

N° d'ordre:

N° de série:

République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université d'El-Oued

Faculté de Sciences et de Technologie

Mémoire fin de cycle présenté en vue de l'obtention du diplôme de

LICENCE ACADEMIQUE

Domaine Mathématiques et Informatique

Département d'Informatique

Filière : Informatique

Thème

**Conception et réalisation d'une application réseau
de gestion de personnel**

Présenté par:

Bebboukha Meriem

Hafouda Chaima

Encadré par:

Hadjadj Ismail

Devant le Jury composé de:

- Yagoub Mohamed Amine
- Djalid Raouf

Année Universitaire 2013/2014

Remerciements

Nos remerciements s'adressent à tout le personnel d'université d'el-oued au niveau des différents départements, où on a effectué ce projet qu'ils trouvent ici l'expression de nos respects

Nous souhaitons encore dire un très grand merci à notre encadreur Mrs **Ismail Hadjadj** pour sa participation et ses conseils prodiges à la réalisation de ce travail

Nous remercierons particulièrement Monsieur Hocine Soltane, et à tous nos collègues en particulier Mrs Yahia Khecha et l'ingénieur d'informatique Ahmed Souda qui nous ont aidé énormément et avec qui nous avons travaillé et qui ont pu le mieux répondre à toutes nos attentes et nos questions

Et enfin, Nos profonds remerciements et notre gratitude vont aussi aux Membres du jury, pour leur accord de juger notre travail

Merci à tous

يتلخص مشروعنا في إنشاء برنامج يسمح بالتسيير الآلي للموظفين لتسهيل عملية البحث والترقية والعطل والمر دودية والعقوبات والغيبات, ففي حالة مراقبة معلومات احد الموظفين عملية البحث في البرنامج تكون على حسب رقم الموظف.

Résumé :

Notre projet consiste à implémenter une application automatisant le processus manuel de gestion des employeurs pour faciliter l'opération de la recherche , l'avancement, les congés ,les rendements , les sanctions et les absences.

.

Il permet de créer et mettre à jour les fiches des opérateurs . Dans les cas de contrôle des informations de tel opérateur, le système effectue la recherche selon le matricule de employeur.

Sommaire :

| | |
|---|-----|
| ملخص..... | II |
| Résumé :..... | III |
| Sommaire :..... | IV |
| Liste des figure : | V |
| Liste du tableaux:..... | VII |
| INTRODUCTION GENERALE | 1 |
| 1. Introduction générale..... | 1 |
| 2. Le Problème:..... | 2 |
| CONCEPTION GESTION DE PERSONNEL | 3 |
| 3. Introduction..... | 3 |
| 4. UML (Unified Modeling Language) [5] | 3 |
| 5. Les diagrammes de l’UML :..... | 4 |
| 6. Le Processus de développement | 4 |
| 7. Processus Unifié (Unified Processus): (UP)..... | 4 |
| 8. Branche architecture technique (droite): | 6 |
| 9. Conception détaillée:..... | 21 |
| REALISATION..... | 24 |
| CONCLUSION GENERALE | 28 |
| Bibliographiques | 29 |

Liste des figure :

| | |
|---|----|
| Figure 1:Organigramme de l'UML | 4 |
| Figure2 :Les principes de Processus Unifié | 5 |
| Figure 3:Le processus de développement en Y | 6 |
| Figure 4:Le diagramme de cas d'utilisation..... | 9 |
| Figure5 :Diagramme de cas d'utilisation « Authentification »..... | 9 |
| Figure 6:Diagramme d'activité «Authentification» | 10 |
| Figure 7:Diagramme de séquence «Authentification» | 10 |
| Figure 8:Diagramme de cas d'utilisation «editer_personnel» | 11 |
| Figure 9:Diagramme d'activité « Ajouter » | 11 |
| Figure 10:Diagramme de séquence «ajouter » | 11 |
| Figure 11:Diagramme d'activité «Modifier» | 12 |
| Figure 12:Diagramme de séquence «Modifier»..... | 12 |
| Figure 13:Diagramme de cas d'utilisation «Avancement» | 13 |
| Figure 14:Diagramme d'activité «Avancement» | 13 |
| Figure 15:Diagramme de séquence «Avancement»..... | 13 |
| Figure 16:Diagramme de cas d'utilisation «congé»..... | 14 |
| Figure 17:Diagramme d'activité «congé» | 14 |
| Figure 18:Diagramme de séquence «congé» | 14 |
| Figure 19::Diagramme de cas d'utilisation «sanction»..... | 15 |
| Figure 20:Diagramme d'activité «sanction» | 15 |
| Figure 21:Diagramme de séquence «sanction» | 15 |
| Figure 22:Diagramme de cas d'utilisation « Absence» | 16 |
| Figure 23:Diagramme d'activité « Absence» | 16 |
| Figure 24:Diagramme de séquence «Absence»..... | 16 |

| | |
|--|----|
| Figure 25:Diagramme de cas d'utilisation «randement»..... | 17 |
| Figure 26:Diagramme d'activité «randement» | 17 |
| Figure 27:Diagramme de séquence «rendement» | 17 |
| Figure 28:Diagramme de classe «Ajouter, Modifier information de personnel »..... | 20 |
| Figure 29:Interface d'authentification. | 26 |
| Figure 30:Interface Accueil..... | 27 |
| Figure 31:Interface d'information. | 27 |

Liste du tableaux:

| | |
|---|----|
| Tableau 1:Identification des acteurs | 8 |
| Tableau 2:Identification des acteurs par cas d'utilisation | 8 |
| Tableau 3:Liste des classes..... | 21 |
| Tableau 4:Equivalences entre les concepts objets et relationnels..... | 22 |
| Tableau 5:Liste des tables de la base de données..... | 23 |

Introduction générale

Face à l'individualisation de plus en plus poussée de la gestion du personnel, les administrations doivent trouver les outils nécessaires pour la gestion des personnels qui s'automatise pour optimiser les ressources humaines. Dans ce but, les administrations s'équipent du logiciel permettant une gestion fine des personnels.[1]

Le contrôle des employés est une problématique classique du monde de l'emploi, il est progressivement adopté par toutes les administrations, et particulièrement par l'organisme du La cour des organisations de jeunesse (Wilaya d'El- oued) qui comporte plus concerne les affaires des ouvriers et employés. Mieux encore, la mise en œuvre des applications de gestion des employés.

Le système de gestion du personnel est une solution plus au moins adoptée à ce problème car cette application va nous permettre de gérer les différentes catégories d'employés (ouvriers, cadre,...) ainsi que leurs avancement et congé, sanction, absence et rendement.

Le Problème:

Dans la vie professionnelle la gestion du personnel pose un grand problème que ce soit pour les organismes ou pour les employés car l'opération de gérer les personnels du calcul d'avancement et concerne le congé, sanction, absence, rendement, Ce qui cause des problèmes de différents types.

Solution Proposée :

Notre application a pour objectif de :

Gérer les ressources humaines existant au sein de la cour des organisations de jeunesse :

- Ajouter et Modifier un nouveau personnel et congé, rendement, sanction et absence.

- Calculer l'avancement de chaque employé. En comptant date avancement et durée avancement à chaque Personnel.

Ce rapport est structuré en deux parties:

La première partie conception est consacrée à la conception où on décrit l'ensemble des diagrammes UML selon la démarche 2TUP relatifs à l'application de notre projet.

Enfin, partie réalisation est consacré à l'exposé de la solution implémentée ainsi que l'environnement logiciel et matériel utilisé.

Je termine par une conclusion générale par laquelle ou je dresse un plan du travail effectué ainsi que les problèmes restant ouverts.

Introduction

Dans ce chapitre nous allons procéder à la conception de notre logiciel gestion personnel. Pour cela, on a opté pour l'utilisation d'UML, qui est un langage de modélisation unifié (Unified Modeling Language). Pour ce faire, on a organisé le chapitre suivant deux parties. La première partie est consacrée à la présentation du langage UML, ses notions de bases ainsi que ses différentes vues et diagrammes. Dans la deuxième partie on entame la conception de notre logiciel avec UML

Partie I : Présentation d'UML:

UML (Unified Modeling Language) [5]

L'UML est un langage standard de modélisation orienté objet. Il permet de construire plusieurs modèles d'un système : certains montrent le système du point de vue des utilisateurs, d'autres montrent sa structure interne, d'autres encore en donnent une vision globale ou détaillée. Les modèles se complètent et peuvent être assemblés.

Ils sont élaborés tout au long du cycle de vie du développement d'un système (depuis Le recueil des besoins jusqu'à la phase de conception).

La naissance d'UML commençait lorsque la programmation par objets prend de l'importance au début des années 1990 et 1995 (Booch, Classe-Relation, Fusion, HOOD, OMT, OOD, OOM, OOSE, etc.) mais aucune ne parvient à s'imposer. En 1994, le consensus se fait autour de trois méthodes :

- ✓ OMT de James Rumbaugh (*General Electric*) fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système ;
- ✓ OOSE d'Ivar Jacobson (*Ericsson*) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases). Les constructeurs de ces méthodes (les trois Amigos) se mirent d'accord pour définir une méthode commune qui fédérerait leurs apports respectifs. [2]
- ✓ OOSE d'Ivar Jacobson (*Ericsson*) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases). Les constructeurs de ces méthodes (les trois Amigos) se mirent d'accord pour définir une méthode commune qui fédérerait leurs apports respectifs. [2]

Les diagrammes de l'UML :

UML 2.0 comporte treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes (diagrammes structurels ou statiques, diagrammes comportementaux ou dynamiques). [2]

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle. Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) et véhicule une sémantique précise (il offre toujours la même vue d'un système). [4]

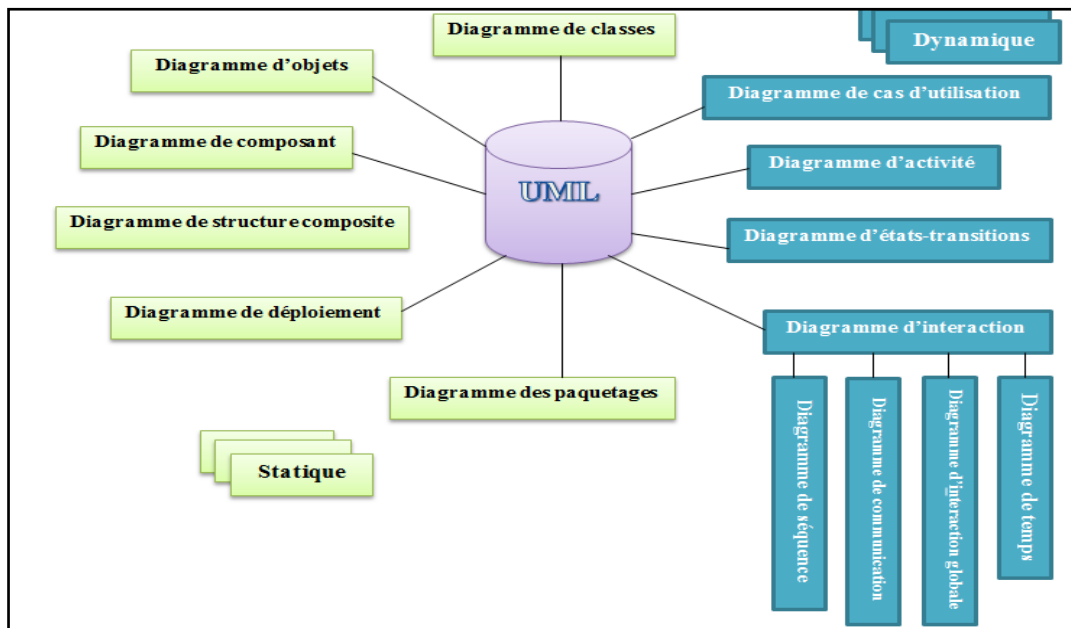


Figure 1: Organigramme de l'UML.

Le Processus de développement

Un processus définit une séquence d'étapes, en partie ordonné, qui concourt à l'obtention d'un système logiciel ou à l'évolution d'un système existant. Pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles. On découpe le processus en deux axes :

- L'axe de développement technique, qui se concentre principalement sur la qualité de production.
- L'axe de gestion du développement, qui permet la mesure et la prévision des coûts et des délais. [3]

Processus Unifié (Unified Process): (UP)

3.1.1. Définition d'UP :

Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. [4]

3.1.2. Les caractéristiques essentielles du processus unifié :

- Le processus unifié utilise le langage UML (ensemble d'outils et de diagramme),
- Le processus unifié est piloté par les cas d'utilisation,
- Centré sur l'architecture,
- Itératif et incrémental.

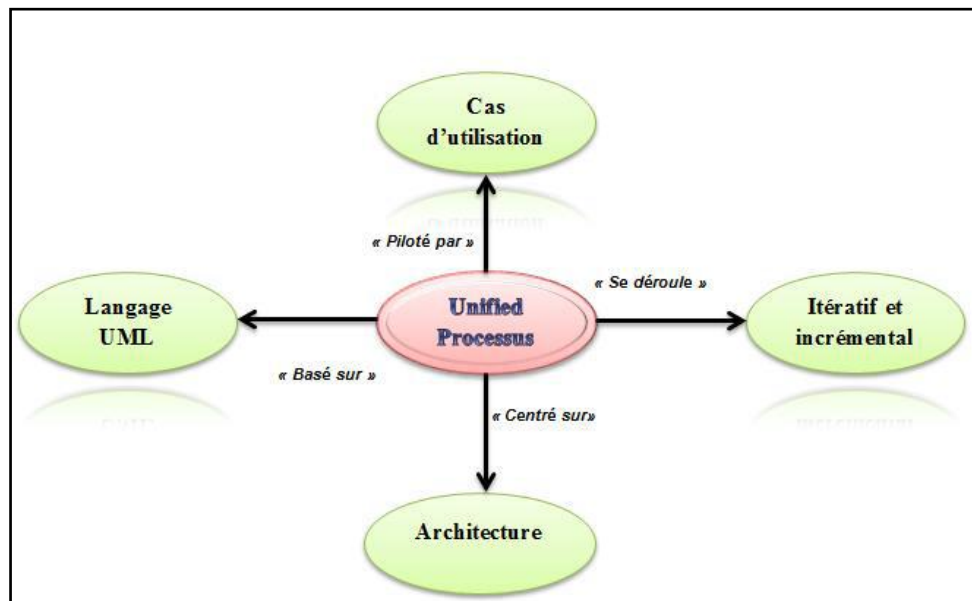


Figure2 :Les principes de Processus Unifié.

3.1.3. Définition :

Le processus 2TUP (2Tracks Unified Processus) apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. [3]

Idée de base du 2TUP : toute évolution imposée au SI peut se décomposer et se traiter parallèlement, suivant 2 axes (« 2 tracks ») : Un axe fonctionnel et Un axe technique . La réalisation du système consiste à fusionner les résultats des deux branches. A l'issue des évolutions du modèle fonctionnel et de l'architecture technique, Cette fusion conduit à l'obtention d'un processus de développement en forme de Y comme illustré dans la figure.II.02 ci-dessous: [4]

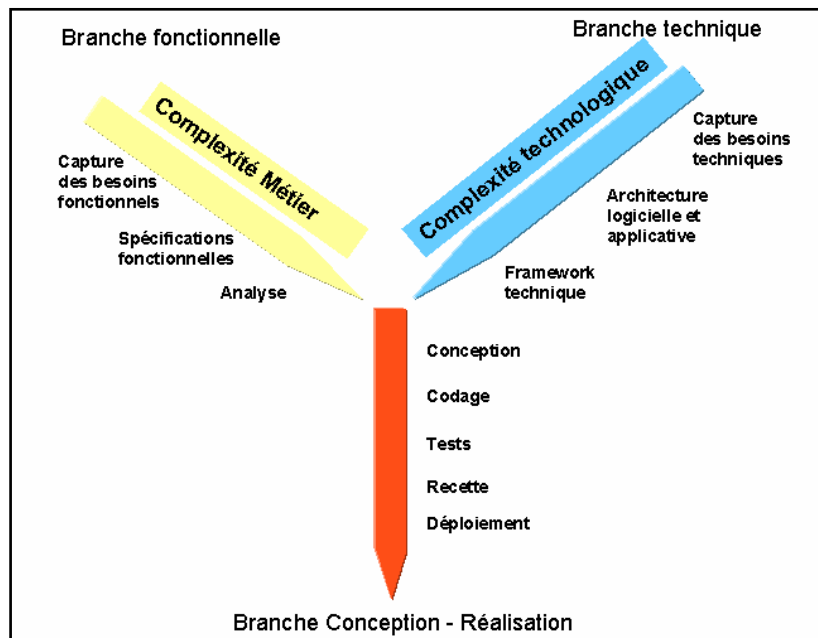


Figure 3: Le processus de développement en Y.

Les branches du 2TUP

La méthode de développement (UML+2TUP).

Branche architecture technique (droite):

Capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser. Cette branche comporte les étapes suivantes :

- La capture des besoins techniques.
- La conception générique. [5]

✓ Capture des besoins techniques :

Qui recense toutes les contraintes sur les choix de dimensionnant et la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique). [3]

✓ Conception générique :

Qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnel. Elle a pour objectif de d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype. [3]

c. Branche conception (milieu) :

A l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l'obtention d'un processus en forme d'Y. [5]

Cette branche comporte les étapes suivantes :

- La conception préliminaire.
- La conception détaillée.
- Le codage.
- L'intégration. La méthode de développement (UML+2TUP)

✓ **Conception préliminaire :**

Qui représente une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer. [3]

✓ **La conception détaillée :** qui étudie ensuite comment réaliser chaque composant. [6]

✓ **L'étape de codage :**

Qui produit ses composants et teste au fur et à mesure les unités de code réalisées. [3]

✓ **L'étape de recette :**

Qui consiste enfin à valider les fonctionnalités du système développé. [3]

3.2.1. Etapes de méthode 2TUP

3.2.1.1. Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

3.2.1.2. Identification des acteurs :

| Acteur | Type |
|----------------|------------|
| Chef-Personnel | Principale |
| Agent | Secondaire |

Tableau 1: Identification des acteurs

Pour administrer la base, Chef-personnel doit tout d'abord s'identifier (saisir son login et mot de passe), puis le système vérifie ces paramètres, ensuite le système lui attribut un ensemble de privilèges, enfin l'administrateur peut agir avec la base selon ses droits donnés.

3.2.1.3. Identification des cas d'utilisation :

Le tableau ci-dessous nous présente les acteurs ainsi que les cas d'utilisation que les appartiennent.

| Acteur | Cas d'utilisation |
|----------------|--|
| Agent | <ul style="list-style-type: none"> - Editer_Personnel - gerer_Congé - gerer_Absance |
| Chef-Personnel | <ul style="list-style-type: none"> - Editer_personnel - Editer_Avancement - gerer_Congé - calculer_Randement - gerer_Sanction -gerer_Absance |

Tableau 2: Identification des acteurs par cas d'utilisation

3.2.1.4. Le diagramme de cas d'utilisation.

Dans ce diagramme, nous allons présenter les différents acteurs ainsi que les cas d'utilisation qui sont liés à ces acteurs, aussi on va présenter les relations entre les cas d'utilisation eux même.

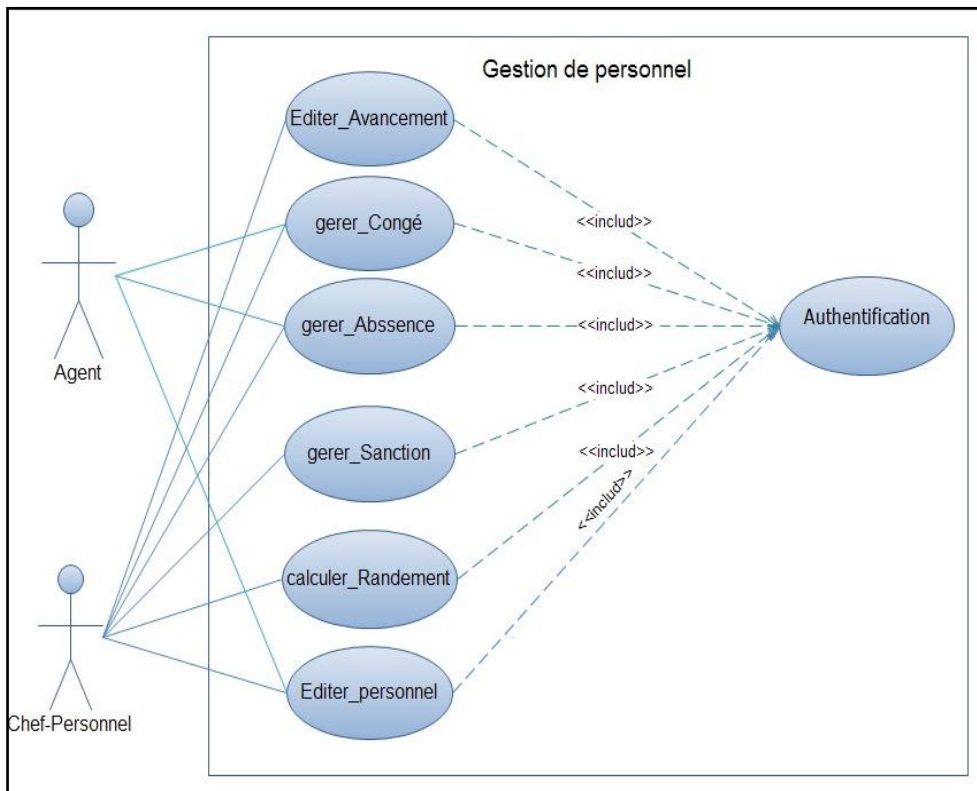


Figure 4:Le diagramme de cas d'utilisation

3.2.1.4.1. Cas d'utilisation «Authentification» :

Description des scénarios:

Pré condition : l'administrateur possède un login et un mot de passe

1. Scénario nominal : 1.Saisir le Login et le mot de passe
 2. Vérifier les coordonnées saise .
 3. Accès au système .

2. Scénario alternatif : Saisi erroné

Enchaînement au point 3

Reprise au point 1

Post Condition : Connexion au système établi

❖ Diagramme de cas d'utilisation « Authentification »:

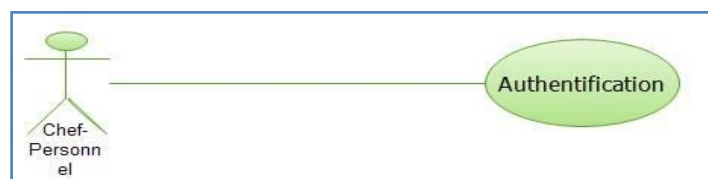


Figure5 :Diagramme de cas d'utilisation « Authentification »

❖ Diagramme d'activité « Authentification »:

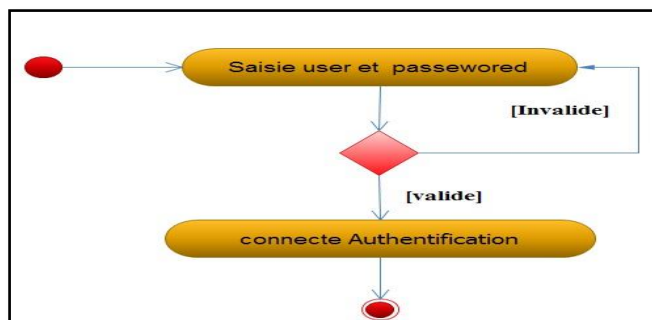


Figure 6:Diagramme d'activité «Authentification».

❖ Diagramme de séquence« Authentification »:

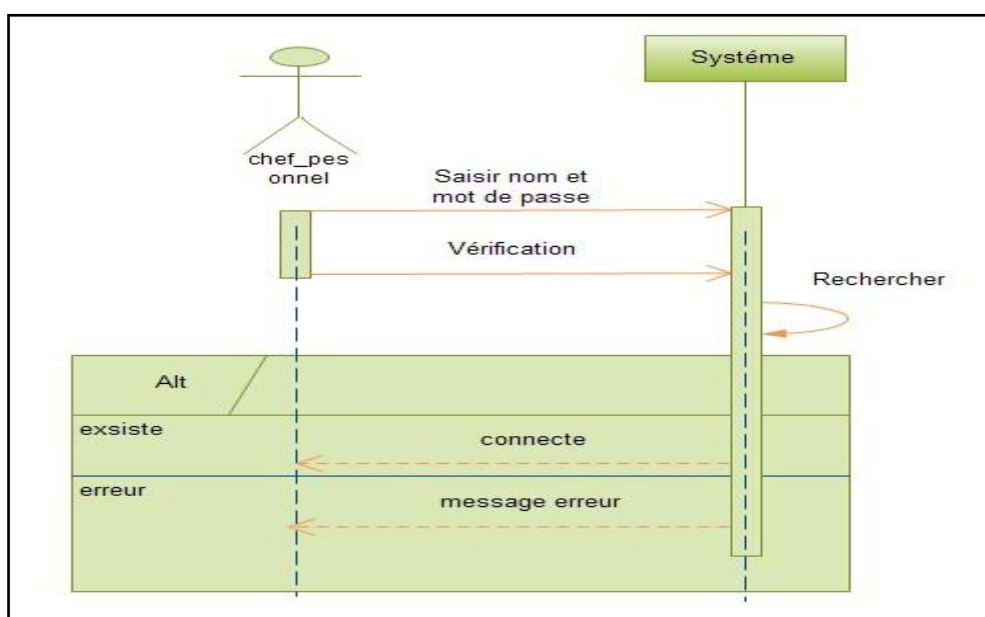


Figure 7:Diagramme de séquence «Authentification».

❖ Cas d'utilisation «éditer-personnel» :

❖ Intention : mise à jour l'info de personnel.

❖ Actions :

- ✚ Ajouter.
- ✚ Modifier.
- ✚ Archivage.

❖ Diagramme de cas d'utilisation «éditer-personnel»:

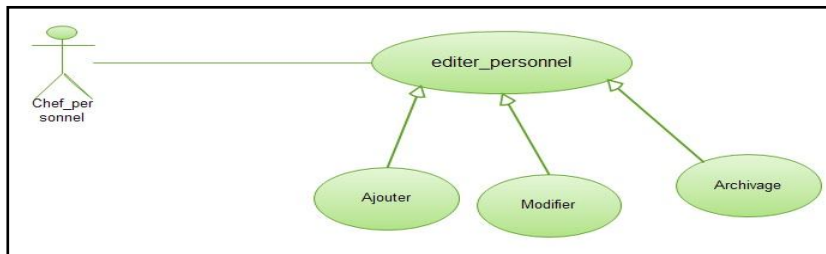


Figure 8:Diagramme de cas d'utilisation «editer_personnel»

❖ Diagramme d'activité « Ajouter ».

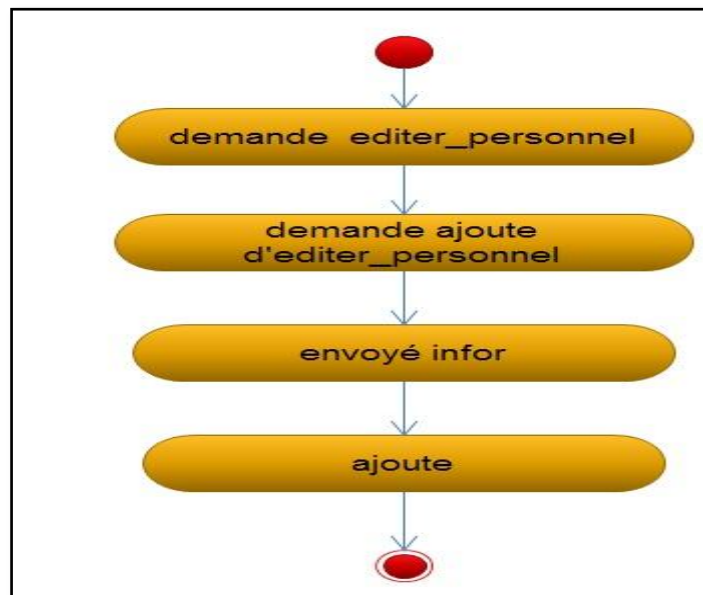


Figure 9:Diagramme d'activité « Ajouter ».

❖ Diagramme de séquence «Ajouter»

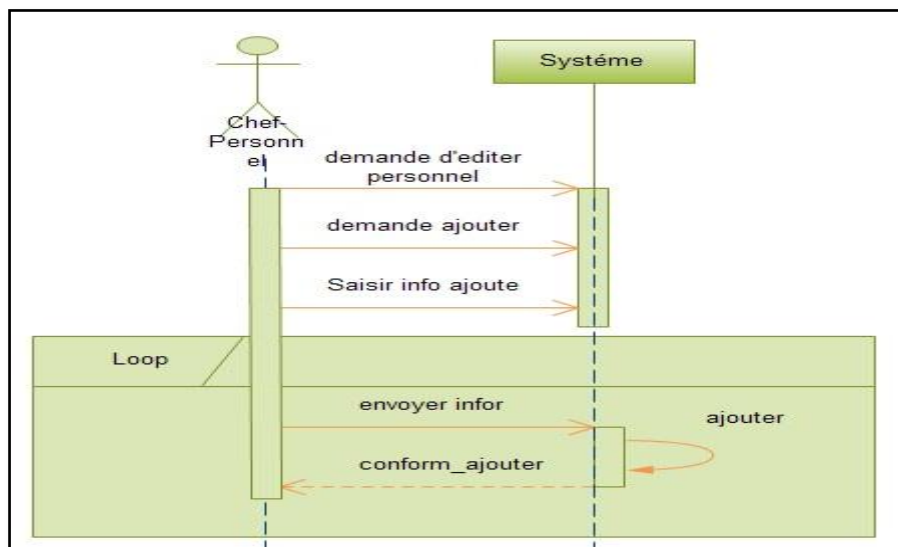


Figure 10:Diagramme de séquence «ajouter ».

❖ Diagramme d'activité «Modifier»

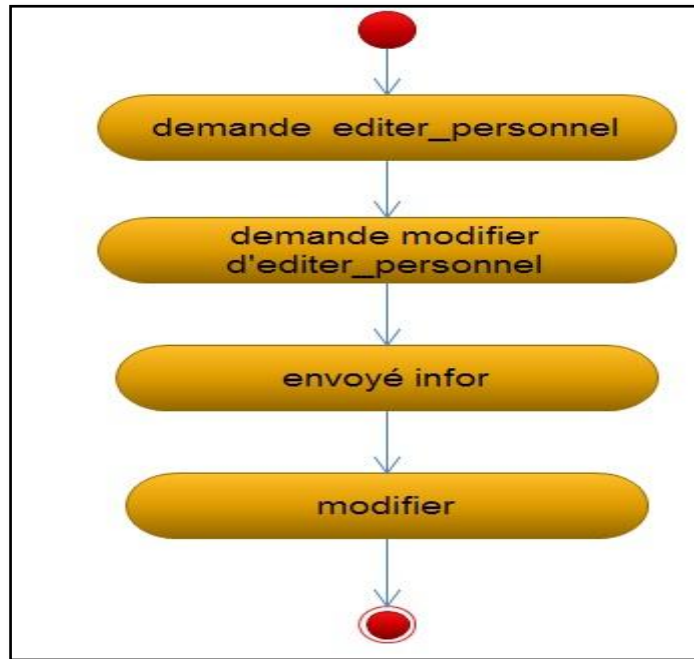


Figure 11:Diagramme d'activité «Modifier»

❖ Diagramme de séquence : «Modifier»

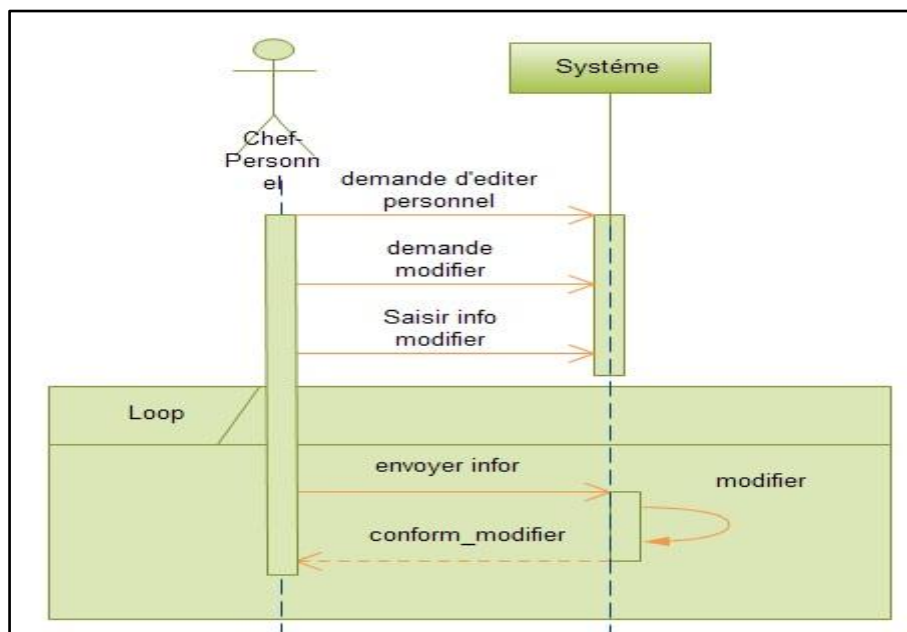


Figure 12:Diagramme de séquence «Modifier»

❖ Diagramme de cas d'utilisation «Avancement»

❖ Intention : mise a jour d'Avancement.

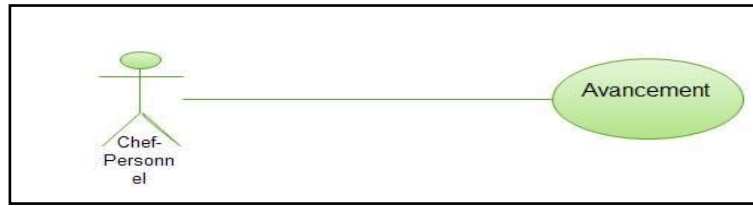


Figure 13: Diagramme de cas d'utilisation «Avancement»

❖ Diagramme d'activité «Avancement»

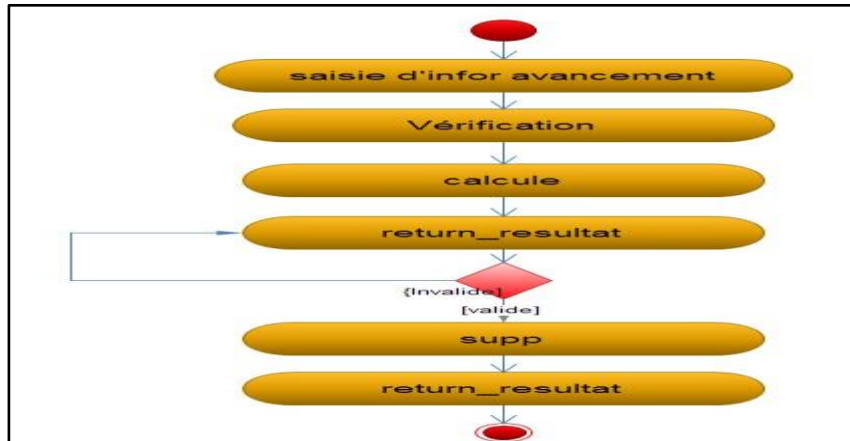


Figure 14: Diagramme d'activité «Avancement»

❖ Diagramme de séquence «Avancement»

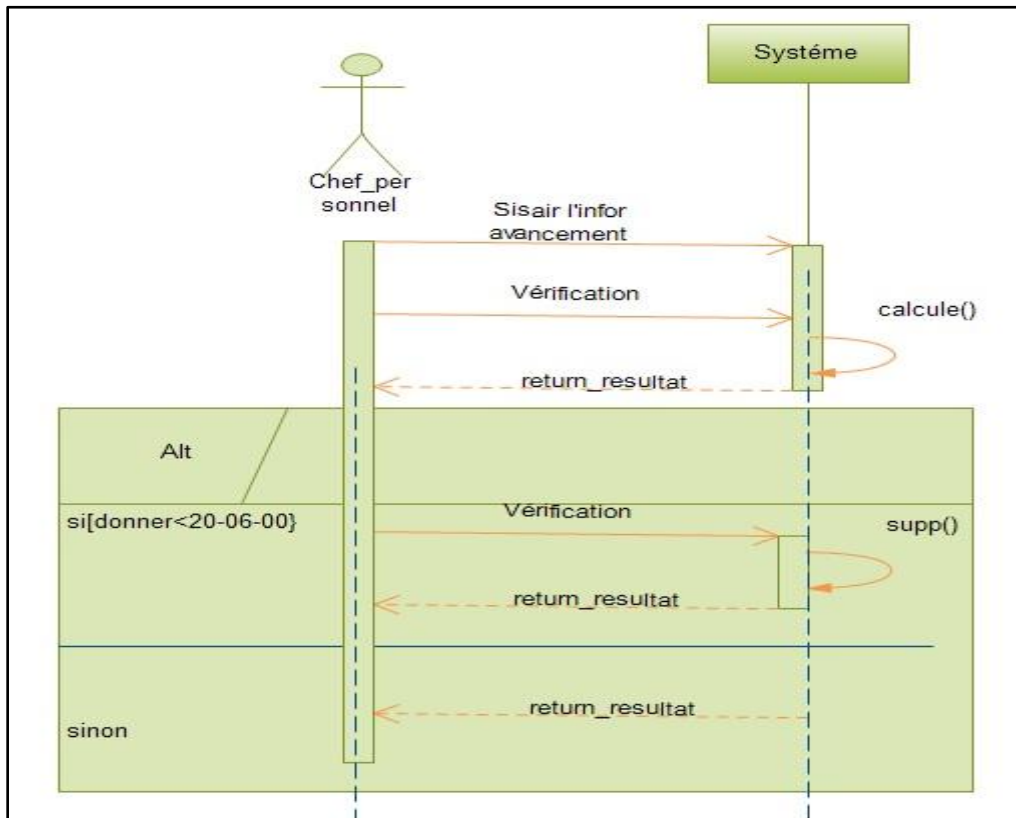


Figure 15: Diagramme de séquence «Avancement»

❖ **Diagramme de cas d'utilisation «congé» :**

❖ Intention : mise à jour **congé**

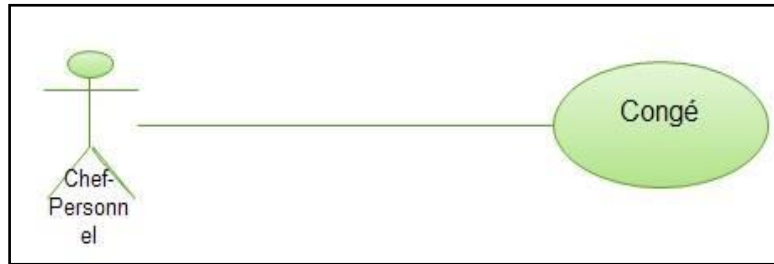


Figure 16:Diagramme de cas d'utilisation «congé»

❖ **Diagramme d'activité «congé»**

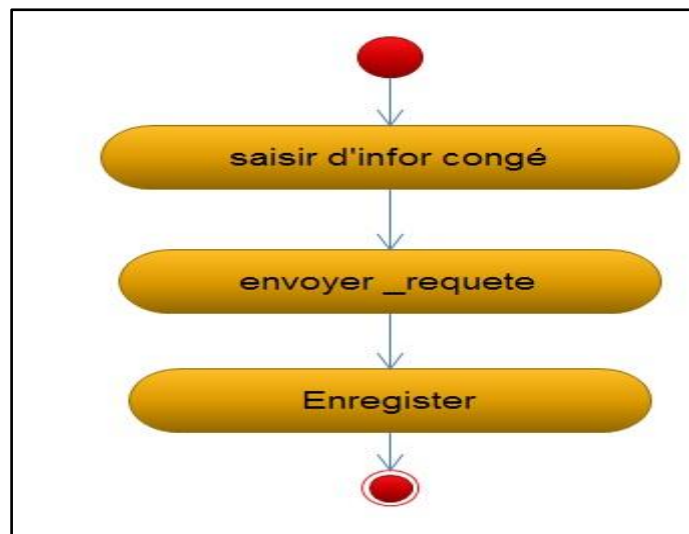


Figure 17:Diagramme d'activité «congé»

❖ **Diagramme de séquence « congé »**

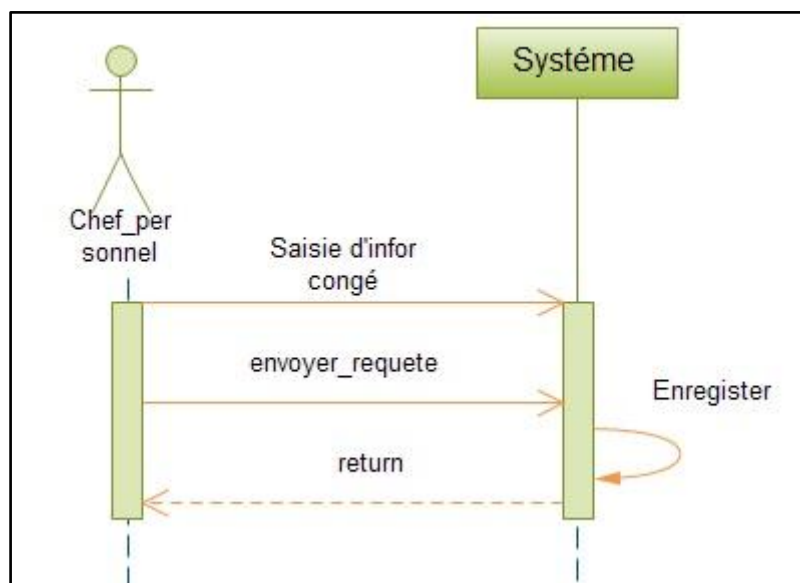


Figure 18:Diagramme de séquence «congé»

❖ **Diagramme de cas d'utilisation «sanction» :**

❖ Intention : mise à jour **sanction**

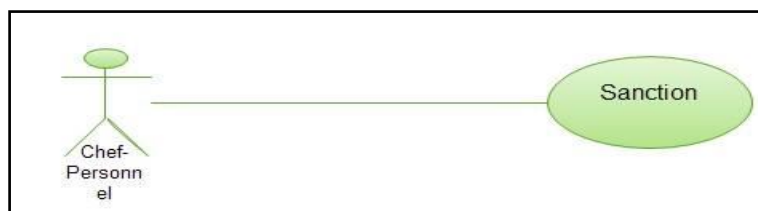


Figure 19: Diagramme de cas d'utilisation «sanction»

❖ Diagramme d'activité «sanction»

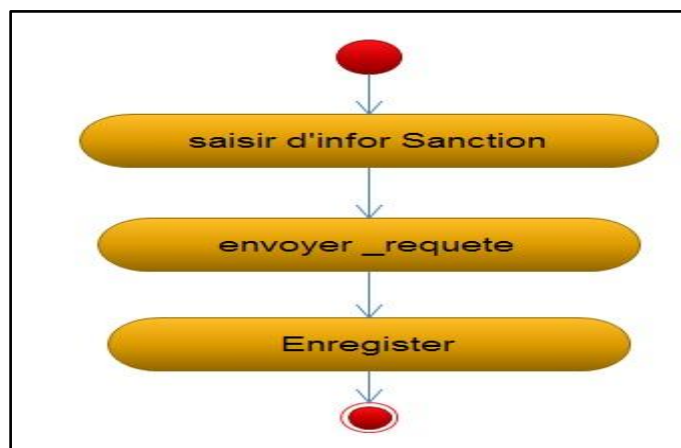


Figure 20: Diagramme d'activité «sanction»

❖ Diagramme de séquence «sanction»:

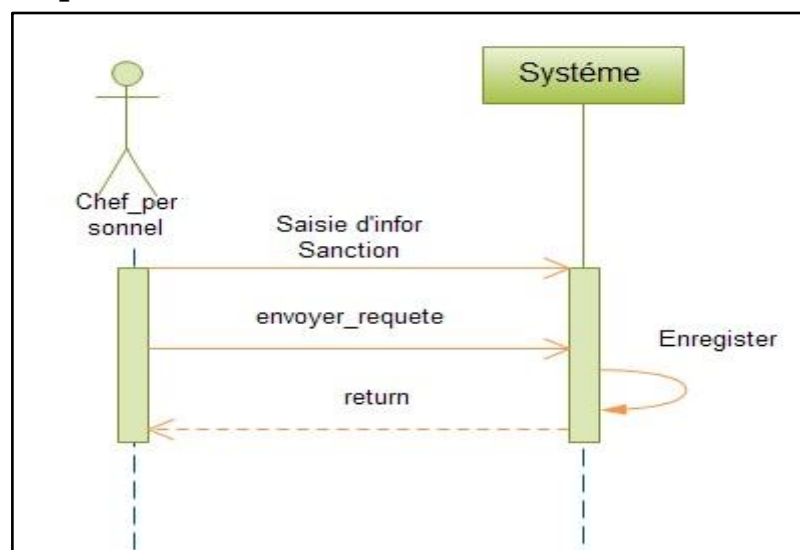


Figure 21: Diagramme de séquence «sanction»

Cas d'utilisation «Absence» :

❖ Intention : mise a jour Absence

❖ Diagramme de cas d'utilisation «Absence »:

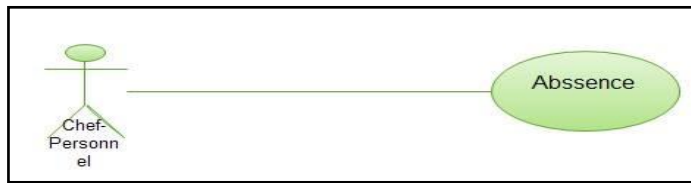


Figure 22: Diagramme de cas d'utilisation « Absence »

❖ Diagramme d'activité «Absence»

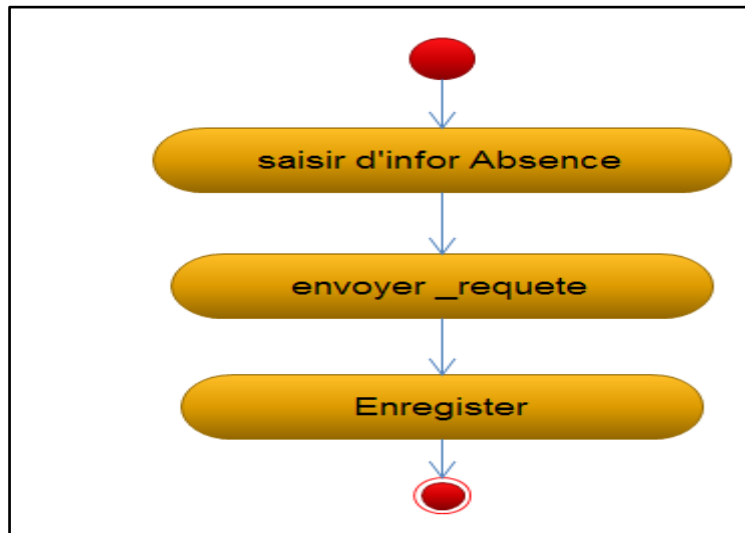


Figure 23: Diagramme d'activité « Absence »

❖ Diagramme de séquence «Absence» :

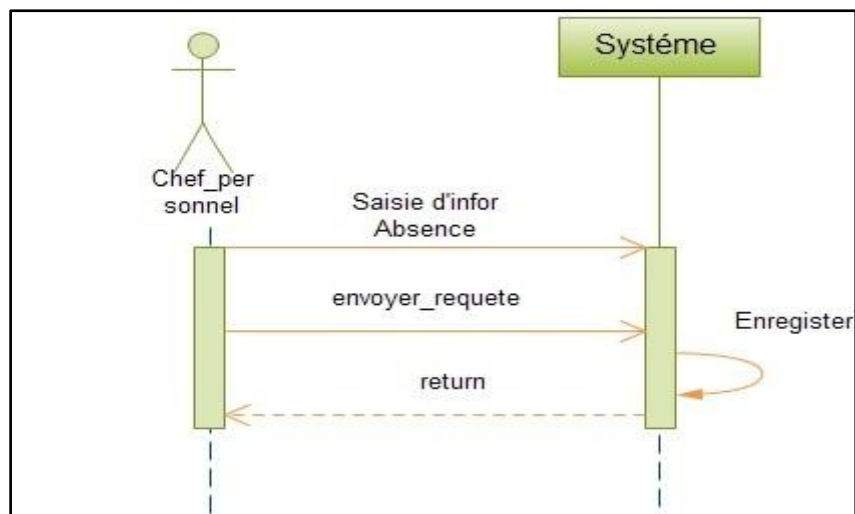


Figure 24: Diagramme de séquence «Absence»

❖ Diagramme de Cas d'utilisation «Randement» :

Intention : mise à jour **randement**

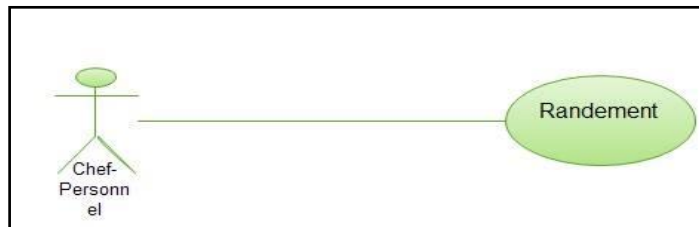


Figure 25: Diagramme de cas d'utilisation «randement»

❖ **Diagramme d'activité «Randement»**

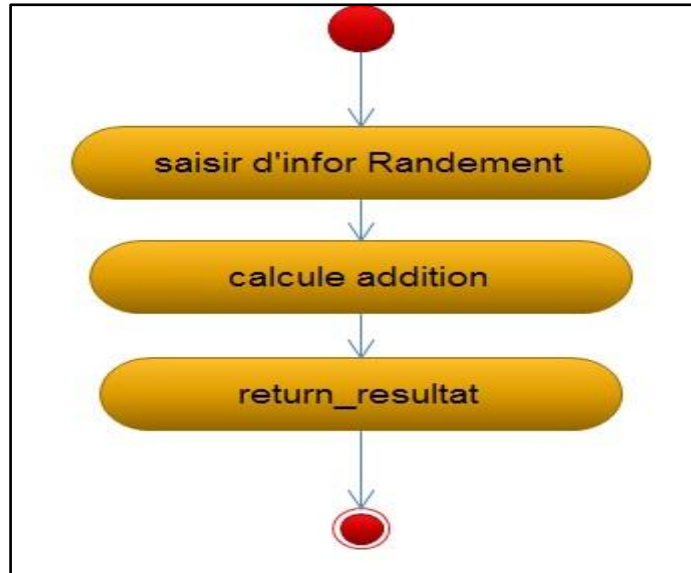


Figure 26: Diagramme d'activité «randement»

❖ **Diagramme de séquence «Randement»**

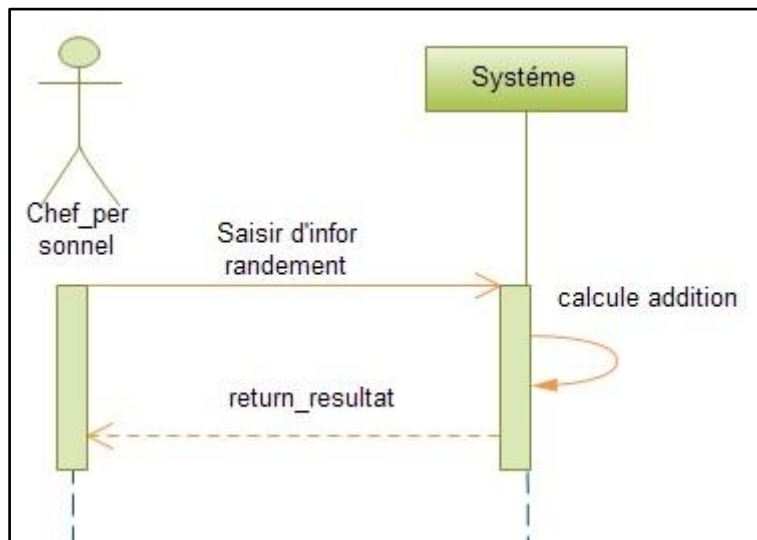


Figure 27: Diagramme de séquence «rendement»

4. Capture de besoins techniques :

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. Le modèle de spécification s'exprime suivant deux points de vue qui sont : la spécification logicielle et la structure du matériel à exploiter. [8]

4.1. Architecture de nouveau système :

L'architecture de nouveau système est basée sur un réseau interne dans le service tel que le serveur d'application serait dans les deux postes de service et le serveur de base de données dans le poste de chef de service, la figure.28 représente l'architecture de nouveau système

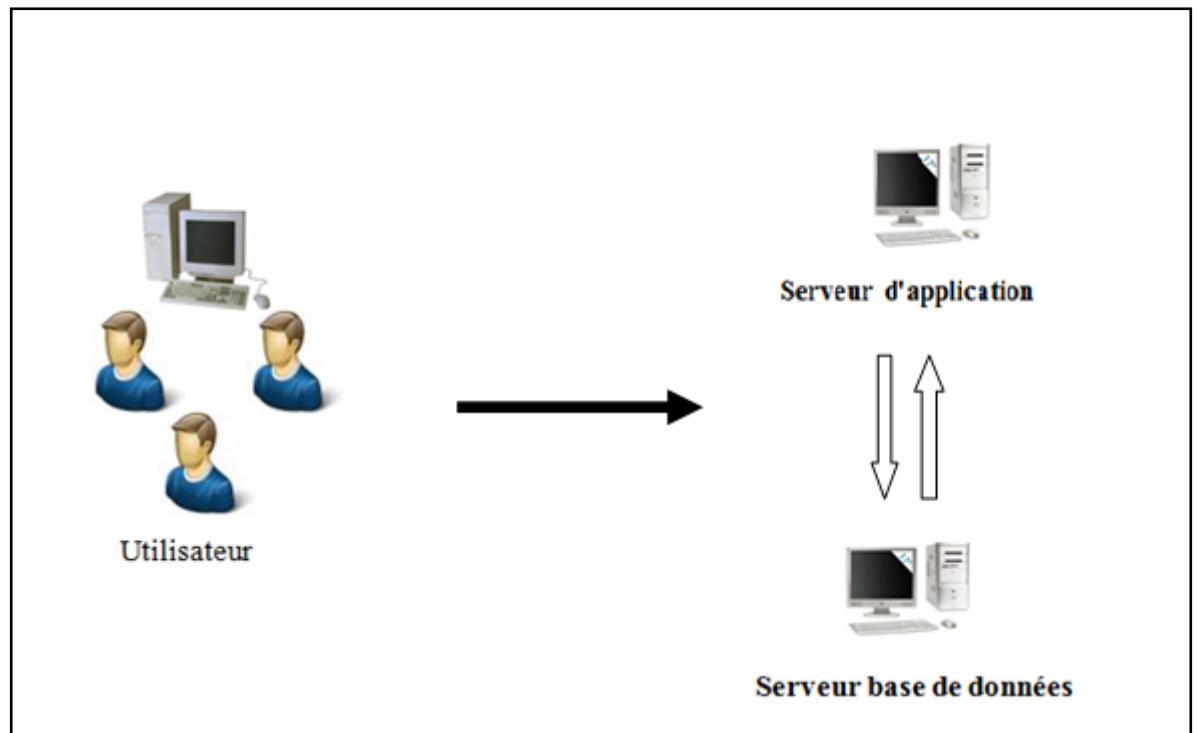


Figure.28: Architecture de nouveau système.

Les avantages de la solution : L'architecture mise en œuvre présente de nombreux avantages qui faciliteront la communication entre les différents postes dans le service. Parmi les points forts du nouveau système, nous citons :

- ✓ L'existence d'un réseau local qui facilite la communication et l'échange d'informations.

4.2. Identification des cas d'utilisation techniques :

Pour l'élaboration du modèle de spécification logicielle, On s'intéressera aux fonctionnalités propres du système en procédant à une spécification logicielle. Dans ce cas, on utilise les cas d'utilisation techniques.

Les exploitants du système sont les suivants :

- L'utilisateur : qui utilise une des applications du système. La majorité des acteurs de la branche fonctionnelle sont donc des utilisateurs dans la dimension technique.

- L'ingénieur d'exploitation, qui est chargé de déployer et de dépanner le système.
- En ce qui concerne les cas d'utilisation du système, on citera :
 - Manipulation des objets par les utilisateurs, ce qui met en œuvre des mécanismes de persistance et de gestion du cycle de vie des objets.
 - Plusieurs utilisateurs travaillent en parallèle. L'intégrité est le mécanisme qui empêche la mise à jour simultanée d'une même entité par deux exploitants différents
- Chaque utilisateur bénéficie d'une gestion des charges au niveau du serveur. Les temps de réponse ne s'en trouvent pas dégradés en fonction du nombre d'utilisateurs connectés.
- L'utilisateur doit se connecter et être reconnu du système. L'authentification est le mécanisme qui protégé le système des intrusions externes.
- Le système doit être exploitable, à ce titre, il faut qu'il soit en mesure de générer des traces et des alertes qui vont faciliter sa maintenance.
- Les exploitants sont soumis à des règles de sécurité qui sont l'authentification, le cryptage, la non-répudiation. Les contraintes d'utilisation technique donnent lieu au modèle de spécification logicielle représenté par le diagramme de cas d'utilisation suivant :

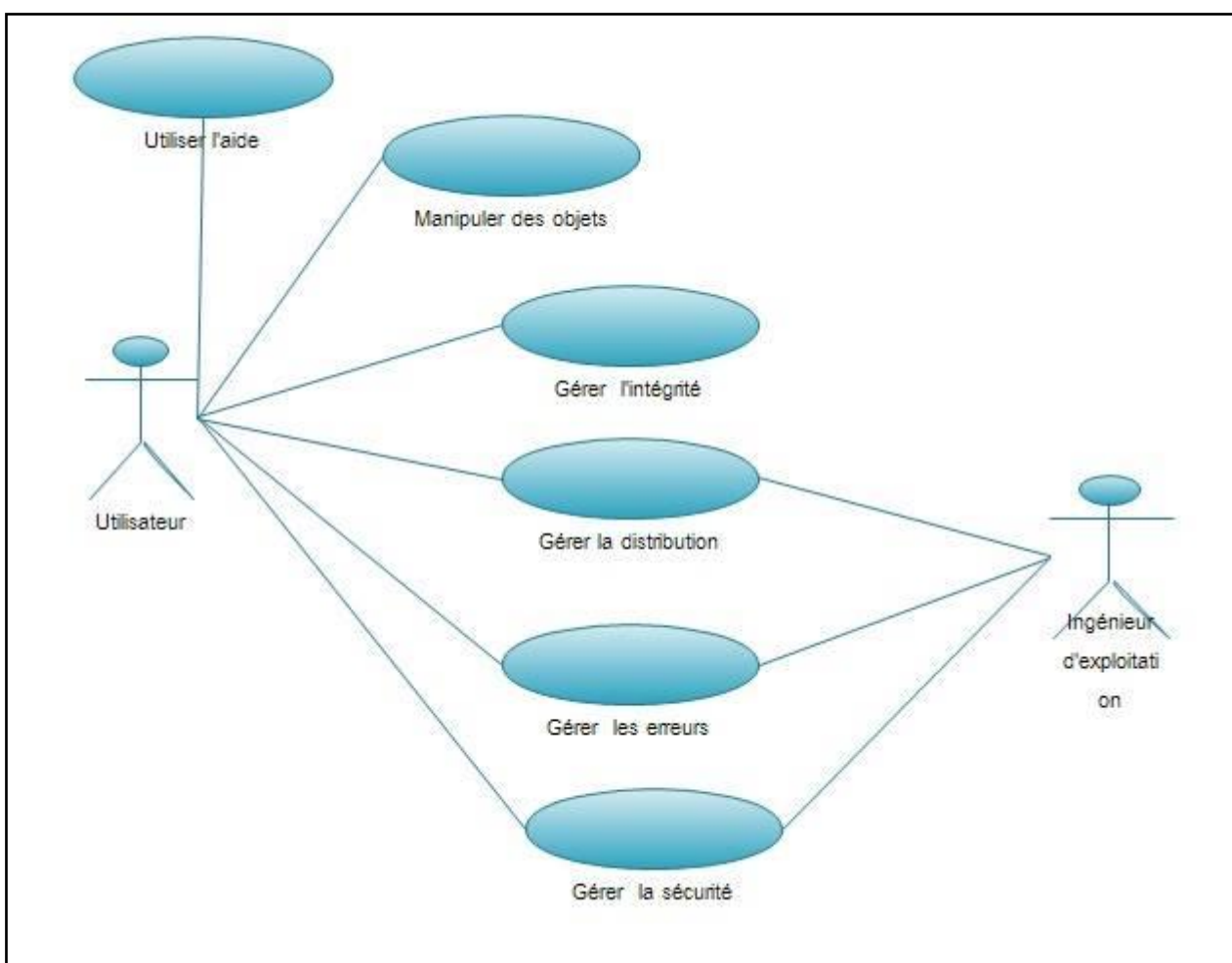


Figure 28: Modèle de spécification logicielle du système.

5. Développement du modèle statique :

Cette étape nous permettra d'illustrer les principales constructions du diagramme de classes. Ces diagrammes de classes sommairement élaborés dans l'étape d'analyse de besoins fonctionnels seront détaillés, complétés et optimisés.

6. Diagramme de classe:

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation.

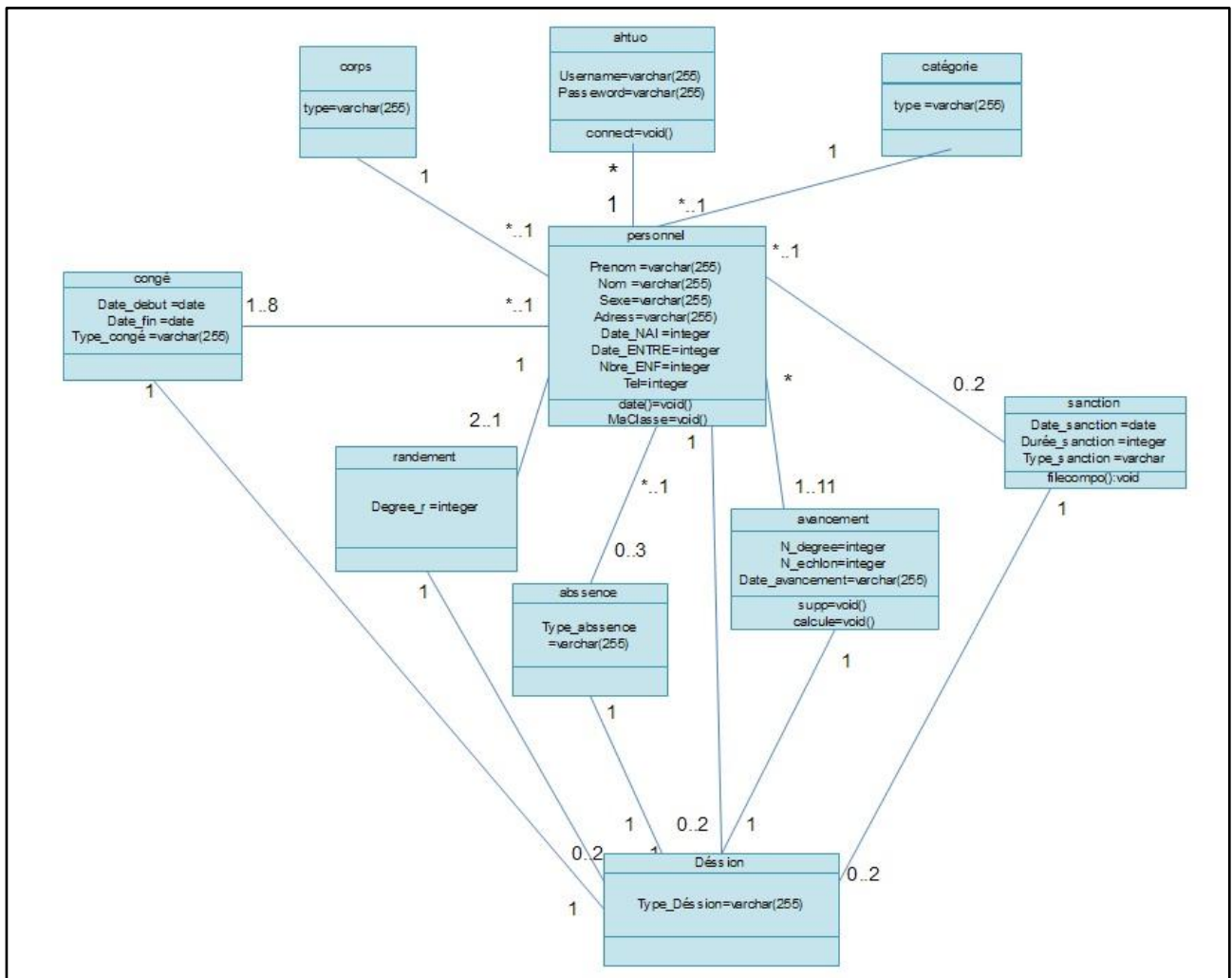


Figure 30: Diagramme de classe «Ajouter, Modifier information de personnel ».

Conception détaillée:

La conception détaillée est la phase ultime de la modélisation qui consiste à construire et à documenter précisément les classes, les tables et les méthodes qui constituent le codage de la solution.

8.1. Description du modèle des classes:

| Classe | Attributs | Désignation | Type [Taille] |
|-------------------|--|--|--|
| personnel | Matricule Prenom Nom Sexe Adress Date_NAI Date_ENTRE Nbre_ENF Nom_femme Prenom_femme Prenom_pere Nom_mere Prenom_mere Tel | Matricule de personnel Prenom de personnel Nom de personnel Sexe de personnel Adress de personnel Date de naissance de personnel Date entre de personnel Nombre de enfant Nom de femme Prenom de femme Prenom de père Nom de mère Prenom de mère Numéro de téléphone de personnel | int S[255] S[255] S[255] S[255] D D S[255] S[255] S[255] S[255] S[255] S[255] S[255] int |
| avancement | N_degree N_echlon Date_avancement Id_avancement Id_avance_perso | Nombre de degree Nombre de echlon Date avancement Identificateur avancement Identificateur avancement de personnel | Integer Integer D Int Int |
| absence | Id_absence Type_absence Dession_absence | Identificateur de absence Type de Absence Dession d'information | Integer S[255] Integer |
| sanction | Date_sanction Id_sanction Durée_sanction Type_sanction | Date de sanction Identificateur de sanction Durée de sanction Type de sanction | D Int int S[255] |
| randement | Id_randement Degree_r | Identificateur de randement Degree de rendement | Int Int |
| congé | Id_congé Date_debut Date_fin Type_congé | Identificateur de congé Date début de congé Date fini de congé Type de congé | Int D D S[255] |

Tableau 3: Liste des classes

5.2. Passage au modèle relationnel :

L'utilisation des SGBDR impose un changement de représentation entre la structure des classes et la structure des données relationnelles. Les équivalences exprimées dans le tableau suivant sont généralement utilisées : [7]

| Modèle objet | Modèle relationnel |
|----------------------------------|---|
| Classe | Table |
| Attribut de type simple | Colonne |
| Attribut de type complexe | Colonnes ou clé étrangère |
| Instance | T-uplet |
| OID | Clé primaire |
| Association | Clé étrangère ou table de liens |
| Héritage | Clé primaire identique sur plusieurs tables |

Tableau 4:Equivalences entre les concepts objets et relationnels.

5. 3.Listes des tables de la base de donnée

| Table | Identifiant | Attributs |
|-------------------------|--------------------|---|
| Personnel | Matricule | <u>Matricule</u> . Nom. Prénom. Sexe. Adresse. Date_NAI. Date_ENRTE.Nbre_ENF. Nom_femme. Nome_mère. Prenom_femme. Prenom_mère. Prenom_pere. Tel. <u>Id_user</u> . |
| ahtuo | Id_user | <u>Id_user</u> . Username. Password. |
| Absence | Id_absence | <u>Id_absence</u> . Durée_absence. Date_absence. <u>Matricule</u> . <u>Id décision</u> . |
| Décision | Id_décision | <u>Id décision</u> .type_ décision |
| Rendement | Id_rendement | <u>Id rendement</u> . Degree_R.Norme_Rendement. <u>Matricule</u> . <u>Id décision</u> |
| Conge | Id_conge | Id_conge. Date_debut.Date_fin. Type_cong. <u>Matricule</u> . <u>Id décision</u> . |
| corps | Id_corps | <u>Id corps</u> . Type. |
| Catégorie | Id_Catégorie | <u>Id Catégorie</u> . Type. |
| Avance_personnel | ID_anvec_perso | <u>ID anvec perso</u> . Dates_avancement. N_degree. Némuro_echelon. Némuro_degree. N_echlon. <u>Matricule</u> . <u>Id tAvancement</u> . |
| T_avancement | Id_tAvancement | <u>Id tAvancement</u> . Designation. <u>Id décision</u> |
| Sanction | Id_sanc | Id_sanc. Type_sanc. Date_sanction. Duree_sanction. <u>Matricule</u> . <u>Id décision</u> . |

Tableau 5:Liste des tables de la base de données.

Conclusion

Dans le but de faciliter l'implémentation de notre système de gestions des personnel, dans ce chapitre, nous avons présenté une capture des digrammes utilisés lors de la modélisation suivi une démarche de développement. Les résultats de ce chapitre seront enrichis par de détails d'implémentation dans le chapitre suivant pour réaliser notre système.

Introduction :

Pour commencer l'implantation de la base de données pour s'interagir avec l'application réalisée il faut tout d'abord choisir un environnement matériel et logiciel adéquat pour le bon déroulement de travail.

II. Environnement Logiciel :



NetBeans 7.1 est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution Licence). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web).



MySQL server 5.5 : MySQL est le SGBDR Open Source le plus populaire au monde. Il permet de mieux répondre aux problématiques d'entreprise.

Au menu des nouveautés fonctionnelles : les vues, les procédures stockées, les déclencheurs, de nouveaux moteurs de stockage, la base de données et diverses petites améliorations



Le logiciel Apache est actuellement le logiciel serveur http le plus utilisé dans l'Internet.

Doté de nombreuses fonctionnalités, performant et gratuit, il constitue un choix très intéressant pour ceux voulant mettre en place un service WWW.



dbForge Studio for MySQL 4.5:

Est un outil d'administration et un environnement de développement pour MySQL. Ses fonctionnalités permettent une exploration rapide des bases de données, la gestion des objets dans les bases, la gestion des comptes et privilèges, le développement de routines et la synchronisation entre bases. Disponible uniquement sous Windows, il nécessite l'installation préalable du .NET Framework 2.0, 3.0 ou 3.5.



Le serveur XAMPP 2.5.8:

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. Il est distribué avec différentes bibliothèques logicielles qui élargissent la palette des services de façon notable : OpenSSL, Expat (parseur XML), PNG, SQLite, zlib, ... ainsi que différents modules Perl et Tomcat. Bon nombre de personnes critiquent la quantité d'extensions ajoutées qui sont pour la plupart inutiles aux débutants. Une version lite a par conséquent été mise en place.

III. Réalisation :

Dans cette phase du rapport, on va afficher un ensemble d'interface Homme Machine qui est nécessaire pour notre application.

➤ **Authentification**

Tout d'abord, l'interface de démarrage est celle de l'authentification



Figure 31: Interface d'authentification.

L'authentification est une étape primordiale que chaque utilisateur de notre système doit y passer pour accéder à l'application. Cette phase assure, en effet, la sécurité de l'application. En demandant l'accès à l'application, l'utilisateur se voit dans l'obligation de s'authentifier à travers son compte. L'application vérifie l'existence de ce compte dans sa base des données de contenu. Si l'utilisateur est identifié dans la base, il accède à l'application selon son mode d'accès fixé par l'administrateur. Une fois les données sont valides, la page d'accueil de l'utilisateur est chargée

➤ **Interface Accueil :**

La forme suivant représente l'interface Accueil de notre application qui apparait après l'authentification. Cette forme comporte les menus principaux de l'application. La Figure. II.2 représente l'interface Accueil du programme.



Figure 32:Interface Accueil.

➤ **Interface information :**

Pour ajouter un nouvel personnel, l'utilisateur clique sur "+" dans l'interface information , puis se remplis les informations obligatoires de personnel.

- ✚ Si les informations correctes, le système sauvegarde personnel.
- ✚ Sinon il informe l'utilisateur sur les erreurs.

Figure.33 présente l'interface de information



Figure 29:Interface d'information.

4. Conclusion

Dans ce chapitre, nous avons présenté la réalisation de notre système et les différentes fonctionnalités ont été développées afin de faciliter l'accès et l'exploitation des données relatives à la gestion de personnel.

CONCLUSION GENERALE

Dans ce rapport de projet de fin d'étude, nous avons conçu et développé une application de gestion de personnel qui permet au responsable de ressource humaine du La Cour des organisations de jeunesse de la Wilaya d'El- oued de bien contrôler et gérer les employés inscrit dans cet organisme . Cette application, permet au gestionnaire d'ajouter des nouveaux personnels, mise à jour des données avancement et congé , randement , absance et sanction un employé .

Sur le plan technique, cette expérience nous a permis de découvrir et d'acquérir des connaissances nouvelles, de renforcer nos connaissances programmation orientée objet et le développement d'une base de données MySQL et d'autre part , particulièrement avec son logiciel NetBeans.

Ce projet nous a permis aussi de maitriser le langage java appliqué au développement d'une application selon la norme J2EE. Un autre aspect assez important de notre expérience était l'opportunité de découvrir le milieu professionnel avec tout ce qu'il exige de responsabilité, de discipline et travail en équipe.

Finalement, l'application de gestion de personnel est une application modeste qu'on peut mieux la développer car c'est une application nécessaire pour chaque administration surtout dans le monde d'aujourd'hui qui est régné par les outils informatique.

[1] :PROJET DE FIN D'ETUDE Licence Appliquée En SCIENCES ET TECHNIQUES DE L'INFORMATIONS ET DE COMMUNICATIONS

[2]: Bali Ahmed /Cours UML/ Chapitre II/ 2013.

[3]: Laurent Piechocki / COURS UML / Document : version 13 / janvier 2003.

[4]: Eric Papet / 2UP_ARCHI_DEV1 / Document: version 1.00/12-12-2000.

[5]: <http://www.cnrc.org.dz/ar/index.php> / 19/03/2013.

[6]: Kazi Aouel Bassim et Rostane Zakaria/Suivie des enseignements du LMD par application de la méthode 2TUP/ Projet de Fin d'Etudes pour l'obtention du Diplôme d'Ingénieur d'Etat en Informatique / 08 -11- 2007.

[7]: Ait-Bachir Ali et Belkada Samir / Conception et réalisation d'un système d'information pour le suivi du parc informatique de B&R-C / Mémoire de fin d'étude pour l'obtention du diplôme d'ingénieur d'état en informatique / 2003-2004.

[8]: Ait-Bachir Ali et Belkada Samir / Conception et réalisation d'un système d'information pour le suivi du parc informatique de B&R-C / Mémoire de fin d'étude pour l'obtention du diplôme d'ingénieur d'état en informatique / 2003-2004.

