Scenario nominal** | 1. L'Agent RH accède à l'onglet **"Affectations internes"**.  2. Le système affiche la liste complète des affectations des employés. |
| **Scenario alternatif** | - **A1 : Liste vide** : Si aucun agent n'est affecté à une fonction ou un lieu de travail, un message "Aucune affectation en cours" s'affiche.  **A2 : Imprimer ou exporter la liste** : Si l'Agent RH souhaite imprimer ou exporter la liste des affectations (PDF ou Excel), il peut le faire après avoir consulté la liste. |
| **Exception** | - **E1 : Erreur technique** : Si une erreur technique survient lors de la récupération des données, un message d'erreur est affiché. |

Tableau 9:Description textuelle du cas Consulter la liste des agents affectés

Cas 7 : Afficher détail agent

|  |  |
| --- | --- |
| **Titre** | Afficher détail agent |
| **Acteur principal** | Agent RH |
| **Résumé** | L’objectif est de permettre à l'Agent RH de consulter les détails d'un agent spécifique, y compris son affectation, ses fonctions passées, et autres informations pertinentes. |
| **Précondition** | -L'Agent RH doit être authentifié. - L'Agent RH doit avoir accès à la page de la liste des affectations ou effectué une recherche d'agent. |
| **Scenario nominal** | 1. L'Agent RH se connecte et accède à la page **"Suivi des affectations"**.  2. L'Agent RH effectue une recherche en entrant le nom de l'agent dans le champ de recherche. 3. Le système filtre la liste des affectations pour afficher les agents correspondant au nom recherché.  4. L'Agent RH clique sur l'icône "Information" à côté de l'agent souhaité pour afficher ses détails.  5. Le système affiche les informations détaillées de l'agent sélectionné (fonction actuelle, historique des affectations, etc.). |
| **Scenario alternatifs** | - **A1 : Aucun agent trouvé** : Si l'Agent RH effectue une recherche et qu'aucun agent ne correspond, un message "Aucun agent trouvé" est affiché.  - **A2 : Liste vide** : Si la liste des affectations est vide, un message "Aucune affectation en cours" s'affiche. |
| **Postconditions** | Les détails de l'agent sont affichés à l'Agent RH |

Tableau 10: Description textuelle du cas Afficher détail agent

Cas 8 : affecter un agent

|  |  |
| --- | --- |
| **Titre** | Affecter un agent |
| **Auteur principal** | Agent RH |
| **Résumé** | Permet à l'Agent RH d'affecter un agent à une nouvelle fonction ou un nouveau lieu de travail dans l'organisation. Le système enregistre cette affectation et effectue des vérifications avant l'enregistrement. |
| **Précondition** | - L'Agent RH doit être authentifié. - L'Agent RH doit avoir accès à la page **"Suivi des affectations"**.  - L'Agent RH doit avoir la liste des agents, fonctions, et lieux de travail disponibles. |
| **Scenario nominal** | 1. L'Agent RH sélectionne un agent dans la liste des affectations.  2. L'Agent RH choisit une nouvelle fonction et/ou un nouveau lieu de travail pour l'agent.  3. L'Agent RH saisit la date de début de l'affectation. 4. L'Agent RH clique sur "Affecter" pour enregistrer la nouvelle affectation. 5. Le système vérifie si un changement a bien été effectué (fonction ou lieu de travail modifié). 6. Le système enregistre l'affectation dans la base de données et met à jour la liste des affectations. |
| **Postcondition** | L'affectation de l'agent est mise à jour avec la nouvelle fonction ou le nouveau lieu de travail, et la liste des affectations est mise à jour. |
| **Exception** | **E1 : Affectation non effectuée** : Si l'agent n'a pas été affecté à une nouvelle fonction ou un nouveau lieu de travail, le système affiche un message d'erreur : "Aucune modification apportée à l'affectation de cet agent." - **E2 : Problème de validation de la date** : Si la date de début de l'affectation est dans le passé (par exemple avant la date actuelle), le système affiche un message : "La date choisie est incorrecte." |

Tableau 11:Description textuelle du cas affecter un agent

Cas **9** : Ajouter un agent

|  |  |
| --- | --- |
| **Titre** | Ajouter un agent |
| **Acteur** | Agent RH |
| **Résumé** | Permet à l'Agent RH d'ajouter un nouvel agent dans le système en saisissant les informations nécessaires. |
| **Précondition** | -Être authentifier  -L'Agent RH doit avoir les autorisations nécessaires. |
| **Scenario nominal** | 1. L'Agent RH se connecte au système.  2. L'Agent RH accède à la section **"Ajouter un agent"**.  3. L'Agent RH remplit les informations nécessaires sur l'agent (nom, prénom, fonction, lieu de travail, etc.).  4. L'Agent RH clique sur **"Ajouter agent "** pour enregistrer l'agent dans le système.  5. Le système vérifie la validité des informations et ajoute l'agent. |
| **Postcondition** | - L'agent est ajouté au système avec un identifiant unique.  - Les informations de l'agent sont enregistrées et visibles dans le système. |
| **Scenarios alternatifs** | - A1 : Informations manquantes : Si des informations sont manquantes, un message d'erreur est affiché pour inviter l'utilisateur à compléter le formulaire.  - A2 : Doublon détecté : Si un agent avec les mêmes informations existe déjà, un message d'erreur est affiché. |
| **Exception** | **- E1 : Erreur de validation** : Si un problème technique survient lors de l'ajout de l'agent (ex. erreur de base de données), un message d'erreur est affiché. |

Tableau 12:Description textuelle du cas Ajouter un agent

Cas 10 : bloquer compte d’un agent

|  |  |
| --- | --- |
| **Titre** | Bloquer compte d’un agent |
| **Résumé** | Permet à l'Agent RH de bloquer l'accès au système d'un agent pour diverses raisons (licenciement, suspension, etc.). |
| **Précondition** | - L'Agent RH doit être authentifié.  - L'Agent RH doit avoir les autorisations nécessaires pour bloquer un agent. |
| **Scenario nominal** | 1. L'Agent RH accède à la liste des agents.  2. L'Agent RH sélectionne l'agent à bloquer.  3. L'Agent RH clique sur "Bloquer le compte" pour suspendre l'accès de l'agent.  5. Le système met à jour le statut de l'agent et bloque l'accès au compte. |
| **Postcondition** | - Le compte de l'agent est bloqué.  - L'agent ne pourra plus se connecter ou accéder au système tant que son compte est bloqué. |
| **Scenario alternatives** | - A1 : Agent déjà bloqué : Si l'agent est déjà bloqué, un message d'erreur est affiché pour indiquer que l'agent est déjà dans l'état "bloqué". |
| **Exception** | **- E1 : Erreur technique** : Si une erreur survient lors du blocage du compte (ex. problème avec la base de données ou mise à jour du statut), un message d'erreur est affiché. |

Tableau 13: Description textuelle du cas bloquer compte d’un agent

Cas 11 : modifier agent

|  |  |
| --- | --- |
| **Titre** | Modifier un agent |
| **Acteur** | Agent RH |
| **Résumé** | Permet à l'Agent RH de modifier les informations d'un agent existant dans le système |
| **Préconditions** | Être authentifier |
| **Scenario nominal** | 1. L'Agent RH ou l'Administrateur se connecte au système.  2. L'Agent RH ou l'Administrateur accède à la liste des agents.  3. L'Agent RH ou l'Administrateur sélectionne l'agent à modifier.  4. L'Agent RH ou l'Administrateur modifie les informations nécessaires (fonction, lieu de travail, etc.).  5. L'Agent RH ou l'Administrateur clique sur "Sauvegarder" pour enregistrer les modifications.  6. Le système valide les modifications et met à jour les informations de l'agent. |
| **Postcondition** | - Les informations de l'agent sont mises à jour dans le système.  - Les modifications sont visibles dans la liste des agents. |
| **Exception** | **- E1 : Informations incorrectes ou manquantes** : Si les informations saisies sont incorrectes ou manquantes, un message d'erreur est affiché pour inviter l'utilisateur à corriger |

Tableau 14: Description textuelle du cas modifier agent

Cas 12 : gérer rôle

|  |  |
| --- | --- |
| **Titre** | Gérer rôle |
| **Acteur** | Administrateur |
| **Résume** | L’administrateur ajoute, modifie, et supprime des rôles dans le système afin de gérer les accès et le permissions des utilisateurs |
| **Préconditions** | Être authentifié |
| **Scenario nominal** | 1.L’administrateur accède à la section "Gestion des rôles".  2. L’administrateur choisit "Ajouter un rôle" pour créer un nouveau rôle.  4. L’administrateur définit les permissions du rôle.  5. Le système crée le rôle et l’associe aux permissions choisies. |
| **Postcondition** | - Le rôle est créé ou modifié et les utilisateurs ont désormais les permissions associées à ce rôle.  - Si un rôle est supprimé, il est retiré du système |
| **Scenarios alternatif** | -A1 : Modification d’un rôle : si l’administrateur veut modifier un rôle existant, il peut ajuster les permissions associées  -A2 : suppression d’un rôle : si l’administrateur veut supprimer un rôle, un message de confirmation s affiche pour éviter une suppression accidentelle |
| **Scenario d’exception** | -**E2 : Rôle déjà crée** : si un rôle est déjà créé |

Tableau 15: Description textuelle du cas gérer rôle

.

Cas 13 : gérer permission

|  |  |
| --- | --- |
| **Titre** | Gérer permission |
| **Acteur principal** | Administrateur |
| **Résumé** | Permet à l’administrateur de définir, modifier ou supprimer des permissions qui contrôlent l’accès à certaines fonctionnalités du système. |
| **Préconditions** | -être authentifier |
| **Scenario nominal** | 1. L’administrateur se connecte au système. 2. L’administrateur accède à la section "**Gestion des permissions".**  3. L’administrateur choisit **"Ajouter une permission".** 4. L’administrateur définit les actions et les rôles utilisateurs autorisés à cette permission. 5. Le système enregistre et applique la nouvelle permission. |
| **Postconditions** | La permission est ajoutée ou mise à jour dans le système. - Les utilisateurs ayant cette permission peuvent désormais accéder aux fonctionnalités associées. |
| **Scenario alternative** | A2 : Modification d’une permission : Si l’administrateur souhaite modifier une permission existante, il peut ajuster ses paramètres et l'affectation des rôles. -A3 : suppression d’une permission : L’administrateur peut supprimer une permission après une confirmation |
| **Scenario d’exception** | **E2 : conflit permission :** Si la suppression ou modification d'une permission crée un conflit avec d'autres permissions ou rôles, un message d’erreur est affiché. |

Tableau 16:Description textuelle du cas gérer permission

Cas 14 : gérer attribution rôle

|  |  |
| --- | --- |
| **Titre** | Gérer attribution rôle |
| **Acteur principal** | Administrateur |
| **Résumé** | L’administrateur attribue des rôles spécifiques aux utilisateurs du système afin de définir leurs droits d'accès et leurs responsabilités. |
| **Précondition** | Être authentifié  Avoir pour rôle l’administrateur |
| **Scenario nominal** | 1.L’administrateur se connecte au système et accède à la section "Gestion des rôles".  2. L’administrateur sélectionne un utilisateur pour lui attribuer un rôle.  3. L’administrateur choisit un rôle existant ou crée un nouveau rôle pour l’attribuer à l’utilisateur.  4. Le système applique le rôle à l’utilisateur sélectionné. |
| **Postcondition** | L’utilisateur dispose désormais du rôle et des permissions associées à ce rôle. |
| **Scenarios d alternatifs** | A1 : Modification de l’attribution d’un rôle : Si l’administrateur souhaite changer le rôle d’un utilisateur, il peut le faire en sélectionnant un autre rôle pour l’utilisateur.  A2 : Suppression d’un rôle attribué : Si l’administrateur souhaite retirer un rôle à un utilisateur, une confirmation est demandée avant de procéder. |
| **Scenario d’exception** | **E1 : Rôle non attribué** : Si l’administrateur essaie d’attribuer un rôle sans permissions suffisantes, un message d’erreur est affiché. |

Cas 15 : demander permission

|  |  |
| --- | --- |
| **Titre** | Demander permission |
| **Acteur principal** | Autre Agent |
| **Résumé** | L’Autre agent soumet une demande de permission pour obtenir l’approbation de son supérieur hiérarchique. La demande peut concerner différents types de congé, comme des congés payés, des congés maladie, etc. |
| **Précondition** | L’employé doit être authentifié dans le système. |
| **Scenario nominal** | 1.L’Autre agent s accède à la section "Demande de permission".  2. L’Autre agent sélectionne le motif d’absence (par exemple, congé payé, congé maladie, etc.) et définit les dates de début et de fin de son congé ou ajoute un nouveau motif si son motif ne figure pas dans liste.  3. L’Autre agent peut ajouter une justification si nécessaire.  4. L’Autre agent soumet la demande de congé pour approbation.  5. Le système envoie la demande de congé au supérieur hiérarchique (qui est aussi un autre agent) pour validation. |
| **Postcondition** | La demande de congé est soumise et attend une validation de la part du supérieur hiérarchique. |
| **Scenario alternatif** | A1 : Modification de la demande : Si l’autre agent souhaite modifier sa demande avant qu’elle ne soit validée, il peut ajuster les dates ou le type de congé. |
| **Scenario d’exception** | **E2 : Demande incomplète :** Si des informations obligatoires sont manquantes, un message d'erreur apparaît pour rappeler à l'employé de compléter la demande. |

Figure 12:Description textuelle du cas demander permission

Cas 16 : valider permission

|  |  |
| --- | --- |
| **Titre** | Valider la permission |
| **Acteur principal** | Supérieur hiérarchique (également un Autre agent) |
| **Résumé** | Le supérieur hiérarchique, qui est aussi un autre agent, examine et approuve ou rejette la demande de congé soumise par un autre employé. |
| **Scenario nominal** | Le supérieur hiérarchique (qui est aussi un autre agent) se connecte au système et accède à la section **"Demandes de congé en attente".**  Le supérieur hiérarchique consulte la demande de congé de l'employé, y compris les dates et le type de congé demandé.  Le supérieur hiérarchique vérifie la disponibilité de l'employé (par exemple, si la période demandée ne chevauche pas d’autres congés etc).  Si la demande est acceptable, le supérieur hiérarchique valide la demande.  Le système notifie l’employé que sa demande a été validée.  Si le supérieur hiérarchique rejette la demande, il saisit une justification et le système notifie l’employé que sa demande a été rejetée. |
| **Post condition** | Si la demande est validée, l’employé peut prendre son congé comme prévu.  Si la demande est rejetée, l’employé est informé et la demande est clôturée. |
| **Scenario alternatif** | A1 : Modification de la demande : Si le supérieur hiérarchique souhaite modifier les dates ou le type de congé avant de le valider (par exemple, en raison d'une situation opérationnelle), il peut ajuster la demande avant de la confirmer. |
| **Scenario d’exception** | **E1 : Conflit de congé** : Si la demande de congé entre en conflit avec d’autres demandes ou avec les besoins de l’entreprise (par exemple, un nombre trop élevé de congés durant une période critique), le supérieur hiérarchique peut rejeter la demande.  **E2 : Absence de droits de validation** : Si le supérieur hiérarchique n’a pas les droits nécessaires pour valider une demande de congé, un message d'erreur est affiché. |

Figure 13:Description textuelle du cas valider permission

1. **Elaboration du Diagramme de séquence**

Le diagramme de séquence est une représentation de façon séquentielle du déroulement des traitements et des interactions entre le système et/ou de ses acteurs

Cas 1 : S’authentifier

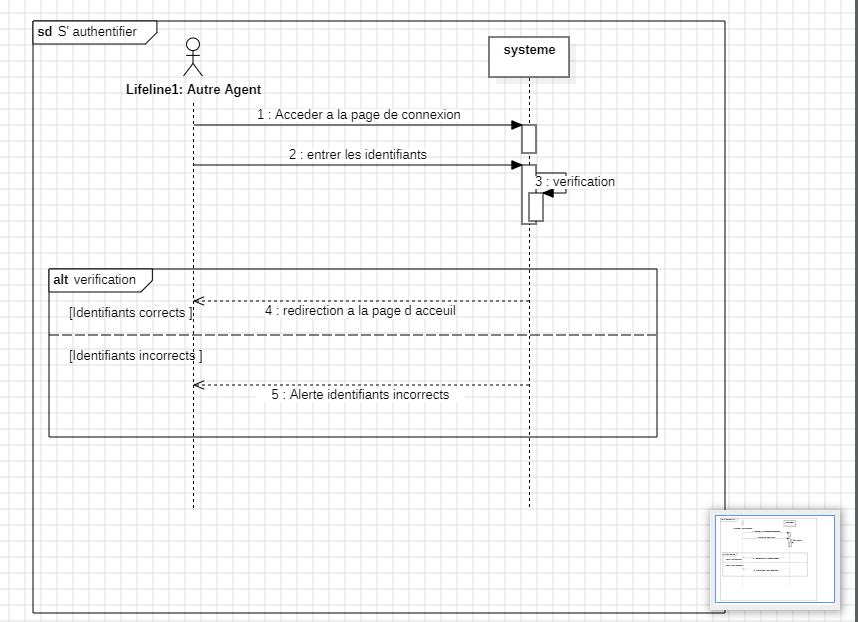


Figure 14:Diagramme de séquence du cas S’authentifier

Cas 2 : Modifier mot de passe

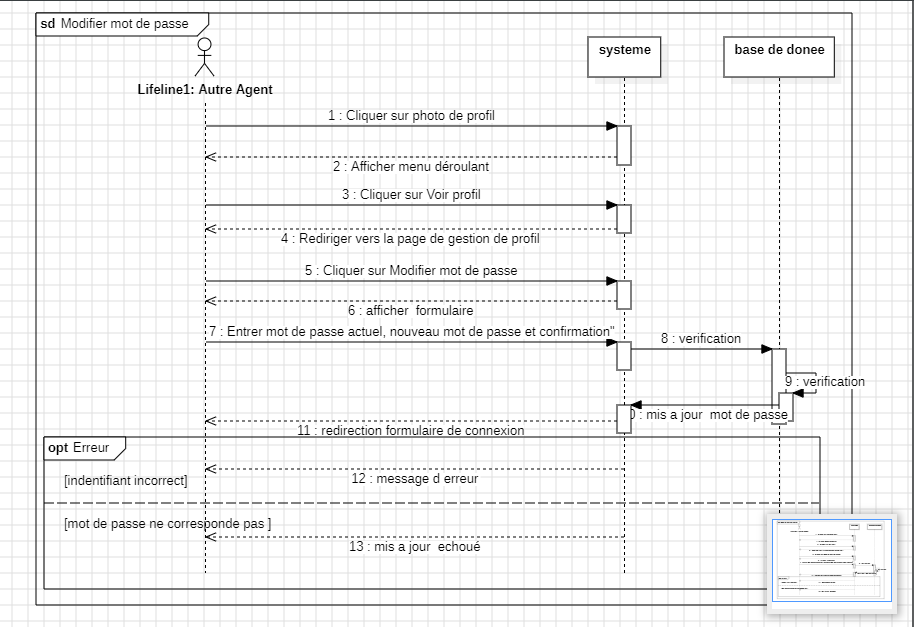


Figure 15:Diagramme de séquence du cas Modifier mot de passe

Cas 3 : Modifier photo de profil

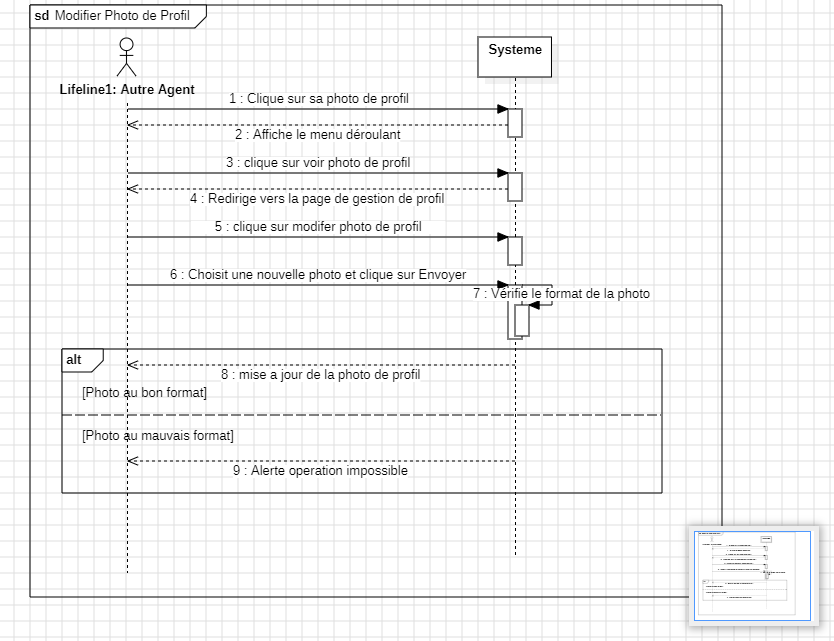


Figure 16:Diagramme de séquence du cas Modifier photo de profil

Cas 4 : Se déconnecter

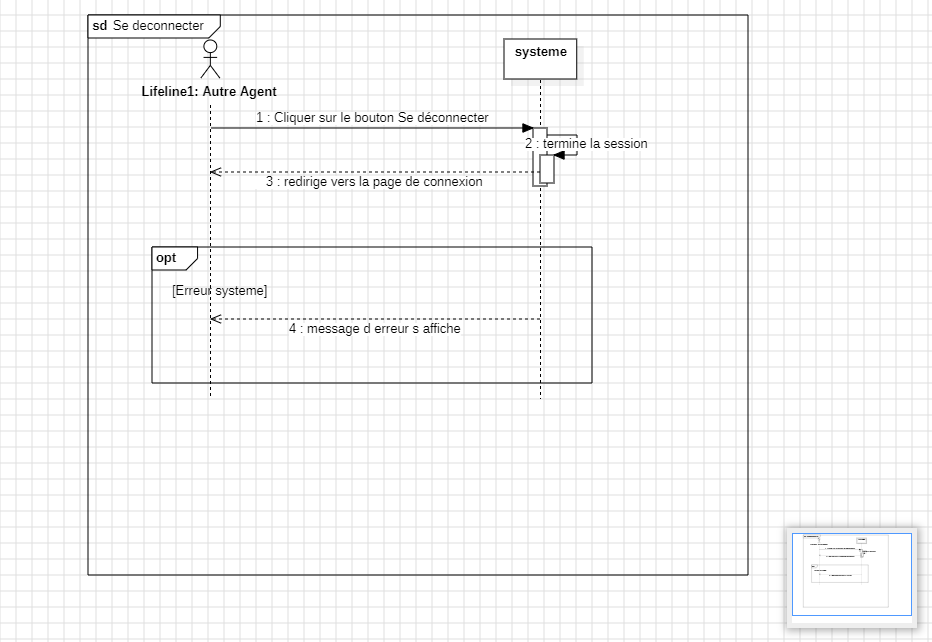


Figure 17:Diagramme de séquence du cas se déconnecter

Cas 5 : Editer document

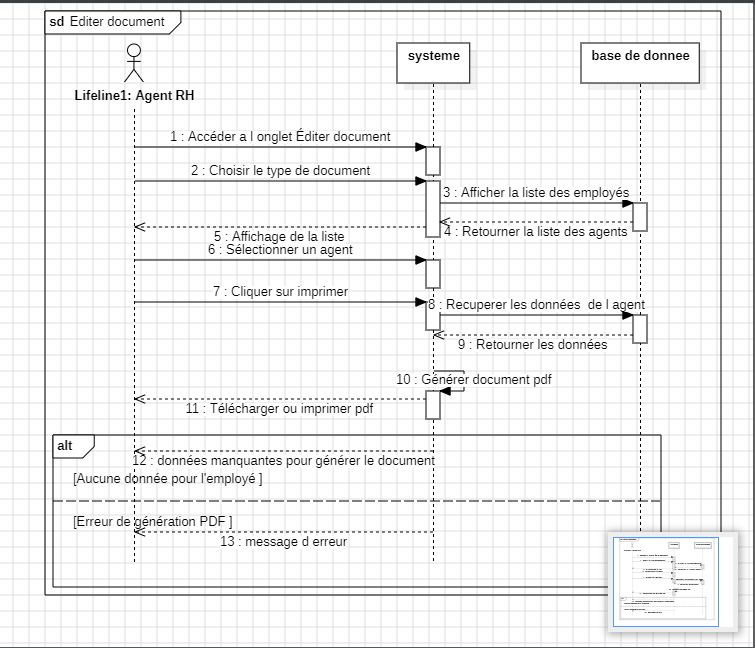


Figure 18:Diagramme de séquence du cas Editer document

Cas 6 : Consulter la liste des agents affectés

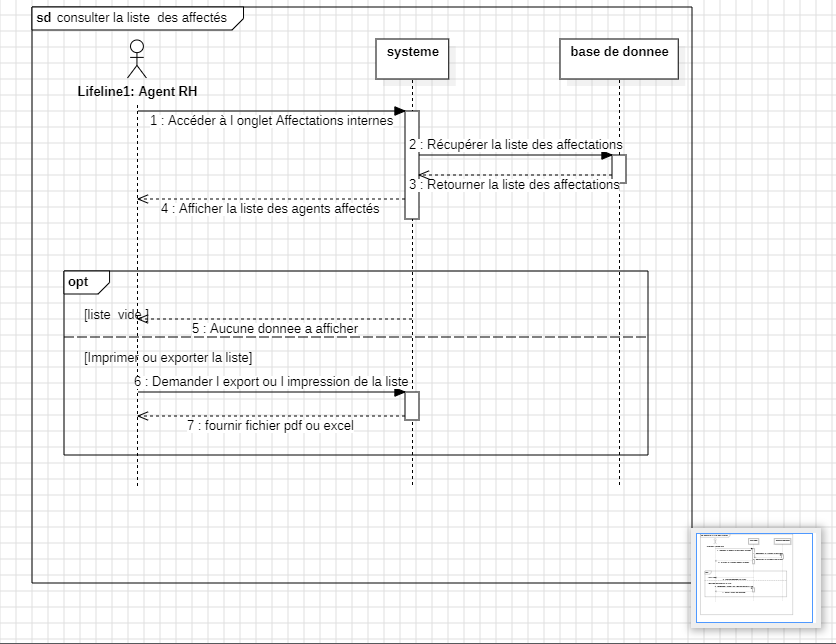


Figure 19:Diagramme de séquence du cas Consulter la liste des agents affectés

Cas 7 : Afficher détail agent

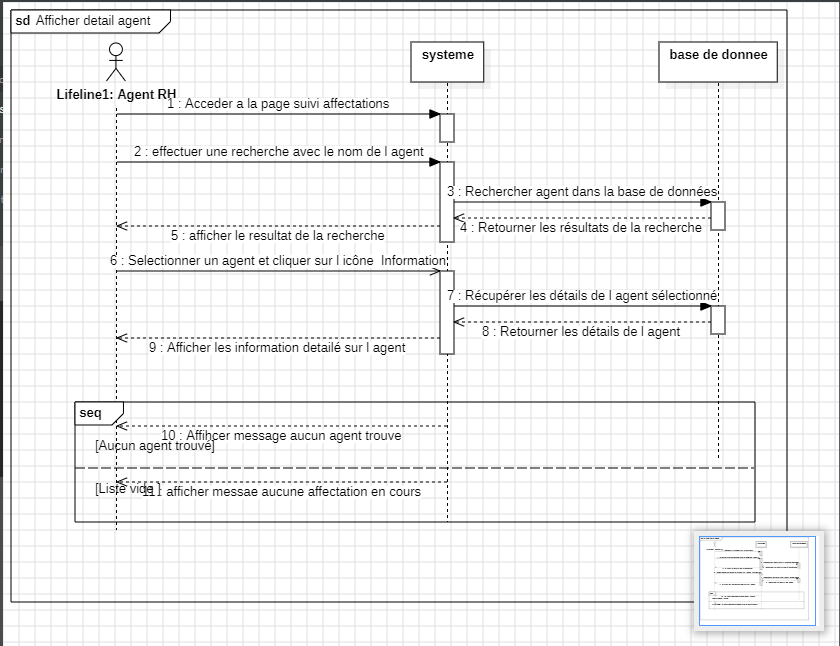


Figure 20:Diagramme de séquence du cas Afficher détail agent

Cas 8 : affecter un agent

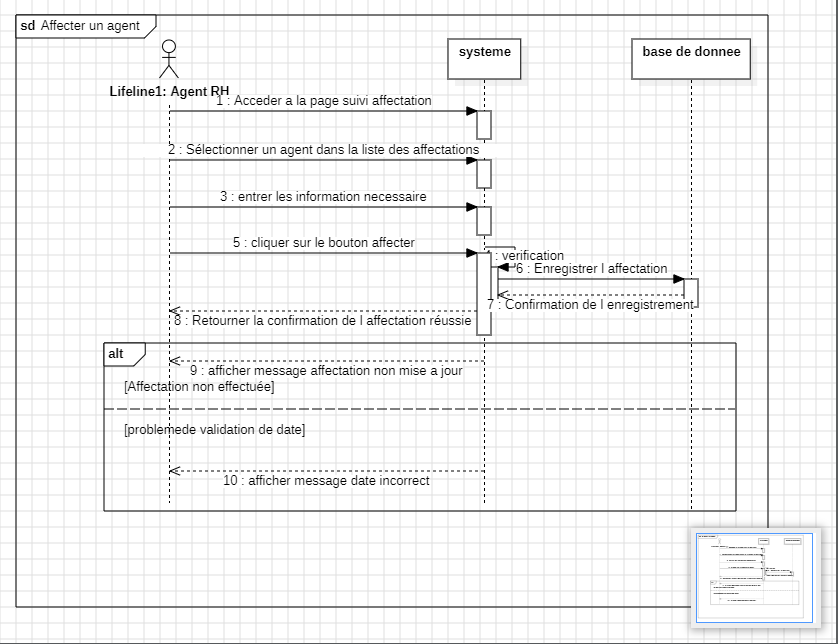


Figure 21: Diagramme de séquence du cas affecter un agent

Cas 9 : Ajouter un agent

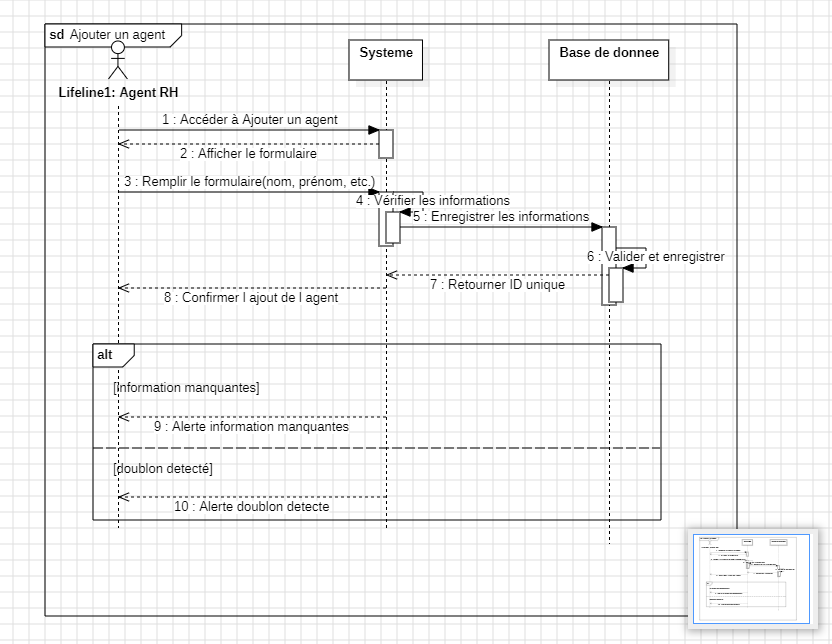


Figure 22:Diagramme de séquence du cas Ajouter un agent

Cas 10 : bloquer compte d’un agent

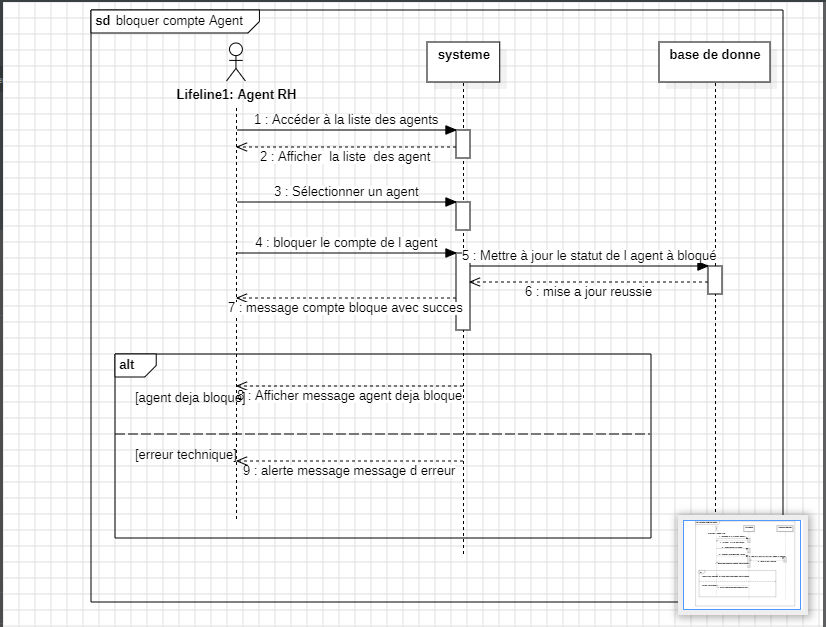


Figure 23:Diagramme de séquence du cas bloquer compte d un agent

Cas 11 : Modifier un agent

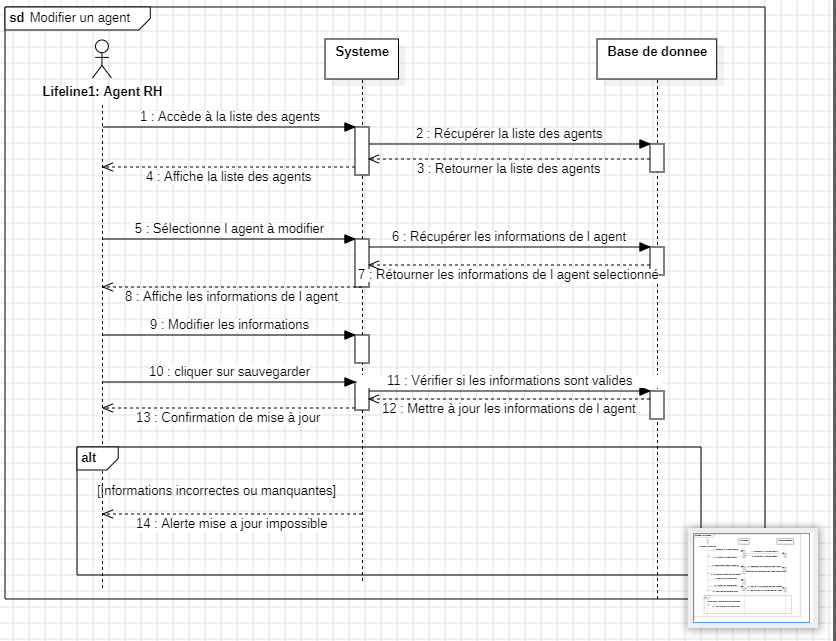


Figure 24:Diagramme de séquence du cas Modifier un agent

1. CONCEPTION
2. **Diagramme de classe**

Le diagramme de classe est le diagramme UML le plus important de la modélisation Objet. C’est une représentation statique des objets du système qui interagissent pour réaliser les cas d’utilisation. Ainsi ci-dessous présenté le diagramme de classe :

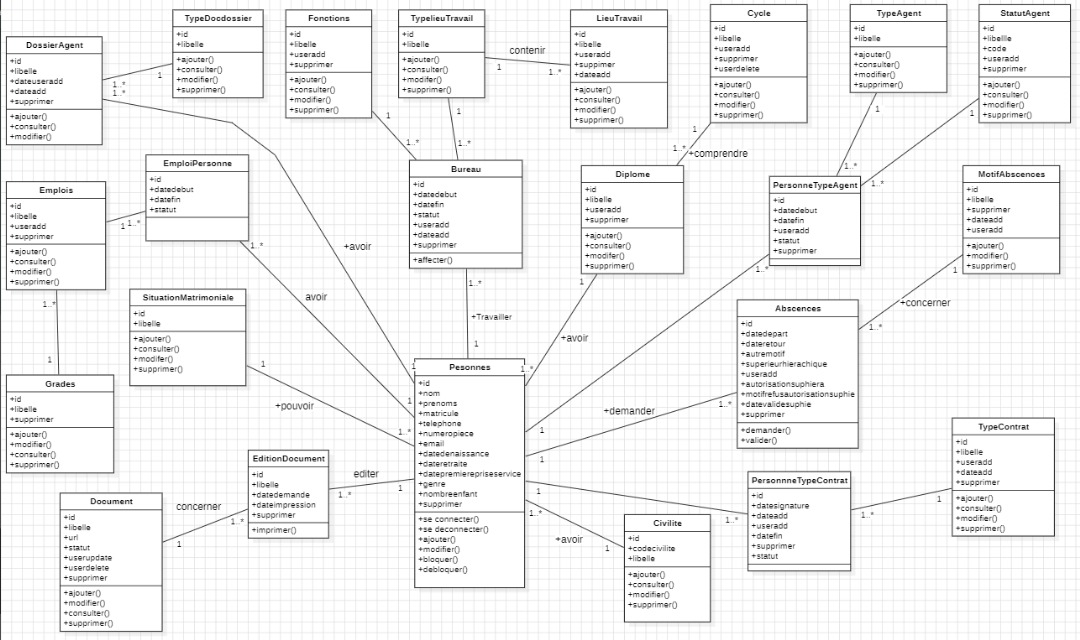


Figure 25:Diagramme de classe

Conclusion

La modélisation à l’aide de diagrammes UML a été essentielle pour clarifier les différentes exigences fonctionnelles et structurelles du système web de gestion des processus RH à l'INP-HB. Le diagramme de cas d'utilisation a permis de cerner les principales fonctionnalités du système, tandis que le diagramme de séquence a illustré la dynamique des processus, et le diagramme de classe a offert une vue d’ensemble de la structure du système. Ces outils de modélisation ont joué un rôle crucial dans la planification et la mise en œuvre du projet, facilitant la communication entre les différentes parties prenantes et assurant la cohérence du système lors de sa réalisation.

PARTIE 3 : ETUDE TECHNIQUE ET REALISATION

CHAPITRE 5 : ETUDE TECHNIQUE

INTRODUCTION

L’implémentation est la phase la plus critique après celle de la conception. Le choix des outils de développement influence considérablement sur le coût en temps de programmation et sur la flexibilité du produit à réaliser. Cette phase consiste à transformer le modèle conceptuel établi en composants logiciels qui constitueront notre système. Dans ce chapitre, nous commencerons par la description de l’environnement de travail, puis nous aborderons l'architecture de l'application et les composants de notre système, ainsi que les choix technologiques effectués.

1. **DESCRIPTION DE L’ENVIRONNEMENT DE TRAVAIL**

Pour garantir une collaboration efficace et un développement fluide, notre environnement de travail a été soigneusement configuré. Les outils utilisés incluent :

* **Ordinateur** : Un PC personnel Dell avec les spécifications suivantes :
* Processeur : Intel® Core™ i3-6006U à 2,00 GHz, avec 2 cœurs et 4 threads.
* Mémoire RAM : 4 Go (3,61 Go utilisables).
* Système d’exploitation : Windows 10 (64 bits, processeur x64), offrant une plateforme stable pour les applications de développement et une compatibilité avec les outils choisis.
* **IDE et éditeurs de code** : Visual Studio Code a été notre éditeur de choix, apprécié pour ses nombreuses extensions utiles pour React et Laravel.
* **Serveur de développement** : WAMP a été installé pour gérer notre environnement PHP et MySQL, permettant une exécution locale de l’application.
* **Navigateur** : Chrome a été utilisé pour le développement et le débogage, grâce à ses outils de développement avancés.

1. **ANALYSE DES BESOINS TECHNIQUES**

Avant de commencer l’implémentation, nous avons identifié les exigences fonctionnelles et non fonctionnelles du projet, qui ont été détaillées dans le **chapitre 2**. Nous y avons présenté les principales fonctionnalités attendues de l'application ainsi que les exigences en termes de performance, de sécurité et de compatibilité. Pour plus de détails sur ces exigences, veuillez-vous référer à la section **III. Objectifs du Projet** et **IV. Cahier des charges** du chapitre 2.

1. **CHOIX DES DIFFERENTS OUTILS, LANGAGE DE PROGRAMMATION ET FRAMEWORK**

L'environnement de développement joue un rôle crucial dans la création d'applications performantes et fiables. Pour notre projet, nous avons choisi une combinaison de technologies modernes qui favorisent l'interactivité et la rapidité de développement.

* Choix d’un SGBD



Figure 26 : Logo MySQL

Les systèmes de gestion de bases de données (SGBD) jouent un rôle essentiel dans la création, la gestion et la manipulation des données. Ces systèmes permettent non seulement de créer de nouvelles tables et d'ajouter des entrées, mais également de mettre à jour des données et de récupérer des informations déjà enregistrées. Parmi les SGBD, les systèmes relationnels, basés sur le langage SQL (Structured Query Language), se distinguent par leur capacité à gérer des transactions complexes et à établir des relations entre les données.

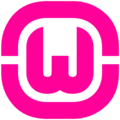
Pour notre projet, nous avons choisi d'utiliser MySQL comme SGBD, intégré dans un environnement de développement WAMP (Windows, Apache, MySQL, PHP).

Ce choix repose sur plusieurs raisons convaincantes :

- Performance et Scalabilité : MySQL est reconnu pour sa robustesse, sa rapidité et sa capacité à gérer efficacement de grandes quantités de données ainsi que de nombreuses connexions simultanées. Cela le rend particulièrement adapté aux applications d'entreprise qui nécessitent une disponibilité constante et une réactivité élevée.

- Support et Communauté : En tant que l'un des SGBD les plus utilisés au monde, MySQL bénéficie d'une vaste communauté de développeurs. Cela se traduit par une abondance de ressources, de tutoriels et de documentation, facilitant ainsi le développement et la résolution de problèmes rencontrés lors du projet.

* Serveur d’application



**Figure 27 : Logo Wamp**

WAMP est une plateforme de développement qui permet de configurer facilement un serveur web local sur un système d'exploitation Windows. Elle inclut Apache comme serveur HTTP, MySQL pour la gestion des bases de données et PHP comme langage de script côté serveur.

L'utilisation de WAMP a permis de créer un serveur local convivial, intégrant Apache pour le serveur web, MySQL pour la gestion des bases de données, et PHP pour le développement back-end. Cet environnement a grandement facilité le test et le débogage de notre application, en offrant un cadre similaire à celui de la production.

* Editeur de code

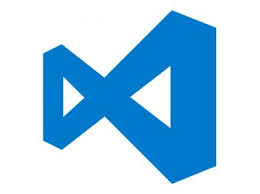


Figure 28 : Logo VS code

Visual Studio Code (VS Code) : est un éditeur de code source développé par Microsoft. Il est léger, open-source et multiplateforme, ce qui signifie qu'il peut être utilisé sur Windows, macOS et Linux. VS Code est conçu pour être extensible, permettant aux développeurs d'ajouter des fonctionnalités via une vaste bibliothèque d'extensions. Il prend en charge une multitude de langages de programmation, notamment JavaScript, PHP, Python et bien d'autres.

L'éditeur offre une interface utilisateur intuitive qui facilite la navigation dans les projets, tout en intégrant des fonctionnalités avancées telles que le débogage, la gestion de version avec Git, et des outils d'autocomplétions. Ces caractéristiques font de VS Code un choix populaire parmi les développeurs pour la création d'applications modernes.

Dans le cadre de notre projet le choix de Visual Studio Code s'est imposé pour plusieurs raisons clés :

- Interface Utilisateur Intuitive : VS Code propose une interface claire et personnalisable, facilitant la gestion des fichiers et la navigation entre les différents composants de l'application, ce qui est crucial pour un projet utilisant plusieurs technologies.

- Support Multi-Langages : Avec sa capacité à gérer plusieurs langages (PHP pour Laravel et JavaScript pour React), VS Code est idéal pour notre stack technologique. Les extensions spécifiques permettent d'améliorer l'expérience de développement pour chacun de ces langages.

-Intégration avec Git : VS Code facilite le contrôle de version grâce à une intégration native avec Git. Cela permet un suivi efficace des modifications et une collaboration fluide entre les membres de l'équipe.

* Outil de collaboration

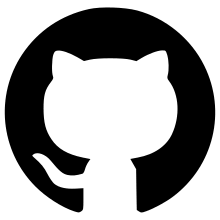


Figure 29 : Logo GitHub

**GitHub** est un service web d'hébergement et de gestion de développement de logiciels, qui repose sur le logiciel de gestion de versions **Git**. Il permet à plusieurs collaborateurs de travailler à distance sur un même projet tout en maintenant un suivi précis des modifications apportées au code. Grâce à **Git**, chaque modification est sauvegardée sous forme de *commit*, créant ainsi une série d'historique de versions. Cela permet aux développeurs de revenir à des versions antérieures du projet en cas de problème, de fusionner des changements en toute sécurité et de résoudre les conflits potentiels qui pourraient survenir lors du développement en parallèle.

Dans le cadre de ce projet, **GitHub** a joué un rôle fondamental dans l'organisation et la gestion des tâches. Il a facilité la collaboration entre les membres de l’équipe, permettant une gestion fluide des versions du code à chaque étape du développement. De plus, grâce aux **branches** et **pull requests**, chaque membre de l’équipe a pu travailler indépendamment sur des fonctionnalités spécifiques, puis proposer ses changements pour les intégrer dans la branche principale après une revue et une validation.

* Framework

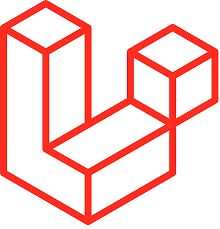


Figure 30 : Logo Laravel

**Laravel** est un framework PHP puissant conçu pour le développement d'applications web robustes et maintenables, en utilisant le modèle **Modèle-Vue-Contrôleur** (MVC). Cette architecture sépare clairement la logique métier (modèle), la présentation (vue) et le traitement des requêtes (contrôleur), facilitant ainsi le développement, la maintenance et l’évolution des applications.

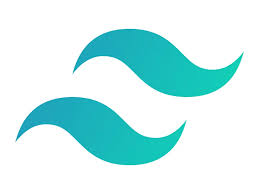


Figure 31 : Logo Tailwind Css

**Tailwind CSS** est un framework CSS basé sur une approche utilitaire-first, conçu pour faciliter la création d'interfaces utilisateur modernes et réactives. Contrairement à d'autres frameworks CSS, comme **Bootstrap**, qui offrent des composants prédéfinis, **Tailwind** propose une série de classes utilitaires permettant de construire des designs personnalisés directement dans le code HTML. Cela donne une flexibilité maximale tout en réduisant la quantité de CSS à écrire.

Dans le cadre de notre projet, nous avons opté pour **Tailwind CSS** pour plusieurs raisons :

* **Flexibilité de conception** : Grâce à son approche utilitaire-first, **Tailwind CSS** permet de créer des designs entièrement personnalisés sans avoir à se conformer à des modèles préexistants. Chaque classe utilitaire est responsable d'une seule propriété CSS (par exemple, p-4 pour des marges internes ou text-center pour centrer le texte), ce qui facilite la création de designs très spécifiques et adaptés aux besoins du projet.
* **Rapidite de développement** : **Tailwind** permet de construire des interfaces rapidement sans avoir à écrire de longs fichiers CSS. En appliquant des classes utilitaires directement sur les éléments HTML, nous avons pu optimiser notre temps de développement, tout en assurant un rendu cohérent et responsive sur différents appareils.
* **Réactivité et compatibilité** : **Tailwind CSS** facilite la création d'interfaces réactives grâce à ses classes intégrées qui gèrent la mise en page pour différents types d'écrans (mobile, tablette, bureau). Cela nous a permis de créer une interface fluide qui s’adapte automatiquement aux différentes tailles d’écrans sans avoir à écrire de media queries complexes.
* **Performance optimisée** : **Tailwind CSS** intègre une fonctionnalité de purge qui élimine les classes CSS inutilisées lors de la production, ce qui réduit la taille du fichier CSS et améliore les performances de l’application.
* Outil de liaison

****

Figure 32 : Logo Inertia

Inertia.js n'est ni une bibliothèque ni un framework à proprement parler, mais plutôt une **"interface"** ou un **"outil de liaison"** qui facilite l'intégration entre un framework côté serveur (comme Laravel) et un framework côté client (comme React, Vue, ou Svelte). Il permet de créer des applications web modernes de type **monolithique** tout en conservant la simplicité et les bonnes pratiques des applications serveur traditionnelles.

Le concept d'Inertia repose sur la création d'applications **single-page (SPA)** tout en s’appuyant sur des pratiques classiques côté serveur. Plutôt que de nécessiter la mise en place complexe d’une API REST ou GraphQL pour la communication entre le client et le serveur, Inertia permet de garder la logique serveur tout en offrant l'interactivité d'une application côté client.

Dans le cadre de notre projet nous avons opté pour inertia.js pour plusieurs raisons :

-**Pas de routage côté client** : Inertia n’utilise pas de système de routage côté client comme d'autres frameworks SPA (par exemple, React Router). Il laisse ce travail au serveur, et la gestion des pages se fait naturellement via les **contrôleurs et vues** côté serveur.

-**Pas d’API nécessaire** : Contrairement aux SPA traditionnelles qui nécessitent une API pour échanger des données entre le client et le serveur, Inertia permet de rendre les pages dynamiques sans avoir à développer une API séparée. Vous pouvez gérer les requêtes et les réponses directement avec des **contrôleurs Laravel** ou d'autres frameworks backend.

-**Connecteur entre le backend et le frontend** : Inertia agit comme une **colle** entre le backend (par exemple Laravel) et le frontend (React, Vue, Svelte). Il facilite l'échange de données entre le serveur et le client tout en permettant d'utiliser les **composants React**, **Vue**, ou **Svelte** côté client sans la complexité habituelle des SPAs.

**-Adapté aux frameworks côté serveur** : Bien qu’Inertia puisse fonctionner avec n'importe quel framework backend, il est **optimisé pour Laravel**, ce qui permet de tirer pleinement parti des outils et des pratiques Laravel tout en obtenant une expérience utilisateur fluide et réactive sur le frontend.

Ces fonctionnalités clés font d’Inertia.js un choix idéal pour notre projet, car elles favorisent une expérience de développement harmonieuse et une interface utilisateur performante.

Avec ces atouts en tête, nous allons explorer la configuration pratique de notre environnement de développement, qui nous permettra d'exploiter pleinement ces fonctionnalités

**Configuration de Notre Projet Laravel avec React et Inertia.js**

Avec les avantages clairs d’Inertia.js en tête, passons maintenant à la configuration de notre projet. Voici les étapes pour mettre en place notre environnement de développement avec Laravel, React et Inertia.js :

**Première Étape : Création d’un Nouveau Projet Laravel**

* **Cas 1 : Laravel n'est pas installé sur la machine**
* composer create-project laravel/laravel nom-projet
* **Cas 2 : Laravel est déjà installé :**
* laravel new nom-projet

Puis, accédez au répertoire du projet :

* cd nom-projet

**Deuxième Étape : Installation de React et React DOM**

Pour installer React et React DOM, exécutez :

* npm install react react-dom

(Note : Cela nécessite que Node.js soit installé sur la machine pour utiliser npm.)

**Configuration de l'Asset Bundling avec Vite :**

Pour utiliser React avec Vite, installez le plugin React :

* npm install --save-dev @vitejs/plugin-react

Ensuite, incluez le plugin dans le fichier vite.config.js :

Configuration Côté Serveur

1. **Installation de l'Adaptateur Laravel pour Inertia :**

* composer require inertiajs/inertia-laravel

1. **Template Racine :** Créez un fichier app.blade.php pour charger les actifs de votre site :

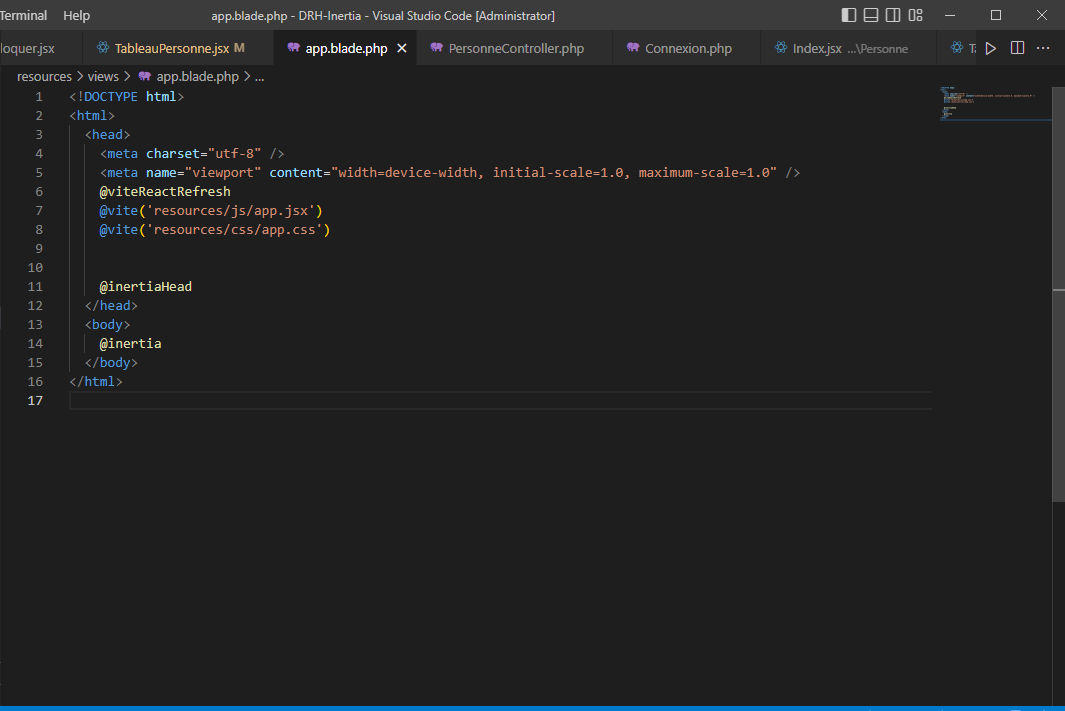


Figure 33 : fichier app.blade.php

1. **Mise en Place du Middleware Inertia :** Exécutez la commande suivante :

* php artisan inertia:middleware

Ensuite, ajoutez le middleware à votre groupe middleware dans bootstrap/app.php :

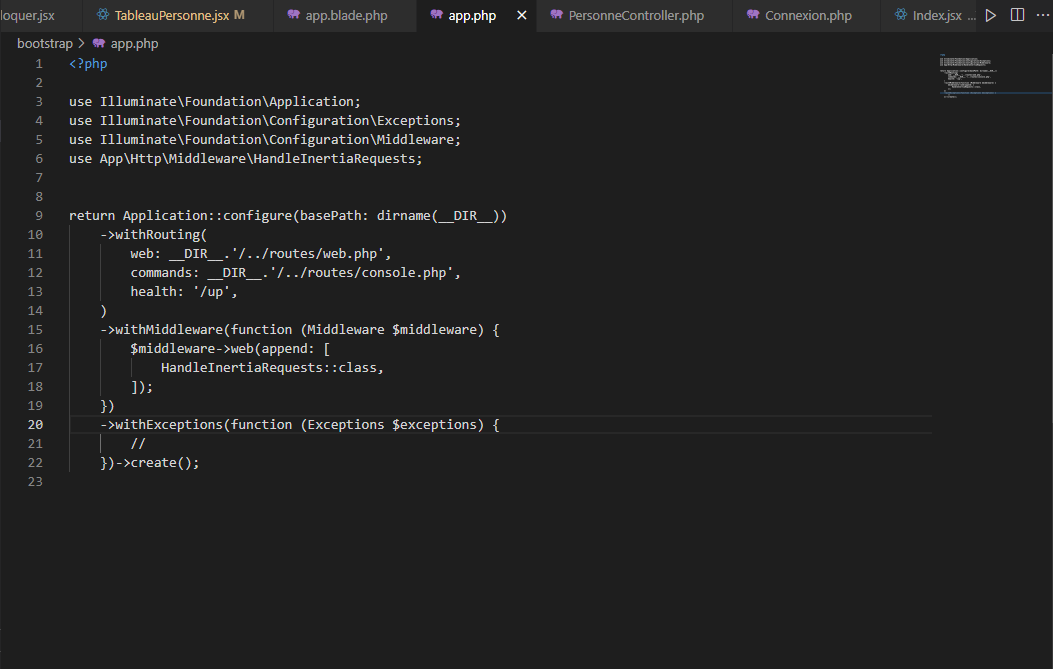


Figure 34 : fichier app.php

**Configuration Côté Client**

1. **Installation de l'Adaptateur Client pour Inertia :**

* npm install @inertiajs/react

1. **Initialisation de l’Application Inertia :** Dans votre fichier JavaScript principal, initialisez votre application Inertia avec le composant de base :

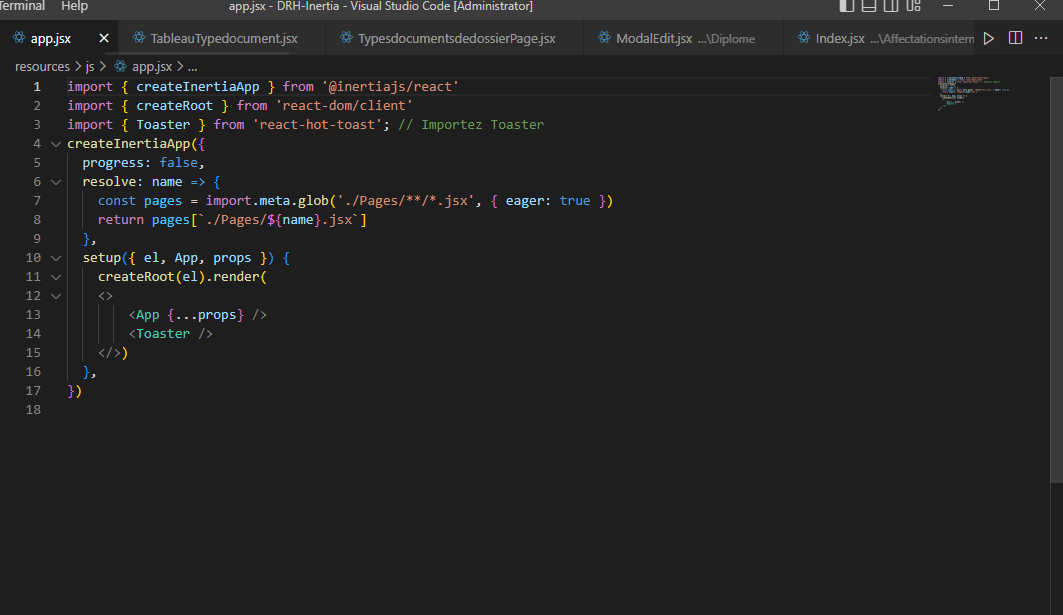


Figure 35 :fichier principal javascript app,jsx

Tandis qu'Inertia.js agit comme un intermédiaire entre Laravel et React, React nous permet de construire des interfaces utilisateur dynamiques et réactives qui exploitent pleinement les capacités d'Inertia.

* Bibliothèque

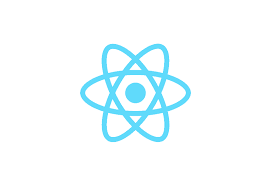


Figure 36 :Logo React

React est une bibliothèque JavaScript conçue pour la création d’interfaces utilisateur dynamiques, fondée sur un modèle de composants réutilisables qui interagissent avec un Virtual DOM.

- Développement Composant: Cette approche permet de créer des composants isolés, favorisant la réutilisabilité à travers l’application et garantissant une cohérence visuelle.

- Interface Utilisateur Dynamique: La gestion de l'état facilitée par React permet des mises à jour en temps réel, améliorant considérablement l'expérience utilisateur par rapport à d'autres bibliothèques.

- Performance Optimisée : Grâce à son Virtual DOM, React réduit les opérations coûteuses sur le DOM réel, offrant ainsi une performance accrue lors des mises à jour de l'interface.

* **Outil de manipulation de fichiers Excel**

****

Figure 37 : Logo PhpSpreadSheat

**PHPSpreadsheet** permet de gérer les données de manière tabulaire, ce qui est très utile dans des applications comme les systèmes de gestion de ressources humaines (SGRH), les systèmes de gestion des employés ou des inventaires, où les données sont souvent stockées et manipulées sous forme de tableaux. Ce type d'outil est essentiel pour offrir une gestion et une exportation de données facile et flexible dans des formats compatibles avec Excel.

1. Architecture de l'application

L'architecture de notre application repose sur un modèle **client-serveur**, qui permet de séparer clairement les responsabilités entre le front-end (client) et le back-end (serveur). Ce modèle facilite la gestion des ressources, améliore la sécurité et permet une évolution aisée de l'application.

* **Client (Front-end)** : Le front-end est développé avec **React**, un framework JavaScript moderne, idéal pour la création d'interfaces utilisateur dynamiques et interactives. En utilisant **Tailwind CSS**, nous avons pu concevoir une interface élégante et réactive qui offre une expérience utilisateur intuitive. Le front-end interagit directement avec les utilisateurs, traitant leurs entrées et affichant les données de manière fluide. Pour optimiser cette interaction, nous avons intégré **Inertia.js**, qui sert de pont entre React et Laravel. Inertia.js permet de gérer les requêtes et les réponses de manière transparente, offrant une expérience semblable à celle d'une application à page unique (SPA), sans nécessiter de rechargement complet des pages.
* **Serveur (Back-end)** : Le back-end est construit sur **Laravel**, un framework PHP robuste qui gère les requêtes HTTP et exécute la logique métier de l'application. Grâce à son ORM (Object-Relational Mapping) appelé Eloquent, Laravel facilite les interactions avec la base de données MySQL. Lorsque le client envoie une requête, le serveur traite cette requête, effectue les opérations nécessaires sur la base de données, puis renvoie des données formatées au front-end, garantissant ainsi une communication efficace.
* **Base de données** : Nous utilisons **MySQL**, un système de gestion de bases de données relationnelles, pour stocker des informations essentielles concernant les utilisateurs et les employés. Intégré dans un environnement de développement WAMP (Windows, Apache, MySQL, PHP), MySQL permet de gérer aisément les connexions et les transactions. Cette structure relationnelle assure l'intégrité des données et optimise les performances lors des opérations de lecture et d'écriture.

Cette architecture modulaire nous permet de gérer les mises à jour et l'ajout de nouvelles fonctionnalités sans affecter l'ensemble du système. Grâce à Inertia.js, la communication entre le front-end et le back-end est fluide, permettant l’échange de données en temps réel sans recharger la page, ce qui améliore considérablement l’expérience utilisateur et la réactivité de l'application.

1. Mise en œuvre des composants

La mise en œuvre des composants s'est déroulée en plusieurs étapes :

1. **Développement des fonctionnalités** : Chaque membre de l'équipe a travaillé sur des modules spécifiques, en utilisant les branches Git pour gérer le code.
2. **Tests et validation** : Nous avons mis en place des tests unitaires et fonctionnels pour garantir la fiabilité des fonctionnalités.
3. **Optimisation** : Nous avons effectué des révisions de code régulières pour améliorer la performance et corriger les erreurs détectées.

Chaque étape a été soigneusement documentée pour assurer une traçabilité et faciliter les ajustements ultérieurs.

CONCLUSION

Le choix de technologies telles que Laravel, WAMP, React.js et Inertia.js pour le Système de Gestion des Ressources Humaines (SGRH) est déterminé par leur capacité à offrir une solution moderne, sécurisée et efficace. Chaque technologie a été soigneusement sélectionnée pour ses caractéristiques spécifiques et sa pertinence aux exigences du projet, assurant ainsi une mise en œuvre réussie. Ensemble, elles contribuent à améliorer l'efficacité des processus RH tout en garantissant une expérience utilisateur fluide et intuitive, prête à répondre aux défis de la gestion des ressources humaines dans un environnement en constante évolution.

**CHAPITRE 6 : REALISATION DE L’APPLICATION**

INTRODUCTION

La phase de réalisation de notre application de gestion des ressources humaines a été une étape cruciale, transformant nos idées en un produit tangible et fonctionnel. Nos objectifs étaient clairs :

* Respecter les délais,
* Garantir la qualité du code
* Développer des fonctionnalités répondant aux besoins des utilisateurs

Étant tous membres de la même équipe, la coordination a été facilitée par des échanges réguliers et une communication ouverte. Notre maître de stage a joué un rôle clé en examinant notre code et nos interfaces de manière hebdomadaire, permettant d'obtenir des retours constructifs et d'ajuster notre travail en conséquence. Des discussions quotidiennes ont également permis de clarifier les attentes et de prioriser les tâches en fonction des défis rencontrés. Cette proximité et ce suivi ont été déterminants pour assurer la fluidité du développement.

1. **DEVELOPPEMENT DES FONCTIONNALITES**

Comme mentionné dans le Chapitre 5, les fonctionnalités principales comprennent :

* **Gestion des employés** : Permet l’ajout, la modification et la suppression d’employés, garantissant une base de données à jour et facile à naviguer.
* **Authentification des utilisateurs** : Un processus sécurisé pour l’inscription et la connexion, intégrant des protocoles de sécurité avancés pour protéger les données sensibles.
* **Suivi des absences** : Permet aux gestionnaires de visualiser et de gérer les absences des employés, facilitant ainsi la planification des ressources.

Chaque fonctionnalité a été développée en utilisant Git pour la gestion du code source. Ce système a permis une collaboration efficace, le suivi des modifications et la gestion des branches pour le développement de nouvelles fonctionnalités. Grâce à Git, nous avons pu effectuer des revues de code systématiques, renforçant ainsi la qualité et la fiabilité du produit.

1. **OPTIMISATION ET DEBOGAGE**

Une fois les fonctionnalités développées, nous avons engagé un processus d’optimisation du code afin d'améliorer la performance globale de l'application. Plusieurs stratégies ont été mises en œuvre :

* **Optimisation de la performance** : Nous avons analysé et optimisé les requêtes SQL pour réduire le temps de réponse des bases de données. Par exemple, nous avons utilisé des index pour accélérer les recherches et avons simplifié les requêtes complexes afin de minimiser leur impact sur la performance.
* **Réduction de la latence** : En identifiant les goulets d’étranglement au sein de l'application, nous avons pu localiser les zones où le temps de chargement était excessif. Grâce à des ajustements de code, nous avons amélioré la fluidité des interactions utilisateur, en particulier lors de la navigation entre les différentes pages.

Pour faciliter la communication entre le backend (Laravel) et le frontend (React), nous avons intégré **Inertia.js**. Ce framework permet une interaction directe entre les deux parties, tout en préservant l'expérience utilisateur d'une application monopage (SPA). Cela a significativement réduit le besoin de rechargements de page et amélioré la réactivité de l'application.

.

### PRESENTATION DE L'APPLICATION

Pour enrichir cette section, nous avons intégré des captures d'écran illustrant les différentes interfaces de l'application. Ces images montrent non seulement l'esthétique visuelle, mais aussi la fonctionnalité des différentes parties de l'application, facilitant ainsi la compréhension de l'expérience utilisateur.

* Fenêtre d’authentification

Si les informations sont invalides ou manquantes, le système affiche un message d’erreur en rouge, si non le système affiche la page d’accueil. S’il y a une tentative d’accès à une page, directement via l’URL de cette page et sans aucune identification, le système fait la redirection automatique vers la page d’authentification.

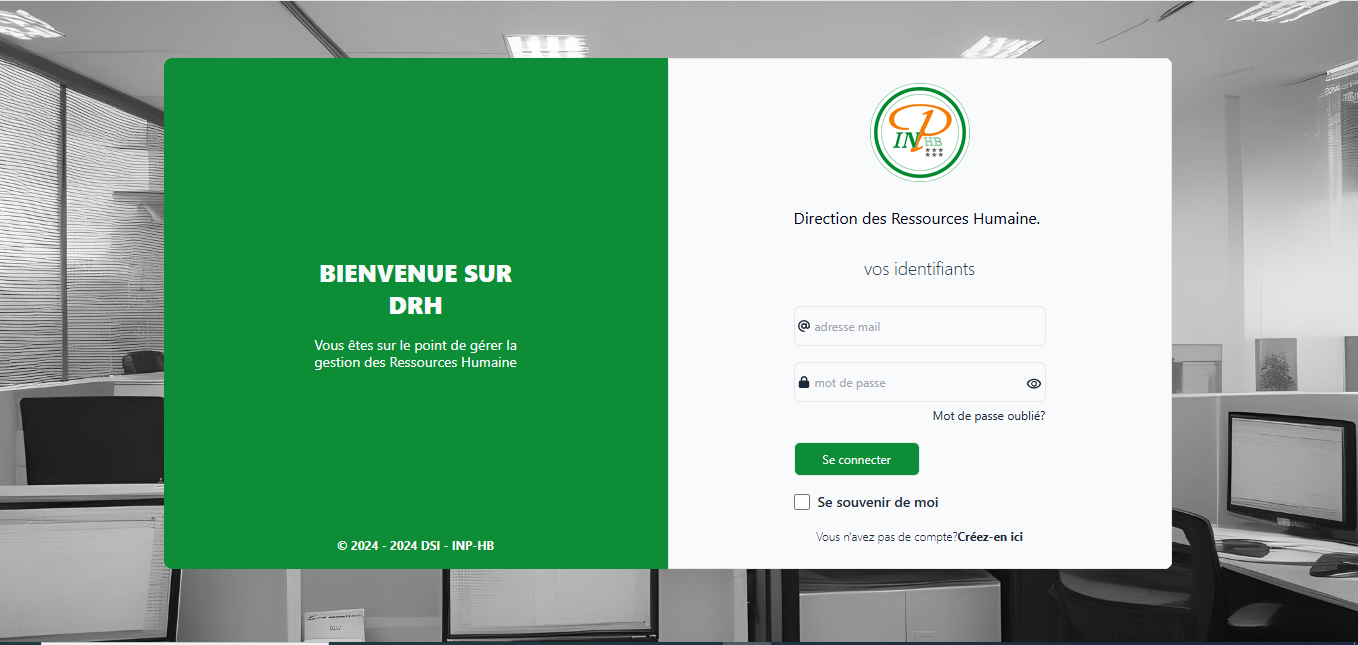


Figure 38:Page d’Authentification

Après l’authentification, et si les informations entrées sont valides, le système affiche la page d’accueil avec une sidebar qui présente les onglet de l’application.

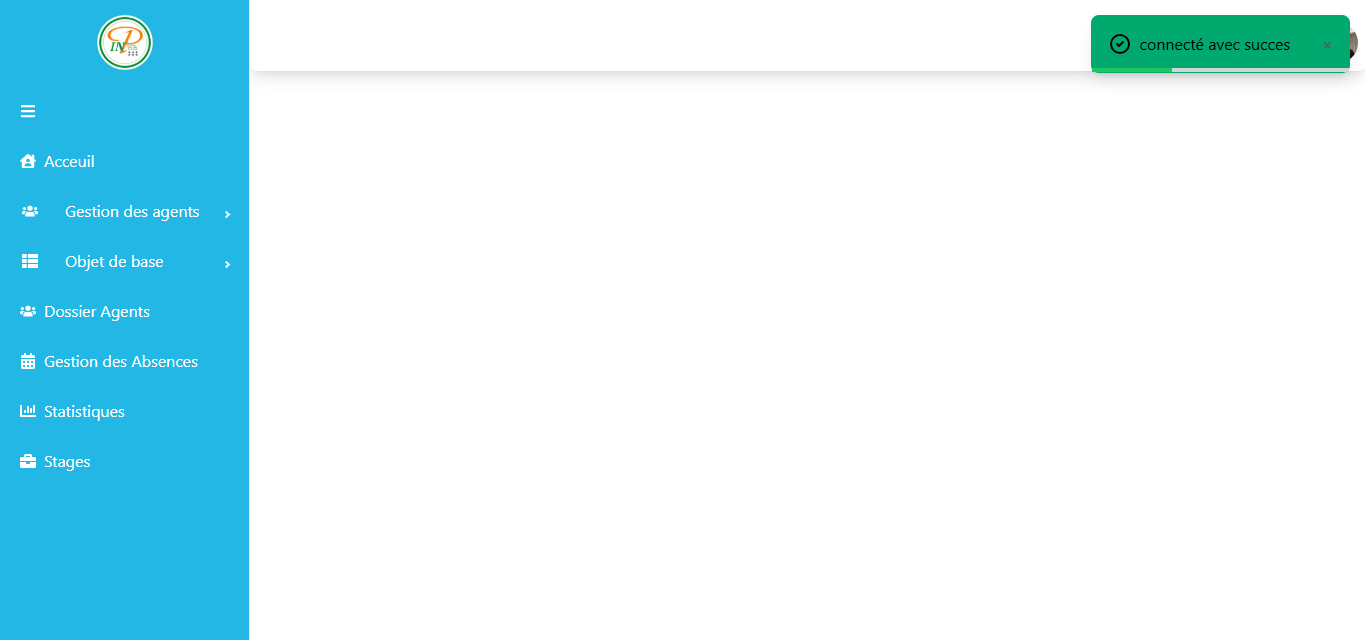


Figure 39:page d’Accueil

En cliquant sur la photo de profil de l’utilisateur connecté un menu déroulant s’affiche comme ci-dessous

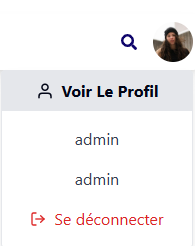


Figure 40 : présentation du menu déroulant

Il est clair que le système change le menu principal selon le type d’accès (Employe / Agent RH, Administrateur )

(figure menu principale de chaque utlisateur)

* Scenarios applicatifs

Nous allons présenter quelques scénarios applicatifs jugés les plus importants.

**-Ajouter un nouvel agent :**

Une clique sur Nouveau agent permet aux Utilisateurs d’accéder à la page d’ajout d’un agent. Ci-dessous la page d’ajout d’un agent, tous les champs doivent se remplir avant le clique sur Ajouter Agent sauf certain champs qu’ils sont pas obligatoire comme l’email institutionnel et le matricule, sinon le système affiche un petite message d’erreur en dessous des champs concernés et un toast d’erreur comme présenter

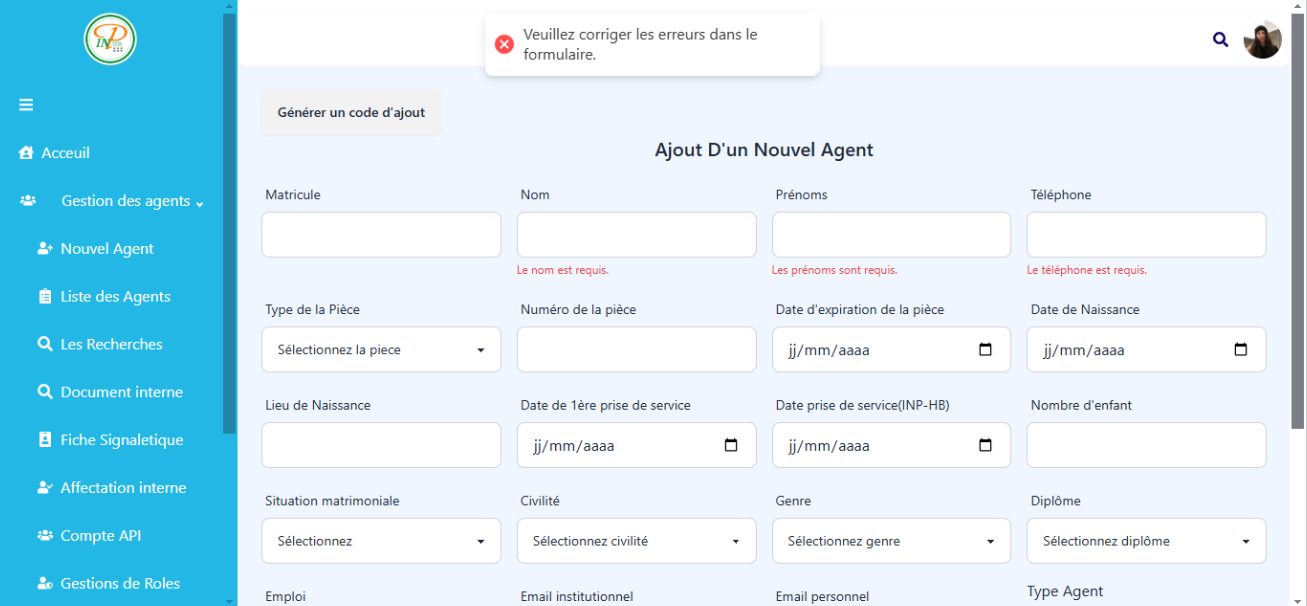


Figure 41: formulaire d’ajout un agent

Si le mail de l’agent est déjà existant, le système affiche un message d’erreur

Une fois l’agent ajouté, tous les champs sont réinitialisés, l’agent pourra ensuite utiliser son mot de passe est son mail institutionnel pour se connecter

S’il veut changer son mot de passe il peut le faire un fois connecter via la page ci-après

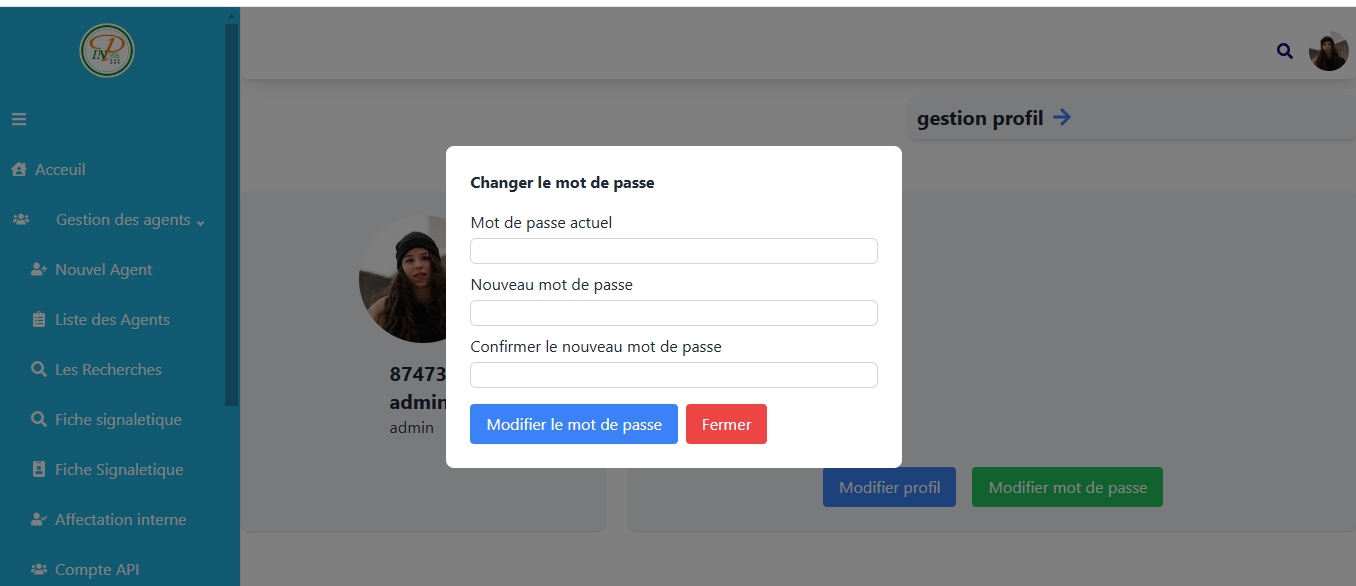


Figure 42:formulaire pour changer le mot de passe

Si l’on n’a pas toutes les informations de l’agent on peut utiliser le formulaire d’ajout rapide un fois les informations rentrées cela va envoyer un message à l’agent avec un lien lui permettant de poursuivre son inscription comme présenter

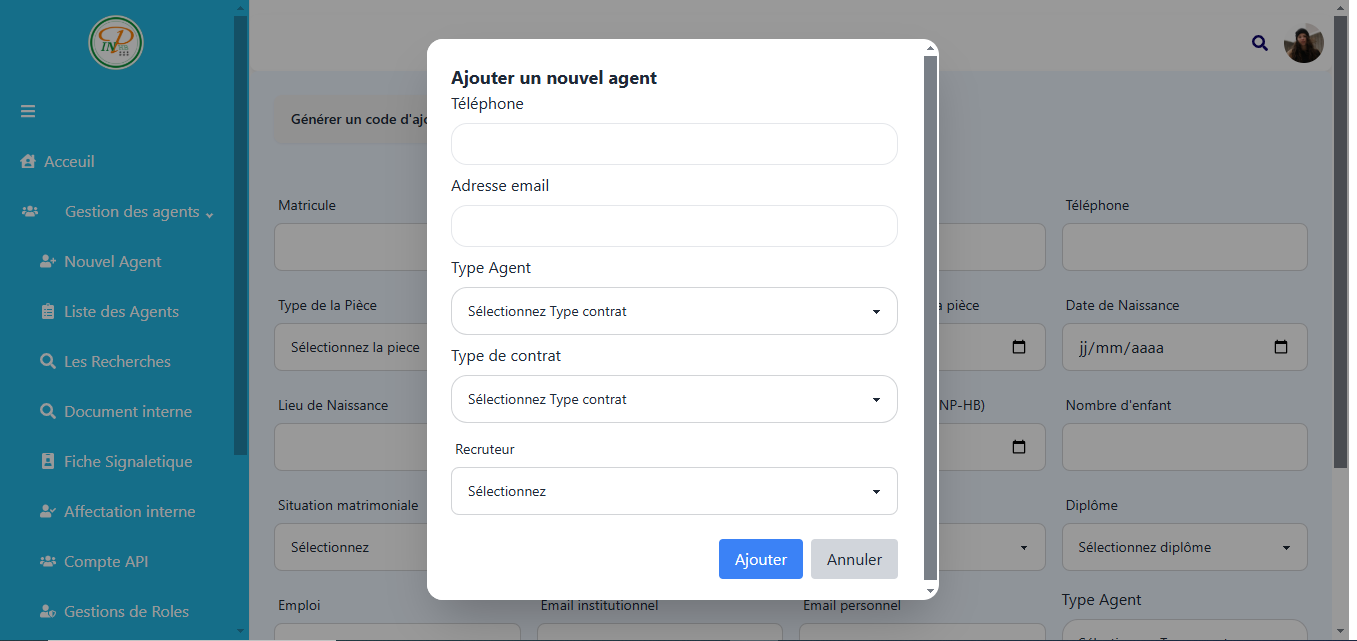


Figure 43:Formulaire d’ajout rapide

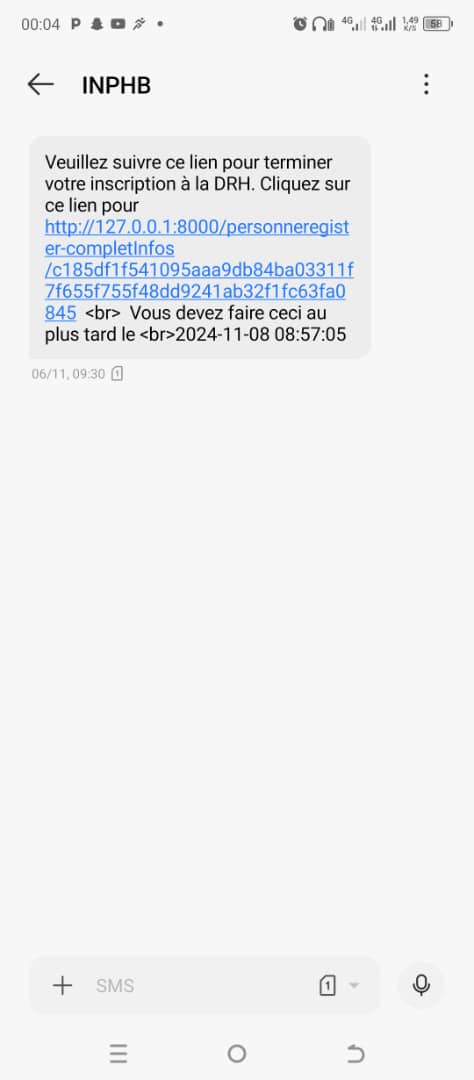


Figure 44 : message envoyé après ajout rapide d’un agent

(NB : lien pas fonctionnelle car notre application est encore en local et non en ligne)

**-Liste des Agents :**

La Liste des Agents présenté la liste de tous les agents enregistrés avec des informations comme (nom, prénoms) C’est à partir de cette liste que nous pouvons modifier les informations d’un agent en particulier affectation et aussi effectuer d’autres opérations comme (bloquer compte, imprimer fiche signalétique)

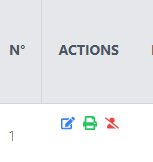


Figure 45:les différentes opérations

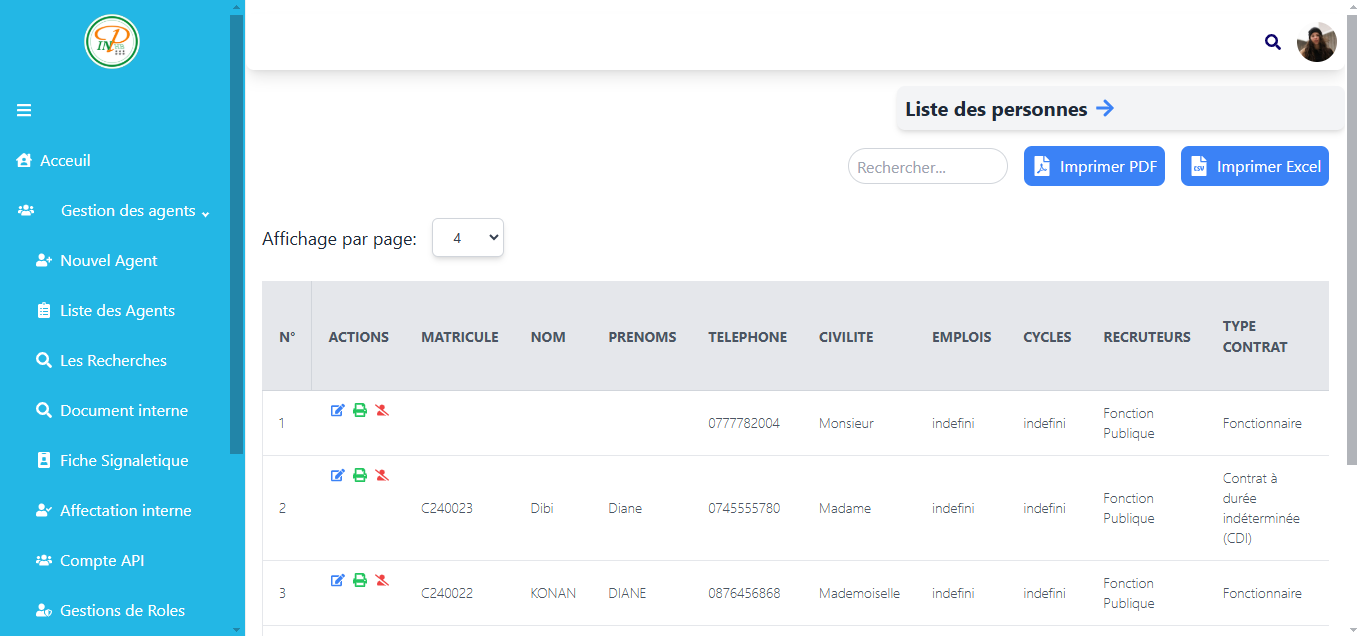


Figure 46 : liste des agents

**-Imprimer fiche signalétique :** on peut le faire de deux manières soit Après avoir cliqué sur le bouton Imprimer fiche signalétique, à partir de la liste des raccourcis. Ou en se rendant sur l’onglet fiche signalétique Après la sélection de l’agent, le système génère automatiquement la fiche signalétique de l’agent concerné avec son nom et prénom, …

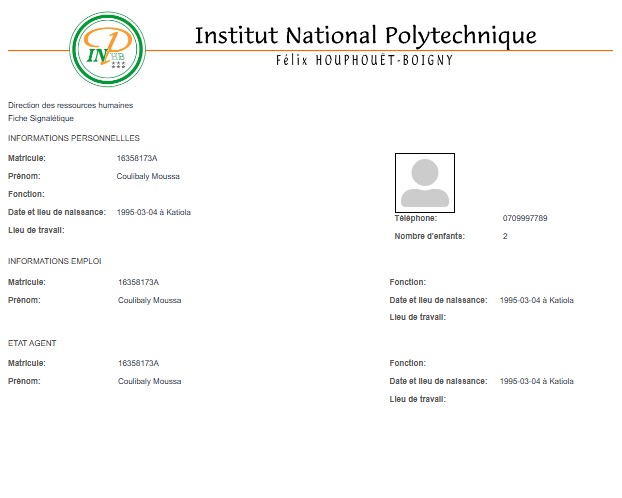


Figure 47: Exemple de fiche signalétique

**-liste des affectés**

Une clique sur Affectation interne permet aux Utilisateurs d’accéder à la page présentant la liste des affectés C’est à partir de cette liste que nous pouvons avoir des détails sur l’agent et ajouter une affectation

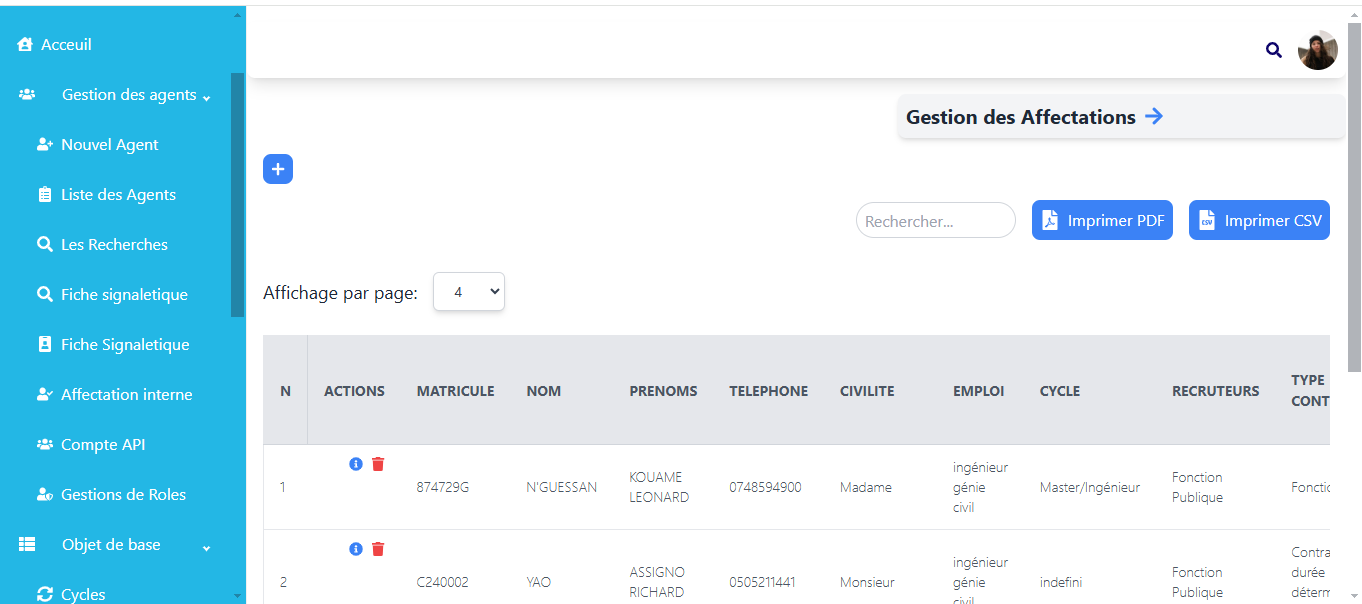


Figure 48:liste des affectés

-**Ajouter affectation :** Pour ajouter une affectation il faut remplir les champs, choisir l’agent choisit la fonction et choisit le lieu de travail. La date de fin doit être supérieure à la date du début si non le système affiche un petit message d’erreur, et lieu de travail et la fonction doit être différente à celle actuelle.

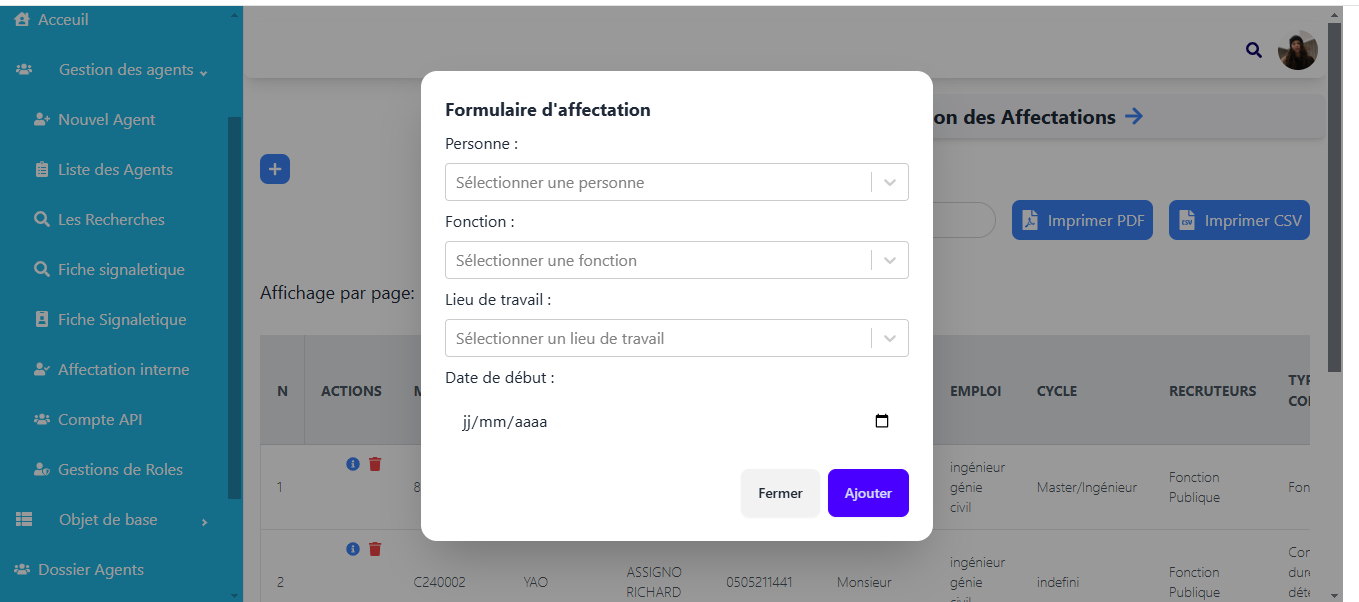
**

Figure 49: formulaire d’ajout d’une nouvelle affectation

**-Afficher détails agent**



**-gestion rôles**

Un clic sur gestion rôle permet d’accéder à un page qui permet à l’utilisateur de gérer les droits d’accès au sein de l’application

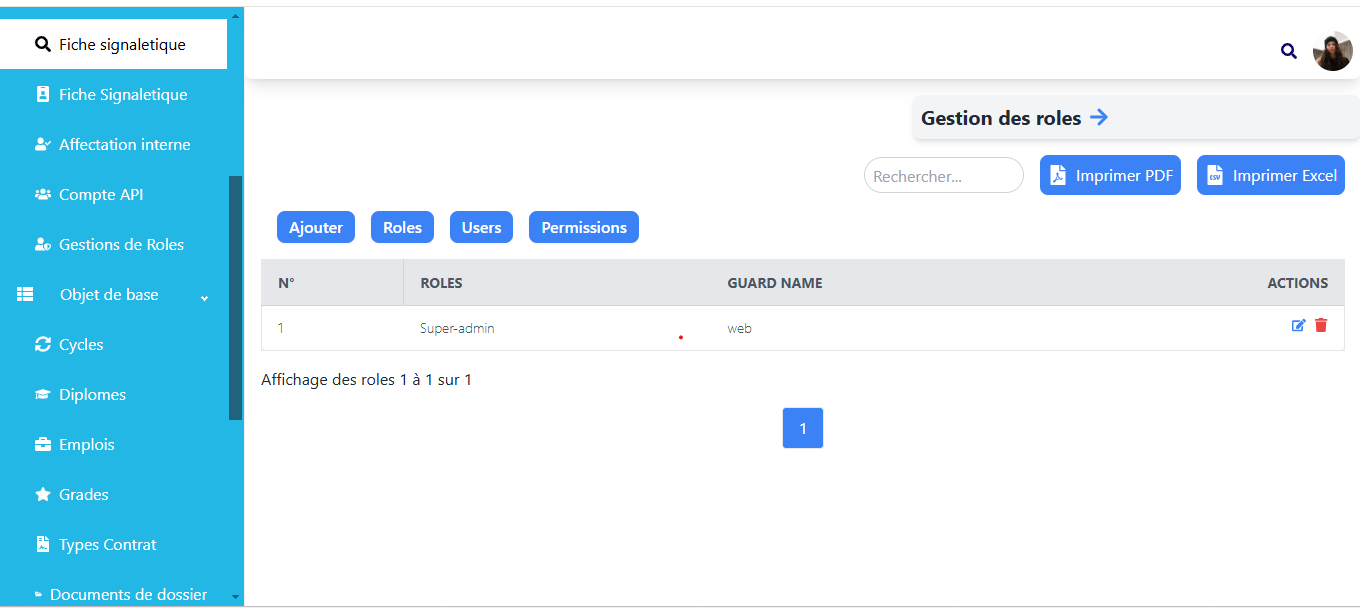
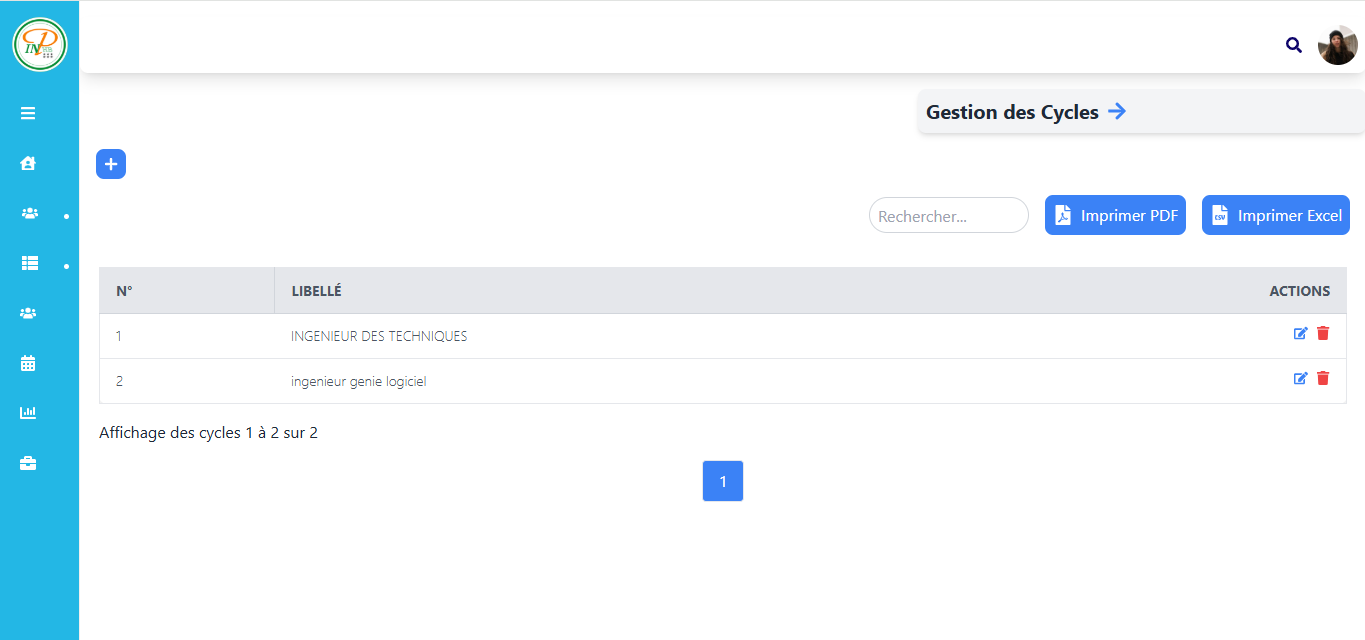


Figure 50 :interface pour la gestion des droits d’accès au sein de l’application

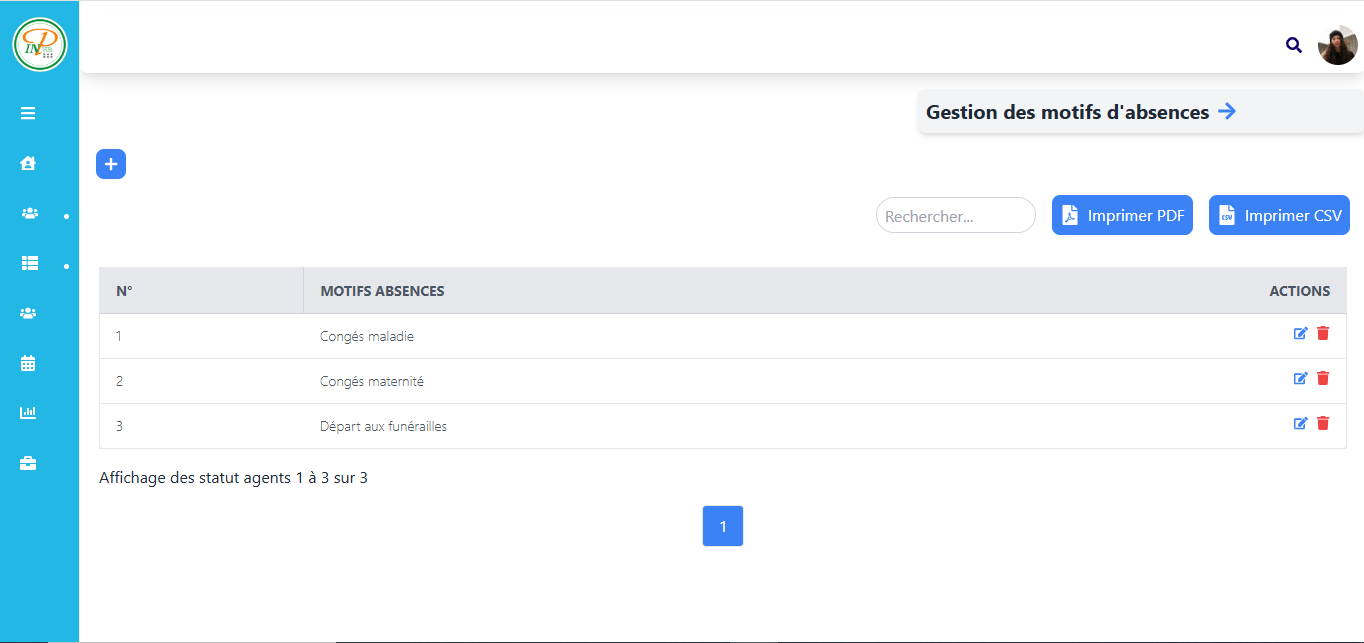
Pour la suite des éléments nous avons les objet base

L'onglet **"Objet de base"** dans l'application permet de gérer des éléments clés associés à chaque agent, facilitant ainsi la gestion des ressources humaines. Ce module est présenté si dessous

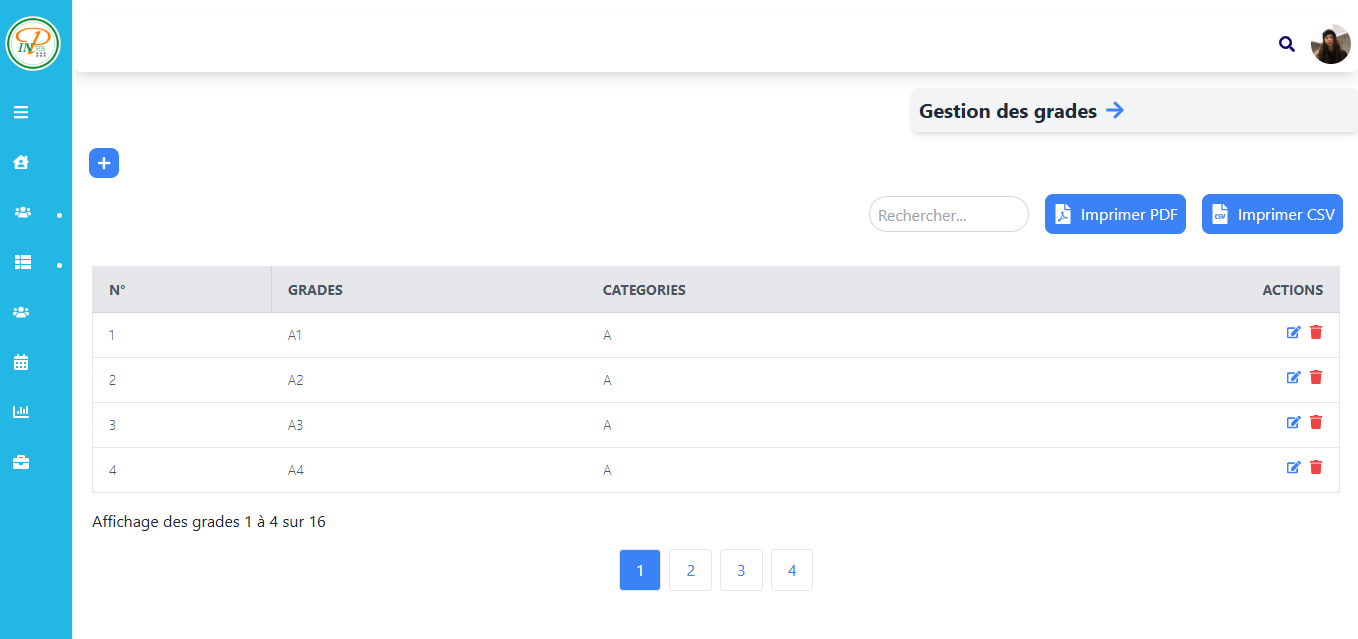
**La section Cycle**



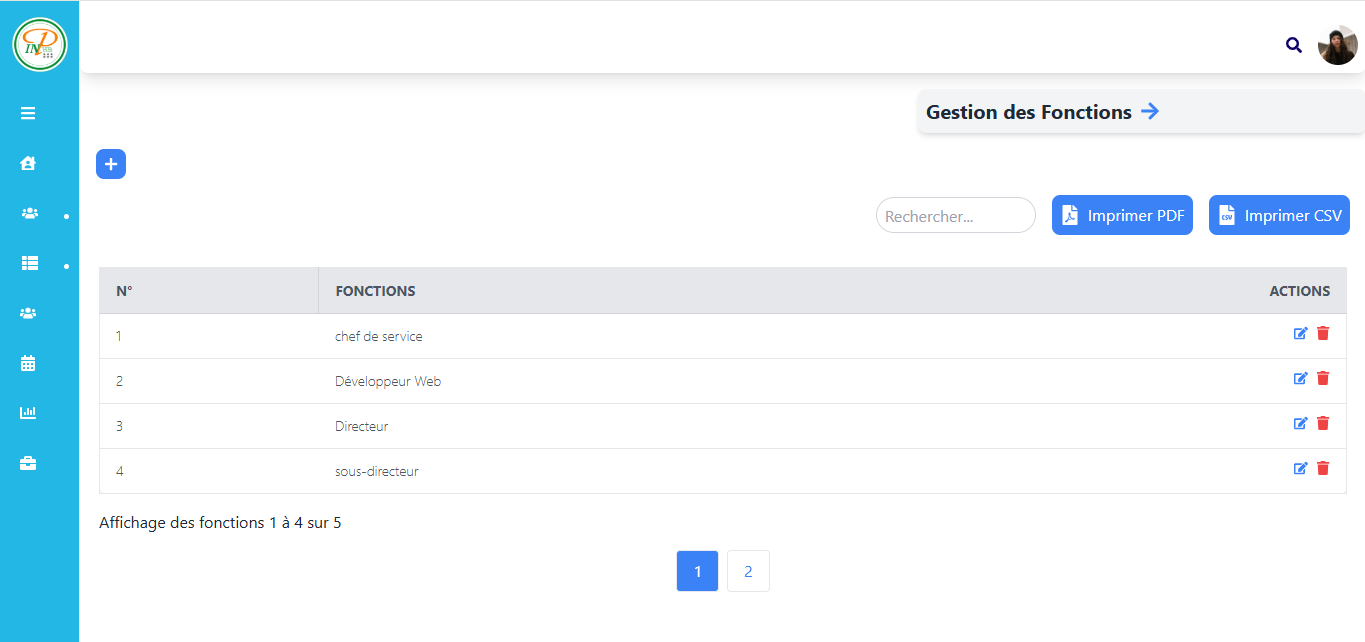
**Motif d'Absence**



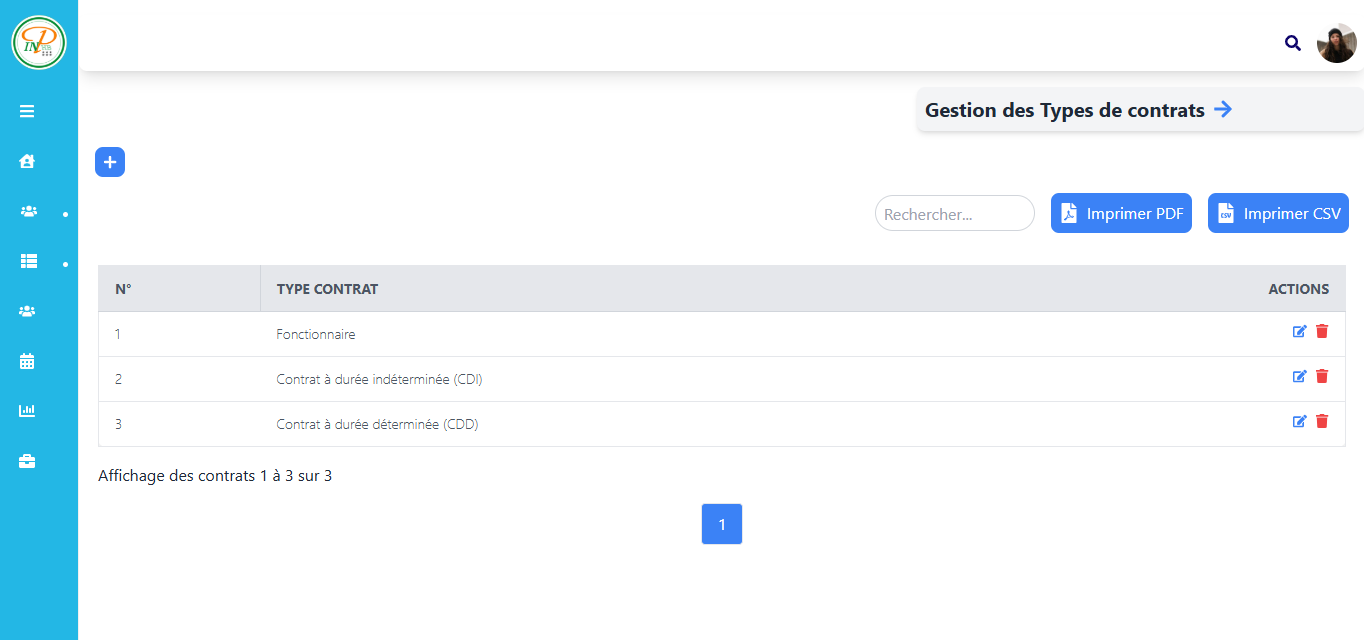
**La section Grade**



**La section fonction**



**La section type contrat**

****

CONCLUSION

Ce chapitre a détaillé les étapes de la réalisation de l’application de gestion des ressources humaines, en présentant les interfaces principales et les fonctionnalités clés mises en place. Nous avons illustré, à travers des exemples concrets et des scénarios applicatifs, le fonctionnement de l’application dans son ensemble. Ces éléments témoignent de l’accomplissement des objectifs du projet, et de l'aboutissement d’une solution répondant aux besoins des utilisateurs en matière de gestion RH.

**CONCLUSION GENERALE**

Au terme de ce rapport de fin d'étude, il convient de rappeler que l'objectif principal de ce stage était de concevoir et de réaliser un système web visant à **optimiser les processus de gestion des ressources humaines** à l'Institut National Polytechnique Félix-Houphouët-Boigny (INP-HB). Ce système a été conçu pour répondre aux besoins spécifiques de gestion des agents, notamment l'automatisation des processus de recrutement, d'affectation, de gestion des absences et de suivi des performances, en utilisant des technologies web modernes.

Pour mener à bien cette tâche, nous avons collaboré avec l’équipe RH de l’INP-HB pour mieux comprendre les enjeux et les besoins spécifiques du personnel administratif. À partir de ces informations, nous avons conçu une **application web intuitive et ergonomique**, permettant de simplifier les tâches administratives, de centraliser les données relatives aux agents et de faciliter l’accès à l’information en temps réel.

L'une des étapes clés de ce projet a été l'élaboration de l'interface utilisateur. Celle-ci a été pensée pour être simple d'utilisation, tout en intégrant des fonctionnalités avancées telles que la gestion des affectations internes, le suivi des absences, et la gestion des grades et fonctions des agents. L’utilisation de bases de données relationnelles pour le stockage des informations et l'implémentation de diverses fonctionnalités ont permis de garantir la sécurité, la fiabilité, et la performance du système.

Bien que la version actuelle de l’application ne couvre pas l’ensemble des processus RH possibles, nous avons pu implémenter les fonctionnalités de base, notamment la gestion des agents, l'affectation des rôles et la gestion des absences. Ces fonctionnalités ont été testées et validées, et le système a montré un taux de fiabilité élevé lors de son utilisation en mode test.

Au cours de ce projet, nous avons pu mettre en pratique une grande partie des compétences théoriques acquises durant notre cursus, notamment dans la gestion de projet, le développement web et la gestion des bases de données. De plus, l’expérience acquise lors de ce stage a permis de renforcer notre maîtrise des outils et technologies modernes utilisés dans la gestion des systèmes d’information.

La **perspective d'avenir** pour ce projet serait de l’étendre et de l’enrichir en intégrant de nouvelles fonctionnalités telles que la gestion des carrières, la planification des formations, ainsi que l’automatisation des évaluations des agents. Il serait également pertinent d'intégrer un module d’analyse des données pour faciliter la prise de décision en matière de gestion des ressources humaines. À terme, il serait possible d'étendre ce système à d'autres institutions ou organisations, en personnalisant les fonctionnalités en fonction des besoins spécifiques de chaque structure.

En conclusion, ce projet a constitué une expérience extrêmement enrichissante tant sur le plan professionnel que personnel, et a contribué de manière significative à l’amélioration de la gestion des ressources humaines au sein de l'INP-HB. Nous espérons que ce système contribuera à simplifier et à rendre plus efficaces les processus administratifs au sein de cette institution, et qu’il pourra être étendu à d’autres contextes dans un avenir proche.

**BIBLIOGRAPHIE**

PASCAL ROQUES, << UML2 par la pratique >>, Eyrolles,5e edition,2006,364 pages

# **WEBOGRAPHIE**

Justine Debret. Les règles de mise en page d’un mémoire ou d’une thèse. Disponible sur <https://www.scribbr.fr/elements-linguistiques/regles-mise-en-page/> consulte le 22 septembre 2024

[https://inphb.ci](https://inphb.ci/portail/missions_vision_valeurs) consulté le 23 septembre 2024

<https://fr.wikipedia.org/wiki/Processus_unifi%C3%A9> consulté le 25septembre 2024

<https://fr.wikipedia.org/wiki/UML_(informatique)> consulté le 24 septembre 2024