

REPUBLIQUE DU SENEGAL
UNIVERSITE CHEIKH ANTA DIOP DE DAKAR



ECOLE SUPERIEURE POLYTECHNIQUE
DEPARTEMENT GENIE INFORMATIQUE

MEMOIRE DE FIN DE CYCLE

Pour l'obtention du :
DIPLOME DE MASTER EN GENIE LOGICIEL ET SYSTEME D'INFORMATION

SUJET :

**Conception et Réalisation d'une Application de
Gestion des Ressources Humaines.**

Lieu de stage : Kya's

Période stage : 03/2018 – 08/2018

*Présenté et soutenu par
Mouhamed DIALLO*

*Professeurs encadreur
Pr. Alassane Bah*

*Maitres de stage
M. Idrissa DIAGNE
M. Daouda NGOM*

REPUBLIQUE DU SENEGAL
UNIVERSITE CHEIKH ANTA DIOP DE DAKAR



ECOLE SUPERIEURE POLYTECHNIQUE
DEPARTEMENT GENIE INFORMATIQUE

MEMOIRE DE FIN DE CYCLE

Pour l'obtention du :
DIPLOME DE MASTER EN GENIE LOGICIEL ET SYSTEME D'INFORMATION

SUJET :

**Conception et Réalisation d'une Application de
Gestion des Ressources Humaines.**

Lieu de stage : Kya's

Période stage : 03/2017 – 08/2018

*Présenté et soutenu par
Mouhamed DIALLO*

*Professeurs encadrateurs
Pr. Alassane Bah*

*Maitres de stage
M. Idrissa DIAGNE
M. Daouda NGOM*

Je dédie ce mémoire :

*A ma chère mère et complice qui a toujours été présente, **Maman Khadizatou Fall** je t'adore.*

*A mon défunt père **Alioune Diallo** et ami pour avoir toujours cru en moi.*

*A mon grand frère **Cheikhou Diallo** pour ses conseils et son soutien,*

*A mes grandes sœurs **Victorine** et **Maimouna Diallo** pour leurs conseils et leur soutien,*

*A mon oncle **Tonton Makhtar Fall** pour son affection et ses précieux conseils*

A toute ma famille : mes tantes, mes oncles, mes cousins et cousines,

A tous mes amis,

A tous les étudiants des promotions 2014, 2015, 2016,2017,2018.



Al Hamdoulah, je rends grâce à Allah.

Je profite par le biais de ce rapport pour adresser mes vifs remerciements à toutes les personnes ayant contribué de près ou de loin à la réalisation de cet humble travail.

Je remercie spécialement ma famille pour leur soutien matériel et moral qu'elle m'a apporté, en particulier ma mère pour son affection, son intérêt, son investissement, sa motivation et sa disponibilité sans faille et mon frère pour sa confiance et ses conseils.

Je tiens à remercier vivement tout le personnel du département Génie Informatique particulièrement tous les enseignants qui se sont investis pour la réussite de ma formation.

*Mes remerciements vont à notre encadreur **Pr Alassane Bah** pour sa disponibilité ses explications et sa contribution à la réalisation de ce modeste projet.*

*Je remercie également **Mr Mamadou Camara** notre responsable pédagogique pour son soutien, ses conseils et l'intérêt qu'il porte à ses étudiants.*

***M. Diop** Directeur Générale de Kya's, **M. Diagne** Directeur Technique de kya's du, mon maitre de stage **M. Daouda Ngom** ainsi que tout le reste du personnel de Kya's.*

Merci à vous tous pour l'encadrement dispensé avec beaucoup de gentillesse et d'humour.

Ce mémoire de fin d'études est l'aboutissement de mon parcours dans l'enseignement supérieur. Il a été rédigé afin de remplir les exigences d'obtention du diplôme de Master à l'Ecole Supérieure Polytechnique. Je présente ce mémoire afin d'obtenir mon Master 2 en Génie Logiciel et Système d'Information.

Le projet a été entrepris à la demande de Kya's., société au sein de laquelle j'ai effectué mon stage. L'entreprise désire avoir une meilleure gestion de son personnel et c'est ainsi que ce sujet de stage m'a été attribué.

Table des matières

AVANT-PROPOS	3
Table des matières	4
Sigles et abréviations	6
Table des figures	7
INTRODUCTION	8
Chapitre1 : Présentation générale	9
1. Présentation de la Structure d'accueil	9
1.1 Présentation de la société kya's	9
1.2 Domaines d'activités	10
2. Contexte du projet	10
2.1 Présentation du cadre d'application du projet	10
2.2 Présentation de l'existant	11
2.2.1 Description de l'existant.....	11
2.2.2 Critique de l'existant	11
2.3 Problématique.....	11
3. Objectifs	12
Chapitre 2 : Méthodologie de développement logiciel	13
1. Choix du processus de développement logiciel	13
1.1 Processus agile vs dirigé par planification.....	13
1.2 Méthodes agiles.....	14
2. Choix d'une méthode d'analyse et de conception.....	17
2.1 UML (Unified Modeling Language).....	17
Utilisation	17
Chapitre 3 : Spécifications fonctionnelles détaillées	20
1. Choix du processus de développement logiciel	20
2. Présentation du domaine	20
3. Capture des Besoins fonctionnels	22
3.1 Diagramme de Cas d'utilisation Gestion du Compte de l'Employé	22
3.2 Fiche de description textuelle et Diagramme d'activité du Cas Création de Compte	
23	
3.2.1 Fiche de Description textuelle du Cas Créer Compte	23
3.2.2 Diagramme d'activité	23
3.3 Diagramme de Cas d'utilisation	24
3.3.1 Raffinement des Cas d'utilisations.....	25
3.3.2 Fiches de description textuelle et Diagrammes d'activités	26
Description des associations :	30
Chapitre4 : Conception de la plateforme et Choix des outils et technologies.....	48
1. Solution technique.....	48
2. Architecture de la plateforme	49
2.1 Différentes couches du 3-tiers.....	49
2.1.1 Présentation des données.....	49
2.1.2 Logique métier	50
2.1.3 Accès aux données	50
2.2 Justification du choix.....	50

3. Les plateformes de développement d'application web	50
3.1 Les différents types d'architecture système	51
3.1.1 Architecture à deux niveaux	51
3.1.2 Architecture à trois niveaux	51
3.1.3 Architecture multi niveaux	51
4. Quelle plateforme de développement pour notre application ?	52
4.1 Introduction à la plateforme Symfony	52
4.1.1 Historique	53
4.1.2 Utilisation	53
4.1.2.1 Le langage PHP	53
4.1.3 Présentation de quelques composants	54
4.1.3.1 HttpFoundation et HttpKernel	54
4.1.3.2 Routing	54
4.1.3.3 Event Dispatcher	54
4.1.3.4 YAML	55
4.1.4 Principe de fonctionnement	55
4.1.5 De l'importance de choisir Symfony	57
Chapitre 5 : Implémentation de la solution	59
1. Architecture de la solution	59
2. Présentation de l'application	60
CONCLUSION	69
<i>webographie</i>	<i>69</i>

Nous présentons ici certains sigles et abréviations que nous utiliserons dans le document.

RH	R essources H umaines
UML	Unified Modeling Language
DRH	Direction des Ressources Humaines
RAD	Développement Rapide d'Applications
GRH	Gestion des Ressources Humaines

Figure1 : Processus Scrum	16
Figure2 : Diagramme de cas d'utilisation Création de Compte	22
Figure4 : Diagramme d'activité Création de Compte	24
Figure5 : Diagramme de cas d'utilisation Gestion Administrative du personnel.....	25
Figure6 : Diagramme de cas d'utilisation détaillé Gérer Employé	26
Figure8 : Diagramme d'activité Créer Employé	27
Figure9 : Diagramme d'activité Modifier un enregistrement.....	29
Figure10 : Diagramme d'activité Créer Employé	30
Figure11 : Les trois couches de l'architecture 3-tiers	49
Figure12 : logo « Symfony »	52
Figure13 : Principe de fonctionnement de Symfony	55
Figure14 : Exemple de route Symfony	56
Figure15 : Diagramme de Déploiement	59
Figure16 : Authentification	60
Figure17 : Page d'accueil.....	61
Figure18 : Page de Création d'un Employé	61
Figure19 : Visualiser la liste des employés	62
Figure20 : Avant Modification de la matricule	63
Figure21 : Apres Modification de la matricule	63
Figure22 : Visualiser l'employé à Supprimer	64
Figure23 : Apres Suppression de l'employé Maty Diallo	65
Figure24 : Formation d'ajout d'un conjoint à un employé.....	66
Figure25 : Apres ajout de la conjointe	67
Figure26 : Détail de l'employé Marieme Sow	67
Figure27 : formulaire d'Ajout de fonction	68
Figure28 : Fonction Ajouté avec Succès	68

D'un point de vue général la gestion désigne l'action ou la manière de gérer, d'administrer, de diriger, d'organiser l'exécution d'une tâche. La **gestion des ressources humaines (GRH)** anciennement la **gestion du personnel** est l'ensemble des pratiques mises en œuvre pour administrer, mobiliser et développer les ressources humaines impliquées dans l'activité d'une organisation.

Au sein de l'Entreprise Kya's la gestion de ressources humaines est un mécanisme très complexe. En effet, les processus de gestion de ressources humaines se font manuellement. Le Directeur Général, de même les agents de Ressources humaines suivent un ensemble de procédure afin de bien gérer l'administration du personnel. Les démarches à suivre pour administrer le personnel, évaluer les employés, évaluer leur performance et compétence de même assurer leur formation sur tel ou tel domaine, sans un système informatique, peuvent être pénible.

Comment faciliter la gestion du personnel de Kya's ?

C'est dans ce sens que s'oriente notre stage effectué au niveau de l'Entreprise Kya's. Au cours de ce stage, il nous a été demandé de mettre en place une application pour la gestion des ressources humaines.

Afin de mener à bien ce travail, nous allons tout d'abord dans le premier chapitre faire une présentation générale de la structure et du sujet pour mieux définir le contexte dans lequel nous allons travailler, puis présenter la problématique et faire une étude de l'existant. Ensuite, dans les chapitres 2 et 3 nous procéderons à l'analyse et à la conception de la solution. Enfin nous terminerons par la réalisation et la mise en œuvre de l'application.

Chapitre1 : Présentation générale

Introduction

Le but de ce chapitre introductif est de mettre le travail dans son contexte général. Nous commençons tout d'abord par une présentation de l'entreprise d'accueil. Puis, nous décrivons le sujet autour duquel s'articule notre projet de fin d'études. Enfin, nous terminons par l'énumération des objectifs.

1. Présentation de la Structure d'accueil

1.1 Présentation de la société kya's

KIYA'S est une Société de Services et d'Ingénierie en Informatique (SSII) spécialisée dans la conception - développement de progiciels et offre des services à forte ajoutée aux entreprises et institutions. KIYA'S est un éditeur de progiciels de gestion et de solutions logicielles métiers spécifiques, fonctionnement dans un environnement web et permettant une utilisation décentralisée et multisite.

L'offre logicielle de gestion porte principalement sur :

- La gestion de paie et des ressources humaines
- La gestion de comptabilité générale et budgétaire
- La gestion des approvisionnements et achats, stocks et immobilisations

Les solutions métiers de KIYA'S sont conçues pour apporter une réponse personnalisée aux exigences de chaque métier à travers une plateforme logicielle intégrée, modulaire et évolutive :

- Gestion d'une institution de sécurité sociale : immatriculation des employeurs et employés, cotisations, prestations familiales, accidents du travail, risques professionnels, pension de vieillesse – invalidité – décès, assurance maladie
- Gestion d'un établissement d'enseignement supérieur, système LMD ou système classique.

KIYA'S, c'est aussi une équipe de professionnels engagés dans la réussite de ses projets. Les équipes de KIYA'S sont issues de formation de haut niveau et sont dotées d'une double compétence métier et technique.

Avec une expérience allant de 10 à 20 ans dans l'informatique et dans les métiers, nos équipes ont acquis des compétences techniques et fonctionnelles qui leur permettent d'aider nos clients à faire les choix technologiques les mieux adaptés. Nous accompagnons nos clients jusqu'à la réalisation totale de leur projet.

1.2 Domaines d'activités


KIYA'S intervient également dans le conseil, l'assistance à maîtrise d'ouvrage de projets et assure la formation et le renforcement de capacités sur les produits et solutions développées.

2. Contexte du projet


La gestion du personnel a toujours existé dans les entreprises mais elle se faisait de façon manuelle, aujourd'hui avec l'avancée de la technologie le besoin d'automatiser les tâches de gestion du personnel s'impose.

2.1 Présentation du cadre d'application du projet


Les différents domaines d'application du projet seront :

 La gestion des carrières et des compétences :

- L'évaluation du personnel
- La gestion de la performance
- La gestion des compétences
- La définition et la gestion des plans de carrière
- La gestion des plans de succession

 La gestion de la formation :

- L'élaboration, la mise en œuvre et le suivi du plan de formation
- Le suivi de l'activité formation
- La formation assistée par(e-learning)
- Le suivi statistique

 La gestion administrative du personnel

- La gestion des données de l'ensemble du personnel
- La gestion des populations multiples

- La gestion des informations diverses (contacts, cv, état civil, antécédents médicaux, état civil, formation scolaire)
- ✚ La gestion de l'organisation :
 - La gestion de l'organisation et de la répartition des ressources
 - La mise en place des outils de pilotages RH
 - La gestion des annuaires d'entreprise
 - La gestion des temps et des activités
 - La gestion documentaire RH
 - La sécurisation électronique des accès

2.2 Présentation de l'existant

2.2.1 Description de l'existant

Comme beaucoup d'entreprise de la place l'Entreprise Kya's gère son personnel de façon manuelle.

Les Agents RH de même que les salariés suivent un ensemble de processus pour une gestion des ressources du personnel.

En effet, l'administration du personnel, la gestion des carrières et compétences, la gestion de la formation de même que d'autres domaines de la RH (Ressources Humaines) restent purement manuelle au sein de la structure.

2.2.2 Critique de l'existant

La procédure existante atteint ces objectifs depuis sa mise en place mais le critère « suivit en temps réel de l'employé » est absent, en effet tout se fait de façon manuelle.

2.3 Problématique

La Gestion des Ressources Humaines est une discipline universelle et indispensable dans le cadre fonctionnel des entreprises afin de pouvoir gérer leur personnel.

Cependant cette dernière reste toujours manuelle dans la plupart des structures particulièrement l'entreprise Kya's d'où la nécessité d'automatiser les différentes tâches de ce domaine.

3. Objectifs

Nous avons comme objectif principal de mettre en œuvre une solution RH permettant la gestion complète du personnel d'une structure données.

Conclusion

Tout au long de ce chapitre, nous avons pu situer le cadre général de la société d'accueil et les objectifs majeurs à prendre en compte. De plus nous avons fait une étude De l'existant. Dans le chapitre suivant nous allons entamer les choix méthodologiques.

CHAPITRE 2

Chapitre 2 : Méthodologie de développement logiciel

Introduction

L'objectif de ce chapitre est d'adopter une méthodologie de développement logiciel de même qu'une méthode d'analyse et de conception pour notre projet. Nous allons choisir dans un premier temps la méthodologie de développement logiciel et dans un second la méthode d'analyse et de conception.

1. Choix du processus de développement logiciel

1.1 Processus agile vs dirigé par planification

Le terme "agile" définit une approche de **gestion de projet** qui prend le contre-pied des approches traditionnelles prédictives et séquentielles de type **cycle en V** ou waterfall (en cascade). La notion même de "**gestion de projet**" est remise en question au profit de "**gestion de produit**". De façon à raisonner davantage "produit" que "projet". Après tout l'objectif d'un projet consiste bien à donner naissance à un produit. [W1]

Une approche dite "traditionnelle" attend généralement du client une expression détaillée et validée du besoin en entrée de réalisation, laissant peu de place au **changement**. La réalisation dure le temps qu'il faut et le rendez-vous est repris avec le client pour la recette. Cet **effet tunnel** peut être très néfaste et conflictuel, on constate souvent un déphasage entre le besoin initial et l'application réalisée. On se rapporte alors aux **spécifications** validées et au **contrat**. Certains projets se terminent dans la douleur (surtout dans le cadre d'un **contrat au forfait** classique) au risque de compromettre la relation client. De plus il n'est pas rare que certaines fonctionnalités demandées se révèlent finalement inutiles à l'usage alors que d'autres, découvertes en cours de route, auraient pu donner plus de valeur au produit.

Une [enquête de 1994 du « Standish Group »](#) (certes controversée, comme toutes les enquêtes qui traitent d'un sujet sensible) fait le constat suivant : « 31 % des projets informatiques sont arrêtés en cours de route, 52 % n'aboutissent qu'au prix d'un important dépassement des délais et du budget tout en offrant moins de fonctionnalités qu'il n'en était demandé ; seuls 16 % des projets peuvent être considérés comme des succès. ».

Cette même enquête renouvelée en 2008 indique un taux de réussite de 35%, ce qui est plutôt positif mais demeure très faible. Le problème reste entier. Parmi les motifs d'échecs, arrivent en tête :

- Manque d'implication des utilisateurs finaux : 12,8 %.
- Changements de spécifications en cours de projet : 11,8 %.

L'approche Agile propose au contraire de réduire considérablement voire complètement cet **effet tunnel** en donnant davantage de visibilité, en impliquant le client du début à la fin du projet et en adoptant un **processus itératif et incrémental**. Elle considère que le besoin ne peut être figé et propose au contraire de s'adapter aux changements de ce dernier. Mais pas sans un minimum de règles.

1.2 Méthodes agiles

Les méthodes agiles partent du principe que spécifier et planifier dans les détails l'intégralité d'un produit avant de le développer (approche prédictive) est contre-productif. Cela revient à planifier dans les détails un trajet "Paris - Narbonne" en voiture par les petites routes. Spécifiant chaque villes et villages traversés, l'heure de passage associée, chaque rue empruntée dans les agglomérations, litres d'essence consommés, kilomètres parcourus, etc. Les imprévus ne manqueront pas d'arriver : embouteillages, déviations, travaux, sens de circulation inversés, voire la panne, etc. Rendrant votre planification et vos spécifications très vite obsolètes. Combien de temps aurez-vous passé à planifier cet itinéraire, comment réagirez-vous face à vos frustrations de ne pas pouvoir appliquer votre plan à la lettre ?

L'idée consiste à se fixer un premier objectif à courts termes (une grande ville par exemple) et se lancer sur la route sans tarder. Une fois ce premier objectif atteint, on marque une courte pause et on adapte son itinéraire en fonction de la situation du moment. Et ainsi de suite jusqu'à atteindre la destination finale. On parle donc d'une **approche empirique**. Dans le cadre d'un projet de développement logiciel, le client élabore sa **vision du produit** à réaliser et liste les **fonctionnalités** ou **exigences** de ce dernier. Il soumet cette liste à l'**équipe de développement**, communique directement avec elle (plutôt que par papier) qui estime le

coût de chaque élément de la liste. On peut ainsi se faire une idée approximative du budget global.

L'équipe sélectionne ensuite une portion des exigences à réaliser dans une portion de temps courte appelée **itération**. Chaque itération inclut des travaux de conception, de spécification fonctionnelle et technique quand c'est nécessaire, de développement et de test. A la fin de chacune de ces itérations, le produit partiel mais utilisable est montré au client. Ce dernier peut alors se rendre compte par lui-même très tôt du travail réalisé, de l'alignement sur le besoin. L'utilisateur final quant à lui peut se projeter dans l'usage du produit et émettre des feedbacks précieux pour les futures itérations. La visibilité ainsi offerte est clef. Cette transparence peut également apporter davantage de confiance et de collaboration dans la relation client/fournisseur. Les risques quant à eux sont levés très tôt.

Si le client a priorisé avec soin son besoin, il peut saisir l'opportunité d'accélérer le "**time to market**" s'il estime que le produit en l'état (partiel) peut aller en production. Économisant ainsi son budget et récoltant un premier **retour sur investissement**. Il a aussi la possibilité de changer en cours de route la priorité des fonctionnalités qui n'ont pas encore été développées (prévues pour les itérations futures). Afin de retarder une fonctionnalité dont le besoin n'est pas mûr, ajouter une nouvelle fonctionnalité cruciale en échange du retrait d'une autre (respectant ainsi budget et délais), etc.

Cette souplesse ainsi offerte est donc un véritable atout pour le client.

✓ **Scrum**

Scrum est considéré comme un cadre ou « framework » de gestion de projet. Ce cadre est constitué d'une définition des rôles, de réunions et d'artefacts.

Scrum définit trois rôles :

- ◆ **Le « Product Owner »** qui porte la vision du produit à réaliser (représentant généralement le client).
- ◆ **Le « Scrum Master »** garant de l'application de la méthodologie Scrum.
- ◆ **L'équipe de développement** qui réalise le produit.

La vie d'un projet Scrum est rythmée par un ensemble de réunions clairement définies et strictement limitées dans le temps (timeboxing) :

- ◆ **Planification du Sprint** (Sprint = itération) : au cours de cette réunion, **l'équipe de développement** sélectionne les éléments prioritaires du « **Product Backlog** » (liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du sprint (en accord avec le « **Product Owner** »).

- ♦ **Revue de Sprint** : au cours de cette réunion qui a lieu à la fin du sprint, l'**équipe de développement** présente les fonctionnalités terminées au cours du sprint et recueille les feedbacks du **Product Owner** et des utilisateurs finaux. C'est également le moment d'anticiper le périmètre des prochains sprints et d'ajuster au besoin la planification de release (nombre de sprints restants).
- ♦ **Rétrospective de Sprint** : la rétrospective qui a généralement lieu après la revue de sprint est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc.) à la lueur du "vécu" sur le sprint écoulé (principe d'**amélioration continue**).
- ♦ **Mêlée quotidienne** : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait debout (elle est aussi appelée "stand up meeting") en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions : « Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ? »



Figure 1 : Processus Scrum

2. Choix d'une méthode d'analyse et de conception

Merise et UML sont deux grands principes de « traduction » ou modélisation d'un système d'information. Néanmoins, ils ne sont pas aussi proches qu'on pourrait le penser. Le choix de l'un ou de l'autre se fait selon trois axes à savoir l'accessibilité, la précision et l'exploitabilité. Pour le premier axe (accessibilité) MERISE présente l'intérêt d'avoir des modèles logiques moins détaillés facilement compréhensibles. Tandis qu'UML conçu pour s'adapter à n'importe quel langage de programmation orientée objet (POO), présente plusieurs modèles (diagrammes) dont leurs compréhensions nécessitent une grande attention. En ce qui concerne le deuxième critère (précision), MERISE est moins préférable. Malgré sa clarté, il manque une précision du fait qu'elle est éloignée du langage donc difficile à implémenter alors qu'UML intègre les éléments communs des différents langages, sa volonté est d'être fidèle à la réalisation finale. Elle est beaucoup plus complète avec ses différents diagrammes.

Pour en finir avec l'exploitabilité, MERISE est une méthode plus généraliste. Elle donne une vue globale de la solution sans autant entrer dans les petits détails. Contrairement à UML qui est conçu pour l'implémentation objet avec ses différents détails et sa portabilité (s'adapte à n'importe quelle plateforme) elle est donc plus exploitable. L'une ou l'autre présente des avantages et des inconvénients. Il est réservé au concepteur de choisir la méthode la mieux adaptée pour son cas. Si on cherche la précision et l'exploitabilité UML devance MERISE. Tandis que, si c'est la clarté et l'accessibilité qui sont en question MERISE est préférable.

Mon application gère des données complexes et utilise l'implémentation objet d'où UML est la mieux recommandée, vue qu'elle rend la modélisation plus simple à implémenter.

2.1 UML (Unified Modeling Language)

UML est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. [W 2]

Utilisation

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

- Activité d'un objet/logiciel
- Acteurs

- Processus
- Schéma de base de données
- Composants logiciels
- Réutilisation de composants

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement tout ou partie du code d'une application logicielle, par exemple en langage Java, à partir des divers documents réalisés. UML 2.3 propose 14 types de diagrammes (25 en UML 1.3). UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun, même si le diagramme de classes est généralement considéré comme l'élément central d'UML ; des méthodologies, telles que l'Unified Process, axent l'analyse en tout premier lieu sur les diagrammes de cas d'utilisation (use case). De même, on peut se contenter de modéliser seulement partiellement un système, par exemple certaines parties critiques.

Une façon de mettre en œuvre UML est de considérer différentes *vues* qui peuvent se superposer pour collaborer à la définition du système :

- Vue logique (logical view) : c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs (c'est le comment).
- Vue d'implémentation (implementation view) : cette vue définit les dépendances entre les modules.
- Vue des processus (process view) : c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation...
- Vue de déploiement (deployment view) : cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est le houx).

UML se décompose en plusieurs parties :

- Les vues : ce sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues, il est possible de définir (ou retrouver) le système complet.
- Les diagrammes : ce sont des ensembles d'éléments graphiques. Ils décrivent le contenu des vues, qui sont des notions abstraites. Ils peuvent faire partie de plusieurs vues.

Les modèles d'élément : ce sont les éléments graphiques des diagrammes.

Conclusion

Dans ce chapitre il a été question de choix de méthodologies. Cependant le choix de méthodologie de développement logiciel et de méthode d'analyse et de conception a été réalisé. Dans le prochain chapitre nous aborderons l'analyse et la spécification des besoins fonctionnels.

CHAPITRE

3

Chapitre 3 : Spécifications fonctionnelles détaillées

Introduction

Dans ce chapitre nous allons faire l'analyse et la spécification fonctionnelle détaillée des besoins.

1. Choix du processus de développement logiciel

Employé : par employé nous désignons tous ceux qui travaillent dans l'entreprise hormis les employeurs.

Agent RH : c'est un employé de l'entreprise qui s'occupe de la gestion des ressources humaines de l'entreprise.

Responsable : c'est un employé ou employeur qui a une autorité sur d'autres employés ou qui est le supérieur à d'autres employés. Il est chargé de diriger un groupe bien défini selon ses qualifications.

2. Présentation du domaine

La GRH couvre de nombreux domaines, intervenant à tous les stades de la « vie » du salarié dans l'organisation, nous trouvons alors :

- ◆ Le recrutement
- ◆ La gestion des carrières
- ◆ La formation
- ◆ L'évaluation des performances
- ◆ La gestion des conflits
- ◆ La concertation sociale
- ◆ La motivation et l'implication du personnel
- ◆ La communication
- ◆ La satisfaction au travail

Nous présenterons ci-dessous les principaux domaines de la GRH :

A. L'administration du personnel :

C'est sous cet aspect que la fonction commence à exister et à être perçue dans l'entreprise :

- ◆ L'enregistrement, le suivi et le contrôle des données individuelles, et collectives du personnel de l'entreprise.
- ◆ L'application des dispositions légales et réglementaires dans l'entreprise.
- ◆ La préparation des commissions et des réunions.
- ◆ Le maintien de l'ordre et du contrôle et les travaux de pointage.

B. La Gestion au sens large :

Cette expression nous recouvre trois domaines :

- ◆ L'acquisition des ressources humaines : par la gestion de l'emploi, programmes de recrutement, plans de carrières et de promotion, analyse des postes et l'évaluation des personnes.
- ◆ La gestion des rémunérations : par l'analyse et l'évolution des postes, grille de salaires, politique de rémunération, intéressement et participation.
- ◆ La gestion de formation : par la détection des besoins, l'élaboration des plans de formation, la mise en œuvre des actions de formation et l'évaluation des résultats.

C. La communication, l'information et les conditions de travail :

Pour la communication et l'information, les tâches de la direction des ressources humaines sont :

- ◆ La définition des publics et de la conception des messages
 - ◆ La gestion des moyens : journal d'entreprise, affichage, audio-visuel, réunion systématique
- Pour l'amélioration des conditions de travail, les principaux thèmes sont :
- ◆ L'hygiène et la sécurité
 - ◆ L'aménagement du temps de travail
 - ◆ La gestion des activités sociales (restaurant, centre de vacances, loisirs)

D. Les relations professionnelles :

Ce domaine consiste en trois activités :

- ◆ La négociation, la conclusion et le renouvellement de la convention d'entreprise
- ◆ La mise en œuvre de la convention collective : La direction ressources humaines est chargée de l'application de la convention collective, et la solution des conflits

individuels et collectifs pour améliorer les relations de travail avec l'aide ou la participation des représentants des travailleurs s'il y a lieu (syndicat)

- ◆ Les relations externes : Le titulaire de la fonction est souvent responsable des relations :

1. avec l'inspection de travail
2. avec la sécurité sociale
3. avec le conseil des prud'hommes et les avocats

3. Capture des Besoins fonctionnels

Nous allons commencer par la gestion du compte de l'employé avant d'entamer la gestion administrative du personnel.

GESTION DU COMPTE DE L'EMPLOYÉ :

Créer un compte, s'authentifier, et modifier son compte. Pour s'authentifier il faut forcément créer un compte au préalable, de même pour modifier son compte il faut d'abord s'authentifier donc l'utilisateur s'inscrit sur la plateforme ensuite il s'authentifie et enfin peut modifier son compte.

3.1 Diagramme de Cas d'utilisation Gestion du Compte de l'Employé

La figure suivante présente le diagramme de cas d'utilisation de la gestion de compte de l'employé :

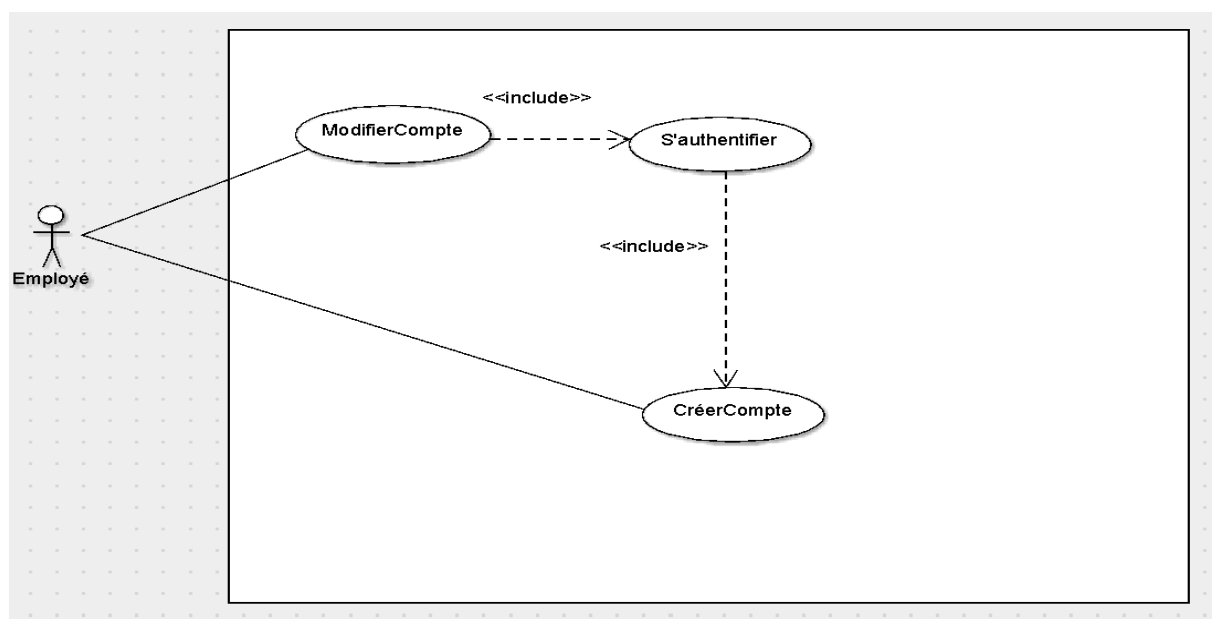


Figure2 : Diagramme de cas d'utilisation Création de Compte

3.2 Fiche de description textuelle et Diagramme d'activité du Cas Création de Compte

3.2.1 Fiche de Description textuelle du Cas Créer Compte

Titre	Créer Compte
Acteurs	Employé
Description	Créer un compte pour avoir un accès au niveau du système
Préconditions	Etre Employé de l'Entreprise Kya's
Scénario nominal	1.L' Employé sélectionne Inscription 2. Le système affiche la page d'inscription 3.L' Employé remplit le formulaire et valide la saisie 4. Le système vérifie les informations de saisie 5. L'inscription est validée avec une notification de compte crée aux administrateurs qui devrons l'activer plus
Scénario alternatif	3.b L'Employé décide d'annuler la saisie Si les informations de saisie ne sont pas valides 4.b L'Employé est redirigé vers la saisie avec un message lui indiquant son erreur
Post Conditions	Inscription réussie

Tableau 1 : Création de Compte

À la suite de la description textuelle les scénarii sont représentés dans le diagramme d'activité ci-dessous

3.2.2 Diagramme d'activité

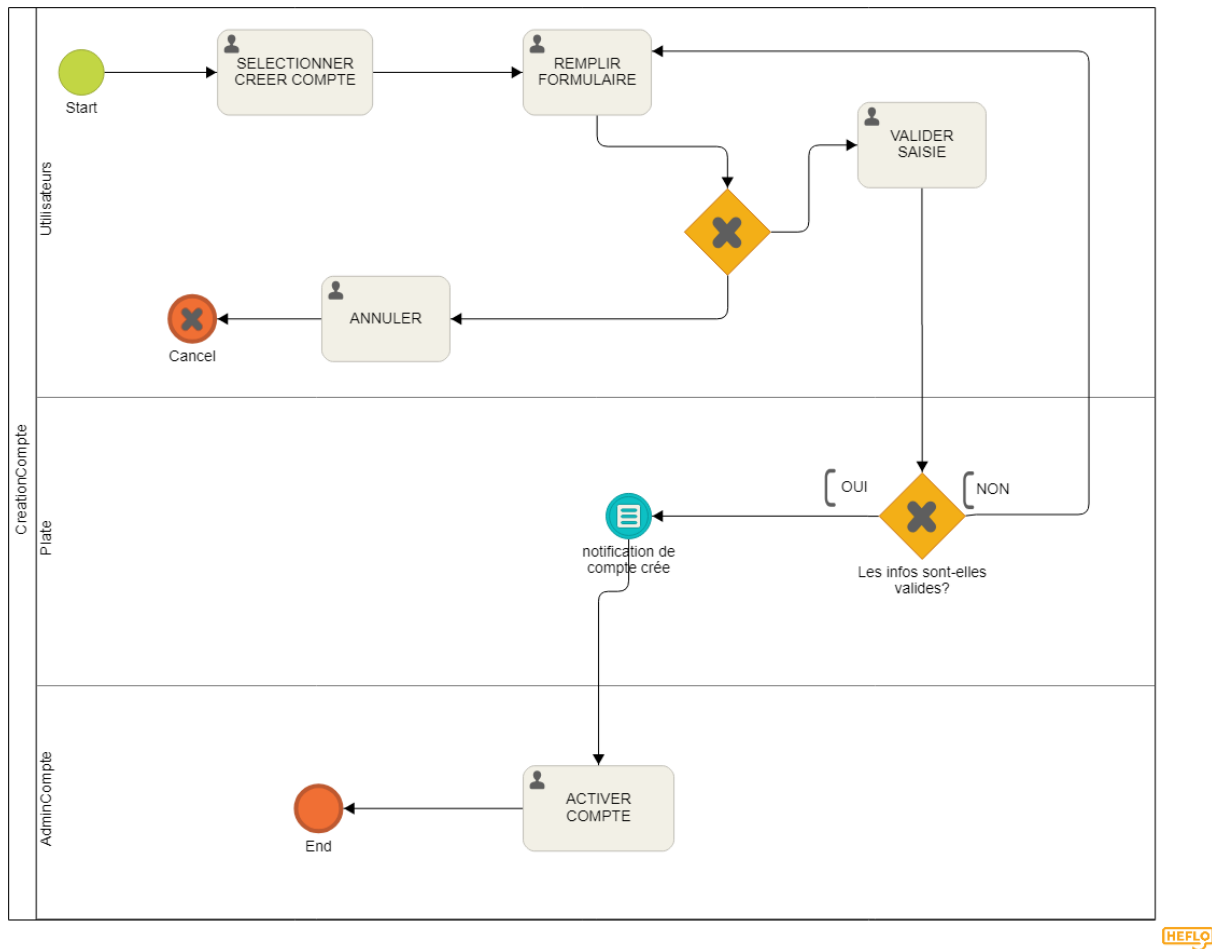


Figure4 : Diagramme d'activité Création de Compte

Nous avons deux acteurs qui interviennent sur ce processus :

Employé : il sélectionne créer compte, ensuite remplit le formulaire. Après remplissage du formulaire il peut choisir d'annuler le processus ou de continuer en validant la saisie puis une vérification des informations de la saisie est faite et en fonction de la validité de ces dernières soit l'utilisateur est renvoyé au formulaire de saisie ou un message est envoyé aux administrateurs de Compte.

Administrateur de Compte : après création d'un compte l'administrateur reçoit une notification et pourra activer le compte de l'utilisateur qui devient un employé.

GESTION ADMINISTRATIVE DU PERSONNEL

3.3 Diagramme de Cas d'utilisation

Nous avons le diagramme de cas d'utilisation du module gestion de cas administrative du personnel

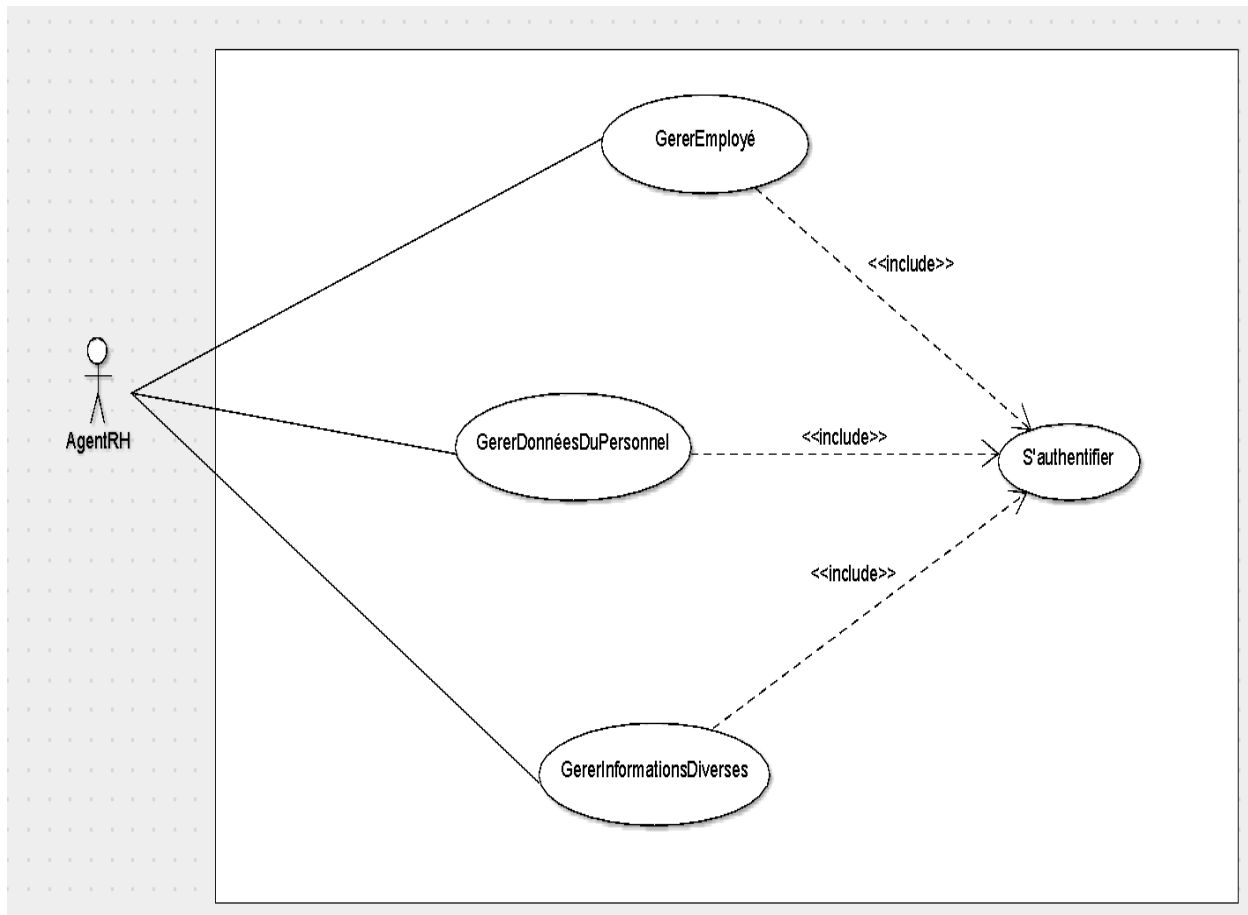


Figure5 : Diagramme de cas d'utilisation Gestion Administrative du personnel

Dans un premier temps nous allons implémenter le cas gestion des populations multiples. La gestion des populations multiples : il s'agit ici de la gestion des employés de l'entreprise, des collaborateurs externes ainsi que la grappe familiale des employés.

La gestion des populations tourne au tour des employés que ce soit la grappe familiale elle est liée à l'employé et pour ce qui est des collaborateurs externes nous avons toujours des employés mais qui sont externes à l'entreprise. Cependant nous considérerons que la gestion des populations multiples est assimilable à la gestion des employés en prenant en compte la grappe familiale et les collaborateurs externes.

Cependant le diagramme de cas d'utilisation du module de la gestion administrative du personnel décrit les package du module qui sont plutôt généraliste sur ce pour mieux comprendre les cas nous allons détailler les plus importants dans ce qui suit.

3.3.1 Raffinement des Cas d'utilisations

❖ Cas : Gérer Employé

◆ Diagramme de cas d'utilisation.

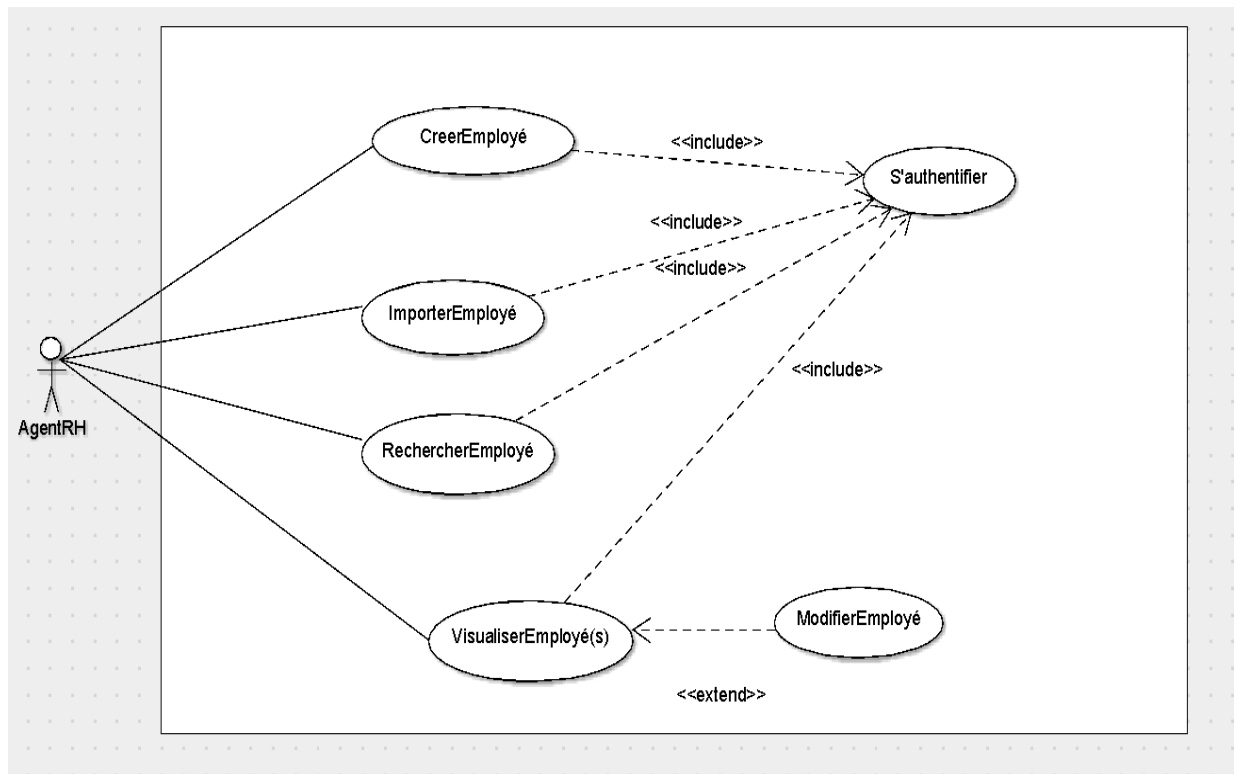


Figure6 : Diagramme de cas d'utilisation détaillé Gérer Employé

3.3.2 Fiches de description textuelle et Diagrammes d'activités

❖ **Cas : Créer Employé**

◆ Fiche de description textuelle

Titre	Créer Employé
Acteurs	Agent RH
Description	Ajouter un employé sur la plateforme
Préconditions	Etre Employé de l'Entreprise Kya's
Scénario nominal	<ol style="list-style-type: none"> 1. l'Agent RH sélection créer un employé 2. Le système vérifie le formulaire 3. Le système affiche le formulaire 4. l'Agent RH remplit le formulaire et valide la saisie 5. Le système vérifie les informations de saisie 6. Le système Affiche la liste des employé

Scénario alternatif	2.b le formulaire ne dispose pas de toutes les informations nécessaires 4.b L'utilisateur peut annuler la saisie 5.b Informations de saisies invalide 6.b Redirection vers le formulaire de création
Post Conditions	Création réussie

Tableau 2 : Fiche de description textuelle Créer Employé

◆ Diagramme d'activité

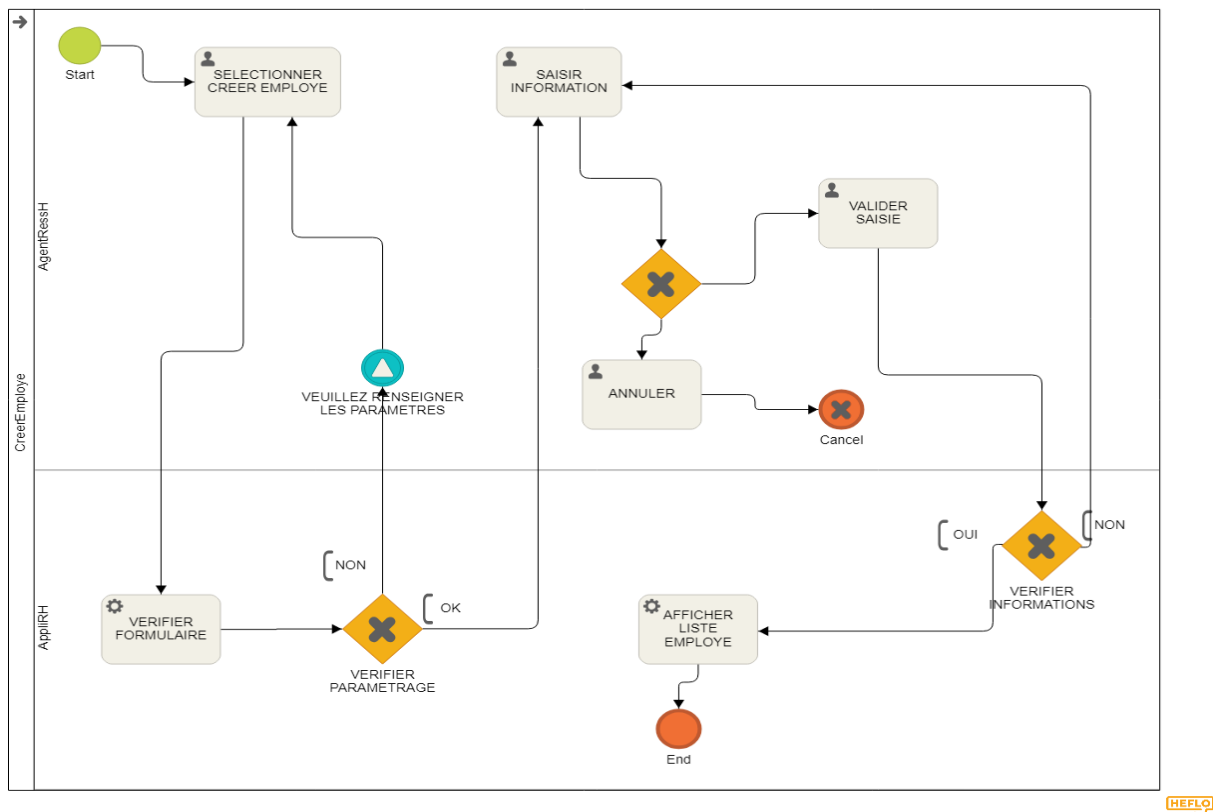


Figure8 : Diagramme d'activité Créer Employé

Administrateur des Ressources Humaines : il sélectionne créer un employé, une vérification du paramétrage est faite si les paramètres nécessaires à la création sont valides il pourra saisir les informations si le paramétrage est correct, ensuite il valide la saisie et enfin après vérification des données la liste des employés récemment créés est affichée au cas échéant il sera redirigé vers la saisie.

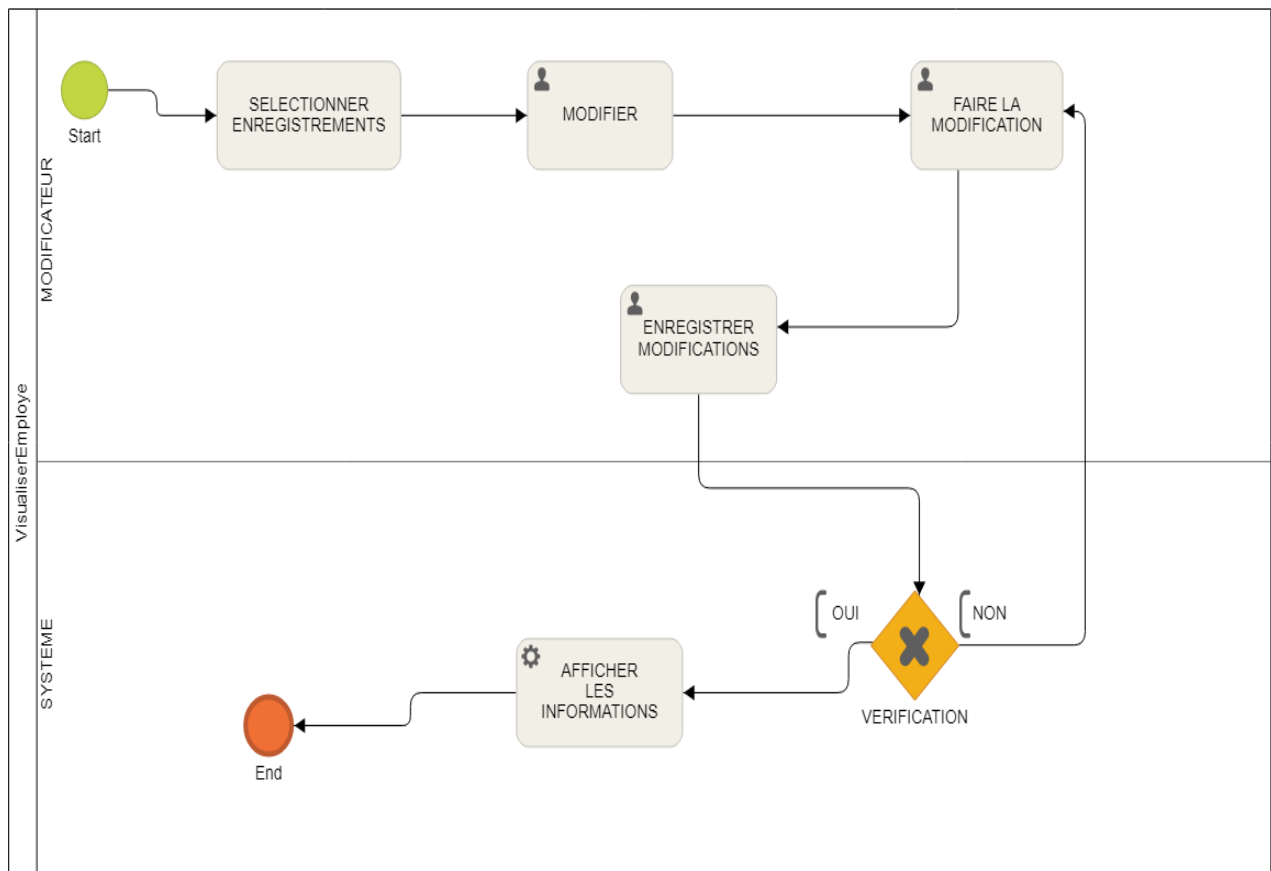
❖ Cas : Modifier Employé

◆ Fiche de description textuelle

Titre	Modifier Employé
Acteurs	Agent RH
Description	Modifier les informations d'un employé donné
Préconditions	Avoir les droits de modification
Scénario nominal	1. Agent RH sélectionne l'employé a modifier 2. Agent RH modifie les informations 3. Enregistre les modifications 5. Le système vérifie les informations modifiées 6. Le système Affiche les informations modifié
Scénario alternatif	5.b Informations invalides 6.b Redirection vers les informations à modifier
Post Conditions	Création réussie

Tableau 3 : Fiche de description textuelle Modifier Employé

◆ Diagramme d'activité



HEFLO

Figure9 : Diagramme d'activité Modifier un enregistrement

Pour modifier un enregistrement quelconque il suffit de le sélectionner ensuite sur modifier puis de faire la modification, après modification on aura la possibilité d'enregistrer les modifications qui entrainera une vérification des enregistrements modifiés s'ils sont valides les enregistrements modifiés seront affichés, en revanche s'ils ne sont pas valides l'utilisateur sera redirigé vers les enregistrements en cours de modification.

1. Diagramme de classe

Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations. Chaque application qui va mettre en œuvre le système sera une instance des différentes classes qui le compose.

A ce titre il faudra bien garder à l'esprit qu'une classe est un modèle et l'objet sa réalisation.

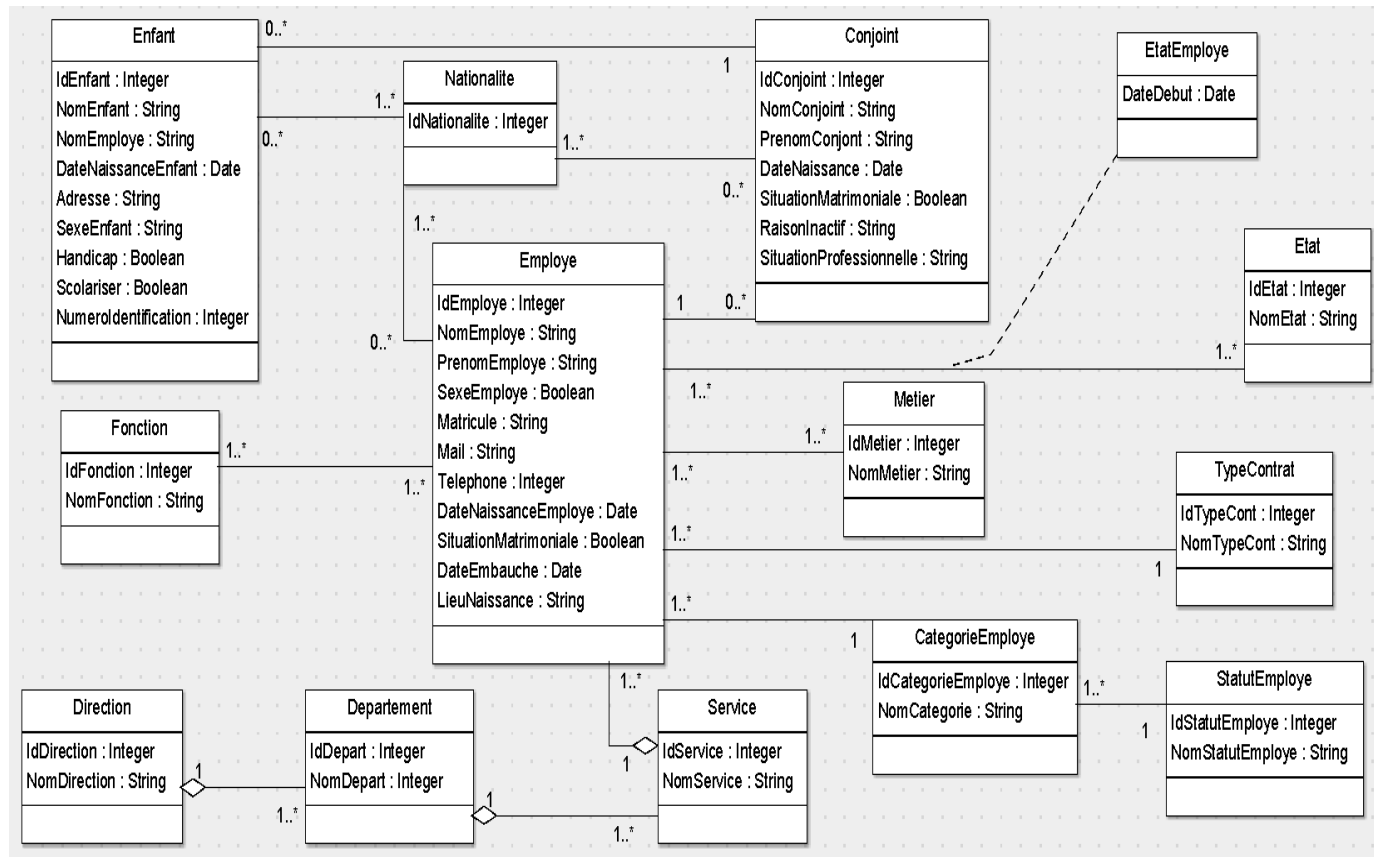


Figure10 : Diagramme d'activité Créer Employé

Description des associations :

Un employé peut avoir :

- Zéro ou plusieurs conjoints.
- Zéro ou plusieurs enfants.
- Une ou plusieurs nationalités.
- Une ou plusieurs fonctions.
- Un ou plusieurs métiers.
- Un type de contrat signé.
- Une catégorie d'employé.
- Un service.

Un Conjoint est lié à :

- Un seul employé.
- Une ou plusieurs nationalités.

Zéro ou plusieurs enfants.

Un enfant est le fils ou la fille d'un employé et d'un conjoint et a une ou plusieurs nationalités.

Une nationalité pour :

- Zéro ou plusieurs employés.
- Zéro ou plusieurs enfants.
- Zéro ou plusieurs conjoints.

Une fonction pour un ou plusieurs employés.

Un métier pour un ou plusieurs employés.

Un type de contrat pour un ou plusieurs employés.

Une Catégorie d'employés pour un ou plusieurs employés.

Une catégorie d'employés ne peut avoir qu'un seul statut et le statut peut être pour plusieurs catégories d'employés.

Une Direction comporte un ou plusieurs départements et un département se trouve que dans une seule Direction.

Un service se trouve dans un seul département tandis qu'un département en compte un ou plusieurs.

Un service peut avoir un ou plusieurs employés.

Conclusion

Au cours de ce chapitre, nous avons commencé par la capture des besoins fonctionnels qui sont décrits en diagramme de cas d'utilisation ensuite pour chaque cas on a fait le diagramme d'activités pour expliciter les actions qui la compose et enfin un diagramme de classe qui fait office de modèle de notre application. La Conception de la plateforme et les choix de technologies dans le chapitre suivant.

Chapitre4 : Conception de la plateforme et Choix des outils et technologies

Introduction

Dans ce chapitre il sera question de solution technique et d'architecture de la plateforme, en outre les choix des outils et technologies seront présentés.

1. Solution technique

La solution technique correspond à l'ensemble des outils techniques corrélés pour aboutir à la Solution obtenue.

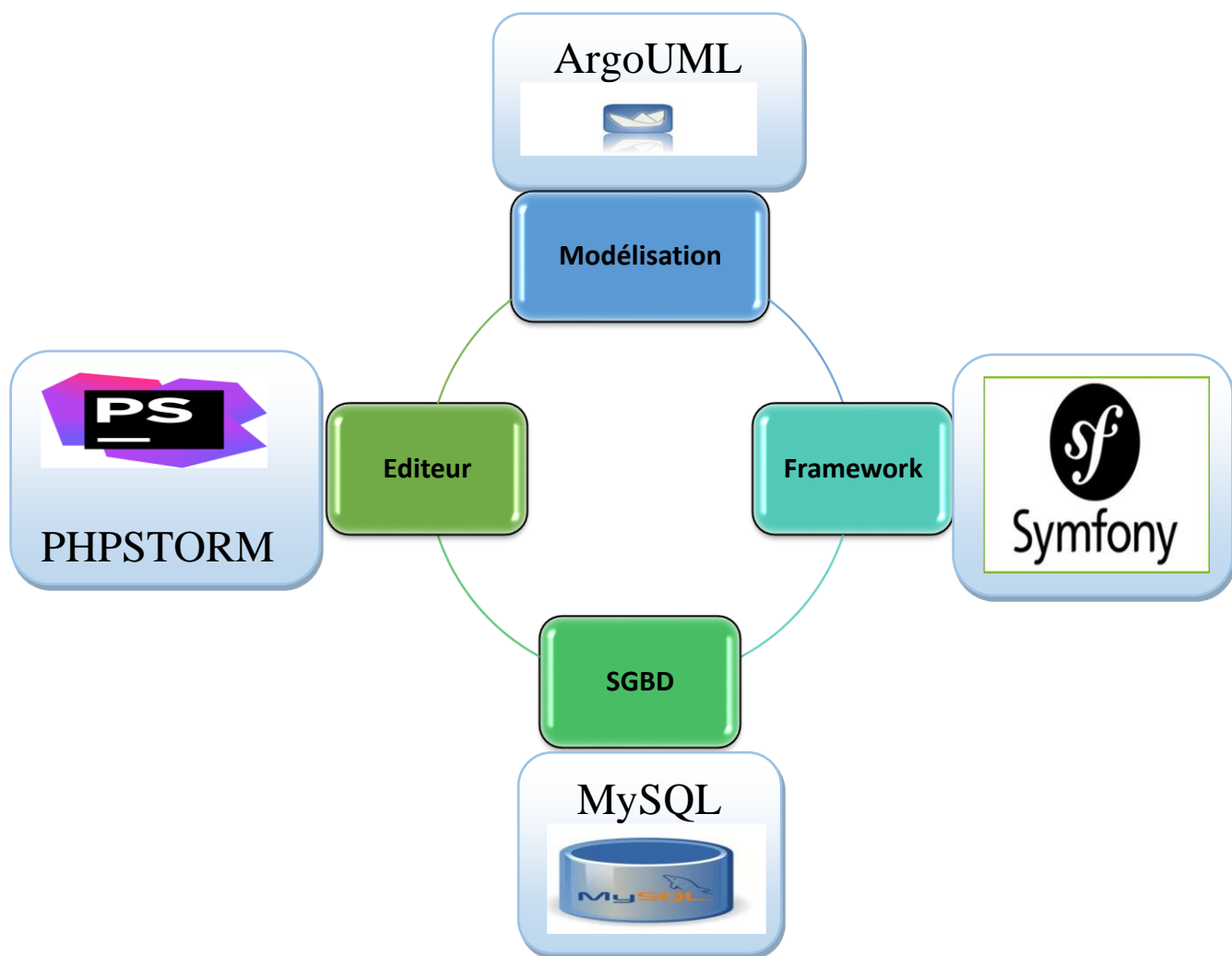


Figure11 : Architecture technique de la Solution

2. Architecture de la plateforme

L'architecture 3-tiers est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et présenter cette application comme un empilement de trois couches dont le rôle est clairement défini.

2.1 Différentes couches du 3-tiers

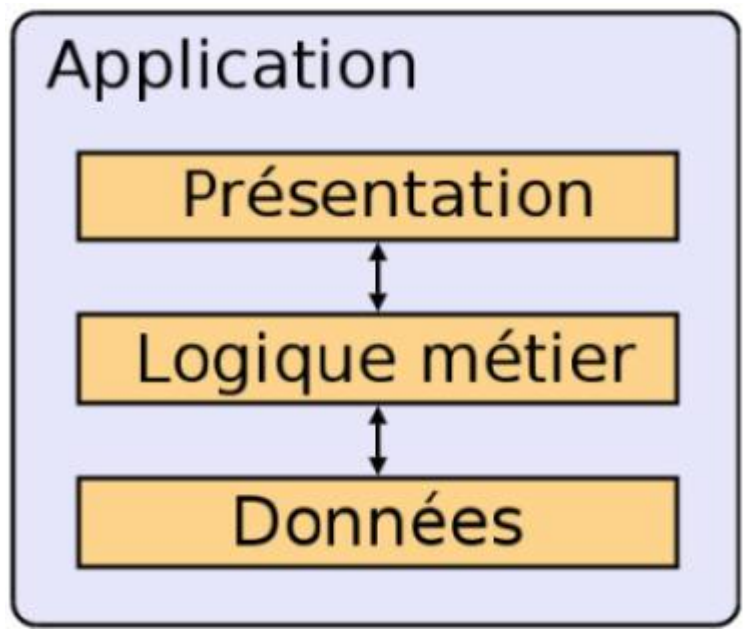


Figure12 : Les trois couches de l'architecture 3-tiers

2.1.1 Présentation des données

Correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur. On parle d'Interface Homme Machine. Elle va être représentée par des interfaces ASP pour être exploitée pour le navigateur web. On conçoit facilement que cette interface peut prendre de multiples facettes sans changer la finalité de l'application.

La couche présentation relaie les requêtes de l'utilisateur à destination de la couche métier, et en retour lui présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche inférieure.

2.1.2 Logique métier

Correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative. Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche. La couche métier offre des services applicatifs et métier à la couche présentation.

Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure. En retour, elle renvoie à la couche présentation les résultats qu'elle a calculés.

2.1.3 Accès aux données

Correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive. Les données peuvent être stockées indifféremment dans de simples fichiers texte, fichiers XML, ou encore dans une base de données (c'est ce qu'on va utiliser dans notre cas).

Quel que soit le support de stockage choisi, l'accès aux données doit être le même. Cette abstraction améliore la maintenance du système.

2.2 Justification du choix

L'approche 3-tiers apporte de réels avantages :

- ◆ Une conception claire et efficace grâce à la séparation des données.
- ◆ Un gain de temps de maintenance et d'évolution du site.
- ◆ Une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données de l'affichage et des actions).
- ◆ Plus de flexibilité dans l'allocation des ressources et dans les requêtes du client vers le serveur

3. Les plateformes de développement d'application web

Une chose est sûre, avec l'avènement de .NET et le combat annoncé entre Sun et Microsoft, il ressort d'ores et déjà un grand gagnant incontestable et incontesté : **l'architecture n-tiers à base d'objets**. Ainsi, la philosophie de cette partie du mémoire est avant tout d'insister sur l'architecture que nous pourrions qualifier de "**logique**" dans le but d'illustrer les différentes solutions du marché.

3.1 Les différents types d'architecture système

Il existe de nombreuses architectures possibles pour les applications d'entreprise, nous allons passer en revue les plus connues.

3.1.1 Architecture à deux niveaux

Pour simplifier, on peut dire que, dans ce type d'architecture, les traitements se font sur le poste client et la base de données est sur le serveur. Le plus gros problème est celui de la maintenance : La moindre modification oblige à mettre à niveau chaque poste client. De plus les performances de l'application dépendent plus des ressources des clients.

3.1.2 Architecture à trois niveaux

À cause des problèmes de l'architecture à deux niveaux, on a rajouté un autre niveau. Ainsi une application est divisée en trois niveaux logiques :

- ◆ **Niveau présentation** : l'interface graphique pour l'utilisateur.
- ◆ **Niveau métier** : Recouvre la logique métier (logique applicative).
- ◆ **Niveau donnée** : Les données de l'application (serveur de base de données, documents XML, serveur LDAP...).

Le niveau métier contient le code appelé (via le niveau présentation) par l'utilisateur pour extraire et traiter les données de la troisième couche. Cette séparation en couches améliore la souplesse de l'application. Il devient facile de déployer de nombreuses interfaces utilisateur sans devoir modifier la logique applicative.

3.1.3 Architecture multi niveaux

Ce type d'architecture se compose de différents niveaux :

- ◆ **Interface utilisateur** : Couche chargée de gérer les interactions entre l'utilisateur et l'application (Application de bureau, navigateur WAP, Browser Internet...)
- ◆ **Logique de présentation** : Elle permet de définir ce que doit afficher l'interface utilisateur et la manière dont les requêtes doivent être traitées.
- ◆ **Logique métier** : Modélise les règles métiers de l'entreprise.
- ◆ **Service d'infrastructure** : Fonctionnalités fournies aux composants (connexions, transactions...).
- ◆ **Données** : Données de l'entreprise.

4. Quelle plateforme de développement pour notre application ?

ASP.NET est la plateforme web principale fonctionnant à l'aide d'un serveur programmé en ASP. Cette plateforme permet une interopérabilité de tous les langages de programmation développés par Microsoft.

Parce qu'elles sont gratuites, les plateformes de développement en PHP abondent. Codeignitor, Symfony ou Cake PHP sont les plus populaires. Flexibles et simples d'utilisation, les plateformes PHP sont une bonne option pour développer des applications web complètes et efficaces.

Enfin du fait de leur gratuité, de leur flexibles et leur facilité d'utilisation, les plateformes PHP sont prioritaires pour notre projet. C'est dans cette optique que nous avons choisi le Framework Symfony version 3, plateforme que nous allons présenter plus en détail.

4.1 Introduction à la plateforme Symfony

Symfony est un ensemble de composants PHP ainsi qu'un Framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web. [W 1]

Son logo est représenté dans la figure suivante



Figure 13 : logo « Symfony »

4.1.1 Historique

L'agence web française **SensioLabs** est à l'origine du Framework Sensio Framework. À force de toujours recréer les mêmes fonctionnalités de gestion d'utilisateurs, gestion **ORM**, etc., elle a développé ce Framework pour ses propres besoins. Comme ces problématiques étaient souvent les mêmes pour d'autres développeurs, le code a été par la suite partagé avec la communauté des développeurs PHP.

Le projet est alors devenu Symfony (conformément à la volonté du créateur de conserver les initiales *S* et *F* de Sensio Framework), puis Symfony2 à partir de la version 2. La version 2 de Symfony casse la compatibilité avec la branche 1.x.

Le 5 septembre 2017, Symfony passe la barre du milliard de téléchargement.

4.1.2 Utilisation

Symfony est utilisé par Askeet et beaucoup d'autres applications, dont Delicious et Yahoo! Bookmarks. En 2009, Dailymotion a porté son code sur Symfony et a continué la transition.

4.1.2.1 Le langage PHP

Tout d'abord, PHP, qui est à la base de Symfony, est aujourd'hui une technologie majeure, adoptée par quasiment l'ensemble de l'industrie logicielle. ***Hypertext Preprocessor***, plus connu sous son sigle **PHP** (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

Aussi Symfony présente de nombreux avantages dont :

➤ La robustesse

En tant que professionnels du web, nous sommes tous exposés aux attaques du web et il est extrêmement important de ne pas se tromper lorsqu'on choisit le socle de son application afin d'être serein une fois l'application en ligne.

Faire le choix de Symfony, c'est faire le choix d'un framework dont la sécurité est pensée dès le début contre les failles les plus dangereuses : XSS, CSRF, injection SQL.

Au de-là de l'approche sécuritaire, choisir Symfony, c'est aussi choisir le socle logiciel le plus puissant de l'écosystème PHP, décomposé en composants comme des outils dont se sert une grande majorité des projets du web moderne :

➤ **La sécurité**

Symfony intègre des mesures de sécurité préventives pour lutter contre les failles et attaques XSS, CSRF et injection SQL. Contrairement à un développement PHP maison où il faut penser systématiquement à protéger chaque requête, formulaire ... Symfony embarque systématiquement ces mécanismes de sécurité, sans avoir à les implémenter à chaque fois.

Autre avantage du développement d'un projet web sur mesure à l'aide du Framework comparé à l'utilisation d'un CMS du marché (comme WordPress, Drupal ou Joomla), il va être moins ciblé par les hackers qui concentrent leurs efforts sur les CMS répandus sur le web et dont l'architecture et les URLs sont prédictibles

4.1.3 Présentation de quelques composants

4.1.3.1 HttpFoundation et HttpKernel

Le premier définit une couche orientée objet pour la spécification HTTP. Il fournit une abstraction pour les requêtes, les réponses, les téléchargements de fichiers, les cookies, les sessions, etc. Quant au composant HttpKernel, il fournit les briques de construction pour bâtir les Framework HTTP souples et performants.

4.1.3.2 Routing

Il permet d'associer une requête HTTP à un ensemble de variables de configuration pouvant être utilisées dans du code en utilisant les fonctions de rappel PHP. Celui-ci simplifie la définition d'URLS facilement indexable sans avoir à utiliser un fichier. htaccess complexe.

4.1.3.3 Event Dispatcher

Il permet l'implémentation du patron de conception Médiateur d'une manière simple et efficace pour rendre les projets PHP facilement extensibles. Ce composant est intéressant pour les programmes riches en classes, car la communication entre ces dernières devient plus simple et facile.

4.1.3.4 YAML

YAML est un format intéressant pour la configuration. N'étant pas un langage de balises et est facilement compréhensible par l'humain pour tous les langages de programmation. Ce composant analyse des chaînes de caractères YAML afin de les convertir en tableaux PHP. Il permet aussi la conversion des tableaux PHP en chaînes de caractères YAML. Notons également que YAML peut être utilisé à la place des fichiers XML ou JSON.

4.1.4 Principe de fonctionnement

Symfony fonctionne grâce à des bundles. Ils sont intégrés à l'aide d'un module. Certains bundles ne peuvent pas fonctionner sans autres bundles. Chaque bundle possède ses vues et ses contrôleurs et sont gérés par le système de route de Symfony.

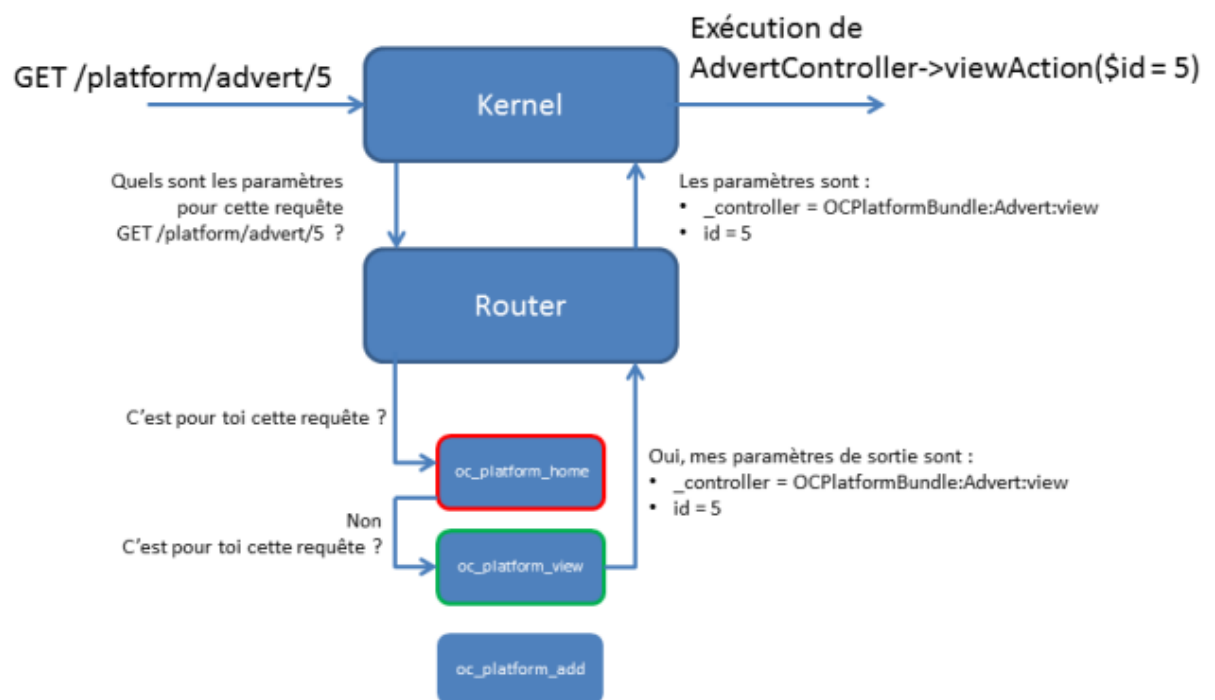


Figure13 : Principe de fonctionnement de Symfony

Le système de routeur va chercher les différentes URL et dès qu'il tombe sur la bonne alors il ne cherche plus.

Exemple d'un code route :

```
1 # src/OC/PlatformBundle/Resources/config/routing.yml
2
3 hello_the_world:
4     path:      /hello-world
5     defaults: { _controller: OCPlatformBundle:Advert:index }
```

Figure14 : Exemple de route Symfony

- hello_the_world est le **nom** de la route. Il est assez arbitraire, et vous permet juste de vous y retrouver par la suite. La seule contrainte est qu'il soit unique.
- path correspond à l'**URL** à laquelle nous souhaitons que notre « Hello World ! » soit accessible. C'est ce qui permet à la route de dire : « Cette URL est pour moi, je prends. »
- Defaults correspond aux **paramètres** de la route, dont :
 - _controller, qui correspond à l'**action** (ici, « index ») que l'on veut exécuter et au **contrôleur** (ici, « Advert ») que l'on va appeler (un contrôleur peut contenir plusieurs actions, c'est-à-dire plusieurs pages).

Parmi les bundles de Symfony, il y en a qui sont incontournables au point où sans eux aucune application ne pourrait fonctionner.

DOCTRINE

Doctrine c'est l'outil incontournable mais optionnel de Symfony qui va vous permettre de faire persister vos objets PHP dans votre base de données mais aussi de les récupérer d'une base de données. C'est alors un travail de mapping qu'effectue Doctrine afin de faire correspondre propriété des classes aux colonnes de la base.

Ce qui va aussi vous rendre la tâche simple sont les actions possibles à partir du bundle comme par exemple générer Getters et Setters avec une simple commandes :

Grâce au mapping vous pouvez dans les deux sens, faire régénérer votre base à partir des objets soit généré des objets à partir de votre base.

COMPOSER

Composer est l'élément utile à l'installation de vos bundles donc de Symfony

C'est une bibliothèque de gestion de dépendances pour PHP qui nous permet d'installer des librairies utiles à nos projets.

Si une librairie a besoin d'une autre librairie pour fonctionner alors Composer va aussi la télécharger ce qui rend Composer indispensable dans le cadre d'un projet Symfony.

Pour fonctionner, il est nécessaire d'utiliser PHP 5.3.2 ou supérieur ainsi que certaines applications dépendant de la source de la librairie. J'ai déjà parlé de Git mais d'autres existent aussi tel SVN ou Hg (Mercurial). Ce sont tous les trois des outils de versioning.

TWIG

Twig est un moteur de templates.

Il va vous obliger à séparer la récupération des données en PHP et le code HTML afin d'être beaucoup plus clair. Cela permet entre autres de savoir quel fichier retoucher en cas de problème.

Le fichier d'affichage ne comportera aucune balise PHP et donc bien plus facile à prendre en main pour ceux qui n'y connaissent rien en PHP comme par exemple les graphistes qui n'ont pas forcément les connaissances.

Les avantages de twig sont sa rapidité sa sécurité et sa flexibilité. Il permet entre autres de créer ses propres filtres et ses propres écrans de chargement.

Il est surtout très facile à apprendre et donc accessible à tous.

PHP 5.2.4 est requis pour mettre en place Twig sur votre framework

4.1.5 De l'importance de choisir Symfony

Grâce aux normes et conventions que chaque développeur sur un projet Symfony doit respecter, on obtient une organisation solide des fichiers et du code. Les avantages sont multiples :

- ◆ Les développeurs Symfony qui maîtrisent le framework pourront facilement intégrer un projet développé à partir du framework, contrairement à un projet développé en PHP "maison", où il n'y a pas de normes ni règles imposées. Dans ce dernier cas, la phase d'apprentissage et reprise du code existant peut demander un effort conséquent pour le nouveau développeur intégrant le projet
- ◆ Les fichiers doivent respecter une syntaxe particulière et doivent se trouver au bon endroit dans l'arborescence du projet. Cela garantit une facilité de maintenance sur le long terme, les développeurs savent rapidement dans quel fichier il faut aller pour apporter des modifications.
- ◆ L'architecture MVC (Modèle Vue Contrôleur) permet de découper le code représentant la logique métier de l'application et le code de présentation des vues. Ainsi, un

intégrateur web voire même un webdesigner n'aura aucun mal à intervenir sur la partie présentation (vues) du projet, sans avoir à intervenir sur des fichiers PHP complexes.

- ♦ Favorise la réutilisation de code, la création de tests automatisés (tests unitaires avec PHPUnit ou Atoum et tests fonctionnels avec phpUnit ou Behat) et le respect des recommandations PHP-FIG (Des recommandations mondiales pour une meilleure interopérabilité entre les projets web PHP). Symfony permet donc de produire du code de qualité.

Conclusion

Au cours de ce chapitre, la conception de la plateforme a été détaillée avec l'ensemble des choix et outils technologiques. La phase de conception est terminée, nous sommes prêts donc à entamer la réalisation de notre projet, cette implémentation sera détaillée dans le prochain chapitre.

Chapitre 5 : Implémentation de la solution

Introduction

Ce chapitre constitue le dernier volet de ce rapport, il traite la phase qui a pour objectif l'implémentation de notre application. Nous commençons, tout d'abord, par l'architecture de la solution partie dans laquelle nous allons présenter l'architecture de la solution. Ensuite nous donnons un aperçu sur le travail réalisé.

1. Architecture de la solution

✓ Diagramme de déploiement

Le diagramme de déploiement montre la disposition physique des matériels et la répartition des composants sur ces matériels. Une ressource matérielle est représentée par un nœud. Les nœuds sont connectés entre eux à l'aide d'un support de communication.

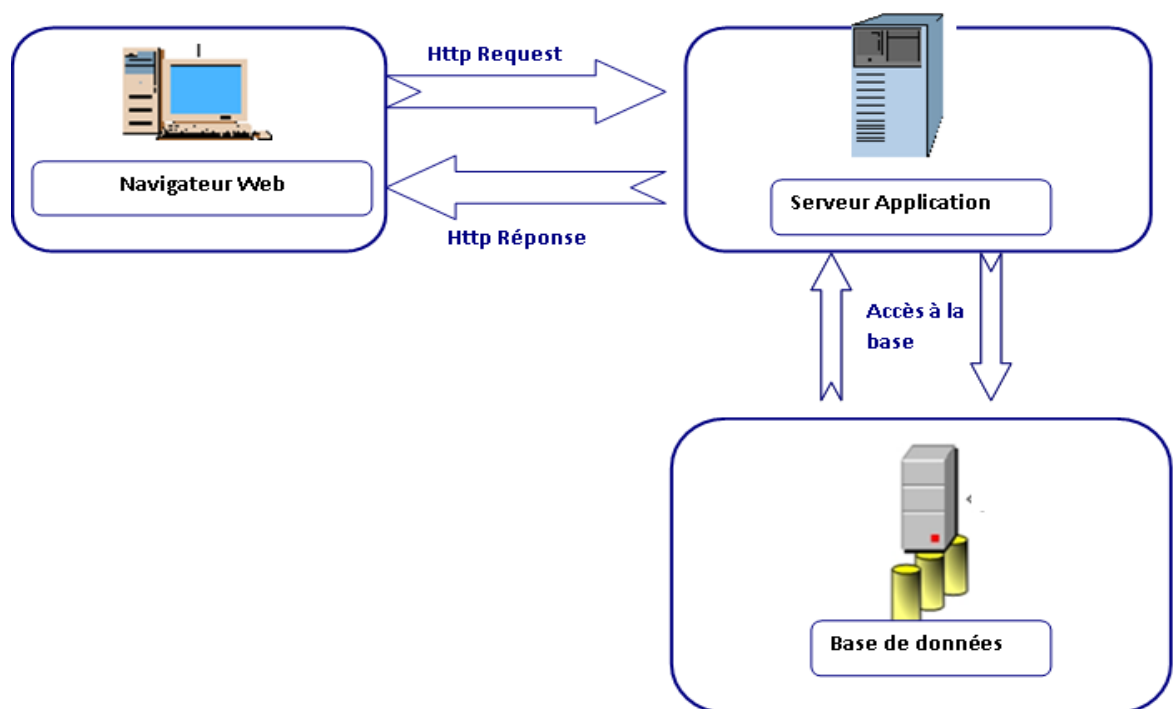


Figure15 : Diagramme de Déploiement

2. Présentation de l'application

Page de Connexion L'utilisateur saisi ses identifiants pour se connecter à la plateforme

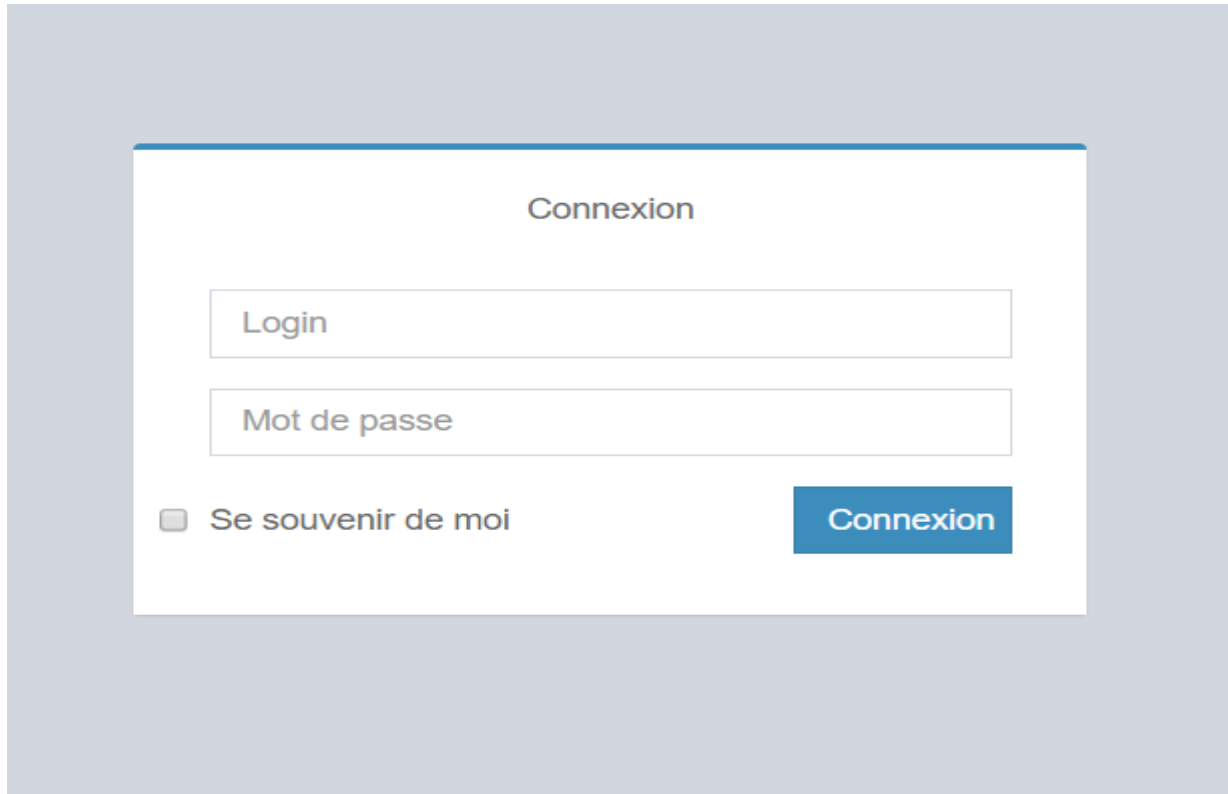
The image shows a login form titled "Connexion" centered on a light gray background. The form is a white rectangle with a thin blue border at the top. It contains two input fields: "Login" and "Mot de passe", both with placeholder text. Below the "Mot de passe" field is a checkbox labeled "Se souvenir de moi". To the right of the checkbox is a blue button with the text "Connexion" in white.

Figure16 : Authentification

Page d'accueil du SIRH, Avec l'Agent RH connecté

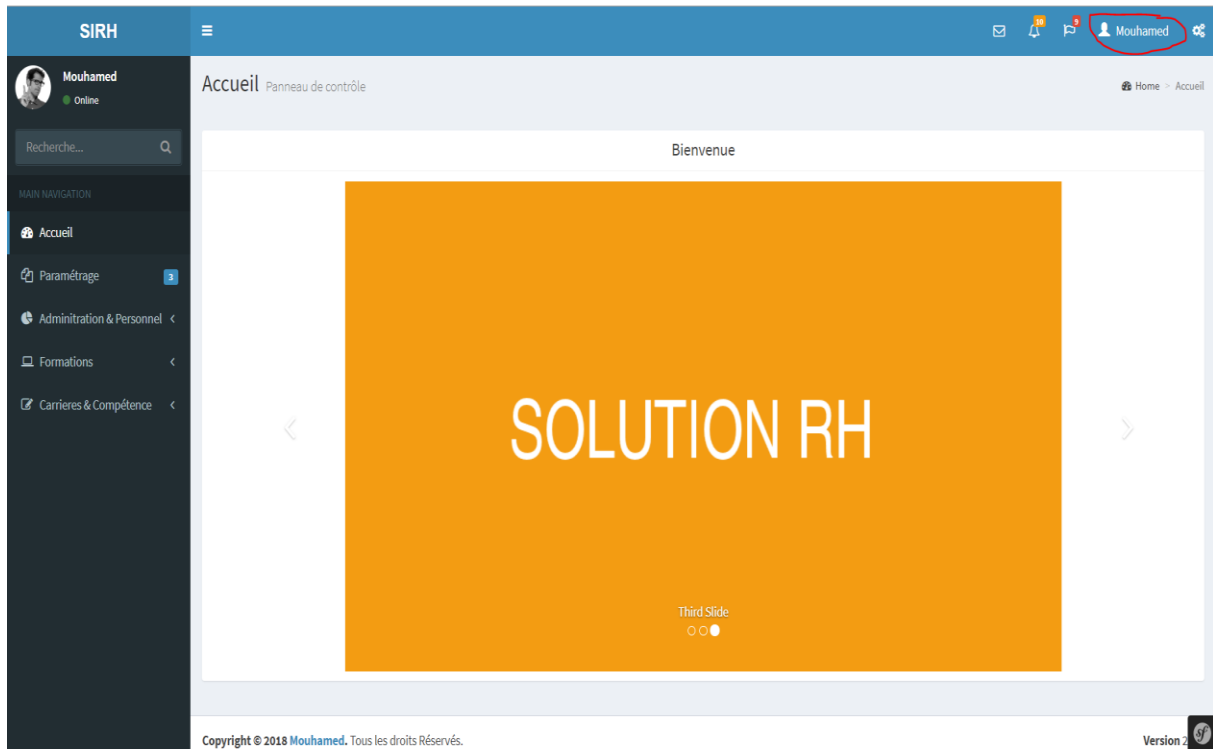


Figure17 : Page d'accueil

Page de Création d'un employé

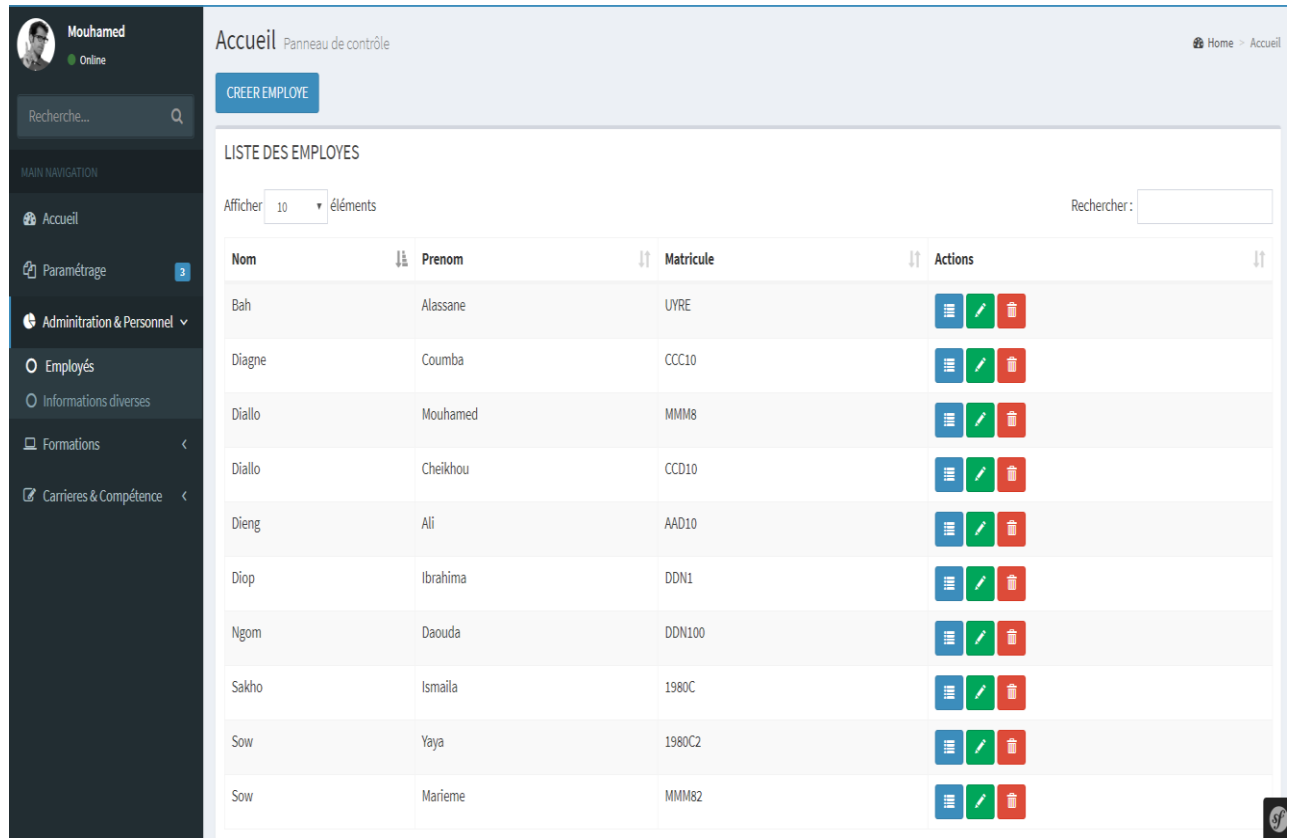
Correspond au Cas d'utilisation Créer Employé décrit lors de l'analyse en [CE].

The screenshot shows the 'CREATION D'EMPLOYE' form. It has a light blue header with the title 'CREATION D'EMPLOYE'. Below the title are several form fields, each with a label and a text input area. The fields are: 'Nom' (Ba), 'Prenom' (Alassane), 'Sexe' (M), 'Matricule' (UYRE), 'Mail' (Alassane@gmail.com), 'Telephone' (778905632), 'Date naissance' (09-09-1991), 'Situation matri' (Marié), 'Date embauche' (20-08-2015), and 'Lieu naissance' (Dakar). Each field is highlighted in yellow. At the bottom left is a blue button labeled 'Creer Employe'. At the bottom right is a small circular logo.

Figure18 : Page de Création d'un Employé

○ **Visualiser Employé**

Nous avons la liste des employés étape correspondant au cas d'utilisation Visualiser Employé.



Accueil Panneau de contrôle

CREER EMPLOYE

LISTE DES EMPLOYES

Afficher 10 éléments

Rechercher :































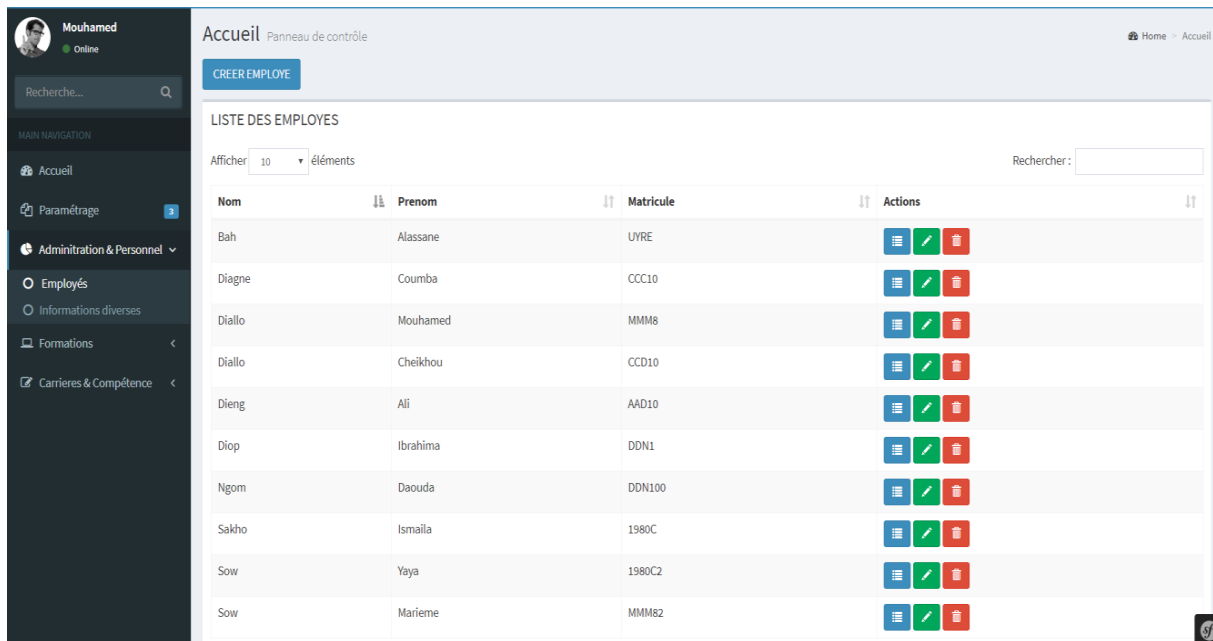
Nom	Prenom	Matricule	Actions
Bah	Alassane	UYRE	  
Diagne	Coumba	CCC10	  
Diallo	Mouhamed	MMM8	  
Diallo	Cheikhou	CCD10	  
Dieng	Ali	AAD10	  
Diop	Ibrahima	DDN1	  
Ngom	Daouda	DDN100	  
Sakho	Ismaila	1980C	  
Sow	Yaya	1980C2	  
Sow	Marieme	MMM82	  

Figure19 : Visualiser la liste des employés

○ Modifier Les informations d'un employé

Nous avons la modification des informations d'un employé qui correspond au cas d'utilisation Modifier Employé.

On constate que le matricule de l'employé Alassane Bah est : URYE



Accueil Panneau de contrôle

CREER EMPLOYE

Recherche...

MAIN NAVIGATION

- Accueil
- Paramétrage
- Administration & Personnel
 - Employés
 - Informations diverses
 - Formations
 - Carrières & Compétence

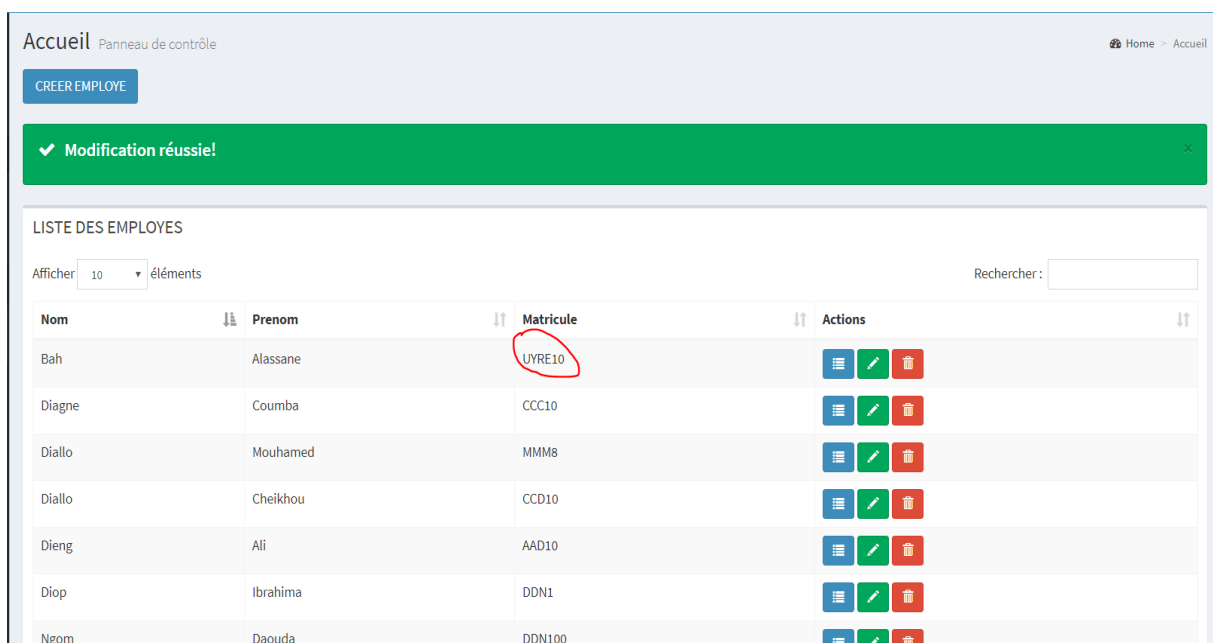
Rechercher :

Afficher 10 éléments

Nom	Prenom	Matricule	Actions
Bah	Alassane	URYE	[Menu] [Edit] [Delete]
Diagne	Coumba	CCC10	[Menu] [Edit] [Delete]
Diallo	Mouhamed	MMM8	[Menu] [Edit] [Delete]
Diallo	Cheikhou	CCD10	[Menu] [Edit] [Delete]
Dieng	Ali	AAD10	[Menu] [Edit] [Delete]
Diop	Ibrahima	DDN1	[Menu] [Edit] [Delete]
Ngom	Daouda	DDN100	[Menu] [Edit] [Delete]
Sakho	Ismaila	1980C	[Menu] [Edit] [Delete]
Sow	Yaya	1980C2	[Menu] [Edit] [Delete]
Sow	Marieme	MMM82	[Menu] [Edit] [Delete]

Figure20 : Avant Modification de la matricule

Après modification on aura le matricule sera UREY10



Accueil Panneau de contrôle

CREER EMPLOYE

✓ Modification réussie!

Rechercher :

Afficher 10 éléments

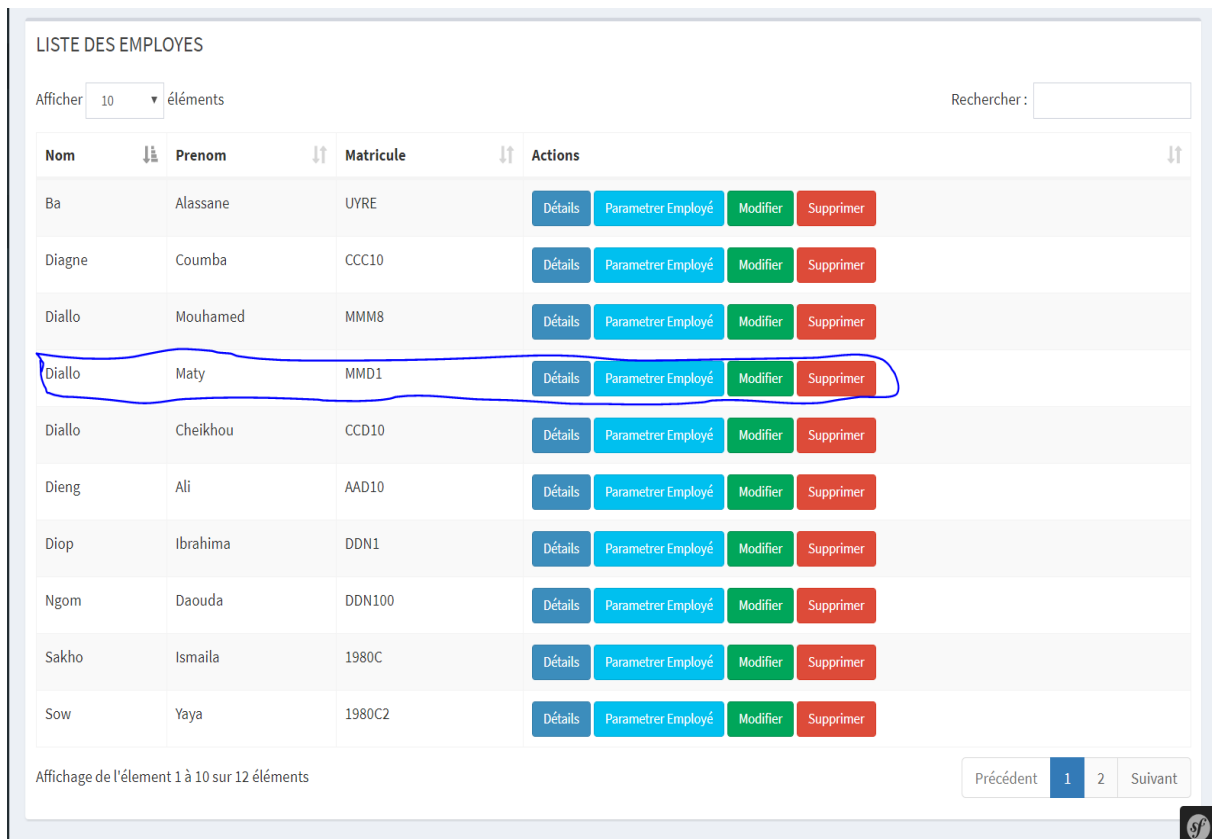
Nom	Prenom	Matricule	Actions
Bah	Alassane	UREY10	[Menu] [Edit] [Delete]
Diagne	Coumba	CCC10	[Menu] [Edit] [Delete]
Diallo	Mouhamed	MMM8	[Menu] [Edit] [Delete]
Diallo	Cheikhou	CCD10	[Menu] [Edit] [Delete]
Dieng	Ali	AAD10	[Menu] [Edit] [Delete]
Diop	Ibrahima	DDN1	[Menu] [Edit] [Delete]
Ngom	Daouda	DDN100	[Menu] [Edit] [Delete]

Figure21 : Apres Modification de la matricule

◆ Suppression

Nous avons la suppression d'un employé qui correspond au cas d'utilisation Supprimer un Employé.

Suppression de l'employé Maty Diallo



Nom	Prenom	Matricule	Actions
Ba	Alassane	UVRE	Détails Parametrer Employé Modifier Supprimer
Diagne	Coumba	CCC10	Détails Parametrer Employé Modifier Supprimer
Diallo	Mouhamed	MMM8	Détails Parametrer Employé Modifier Supprimer
Diallo	Maty	MMD1	Détails Parametrer Employé Modifier Supprimer
Diallo	Cheikhou	CCD10	Détails Parametrer Employé Modifier Supprimer
Dieng	Ali	AAD10	Détails Parametrer Employé Modifier Supprimer
Diop	Ibrahima	DDN1	Détails Parametrer Employé Modifier Supprimer
Ngom	Daouda	DDN100	Détails Parametrer Employé Modifier Supprimer
Sakho	Ismaila	1980C	Détails Parametrer Employé Modifier Supprimer
Sow	Yaya	1980C2	Détails Parametrer Employé Modifier Supprimer

Figure22 : Visualiser l'employé à Supprimer

Après suppression on constate que l'employé ne figure plus sur la liste des employés

🚫 Employé Supprimé!
×

LISTE DES EMPLOYES

Afficher 10 éléments
Rechercher :

Nom	Prenom	Matricule	Actions
Ba	Alassane	UYRE	Détails Parametrer Employé Modifier Supprimer
Diagne	Coumba	CCC10	Détails Parametrer Employé Modifier Supprimer
Diallo	Mouhamed	MMM8	Détails Parametrer Employé Modifier Supprimer
Diallo	Cheikhou	CCD10	Détails Parametrer Employé Modifier Supprimer
Dieng	Ali	AAD10	Détails Parametrer Employé Modifier Supprimer
Diop	Ibrahima	DDN1	Détails Parametrer Employé Modifier Supprimer
Ngom	Daouda	DDN100	Détails Parametrer Employé Modifier Supprimer
Sakho	Ismaila	1980C	Détails Parametrer Employé Modifier Supprimer
Sow	Yaya	1980C2	Détails Parametrer Employé Modifier Supprimer
Sow	Marieme	MMM82	Détails Parametrer Employé Modifier Supprimer

Figure23 : Apres Suppression de l'employé Maty Diallo

♦ Ajout d'un Conjoint à un Employé

Créer un Conjoint

Nom du Conjoint:

Sy

Prenom du Conjoint:

Ndeye

Date de Naissance:

11-07-2018

Situation Matrimoniale:

Marié

Raison Inactif:

Congés

Nom:

Inactif

Annuler

Enregistrer

Figure24 : Formation d'ajout d'un conjoint à un employé

✓ Conjoint Ajouté

Détails de l'Employé Mouhamed Sy

Informations Personnelles

Sexe :	M
Matricule :	SS90
Mail :	SY@gmail.com
Telephone :	77895640
Date de Naissance :	06/20/2018
Situation Matrimoniale :	Célibataire
Date D'embauche :	08/27/2014
Lieu de Naissance :	Saint Louis
Nationalité :	Algérien
Conjoint :	Ndeye Sy

Informations Professionnelles

Figure25 : Apres ajout de la conjointe

◆ **Ajout d'une Fonction à un Employé**

Détails de l'Employé Marieme Sow

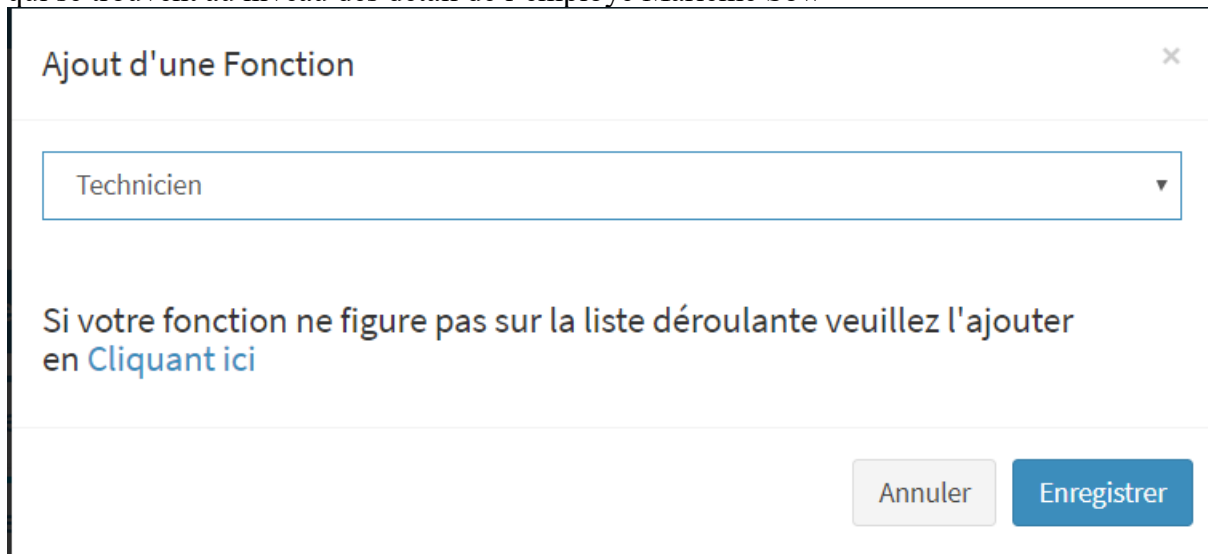
Informations Personnelles

Informations Professionnelles

Fonction	Metier	Etat	Contrat	Catégorie	Service
<div>Ajouter Fonction</div>					

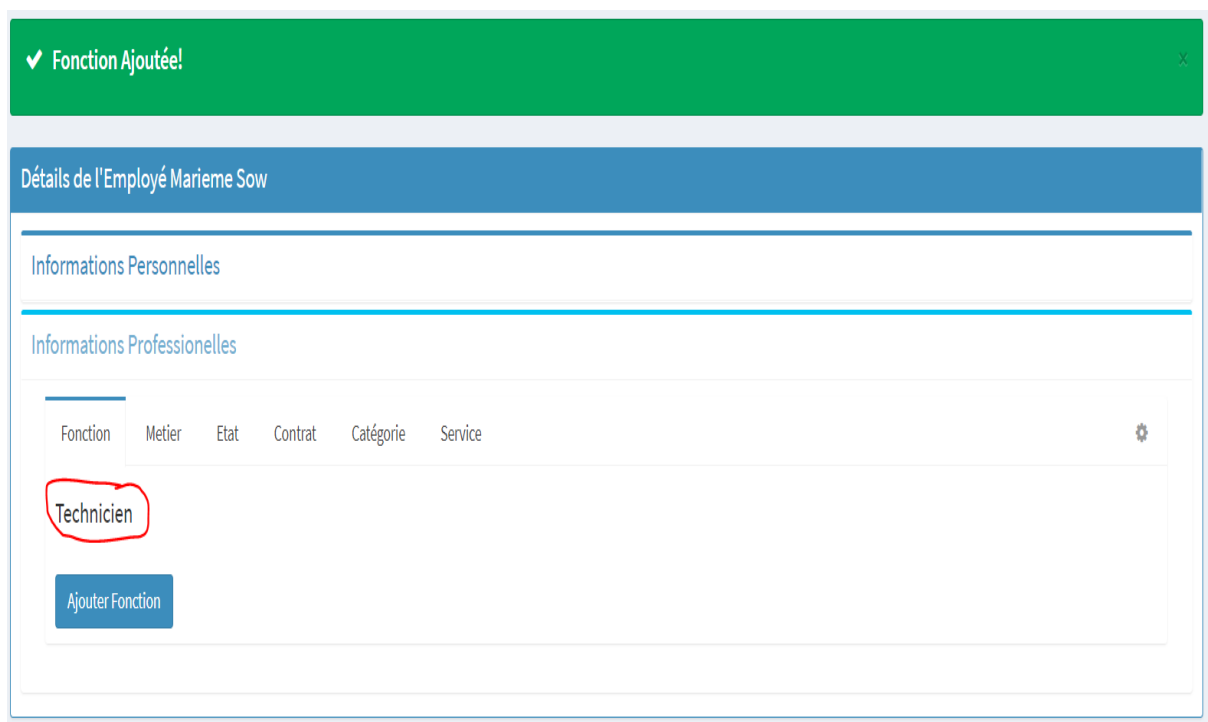
Figure26 : Détail de l'employé Marieme Sow

Nous allons ajouter une fonction à l'employé Marieme Sow en cliquant sur Ajouter Fonction qui se trouvent au niveau des détail de l'employé Marieme Sow



A modal form titled "Ajout d'une Fonction" with a close button (X) in the top right corner. It features a dropdown menu with "Technicien" selected. Below the dropdown, a message states: "Si votre fonction ne figure pas sur la liste déroulante veuillez l'ajouter en [Cliquant ici](#)". At the bottom right, there are two buttons: "Annuler" (grey) and "Enregistrer" (blue).

Figure27 : formulaire d'Ajout de fonction



The screenshot shows the "Détails de l'Employé Marieme Sow" page. At the top, a green banner displays a success message: "✓ Fonction Ajoutée!". Below this, the page is divided into "Informations Personnelles" and "Informations Professionnelles". The "Informations Professionnelles" section contains a table with columns: Fonction, Metier, Etat, Contrat, Catégorie, and Service. The "Fonction" column has a red circle around the value "Technicien". Below the table is a blue button labeled "Ajouter Fonction".

Figure28 : Fonction Ajouté avec Succès

L'ajout de Métier, Etat, Contrat, Catégorie et service se fait de la même façon que l'ajout de la fonction.

Conclusion

A ce stade, notre projet d'études atteint sa fin. Tout au long de ce chapitre, nous avons commencé par l'architecture de la solution ensuite par présenter les différentes principales interfaces de notre application réalisée.

CONCLUSION

Notre projet a été réalisé dans le cadre d'un projet de fin d'études qui a pour objectif la mise en place d'une application de gestion des ressources humaines.

Notre travail se résume en la conception et la réalisation d'une application permettant la gestion des ressources humaines. Dans ce contexte, nous avons cherché à développer une application flexible et évolutive permettant une amélioration de la gestion des salariés de l'Entreprise Kya's.

Cette application va permettre, en premier lieu, d'améliorer l'administration du personnel, l'évaluation du personnel, la gestion des carrières et compétences ainsi que la gestion de la formation du personnel.

Pour l'analyse et la conception de notre application, nous avons eu recours à la méthodologie Scrum. Cette approche nous a permis de bien comprendre la problématique et de bien modéliser les objectifs à atteindre. Ceci nous a donné la possibilité de réaliser un système stable et évolutif.

Le projet s'est déroulé selon trois axes principaux afin de passer par les étapes essentielles de tout projet : l'analyse, la conception et la réalisation. Pour la réalisation, nous avons utilisé Symfony dans sa version 3 comme framework qui est un framework PHP et MySQL comme système de gestion de base de données.

En outre, ce projet était une opportunité pour bien maîtriser le développement web et apprendre le framework Symfony.

En guise de perspective, nous comptons améliorer ce travail en ajoutant certaines fonctionnalités telles que gérer le recrutement d'employé, la gestion de la paie.

⇒ **Webographie**

[W 1] => [https://fr.wikipedia.org/wiki/UML \(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique)) Consulter Juin 2018

[W2] => <https://www.novaway.fr/notre-expertise/symfony> consulter Juillet 2018

[W3] => <https://www.agiliste.fr/introduction-methodes-agiles/> Consulter le 12 juillet 2018