Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

**Dédicaces**

A tous ceux qui nous ont éclairé le chemin de la connaissance.

A nos chers parents qui nous ont déployé tous les moyens moraux et matériels pour faire de nous ce que nous sommes devenu.

A nos chers enseignants pour leurs remarques qui nous ont permis de mener cette mémoire à bon terme.

A vous cher lecteur.

**Remerciements**

Avant d’entamer ce rapport, nous profitons de l’occasion pour exprimer nos profonds remerciements à toutes les personnes qui ont contribués de près ou de loin à la réalisation et l’élaboration de ce modeste rapport.

Nous tenons à remercier infiniment l’ensemble du personnel de l’Ecole Marocaine des Sciences de l’Ingénieure et spécialement Mr GHAZOUANI Mohammed notre enseignant de Programmation PHP et Frameworks, Mr AZOUAOUI Ahmed notre encadrant, Mme MOUTTALIB Houda notre enseignante de Recherche Scientifique, Mr BOURAHI Tarik Responsable du site, Mr ERRAJI Naoufal Responsable de la filière et Mme BADRITIJANE Fatimazohra chef de département pour le soutien qu’ils nous ont apporté. Nos surveillants Mr ATTAR Jalal, Mr BENHMAMA Hicham, Mr BOUTRABA Ayoub et Mr JOBAIR Morad.

Jamais nous saurons présenter ce travail sans adresser nos profonds et sincères remerciements à tous nos chers enseignants.

**Table des matières**

**Introduction générale1**

Chapitre 1 : Présentation du cadre de projet2

1. Introduction3

**2. Présentation du cadre de projet3**

3. Etude de l’existant3

**3. 1. Description de l’existant3**

**3. 2. Critique de l’existant4**

**3. 3. Solution proposée4**

**4. Choix de modèle de développement4**

**5. Planning Prévisionnel5**

**6. Conclusion6**

Chapitre 2 : Spécifications des Besoins7

1. Introduction8

2. Spécification des besoins fonctionnels8

**2. 1. Besoin fonctionnel 1 : Gestion des employés8**

**2. 2. Besoin fonctionnel 2 : Gestion des contrats9**

**2. 3. Besoin fonctionnel 3 : Gestion des demandes de congés9**

**2. 4. Besoin fonctionnel 4 : Gestion des feuilles de temps 9**

**2. 5. Besoin fonctionnel 5 : Gestion des candidatures9**

**3. Spécification des besoins non fonctionnels9**

**4. Présentation des cas d’utilisation10**

**4. 1. Présentation des acteurs10**

**4. 2. Description des cas d’utilisation10**

**4. 3. Diagramme des cas d’utilisation global12**

Chapitre 3 : Conception du système 13

1. Introduction14

2. Modélisation dynamique14

**3. 1. Diagramme de séquences14**

**3. 2. Diagramme de collaboration15**

**3. 3. Diagrammes d’états17**

**3. 2. Diagramme d’état-transition19**

**3. 3. Diagrammes d’activité23**

**3. Modélisation statique24**

**3. 1. Diagramme de classes24**

**3. 2. Modèle relationnel24**

**3. 3. Dictionnaire de données 24**

**3. 4. Architecture de l’application25**

3.4.1. Architecture logicielle25

3.4.1. Architecture matérielle27

4. Conclusion28

Chapitre 4 : Réalisation du Système29

1. Introduction30

2. Environnement de développement30

**3. 1. Environnement matériel30**

**3. 2. Environnement logiciel30**

3. Principales interfaces graphiques31

**4. Conclusion37**

**Conclusion générale38**

**Bibliographie39**

**Liste des figures**

**Figure 1 : Diagramme de cas d’utilisation11**

**Figure 2 : Diagramme de séquences14**

**Figure 3 : Diagramme de collaboration15**

**Figure 4 : Diagramme d’états17**

**Figure 5 : Diagramme d’état-transition – Partie 118**

**Figure 6 : Diagramme d’état-transition – Partie 219**

**Figure 7 : Diagramme d’état-transition – Partie 319**

**Figure 8 : Diagramme d’activité21**

**Figure 9 : Diagramme de classes 22**

**Figure 10 : Diagramme de composant25**

**Figure 11 : Diagramme de déploiement27**

**Figure 12 : Home - Partie 130**

**Figure 13 : Home - Partie 230**

**Figure 14 : Home - Partie 331**

**Figure 15 : Exemple de table Employé31**

**Figure 16 : Exemple du formulaire de modification32**

**Figure 17 : Exemple de formulaire d’ajout32**

**Figure 18 : Historique des contrats33**

**Figure 19 : Extrait de la table Candidature33**

**Figure 20 : Dashboard – Partie 134**

**Figure 21 : Dashboard – Partie 234**

**Figure 22 : Dashboard – Partie 335**

**Figure 23 : Validation de suppression35**

**Figure 24 : Dashboard – Partie 436**

**Liste des tableaux**

**Tableau 1 : Dictionnaire de données23**

**Introduction générale**

Ce rapport présente l’étude et la mise en œuvre d’un serveur Linux Fedora 41 destiné à héberger une application de gestion des ressources humaines (GRH) développée avec le framework Symfony. Ce projet s’inscrit dans un contexte où la digitalisation des processus RH est essentielle pour optimiser la gestion des employés, des candidatures, des contrats, des feuilles de temps et des demandes de congés. L’objectif est de proposer une solution intégrée, sécurisée et performante, répondant aux besoins spécifiques des services RH tout en garantissant la confidentialité et l’intégrité des données.

Le choix de Fedora 41 repose sur ses qualités de stabilité, de sécurité et sa bonne compatibilité avec les technologies PHP et Symfony. L’infrastructure serveur est pensée pour supporter l’application web et assurer la gestion des services réseau indispensables à son bon fonctionnement. Par ailleurs, l’application Symfony est conçue selon les principes de l’architecture MVC, favorisant la modularité et la maintenabilité du code, tout en offrant une interface utilisateur ergonomique accessible via un navigateur web. La problématique centrale de ce projet est donc de concevoir un système informatique cohérent, capable d’automatiser et de centraliser les processus RH, tout en respectant les contraintes techniques liées à la sécurité des données et à la gestion dynamique du réseau. Ce rapport détaille les différentes phases du projet, depuis la définition du cadre et des besoins jusqu’à la conception, la réalisation et la validation du système.

Le rapport est structuré en quatre chapitres principaux :

Le chapitre 1 présente le cadre du projet en introduisant son contexte général. Il expose les objectifs visés et explique les choix technologiques retenus pour sa réalisation.

Le chapitre 2 décrit les spécifications des besoins, en détaillant les exigences fonctionnelles et non fonctionnelles. Il précise aussi les besoins pour chaque composant de l’application.

Le chapitre 3 explique la conception de l’application, en présentant l’architecture logicielle, les modèles de données, ainsi que les choix de conception adoptés pour répondre aux besoins.

Le chapitre 4 traite de la réalisation du système, en détaillant la mise en œuvre technique, la configuration du serveur, le déploiement de l’application et les tests effectués pour valider la solution.

* Une image contenant texte, capture d’écran, Police, cercle

  Description générée automatiquementChapitre 4 : Réalisation du système : Il présente la mise en œuvre technique, la configuration du serveur, le déploiement de l’application et les tests effectués pour valider la solution.

1. Introduction

Ce premier chapitre pose les bases du projet en situant son contexte général et en justifiant la nécessité de développer une application de gestion des ressources humaines sur un serveur Linux Fedora 41. Il présente la situation actuelle, marquée par l’absence d’un système automatisé, et souligne les limites des méthodes manuelles, notamment en termes d’erreurs et de manque de centralisation. L’étude de l’existant permet ainsi de mettre en lumière les besoins précis auxquels le projet doit répondre. Ce chapitre détaille également les choix méthodologiques, notamment l’adoption d’un modèle de développement incrémental qui favorise une intégration progressive et contrôlée des composants. Enfin, il propose un planning prévisionnel réaliste qui organise les différentes étapes du projet, de l’installation du serveur à la mise en production. Cette introduction vise à fournir au lecteur une vision claire du cadre dans lequel s’inscrit le projet, des objectifs poursuivis et des contraintes à prendre en compte pour garantir son succès.

1. Présentation du cadre du projet

Ce projet s’inscrit dans le cadre du projet de fin d’année de la troisième année de la filière Ingénierie Informatique à l’École Marocaine des Sciences de l’Ingénieur (EMSI). Il constitue une étape importante dans la formation, visant à mettre en pratique les connaissances théoriques et techniques acquises au cours du cursus. L’objectif principal est de concevoir et de développer une application de gestion des ressources humaines, répondant aux besoins spécifiques d’une organisation, tout en respectant les contraintes de sécurité, de performance et de maintenabilité.

L’EMSI, reconnue pour la qualité de son enseignement et son engagement dans la formation d’ingénieurs compétents, offre un cadre propice à la réalisation de projets innovants et professionnalisants. Ce projet permet ainsi de consolider les compétences en développement logiciel, en administration système et en gestion de bases de données, tout en favorisant l’autonomie et la rigueur dans la conduite d’un projet informatique complet.

1. Etude de l’existant

3.1. Description de l’existant

Bien que l'organisation puisse disposer de certains outils ou systèmes pour la gestion des ressources humaines, ceux-ci ne sont pas intégrés et ne couvrent pas l'ensemble des processus RH de manière automatisée. Par exemple, un logiciel de paie distinct peut être utilisé, ou des feuilles de calcul peuvent être employées pour suivre les congés et les feuilles de temps. Cependant, il n'existe pas de système centralisé permettant de gérer de manière cohérente et efficace les informations des employés, les candidatures, les contrats, les demandes de congés et les feuilles de temps. Cette absence d'intégration entraîne une fragmentation des données, une complexité accrue dans la gestion des opérations RH et un manque de visibilité globale sur les ressources humaines de l'organisation. Les processus manuels restent prédominants dans de nombreux domaines, ce qui limite l'efficacité et augmente le risque d'erreurs.

**3. 2. Critique de l’existant**

Le système manuel actuel est inefficace en raison de plusieurs facteurs. Premièrement, il est sujet aux erreurs humaines lors de la saisie et du traitement des données. Deuxièmement, il est chronophage, car il nécessite une manipulation manuelle des informations et une recherche fastidieuse des données. Troisièmement, il est difficile de maintenir la cohérence et l'exactitude des données à travers différents systèmes. Quatrièmement, il manque de fonctionnalités d'analyse et de reporting, ce qui rend difficile l'identification des tendances et des problèmes potentiels. Enfin, il est peu évolutif et ne peut pas s'adapter facilement aux besoins changeants de l'organisation.

**3. 3. Solution proposée**

Pour résoudre les problèmes identifiés dans le système actuel, nous proposons de développer une application web basée sur le framework Symfony pour la gestion des ressources humaines. Cette application sera hébergée sur un serveur Linux Fedora 41 avec les services nécessaires (httpd, dhcpd, bind, mariadb). Elle permettra de centraliser toutes les informations des employés, de gérer les candidatures, les contrats, les demandes de congés et les feuilles de temps de manière efficace et intégrée. L'application offrira également des fonctionnalités d'analyse et de reporting pour aider à la prise de décision.

1. Choix de modèle de développement

Le modèle de développement choisi est une approche incrémentale, où chaque composant de l'application est développé et testé séparément avant d'être intégré au système global. Cette approche permet de diviser le projet en tâches plus petites et plus faciles à gérer, de réduire les risques et d'améliorer la qualité du produit final. Elle permet également de recueillir les commentaires des utilisateurs à chaque étape du développement, ce qui permet d'adapter l'application à leurs besoins réels.

1. Planning Prévisionnel

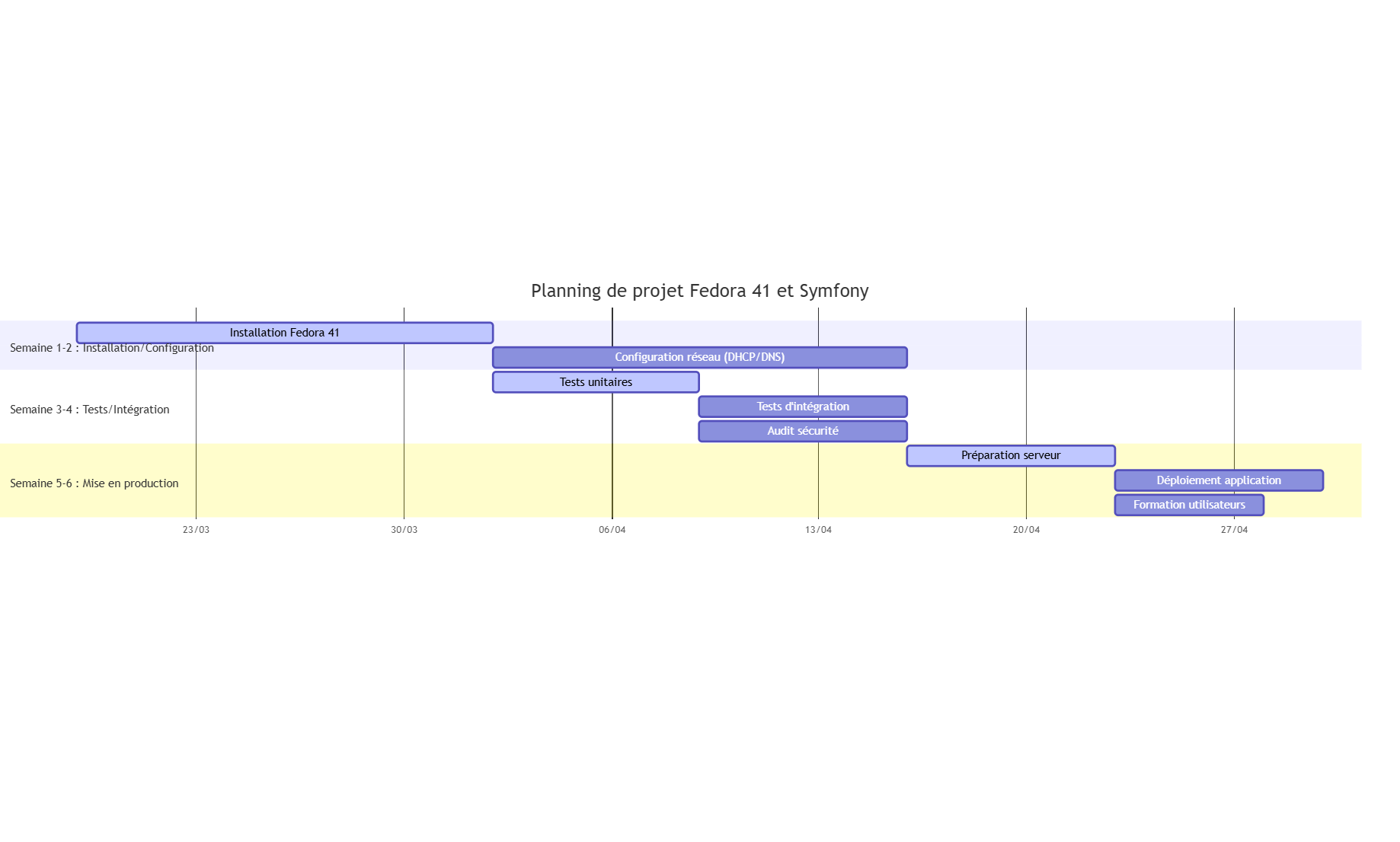


Figure : Planning Prévisionnel

1. Conclusion

Ce chapitre a permis de définir le contexte du projet, de justifier la nécessité d'une application de gestion des ressources humaines centralisée et automatisée. L'étude de l'existant a mis en évidence les limites des méthodes manuelles et l'absence d'intégration des systèmes existants. La solution proposée, basée sur Symfony et Fedora 41, répond aux besoins identifiés et s'inscrit dans une démarche de modernisation des processus RH. Le choix du modèle de développement incrémental assure une approche progressive et maîtrisée, tandis que le planning prévisionnel offre un cadre temporel réaliste pour la réalisation du projet. Les prochaines étapes consisteront à spécifier en détail les besoins fonctionnels et non fonctionnels, afin de guider la conception et le développement de l'application.



1. Introduction

Ce chapitre est consacré à la définition précise des besoins fonctionnels et non fonctionnels de l’application de gestion des ressources humaines. La phase de spécification est une étape clé qui conditionne la qualité et la pertinence du système final. Elle permet de traduire les attentes des utilisateurs en exigences claires et mesurables, assurant ainsi que le développement réponde effectivement aux besoins identifiés. Les besoins fonctionnels décrivent les fonctionnalités essentielles telles que la gestion des employés, des contrats, des demandes de congés, des feuilles de temps et des candidatures, tandis que les besoins non fonctionnels englobent des aspects cruciaux comme la sécurité, la performance et l’ergonomie. Ce chapitre présente également les cas d’utilisation, qui illustrent les interactions entre les utilisateurs et le système, facilitant la compréhension des processus métier. Cette étape garantit une base solide pour la conception technique et le développement ultérieur, en alignant les objectifs du projet avec les attentes réelles des utilisateurs.

1. Spécification des besoins fonctionnels

Chaque besoin fonctionnel a été défini en tenant compte des exigences métier et des attentes des utilisateurs finaux. Par exemple, la gestion des employés ne se limite pas à un simple enregistrement des données, mais inclut la possibilité de modifier, supprimer et visualiser les informations de manière sécurisée et intuitive. La gestion des contrats intègre des règles spécifiques liées aux types de contrats et aux périodes de validité, garantissant la conformité légale. Les demandes de congés sont conçues pour refléter les différents types de congés et leur statut, facilitant le suivi par le service RH. La gestion des feuilles de temps permet un suivi précis des heures travaillées et des heures supplémentaires, indispensable pour le calcul des salaires et la planification. Enfin, la gestion des candidatures est pensée pour optimiser le processus de recrutement, avec un suivi clair des statuts.

**2.1. Besoin fonctionnel 1 : Gestion des employés**

L'application doit permettre une gestion complète des employés. Cela inclut l'enregistrement structuré des informations des employés, telles que les informations personnelles (nom, prénom, adresse e-mail) et professionnelles (poste, date d'embauche, salaire, type de contrat). Elle doit aussi afficher ces informations de manière claire et organisée, regroupées par catégories dans un format facile à lire. De plus, l'application doit permettre de modifier les informations des employés existants de manière sécurisée, avec enregistrement des modifications, et de supprimer les enregistrements d'employés de manière sécurisée.

**2. 2. Besoin fonctionnel 2 : Gestion des contrats**

L'application doit gérer les contrats des employés de manière structurée et complète. Cela comprend la création et la gestion des contrats, avec des informations telles que le type de contrat (CDI, CDD, etc.), la date de début et de fin (si applicable) et le salaire. Elle doit afficher les contrats de manière détaillée, incluant toutes les informations pertinentes. De plus, l'application doit permettre de modifier les contrats existants de manière sécurisée, avec enregistrement des modifications, et de supprimer les contrats de manière sécurisée.

**2. 3. Besoin fonctionnel 3 : Gestion des demandes de congés**

L'application doit faciliter la gestion des demandes de congés. Elle doit permettre aux employés de soumettre des demandes de congés de manière simple et intuitive, avec des informations telles que le type de congé (congés payés, RTT, etc.), la date de début et de fin, et le statut. Elle doit afficher les demandes de congés avec leur statut (en attente, approuvée, rejetée) et les informations pertinentes. De plus, l'application doit permettre aux administrateurs de modifier le statut des demandes de congés (approuvée, rejetée).

**2. 4. Besoin fonctionnel 4 : Gestion des feuilles de temps**

L'application doit gérer les feuilles de temps des employés. Elle doit permettre aux employés d'enregistrer leurs heures travaillées et leurs heures supplémentaires de manière simple et précise, avec des informations telles que la date, les heures de début et de fin de travail, et les heures supplémentaires (si applicable). Elle doit afficher les feuilles de temps des employés avec les informations pertinentes.

**2. 5. Besoin fonctionnel 5 : Gestion des candidatures**

L'application doit gérer les candidatures. Elle doit permettre aux candidats de soumettre leurs candidatures de manière simple et intuitive, avec des informations telles que le poste souhaité, le nom, l'adresse e-mail et le statut. Elle doit afficher les candidatures avec leur statut (reçue, en cours d'examen, acceptée, rejetée) et les informations pertinentes. De plus, l'application doit permettre aux administrateurs de modifier le statut des candidatures.

**3. Spécification des besoins non fonctionnels**

Les besoins non fonctionnels sont essentiels pour assurer la qualité globale du système. La sécurité des données est primordiale, notamment pour protéger les informations personnelles des employés et respecter les réglementations en vigueur. La performance doit garantir une réactivité optimale même en cas de forte charge utilisateur. L’ergonomie de l’interface vise à offrir une expérience utilisateur fluide, accessible à différents profils, avec une compatibilité multi-navigateurs assurant l’accès sur divers équipements. La maintenabilité du système est également prise en compte pour faciliter les évolutions futures.

**4. Présentation des cas d’utilisation**

Les cas d’utilisation ont été élaborés pour représenter de manière claire et synthétique les interactions entre les utilisateurs et le système. Le Responsable RH, acteur principal, est au cœur des processus métier, intervenant dans la gestion des employés, des candidatures, des contrats, des congés et des feuilles de temps. Chaque cas d’utilisation décrit un scénario précis, avec ses préconditions, ses étapes et ses résultats attendus, facilitant la compréhension des besoins par les équipes techniques. L’inclusion des sous-cas d’utilisation permet de modéliser la complexité des tâches tout en assurant une organisation cohérente et modulaire du système.

**4. 1. Présentation des acteurs**

L'acteur principal est le Responsable RH (RH). Cet acteur interagit directement avec l'application GRH (Gestion des Ressources Humaines) pour effectuer toutes les opérations de gestion des employés, des candidatures, des contrats, des demandes de congés et des feuilles de temps. Le Responsable RH utilise également le serveur HTTPD pour accéder à l'application via un navigateur web.

**4.2. Description des cas d’utilisation**

Le Responsable RH interagit avec le système pour effectuer les tâches suivantes :

* Gérer les Candidatures : Permet d'ajouter, de modifier, de consulter et de supprimer les informations des candidatures reçues. Le Responsable RH peut également suivre l'état des candidatures (reçue, en cours d'examen, acceptée, rejetée). Ce cas d'utilisation communique avec le serveur HTTPD pour l'affichage des informations et la gestion des données.
* Gérer les Employés : Permet d'ajouter, de modifier, de consulter et de supprimer les informations des employés (nom, prénom, poste, département, date d'embauche, etc.). Ce cas d'utilisation est central car il inclut plusieurs sous-tâches. Il communique également avec le serveur HTTPD pour l'affichage des informations et la gestion des données.
* Gérer les Contrats : Permet d'ajouter, de modifier, de consulter et de supprimer les contrats de travail des employés. Ce cas d'utilisation est inclus dans "Gérer les Employés", car la gestion des contrats est une sous-tâche de la gestion des employés. Il communique également avec le serveur HTTPD pour l'affichage des informations et la gestion des données.
* Gérer les Demandes de Congés : Permet aux employés de soumettre et de modifier leurs demandes de congés. Permet aux managers de valider ou de rejeter les demandes de congés et de suivre les congés utilisés et restants pour chaque employé. Ce cas d'utilisation est inclus dans "Gérer les Employés", car la gestion des congés est une sous-tâche de la gestion des employés. Il communique également avec le serveur HTTPD pour l'affichage des informations et la gestion des données.
* Gérer Timesheet : Permet d'ajouter, de modifier, de consulter et de supprimer les heures de travail et les heures supplémentaires des employés. Ce cas d'utilisation est inclus dans "Gérer les Employés", car la gestion des feuilles de temps est une sous-tâche de la gestion des employés. Il communique également avec le serveur HTTPD pour l'affichage des informations et la gestion des données.

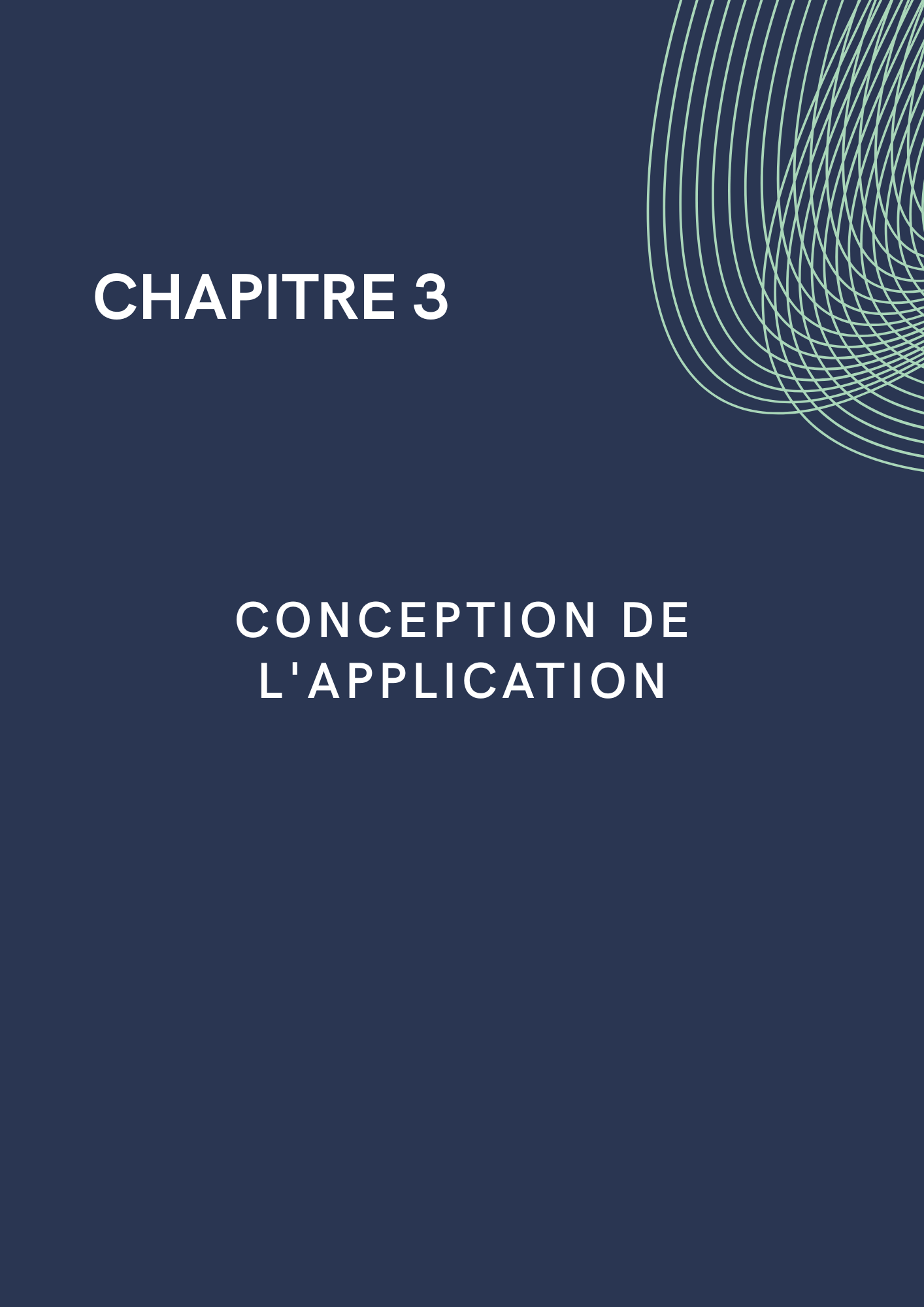
En résumé, le Responsable RH utilise l'application GRH, hébergée sur le serveur Fedora 41 et accessible via le serveur HTTPD, pour effectuer toutes les tâches de gestion des ressources humaines. Les cas d'utilisation sont organisés de manière modulaire, avec un cas d'utilisation principal ("Gérer les Employés") qui inclut plusieurs sous-tâches (gestion des contrats, des congés et des feuilles de temps).

**4. 3. Diagramme des cas d’utilisation global**

Une image contenant texte, diagramme, ligne, Tracé

Description générée automatiquement

Figure : Diagramme de cas d'utilisation



1. Introduction

La phase de conception, détaillée dans ce chapitre, constitue le pont entre la définition des besoins et la réalisation technique du système. Elle vise à structurer et modéliser le système de gestion des ressources humaines de manière à assurer cohérence, modularité et évolutivité. Ce chapitre présente les différentes modélisations dynamiques et statiques, notamment les diagrammes UML qui permettent de visualiser les interactions entre les composants, les états des objets et la structure des données. L’architecture logicielle est explicitée, mettant en avant l’utilisation du framework Symfony selon le modèle MVC, garantissant une séparation claire des responsabilités et facilitant la maintenance. L’architecture matérielle est également décrite, soulignant le rôle du serveur Fedora 41 et des services associés (httpd, mariadb, bind, dhcpd) dans le déploiement de l’application. Cette conception détaillée est essentielle pour anticiper les défis techniques, optimiser les performances et assurer la sécurité du système, tout en répondant aux besoins fonctionnels spécifiés.

2. Modélisation dynamique

Les diagrammes dynamiques jouent un rôle fondamental dans la compréhension des comportements du système au fil du temps. Les diagrammes de séquences détaillent l’enchaînement précis des interactions entre objets, mettant en lumière les flux d’informations et les appels de méthodes. Les diagrammes de collaboration illustrent la coopération entre objets pour atteindre un objectif commun, soulignant la répartition des responsabilités. Les diagrammes d’états et d’état-transition permettent de modéliser les cycles de vie des entités, essentiels pour gérer les statuts des demandes de congés ou des candidatures. Enfin, les diagrammes d’activités décrivent les processus métier de manière visuelle, facilitant la validation des workflows auprès des utilisateurs.

**3. 1. Diagramme de séquences**

Il met en évidence la séquence des interactions entre l'Utilisateur et le Système pour un cas d'utilisation spécifique, comme l'inscription à une mission. Il illustre les messages échangés entre l'Utilisateur et le Système, ainsi que les opérations réalisées par le Système pour répondre à la requête de l'Utilisateur.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementFigure 3 : Diagramme de séquence

Ce diagramme de séquence illustre les échanges entre le Responsable RH, le serveur HTTPD, l’application Symfony et la base de données MariaDB lors de l’utilisation de l’application de gestion des ressources humaines. Le Responsable RH accède d’abord à l’application via le serveur HTTPD, qui transmet la requête à Symfony. L’application interroge alors la base de données pour récupérer les informations des employés, puis affiche les formulaires nécessaires à l’utilisateur. Une fois les demandes saisies et soumises par le Responsable RH, celles-ci sont traitées par Symfony, enregistrées dans la base de données, puis une confirmation est renvoyée à l’utilisateur via le serveur HTTPD. Ce processus garantit une interaction fluide et sécurisée entre les différents composants du système.

**3. 2. Diagramme de collaboration**

Les diagrammes de collaboration montrent les interactions entre les objets du système, en mettant l'accent sur les relations entre les objets et les rôles qu'ils jouent dans les interactions. Ils permettent de comprendre comment les différents composants (par exemple, le contrôleur, le modèle et la vue) travaillent ensemble pour réaliser une tâche.

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

Figure : Diagramme de collaboration

Ce diagramme de collaboration présente les principales classes et leurs interactions au sein du système de gestion des ressources humaines. Le Responsable RH interagit avec le serveur HTTPD, qui sert d’interface entre l’utilisateur et l’application. Le serveur HTTPD communique avec l’application Symfony, responsable du traitement des requêtes, de la récupération, de la mise à jour et de l’enregistrement des données. Symfony interagit directement avec la base de données MariaDB pour obtenir ou stocker les informations nécessaires. Par ailleurs, Symfony gère les différentes entités métier telles que les Employés, les Candidatures, les Contrats, les Demandes de congés et les Feuilles de temps, en appelant leurs méthodes spécifiques pour manipuler les données. Ce modèle met en évidence la structure modulaire du système, où chaque composant a un rôle clair, facilitant ainsi la maintenance et l’évolution de l’application.

**3. 3. Diagrammes d’états**

Les diagrammes d'états décrivent les différents états possibles d'un objet et les transitions entre ces états en réponse à des événements. Ils sont particulièrement utiles pour modéliser le cycle de vie d'une demande de congé (en attente, approuvée, rejetée) ou l'état d'une candidature (reçue, en cours d'examen, acceptée, rejetée).

Figure : Diagramme d'états

Une image contenant texte, ligne, capture d’écran, diagramme

Description générée automatiquement

Ce diagramme d’états présente les différents statuts et transitions des trois principales entités du système : les demandes de congés, les candidatures et les contrats. Pour les demandes de congés, le cycle débute à l’état « En attente », où la demande attend l’approbation du Responsable RH. Selon la décision, la demande peut être approuvée ou rejetée, avec la possibilité de revenir en « En attente » pour réouverture ou réexamen. Les candidatures suivent un chemin similaire, passant de l’état « Soumise » à « En cours d’examen », puis à un état final d’« Approuvée » ou de « Rejetée », avec la possibilité de réexamen à partir des états finaux. Enfin, les contrats évoluent à travers les états « Proposé », « Signé », « En cours » et « Terminé », reflétant les phases du cycle de vie contractuel, de la proposition initiale à la fin ou à la résiliation du contrat. Ce diagramme met en lumière la gestion dynamique des processus métiers et les règles associées à chaque étape.

**3. 2. Diagramme d’état-transition**

Il modélise le comportement d'un objet (par exemple, un utilisateur) en décrivant les différents états possibles de cet objet et les transitions entre ces états. Il permet de visualiser les différentes étapes par lesquelles passe un objet au cours de son cycle de vie et les événements qui déclenchent ces transitions.

Figure 6 : Diagramme d'états-transition – Partie 1

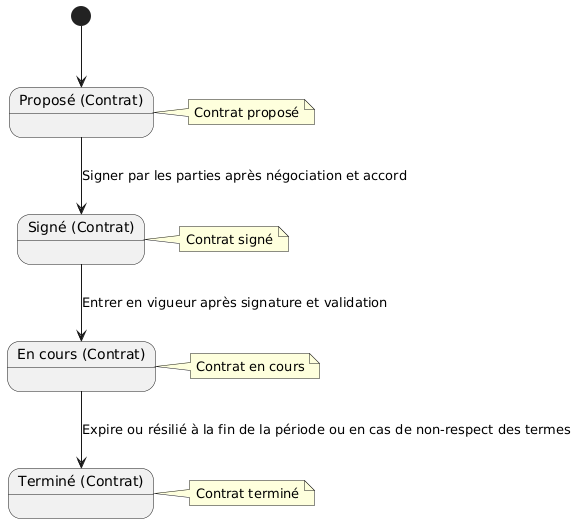
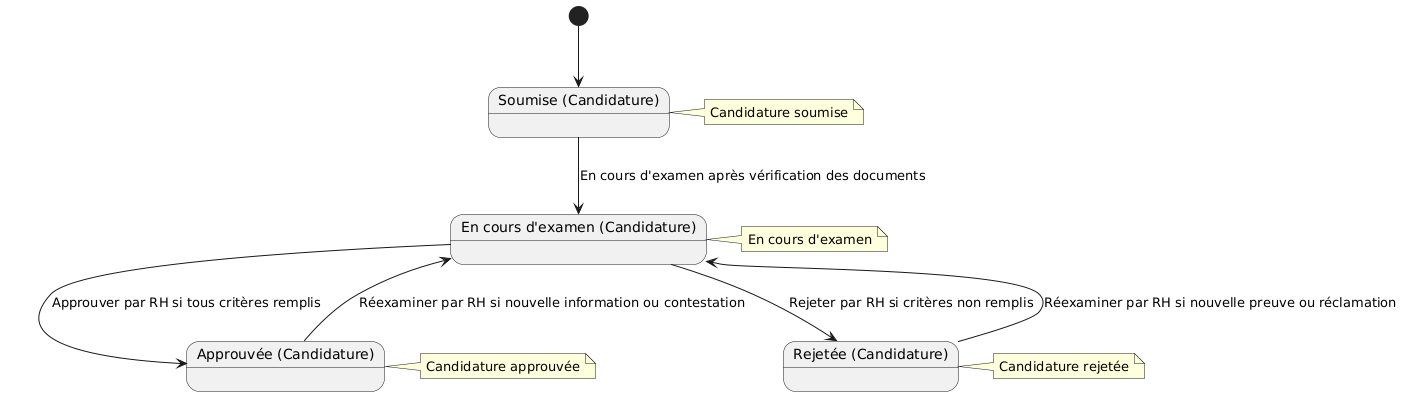
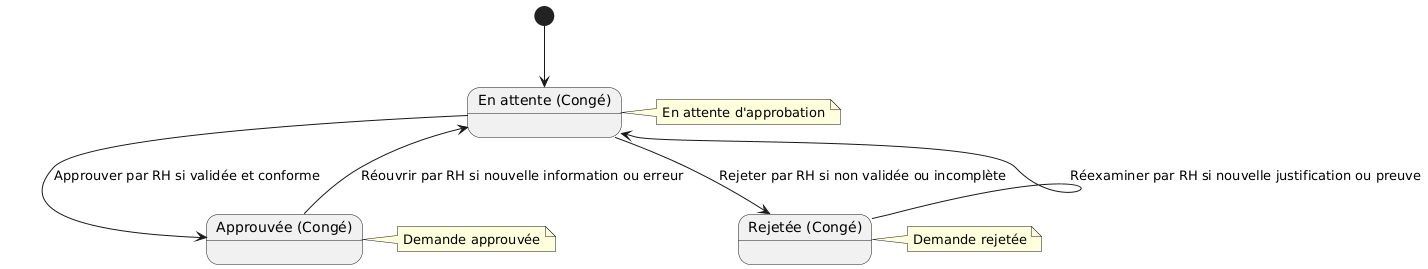


Figure : Diagramme d'états-transition – Partie 3

Figure : Diagramme d'états-transition – Partie 2





Ce diagramme d’état-transition illustre les différents états et transitions des principales entités du système, à savoir les demandes de congés, les candidatures et les contrats. Pour les demandes de congés, le cycle commence par un état « En attente », où la demande attend l’approbation du Responsable RH. Selon la décision, la demande peut passer à l’état « Approuvée » ou « Rejetée », avec la possibilité de revenir en « En attente » en cas de réexamen. De même, les candidatures évoluent depuis l’état « Soumise » vers « En cours d’examen », puis vers un état final d’« Approuvée » ou de « Rejetée », avec des transitions possibles pour réexaminer la candidature si de nouvelles informations apparaissent. Enfin, les contrats passent par les états « Proposé », « Signé », « En cours » et « Terminé », reflétant les différentes phases du cycle de vie contractuel, depuis la proposition initiale jusqu’à la fin ou la résiliation du contrat. Ce diagramme met en lumière la dynamique des processus métier et les règles de gestion associées à chaque entité.

**3. 3. Diagrammes d’activité**

 Il décrit le flux d'activité pour un processus spécifique, comme l'inscription d'un bénévole à une mission. Il illustre les différentes étapes du processus, les décisions à prendre et les actions à réaliser, en mettant en évidence le déroulement des opérations et les alternatives possibles.

Figure 9 : Diagramme d'activité

Une image contenant texte, diagramme, reçu

Description générée automatiquement

Ce diagramme d’activité décrit le flux des actions possibles au sein de l’application de gestion des ressources humaines. L’utilisateur commence par accéder à l’application, puis choisit la gestion des candidatures ou des employés. S’il opte pour la gestion des candidatures, il peut afficher la liste des candidatures et décider d’approuver ou de rejeter une candidature, ce qui entraîne la mise à jour de son statut. Dans le cas où l’utilisateur choisit de gérer les employés, il peut consulter leurs informations, puis, selon les besoins, créer ou modifier des contrats, approuver ou rejeter des demandes de congés, ou encore consulter les feuilles de temps. Ce diagramme met en évidence la diversité des fonctionnalités offertes par l’application et la manière dont elles s’enchaînent selon les choix de l’utilisateur.

**3. Modélisation statique**

La modélisation statique, à travers le diagramme de classes, offre une représentation claire des entités du système et de leurs relations, ce qui est crucial pour la conception de la base de données et la structuration du code. Le modèle relationnel traduit cette conception en tables et clés étrangères, garantissant l’intégrité des données. Le dictionnaire de données précise les types et contraintes des champs, assurant une cohérence des données. L’architecture logicielle basée sur Symfony MVC garantit une séparation des préoccupations, facilitant le développement, la maintenance et l’extensibilité. L’architecture matérielle est pensée pour optimiser la disponibilité, la sécurité et la performance du serveur, en intégrant des services réseau indispensables au bon fonctionnement de l’application.

**3. 1. Diagramme de classes**

Le diagramme de classes montre les classes du système, leurs attributs, leurs méthodes et les relations entre les classes (héritage, association, agrégation, composition). Il permet de visualiser la structure des données et les relations entre les entités (Employé, Contrat, Candidature, etc.).

Figure 10 : Diagramme de classes

Une image contenant texte, capture d’écran, Police, diagramme

Description générée automatiquement

Ce diagramme de classes présente les principales entités du système de gestion des ressources humaines ainsi que leurs attributs et relations. La classe Candidature représente les candidatures des postulants, avec des informations telles que le poste souhaité, le nom du candidat, son e-mail et le statut de la candidature. Un candidat peut potentiellement devenir un Employé, représenté par la classe correspondante, qui contient les données personnelles et professionnelles, notamment le nom, prénom, poste, salaire, date d’embauche et une référence vers son contrat. La classe Contrat détaille les informations relatives au type de contrat, ses dates de début et de fin, ainsi que le salaire associé. Les employés peuvent également soumettre plusieurs Demandes de congés, chacune caractérisée par le type de congé, les dates concernées et le statut de la demande. Enfin, la classe Timesheet permet d’enregistrer les heures travaillées et les heures supplémentaires des employés sur une date donnée. Les relations entre ces classes reflètent la dynamique du système : un employé possède un contrat, peut effectuer plusieurs demandes de congés et enregistrer plusieurs feuilles de temps, tandis qu’une candidature peut évoluer vers un employé.

**3. 2. Modèle relationnel**

Le modèle relationnel décrit la structure de la base de données MariaDB, en spécifiant les tables, les colonnes, les types de données (VARCHAR), les clés primaires et les clés étrangères. Conformément au cahier des charges, le schéma de base de données se nomme "grh" et contient les tables suivantes :

* Employes (id, nom, prénom, email, poste, salaire, date\_embauche, contrat\_id)
* Candidatures (id, nom, email, poste\_souhaite, etat\_candidature)
* Contrats (id, type, salaire, date\_debut, date\_fin, employe\_id)
* FeuillesDeTemps (id, employe\_id, date, heures\_travaillees, heures\_supplementaires)
* DemandesConge (id, employe\_id, type\_conge, date\_debut, date\_fin, statut)

**3. 3. Dictionnaire de données**

Tableau 1 : Dictionnaire de données

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Champ | Type | Description |
| Employes | **id** | VARCHAR(255) | Identifiant unique de l'employé (Clé primaire) |
|  | **nom** | VARCHAR(255) | Nom de l'employé |
|  | **prénom** | VARCHAR(255) | Prénom de l'employé |
|  | **poste** | VARCHAR(255) | Poste de l'employé |
|  | **email** | VARCHAR(255) | Email de l’employé |
|  | **salaire** | VARCHAR(255) | Salaire de l'employé |
|  | **date\_embauche** | VARCHAR(255) | Date d'embauche de l'employé |
|  | **Contrat\_id** | VARCHAR(255) | Identifiant du contrat é lié à l’employé (Clé étrangère vers Contrat) |
| Candidatures | **id** | VARCHAR(255) | Identifiant unique de la candidature (Clé primaire) |
|  | **nom** | VARCHAR(255) | Nom du candidat |
|  | **email** | VARCHAR(255) | Email du candidat |
|  | **poste\_souhaite** | VARCHAR(255) | Poste souhaité par le candidat |
|  | **etat\_candidature** | VARCHAR(255) | État de la candidature (reçue, en cours, acceptée, rejetée) |
| Contrats | **id** | VARCHAR(255) | Identifiant unique du contrat (Clé primaire) |
|  | **type** | VARCHAR(255) | Type de contrat (CDI, CDD, etc.) |
|  | **salaire** |  | Salaire de l’employé |
|  | **date\_debut** | VARCHAR(255) | Date de début du contrat |
|  | **date\_fin** | VARCHAR(255) | Date de fin du contrat (si applicable) |
|  | **employe\_id** | VARCHAR(255) | Identifiant de l'employé lié au contrat (Clé étrangère vers Employes) |
| FeuillesDeTemps | **id** | VARCHAR(255) | Identifiant unique de la feuille de temps (Clé primaire) |
|  | **employe\_id** | VARCHAR(255) | Identifiant de l'employé lié à la feuille de temps (Clé étrangère vers Employes) |
|  | **date** | VARCHAR(255) | Date de la feuille de temps |
|  | **heures\_travaillees** | VARCHAR(255) | Nombre d'heures travaillées |
|  | **heures\_supplementaires** | VARCHAR(255) | Nombre d'heures supplémentaires |
| DemandesConge | **id** | VARCHAR(255) | Identifiant unique de la demande de congé (Clé primaire) |
|  | **employe\_id** | VARCHAR(255) | Identifiant de l'employé lié à la demande de congé (Clé étrangère vers Employes) |
|  | **type\_conge** | VARCHAR(255) | Type de congé (congés payés, RTT, etc.) |
|  | **date\_debut** | VARCHAR(255) | Date de début du congé |
|  | **date\_fin** | VARCHAR(255) | Date de fin du congé |
|  | **etat\_demande** | VARCHAR(255) | État de la demande (en attente, approuvée, rejetée) |

**3. 4. Architecture de l’application**

3.4.1. Architecture logicielle

L'application est développée en utilisant le framework Symfony, qui suit une architecture Modèle-Vue-Contrôleur (MVC). Le Modèle gère les données et la logique métier, la Vue est responsable de l'interface utilisateur, et le Contrôleur gère les interactions entre le Modèle et la Vue. Composer est utilisé pour gérer les dépendances de l'application. Symfony Security est utilisé pour gérer l'authentification et l'autorisation des utilisateurs.

Figure : Diagramme de Composant

Une image contenant texte, ligne, diagramme, Tracé

Description générée automatiquement

Ce diagramme de composants présente l’architecture modulaire de l’application de gestion des ressources humaines développée avec Symfony. Il met en évidence les différents modules fonctionnels de l’application, tels que la gestion des candidatures, des employés, des contrats, des congés et des feuilles de temps, regroupés au sein du package « Application Symfony ». Ces modules communiquent avec la base de données MariaDB pour stocker et récupérer les données nécessaires. Le serveur web HTTPD joue le rôle d’interface entre les utilisateurs et les modules applicatifs, recevant les requêtes et les transmettant aux composants appropriés. Par ailleurs, les serveurs DNS (BIND) et DHCP (DHCPD) assurent respectivement la résolution des noms de domaine et la gestion dynamique des adresses IP, facilitant ainsi l’accès et la connectivité au sein du réseau. Ce diagramme illustre clairement la séparation des responsabilités entre les composants logiciels et les services d’infrastructure, garantissant une architecture cohérente, évolutive et maintenable.

3.4.1. Architecture matérielle

L'architecture matérielle est basée sur un ordinateur Linux Fedora 41, hébergeant les services nécessaires au fonctionnement de l'application, conformément au cahier des charges :

* Serveur : Un ordinateur Linux Fedora 41, avec une configuration matérielle adaptée à la charge de l'application (Processeur 2 GHz x86-64 minimum, 8 Go de RAM recommandés, 50 Go de stockage minimum).
* Système d'exploitation : Fedora 41 Server, choisi pour sa stabilité, sa sécurité et sa compatibilité avec les technologies utilisées.
* Serveur Web (HTTPD) : Apache HTTP Server, configuré pour servir les fichiers statiques et les requêtes dynamiques. Configuration du VirtualHost pour Symfony dans /etc/httpd/conf/httpd.conf.
* Serveur de base de données (MariaDB) : MariaDB, utilisé pour stocker les données de l'application (schéma "grh" contenant les tables Employes, Candidatures, Contrats, FeuillesDeTemps, DemandesConge).
* Gestion DHCP : ISC DHCP Server pour l'attribution dynamique d'adresses IP aux clients. Configuration manuelle des plages IP dans /etc/dhcp/dhcpd.conf.
* Gestion DNS : Bind pour la gestion des résolutions de noms. Création des zones DNS dans /etc/named.conf.

Une image contenant texte, capture d’écran, ligne, Police

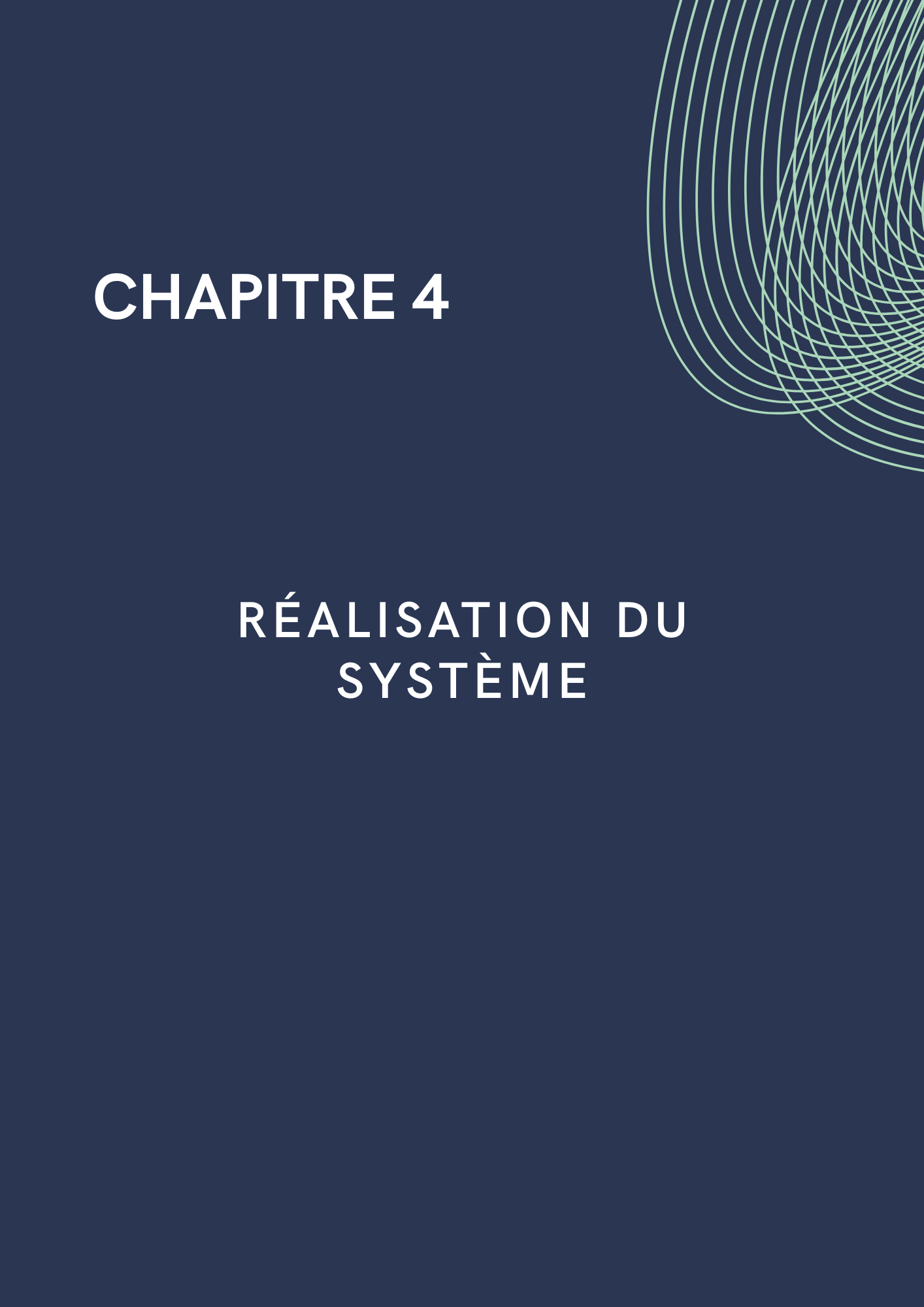
Description générée automatiquementL'architecture matérielle est conçue pour assurer la disponibilité, la sécurité et la performance du système, tout en respectant les contraintes techniques mentionnées dans le cahier des charges. L'attribution dynamique des adresses IP par le serveur DHCP et la résolution des noms de domaine par le serveur Bind sont des éléments clés de l'infrastructure réseau.

Figure : Diagramme de Déploiement

Ce diagramme de déploiement UML représente l’architecture physique du système de gestion des ressources humaines. Il montre les nœuds matériels principaux, à savoir le serveur Fedora 41 et le client utilisateur, ainsi que le réseau local qui les interconnecte. Sur le serveur Fedora 41 sont déployés plusieurs artefacts essentiels : l’application Symfony, le serveur web HTTPD, les serveurs DNS (BIND) et DHCP (DHCPD), ainsi que la base de données MariaDB. Le client utilisateur accède à l’application via un navigateur web. Ce schéma met en évidence la répartition des composants logiciels sur les infrastructures matérielles et réseau, illustrant ainsi la configuration technique nécessaire pour assurer le fonctionnement sécurisé et performant de l’application. Il reflète également la communication entre les différents nœuds, essentielle pour le bon déroulement des échanges entre l’utilisateur et le système.

**4. Conclusion**

Ce chapitre a présenté la conception architecturale du système de gestion des ressources humaines, en traduisant les besoins spécifiés au chapitre précédent en une solution technique concrète. La modélisation dynamique et statique a permis de définir les interactions entre les composants, les flux de données et la structure de la base de données MariaDB. L'architecture logicielle, basée sur le framework Symfony et le modèle MVC, assure une séparation claire des responsabilités et une maintenabilité accrue. L'architecture matérielle, basée sur un serveur Fedora 41 et les services essentiels (httpd, dhcpd, bind), garantit la disponibilité, la sécurité et la performance du système. Les diagrammes UML et les schémas de base de données ont fourni une vision claire et précise de la structure du système. La prochaine étape consistera à mettre en œuvre cette conception en développant les composants logiciels, en configurant les services réseau et en créant les interfaces utilisateur.



1. **Introduction**

Ce chapitre décrit la concrétisation du projet à travers le développement et le déploiement effectifs de l’application de gestion des ressources humaines. Il présente l’environnement matériel et logiciel utilisé, justifiant les choix techniques en fonction des contraintes et des besoins du projet. Le matériel, bien que modeste, est adapté pour un environnement de test et de développement, tandis que l’environnement logiciel intègre les technologies clés telles que Fedora 41, Apache httpd, MariaDB, PHP-FPM et Symfony. Ce chapitre met également en lumière les principales interfaces graphiques développées, qui constituent le point de contact entre les utilisateurs et le système. Ces interfaces ont été conçues pour offrir une expérience utilisateur intuitive, facilitant la gestion des employés, des candidatures, des feuilles de temps et des autres fonctionnalités. La réalisation s’inscrit dans une démarche rigoureuse visant à assurer la conformité aux spécifications, la robustesse du système et la satisfaction des utilisateurs finaux.

2. Environnement de développement

**3. 1. Environnement matériel**

Le choix d’un processeur Intel Celeron à 1.19 GHz et d’une mémoire vive de 4 Go reflète un compromis entre coût et performance, adapté à un environnement de développement et de test. Bien que modeste, cette configuration permet d’exécuter simultanément le serveur Fedora, les services réseau et l’application Symfony avec une performance acceptable. L’espace disque de 450 Go offre une marge confortable pour stocker le système, les bases de données, les fichiers de configuration et les logs, tout en permettant des sauvegardes régulières.

**3. 2. Environnement logiciel**

La sélection des composants logiciels repose sur leur stabilité, leur compatibilité et leur communauté active. Fedora 41 assure un socle sécurisé et à jour. Apache httpd, couplé à PHP-FPM, permet une gestion efficace des requêtes web dynamiques. MariaDB offre une base de données relationnelle robuste et performante. Symfony, avec Composer, facilite le développement modulaire et la gestion des dépendances. L’utilisation d’outils comme VSCode et Git garantit un environnement de travail productif et collaboratif.

3. Principales interfaces graphiques

Les interfaces graphiques ont été conçues pour offrir une expérience utilisateur intuitive et fonctionnelle. Le tableau de bord synthétise les informations clés, permettant au Responsable RH d’avoir une vision globale immédiate. L’interface de gestion des employés centralise toutes les données nécessaires, avec des fonctionnalités de création, modification et suppression, tout en assurant la cohérence avec les contrats et demandes de congés. L’interface de gestion des candidatures facilite le suivi du processus de recrutement, avec des filtres et statuts clairs. L’interface des feuilles de temps permet un enregistrement précis des heures, indispensable pour la gestion administrative et la paie. Ces interfaces ont été développées avec une attention particulière à l’ergonomie et à la sécurité.

Figure : Home - Partie 1

Une image contenant texte, capture d’écran, Police, Page web

Le contenu généré par l’IA peut être incorrect.

Figure : Home - Partie 2

Une image contenant texte, capture d’écran, logiciel, Police

Le contenu généré par l’IA peut être incorrect.

Figure : Home - Partie 3

Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.

Figure : Exemple de table Employé

Une image contenant texte, capture d’écran, logiciel, nombre

Le contenu généré par l’IA peut être incorrect.

Figure : Exemple du formulaire de modification

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Le contenu généré par l’IA peut être incorrect.

Figure : Exemple de formulaire d'ajout

Une image contenant texte, capture d’écran, logiciel, Police

Le contenu généré par l’IA peut être incorrect.

Figure : Historique des contrats

Une image contenant texte, capture d’écran, logiciel, Page web

Le contenu généré par l’IA peut être incorrect.

Figure : Extrait de la table Candidature

Une image contenant texte, capture d’écran, logiciel, Page web

Le contenu généré par l’IA peut être incorrect.

Figure : Dashboard - Partie 1

Une image contenant texte, capture d’écran, nombre, logiciel

Le contenu généré par l’IA peut être incorrect.

Figure : Dashboard - Partie 2

Une image contenant texte, capture d’écran, diagramme, Police

Le contenu généré par l’IA peut être incorrect.

Figure : Dashboard - Partie 3

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Le contenu généré par l’IA peut être incorrect.

Figure : Validation de suppression

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Le contenu généré par l’IA peut être incorrect.

Figure : Dashboard - Partie 4

Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.

**4. Conclusion**

Ce chapitre a décrit la phase de réalisation du système de gestion des ressources humaines, en détaillant l'environnement de développement, les outils utilisés et les principales interfaces graphiques. L'environnement matériel, bien que modeste, a permis de mener à bien les tests et le développement de l'application. L'environnement logiciel, basé sur Fedora 41, Apache, MariaDB et Symfony, a fourni une base solide pour la mise en œuvre des fonctionnalités. Les interfaces graphiques ont été conçues pour offrir une expérience utilisateur intuitive et conviviale, en facilitant l'accès aux données et aux fonctionnalités essentielles. Les prochaines étapes consisteront à effectuer des tests d'intégration, à valider le bon fonctionnement du système et à préparer la mise en production.

**Conclusion générale**

L'objectif principal de ce PFA était de concevoir et de déployer une application de gestion des ressources humaines (GRH) sur un serveur Linux Fedora 41. Ce projet visait à moderniser et à centraliser la gestion des informations relatives aux employés, aux contrats, aux demandes de congés, aux feuilles de temps et aux candidatures, offrant ainsi une solution intégrée et efficace pour le Responsable RH. Les problèmes posés au début du projet concernaient l'absence d'un système automatisé, la difficulté à gérer manuellement les données et le manque de visibilité sur les informations clés. Grâce à la réalisation de ce projet, nous avons pu apporter des réponses concrètes à ces problèmes. L'application GRH, développée avec le framework Symfony et hébergée sur un serveur Fedora 41 configuré avec les services HTTPD, DHCPD, BIND et MariaDB, permet désormais une gestion centralisée et efficace des ressources humaines. Le Responsable RH peut gérer les employés, les contrats, les candidatures et les feuilles de temps, et suivre l'état des demandes de congés. Les résultats obtenus démontrent la faisabilité et la pertinence de la solution proposée. Cependant, il est important de noter certaines insuffisances et améliorations possibles. L'interface utilisateur pourrait être améliorée pour offrir une expérience plus intuitive et conviviale. De plus, l'application pourrait être enrichie de fonctionnalités supplémentaires, telles que la génération de rapports personnalisés, l'intégration avec d'autres systèmes (paie, comptabilité) et la gestion des formations. Enfin, il serait intéressant d'optimiser les performances de l'application pour garantir une réactivité optimale, même en cas de forte charge. En conclusion, ce projet a permis d'atteindre les objectifs fixés et de développer une solution viable et prometteuse pour la gestion des ressources humaines, tout en identifiant des pistes d'amélioration pour les développements futurs.

**Bibliographie**

BASS, Len, Paul CLEMENTS et Rick KAZMAN, Software Architecture in Practice, 3e éd., Boston, Addison-Wesley, 2012.

Fedora Project, Fedora 41 Release Notes [en ligne], 2024, disponible sur : https://docs.fedoraproject.org/en-US/fedora/f41/release-notes/.

FOWLER, Martin, Patterns of Enterprise Application Architecture, Boston, Addison-Wesley, 2003.

MariaDB Foundation, MariaDB Knowledge Base [en ligne], 2024, disponible sur : https://mariadb.com/kb/en/.

OWASP Foundation, OWASP Top Ten Web Application Security Risks [en ligne], 2024, disponible sur : https://owasp.org/www-project-top-ten/ (consulté le 10 avril 2025).

PHP Group, PHP Manual [en ligne], 2024, disponible sur : https://www.php.net/manual/en/ (consulté le 14 avril 2025).

POTENCIER, Fabien, Symfony 5: The Fast Track, Paris, Editions ENI, 2021.

Red Hat, SELinux User Guide [en ligne], 2024, disponible sur : https://access.redhat.com/documentation/en-us/red\_hat\_enterprise\_linux/8/html/security\_hardening/using-selinux-to-control-access\_security-hardening (consulté le 21 mars 2025).

Symfony, Symfony Documentation [en ligne], 2024, disponible sur : https://symfony.com/doc/current/index.html (consulté le 25 mars 2025).

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, ISO 690:2010 – Information and documentation – Guidelines for bibliographic references and citations to information resources, Genève, ISO, 2010.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, ISO/IEC 27001:2013 – Information technology – Security techniques – Information security management systems – Requirements, Genève, ISO, 2013.

Stack Overflow, Symfony Tag [en ligne], 2024, disponible sur : https://stackoverflow.com/questions/tagged/symfony (consulté le 30 avril 2025).

DigitalOcean, How To Install and Secure MariaDB on Fedora 41 [en ligne], 2024, disponible sur : https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-fedora-41 (consulté le 19 mars 2025).