

ROYAUME DU MAROC

FORCES ARMEES ROYALES

ACADEMIE ROYALE MILITAIRE

**PROJET DE FIN D’ETUDES**

POUR L’OBTENTION DE

**La Licence fondamentale**

Filière : Sciences et techniques

Option : Génie informatique

**GESTION ET AFFECTATION DU PERSONNEL**

Rédigé sous la direction de Monsieur le professeur

**CHAKIR LOQMAN**

Par les élèves officiers :

* Saad EL MEJDOUBI
* Zakaria ABDELFATTAH
* El Mehdi HAMIRIFOU
* Zakaria HRIDOUCH

ROYAUME DU MAROC

FORCES ARMEES ROYALES

ACADEMIE ROYALE MILITAIRE

**PROJET DE FIN D’ETUDES**

POUR L’OBTENTION DE

**La Licence fondamentale**

Filière : Sciences et techniques

Option : Génie informatique

**GESTION ET AFFECTATION DU PERSONNEL**

*Rédigé sous la direction de Monsieur le professeur*

**CHAKIR LOQMAN**

Par les élèves officiers :

* Saad EL MEJDOUBI
* Zakaria ABDELFATTAH
* El Mehdi HAMIRIFOU
* Zakaria HRIDOUCH

**Dédicaces**

A mes chers parents

A mes respectueux professeurs

A tous les Officiers de

L’Académie Royale Militaire

De prime à bord, nous tenons à exprimer notre gratitude et présenter notre chaleureux remerciement à:

* Monsieur Chakir LOQMAN notre professeur encadrant, qui n'a pas cessé de nous prodiguer ses précieux conseils, pour son acceuil, disponibilité et son soutien surtout en ce qui concerne les interfaces JavaFX,
* A tous nos professeurs qui ont assuré notre formation pour leur patience et plus particulièrement les membres de jury qui ont accepté de juger notre travail,
* Notre Ecole Académie Royale Militaire qui nous a donnée l'occasion d'acquérir une formation professionnelle,
* toute notre reconnaissance est adressée à nos parents, pour leurs sacrifices et leur soutien, même à distance, durant ces années d’études. Ce n’est que grâce à eux que tout cela a été rendu possible,

Toutes personnes ayant contribué de près ou de loin à l'élaboration de ce modeste travail.

**R E S U M E**

Un des buts de l’ingénieur informaticien est l’élaboration de tels systèmes et de pouvoir, par la suite, en assurer le maintien et le bon fonctionnement, qui, en plus de la securité, constituent des critères primordiaux.

Actuellement, l’informatique représente un outil inévitable de gestion et de pilotage des systèmes d’information, surtout au niveau des organismes possédant des systèmes qui reposent sur la gestion de bases de données, dans notre projet on s’occupera de l’informatisation du domaine militaire, qui sera notre surface d’application.

A ce propos, la gestion de personnel et leur affectation sont devenues un véritable enjeu stratégique. Afin de créer ou de maintenir ses avantages concurrentiels, l’unité est continuellement menée à améliorer ses performances, ce qui se traduit souvent par l’optimisation de ses processus et donc par une maitrise de ses ressources, que celles-ci soient matérielles ou humaines. Or, bien gérer son capital humain passe par un suivi adéquat de ses activités (ses absences, formations ..).

La complexité de cette tâche varie en fonction de l’organisation, de la taille de l’unité, mais aussi selon la législation en vigueur. L'affectation se jette dans le domaine de la recherche opérationnelle, s’agissant de l'affectation proprement dis mais sous un aspect militaire, elle a été sujet d'intérêt de plusieurs armées du monde (US ARMY, L’URSS, armée française) et ce pour des raisons majeurs d'optimisation et pour ressortir une disposition du personnel optimale pour une seule finalité; avoir des corps soudés avec dans chacun d'eux le personnel adéquat.

Cette étude nous a permis de mettre en valeur notre sens de l’analyse et de réflexion. De plus, ce projet qui est un véritable travail d’équipe nous a permis d’acquérir l’expérience d’un projet mené de bout en bout, en matière de partage des tâches à réaliser, le travail de groupe et la gestion de l’échéancier. Toutefois de nombreuses difficultés ont fortement mis à contribution notre esprit d’équipe et d’initiative. Nous avons également beaucoup appris au contact de toutes les personnes qui nous ont aidés et soutenus dans la réussite de notre projet.

\* \* \*

**A B S T R A C T**

As part of our final project study, we focus on developing an ergonomic application that can be integrated into the portal of a unit for the management and assignement of personnel, this application should be avaialable for use and accessible to all type of people, even novice in data processing.

This application supports two major functions: Personnel Management beloging to a unit, and their allocation or assignement in certain services or spots according to criteria that we'll see eventually. This study has allowed us to enhance our sense of analysis and reflection. In addition, this project has allowed us to gain the experience of a project from the begining, in the sharing of the tasks, the working group and the management timetable. However many difficulties have severely enhanced our team spirit and initiative. We also learned a lot from all the people who helped and supported us in the success of our project.

Personnel management is an essential activity for the success of a business, this task is motivated by the fact that the staff is the most important resource in companies. It is not universal. Every company must consider its needs and its industry to establish a good personnel management. In our case, we will not discuss companies but military units that require the organization of human resources to ensure the performance of all tasks. To ensure that the work is done correctly and on time, you have to train staff, ensure productivity, control the work, correct, thank, motivate and sometimes punishing, etc. To simplify these tasks, a personnel management software must be available to supervisors.

The military assignment flows in the field of operations research, with regard to the assignment properly say but under a military aspect, it has been subject of interest of many armies in the world (US ARMY, USSR , French military) and for major reasons of optimization and highlight a provision of optimal staff to a single purpose; have bodies welded in each of them the right staff.

**TABLE DES MATIERES**

INTRODUCTION

1. L’Informatique en domaine militaire

2. Motivation et Objectifs

3. Organisation

CHAPITRE I : PRESENTATION DU PROJET

I. Méthodologie générale

I.1. Méthodologie de conception

I.1.1 Introduction

I.1.2 Cycles de vie du projet

I.2. Langage de conception

II- Comprendre la problématique

II.1. Etat actuel du système

II.2. Organisation

III. La solution proposée

III.1. Apport sur le plan technique

III.2. Apport sur le plan fonctionnel

IV.Conclusion

CHAPITRE II: SPECIFICATIONS ET ANALYSE DES BESOINS

I. Capture des besoins

I.1. Identification des acteurs

I.1.1. Acteurs

I.1.2. Diagramme de contexte statique

I.2. Les besoins fonctionnels

I.3. Besoins non fonctionnels

I.4. Scenario

II. Diagramme de cas d’utilisation global

III. Problème d’Affectation :

III.1 Formulation du problème

III.2 Modélisation du problème d’affectation standard

III.3 Modélisation du problème d’affectation standard:

III.3.1 Méthode hongroise

IV. Modélisation du point de vue utilisateur

IV.1. Processus d’authentification

IV.3. Processus de modification :

IV.4.Processus de congé

IV.5.Processus de formation

IV.6.Processus d’affectation

CHAPITRE III: ANALYSE ET CONCEPTION

I. Considérations préalables

I.1 Outil de modélisation

I.2. Architecture du futur système

II. Conception de la base de données

II.1. Présentation de la méthode MERISE

II.2. L’élaboration du dictionnaire des données

II.3. Dépendances fonctionnelles

II.4. Le Modèle conceptuel de données

CHAPITRE IV : IMPLEMENTATION DU PROTOTYPE

I. ENVIRONNEMENT DE DEVELOPPEMENT

I.1. Environnement matériel

I.1.1 Environnement de développement et du test

I.2. Environnement logiciel

I.3. Langages de programmation

II. Réalisation

IV. Conclusion

**INTRODUCTION**

**1. L’Informatique en domaine militaire**

Au fil de l'histoire, la collaboration entre les scientifiques et les militaires a été très fréquente à fin de prendre les meilleures décisions dans les batailles pour atteindre la victoire. C'est pour cela que beaucoup d'experts dans le domaine considèrent le début de la Recherche Opérationnelle dans le IIIe siècle av. J.-C, pendant les Guerres puniques, avec l'analyse et la solution qu'Archimède a proposé pour la défense de la ville de Syracuse, assiégée par les romains. pendant la Première Guerre mondiale en Angleterre, l'étude mathématique de Frederick William Lanchester sur la puissance balistique des forces opposantes ; développant , à partir d'un système d'équations différentielles, la Loi quadratique de Lanchester, avec laquelle c'était possible de déterminer le dénouement d'une bataille militaire en fonction de la force numérique relative et la capacité relative de feu des combattants.Thomas Alva Edison a aussi employé la Recherche Opérationnelle pour contribuer dans la lutte anti-sous-marine, en développant des techniques afin que les vaisseaux puissent éviter et détruire les sous-marins ennemis avec une protection anti-torpille.Du point de vue mathématique, dans les XVIIe et XVIIIe siècles, Newton, Leibnitz, Bernoulli et Lagrange, ont travaillé pour obtenir le maximum et le minimum conditionnés de certaines fonctions. Le mathématicien français Jean Baptiste-Joseph Fourier a ébauché les méthodes de l'actuelle optimisation linéaire. Dans les dernières années du XVIIIe siècle, Gaspar Monge a établi les précédents de la méthode graphique grâce au développement de la Géométrie Descriptive.

Durant la Guerre froide, l'ancienne Union soviétique (URSS), refusant le plan Marshall, a voulu contrôler les communications terrestres, aussi comme les routes fluviales, de Berlin. Afin d'éviter la reddition de la ville, et sa soumission pour faire partie de la zone communiste allemande, l'Angleterre et les États-Unis ont décidé d'approvisionner la ville, bien grâce à convois escorté (ce que peut aboutir à nouveaux affrontements), bien avec un pont aérien, en rompant ou en évitant, en tout cas, le blocus de Berlin. Ils ont opté pour la deuxième option, en ouvrant le Luftbrücke (pont aérienne) le 25 Juin de 1948. Le groupe SCOOP avait aussi participé dans ce problème, en Décembre de cette année on avait réussi d'approvisionner 4500 tonnes par jour, et après l'investigation de la Recherche Opérationnelle on a optimisé l'approvisionnement jusqu'aux 8000 ou 9000 tonnes par jour en Mars 1949. C'était le même chiffre que si on aurait transporté par terre, ce pour cela que les soviétiques ont levé le blocus le 12 mai 1949.

Après la Seconde Guerre mondiale, on a déterminé d'organiser les ressources des États-Unis (énergie, armement, et tous types de distribution) grâce aux exemples d'optimisation, résolus par l'Optimisation linéaire.

Au même temps que la doctrine de la Recherche Opérationnelle, on a développé aussi les techniques de computation, lesquelles ont fortement réduit le temps de résolution sur les problèmes.

Le premier résultat a été obtenu en 1952, avec un ordinateur SEAC du National Bureau of Standars pour obtenir la solution d'un problème. Le succès dans le temps de résolution a été tellement encourageant qu'elle a été immédiatement utilisée pour tous les types de problèmes militaires, telle que la gestion des fonds monétaires pour la logistique et l'armement, déterminer la hauteur optimal dans laquelle les avions devraient voler pour localiser les sous-marins ennemis, et même, la profondeur dans laquelle on devait envoyer les charges pour atteindre les sous-marins ennemis à l'effet de produire le maximum de pertes. Tout cela se traduise par une augmentation cinq fois plus dans la efficience de la force aérienne.

Pendant les années 50 et 60, l'intérêt et le développement de la Recherche Opérationnelle a agrandi dans la suite de l'application sur le commerce et l'industrie. Un exemple c'est le problème de calcul du projet optimal de transport de sable de la construction aux travaux d'édification à Moscou, où il y avait 10 points d'origine et 230 de destination. Pour la résolution ils ont fait usage d'un ordinateur Strena dans le mois de juin de 1958, et après 10 jours de travail, il a fourni une réduction du 11% de coûts par rapport aux coûts originaux prévus.( problème d'affectation )

Auparavant on avait déjà posé ces problèmes dans une discipline connu comme Analyse d'Entreprise, qui n'avaient pas des méthodes si effectifs que celles de la Seconde Guerre mondiale (par exemple la méthode du Simplexe). Les applications pas de guerre de la Recherche Opérationnelle s'étendent par tous les cadres, avec des problèmes de l'alimentation, élevage, distribution de culture dans l'agriculture, transport de marchandise, localisation, distribution du personnelle, problèmes de réseaux, de files d'attente, de graphes, etc. Tout ce développement époustouflant, et toute victoire depuis les guerres puniques jusqu’à la 2e guerre mondiale ont été étroitement liées à l'emploi judicieux de la recherche opérationnelle.

Comme on dit '' la guerre se prépare en temps de paix " donc la Recherche opérationnelle continue à s'épanouir et surtout dans deux de ces aspects qui ont été les plus frappants dans l'histoire de la guerre : le problème de transport et celui de l'affectation ; qui ne cesse de prendre une grande envergure dans toute les armées du monde .

**2. Motivation et Objectifs**

Dans le cadre de notre projet de fin d’étude, nous nous intéressons à développer une application ergonomique capable d’être intégrée au portail d’une unité militaire pour assurer la gestion et l’affectation du personnel, soit facile à utiliser et accessible à tout type de personnes, même novice en Informatique.  
Cette application prend en charge deux grandes fonctions: Gestion de personnel appartenant à une unité, et leur affectation dans certains services ou postes selon des critères que l'on verra par la suite.

La gestion de personnel présente une activité essentielle à la réussite d’une entreprise, cette mission est motivée par le fait que le personnel constitue la ressource la plus importante des entreprises. Elle n’est pas universelle. Chaque entreprise doit tenir compte des ses besoins et de son secteur d’activité pour établir une bonne gestion du personnel. Dans notre cas, on va plus parler des entreprises mais des unités militaires qui demandent l'organisation des moyens humains afin de garantir l'exécution de toutes les tâches.

Pour s'assurer que le travail est effectué correctement et dans les délais impartis, il faut former le personnel, s'assurer de leur productivité , contrôler le travail, savoir corriger, remercier, motiver et parfois sanctionner, etc. Pour simplifier ces tâches professionnelles, un logiciel de gestion et affectation du personnel doit être mis à la disposition des supérieurs hiérarchiques.

**3. Organisation**

Le premier chapitre de ce rapport définit la présentation du projet. Puis le deuxième Il y sera traité brièvement la problématique où on exposera l’état actuel de la gestion et affectation de personnel et les moyens utilisés.

Par la suite, un cahier des charges simplifié, regroupant les modèles d’expression des besoins et d’analyse du futur SI, sera présenté dans le deuxième chapitre. Dans le 3 e chapitre, on verra la modélisation qui sera menée en gardant à l’esprit le principe de représentation du système en trois niveaux: fonctionnel, logique et technique.

La réalisation du prototype fera l’objet du chapitre 4, dans lequel les fonctionnalités du système seront présentées tout d’abord du point de vue du programmeur, puis du point de vue utilisateur final, sous forme d’un miniguide d’utilisation. Quelques cas de tests seront donnés pour vérifier le bon fonctionnement du prototype.

Finalement, une rétrospective critique et des réflexions d’améliorations futures seront données en guise de conclusion de ce travail.

**CHAPITRE I :**

**PRESENTATION DU PROJET**

### **I. Méthodologie générale I.1. Méthodologie de conception**

**I.1.1 Introduction**

### La méthodologie est une démarche organisée rationnellement pour aboutir à un résultat. Parmi les différentes méthodologies existantes, nous pouvons citer le modèle en cascade utilisée souvent dans les simples projets dont les besoins sont clairs et bien définis dès le début, le modèle en **Y** utiliser pour le développement des applications mobiles, ainsi que le processus unifié et les méthodologies agiles (Scrum & extrême programming) caractérisées par leurs souplesses et utilisées dans des grands projets.

Pour bien conduire notre projet et nous assurer du bon déroulement des différentes phases, nous avons opté pour le Rational Unified Process (Unified Process) comme une méthodologie de conception et de développement, cette méthode est l’une des plus célèbres implémentations (approche applicative) de la méthode PU (Unifed Process), livrée clés en main, permettant de donner un cadre au développement logiciel, répondant aux exigences fondamentales préconisées par les créateurs d’UML :

* une méthode de développement doit être guidée par les besoins des utilisateurs
* elle doit être centrée sur l’architecture logicielle
* elle doit être itérative et incrémentale

**I.1.2 Cycles de vie du projet**

Le Processus Unifié de Rational comprend 4 phases :

* + **Démarrage** (Inception)- Définit le champ d’action du projet
  + **Élaboration** – Le plan du projet, il spécifie les exigences, les bases de l’architecture
  + **Construction** – Réalise le produit
  + **Transition** - Transfère le produit vers les utilisateurs finaux

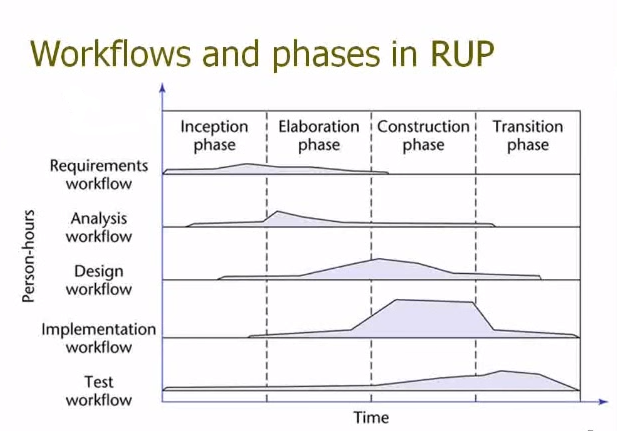
Une phase se compose d'une ou plusieurs itérations et se termine par une étape où l'avancement du projet est évalué et les principales décisions sont prises. Les itérations dans chaque phase se concentrent sur la production des techniques livrables qui répondent aux objectifs de la phase.

* Durant l’étude d’opportunité (démarrage), nous définissons l’objectif du projet.
  + En identifiant tous les acteurs et les cas d’utilisation, et en dessinant les cas d’utilisation essentiels.
  + Un plan de gestion de projet est fait pour déterminer les ressources nécessaires pour le projet.
* Durant l’élaboration, on se concentre sur deux choses :
  + *avoir une bonne connaissance des besoins (90%)*
  + *établir une base de l’architecture.*

Ainsi, on peut éliminer beaucoup de risques, avoir une bonne idée de ce qui doit être fait, et une bonne estimation des ressources et des coûts.

* Durant la Construction, on développe le produit en plusieurs itérations pour une version bêta.
* Durant la Transition, on prépare le produit pour l’utilisateur final et la formation, l’installation, le support.

Pour un projet très complexe l’élaboration peut inclure jusqu’à 3-5 itérations.

  
*Figure : Le cycle de vie RUP*

Le travail de chaque itération comprend une partie ou tout le flux de travail du développment.

1. **Capture des Besoins** : *Vision du système et besoins détaillés des utilisateurs.*
2. **Analyse** : *Manière dont sera réalisé le projet au cours de la phase 3 et 4.*
3. **Design** : *C’est la Conception du projet*
4. **Implémentation**: *Production et acquisition des composants du système et des exécutables. (Réalisation)*
5. **Tests et déploiement**: *Vérification du système dans son ensemble*, la *livraison du système et formation des utilisateurs*.

Comme les phases d'un projet progressent, la quantité relative de travail dans chaque flux de travail varie. Par exemple, Dans la phase de démarrage, la plupart des travaux seront concentrés sur la capture des besoins, alors que dans la phase de Construction, la plupart de ces travaux seront concentrés sur l’implémentation.

Chaque itération se concentre sur l'implémentation des cas d'utilisation spécifiques ou de traiter des questions techniques spécifiques.

**1.2. Langage de conception**

Après le choix de la méthodologie, nous avons besoins d’un langage de modélisation unifiée pour la modélisation de notre projet. Pour concevoir notre système, nous avons choisi UML comme un langage de modélisation.

Notre choix s'est basé sur les points forts de ce langage notamment sa standardisation et les divers diagrammes qu’il propose. Aussi UML présente le meilleur outil pour schématiser des systèmes complexes sous un format graphique et textuel simplifié et normalisé.

En effet UML n'est ni un processus ni une démarche, d'où il fallait choisir une méthodologie de conception et de développement que nous devons l'adopter.

Toutefois, dans le 3eme chapitre, on utilisera la Méthode MERISE pour la conception de la base de données (MCD, MLD et MPD) parce que cette méthode permet une meilleure modélisation des traitements et la conception d’un modèle de base donnée. Aussi que par expérience, Merise permet également de se faire aisément comprendre.

**II- Comprendre la problématique**

**II.1. Etat actuel du système**

Le travail dans ce chapitre est basé sur des renseignements obtenus auprès de l'ARM. Les informations présentées viennent principalement des supérieurs hiérarchiques de l’ARM, qui nous ont présentés les problémes envisagés.

Le processus de gestion et affectation actuel présente une activité lourde et onéreuse pour les supérieurs, qui sont toujours très sensibles à l’organisation de leur temps et aux problèmes d’optimisations.

La gestion et affectation actuelle des personnels rencontre plusieurs difficultés à savoir :

* Processus de mise à jour est lent.
* Recherche d'un dossier d’un personnel archivé prend beaucoup de temps.
* Le format physique des données sur papier rend sa gestion beaucoup plus difficile.
* Manque de vivacité dans la gestion des formations.
* Manque de précision dans la gestion des congés.
* Manque de clarté dans la gestion des sanctions.
* Lenteur de processus d'affectation.
* Le choix du personnel adéquat dans un poste prend une très longue durée.
* La vérification des critères, qui présentent les coûts d'affectation, prend beaucoup de temps.
* Etc.

**II.2. Organisation**

Au sein d’une base militaire, on trouve que l’unité qui commande et y dirige est organiquement constituée, dotée de sa propre autonomie administrative et gérée par les [militaires](https://fr.wikipedia.org/wiki/Militaire), également des personnels civils qui sont recrutés par la Défense Nationale selon des contrats dans des domaines spécifiques ou dans des métiers pour lesquels on ne trouve pas de compétences chez les fonctionnaires (informatique, électrotechnique, finances..). Dans notre cas l'Académie Royale Militaire recours aux enseignants civils pour assurer la formation universitaire des Elèves-Officiers, même chose pour les installations et les bâtiments qui sont confectionnés par des architectes civils.

Dans chaque corps militaire, les positions sont en général organisées en niveaux. C’est la notion du niveau hiérarchique. Le chef de corps [désigne](http://www.linternaute.com/dictionnaire/fr/definition/designer/) [le](http://www.linternaute.com/dictionnaire/fr/definition/le/) [plus haut](http://www.linternaute.com/dictionnaire/fr/definition/plus-haut/) [niveau](http://www.linternaute.com/dictionnaire/fr/definition/niveau/) [d](http://www.linternaute.com/dictionnaire/fr/definition/d/)'[autorité](http://www.linternaute.com/dictionnaire/fr/definition/autorite/), il est chargé de la direction, l’organisation et de la répartition des tâches au sein du corps de police local. Il est responsable de l’exécution et de la gestion de ce Corps, des missions à caractère local, des directives concernant les missions à caractère fédéral et des réquisitions. Il est suivi par les chefs des services ou chefs des departements. Un chef de service est, d'une manière générale, la personne qui dirige un service plus ou moins important (service materiel, effectif, renseignement..), et qui dispose à cet égard de prérogatives hiérarchiques à l'égard du personnel du service (distribution du travail, fixation des horaires, évaluation), il a donc sur le reste de personnel qui y travaillent. Enfin, on trouve le flux de personnel qui présente les militaires concernés par la gestion.

**Figure1 : Organisation générale d’une unité militaire**

## III. La solution proposée

Suite aux inconvénients cités dans le paragraphe précédent, nous proposons la mise en place d'une application bureau qui automatise la gestion du personnel à l’instar de la gestion des congés et de formations de la base militaire ainsi que l’affectation du personnel. Cette application aura plusieurs apports pour l’unité aussi bien sur le plan technique que sur le plan fonctionnel.

**3.1. Apport sur le plan technique**

Sur le plan technique, ce projet permet de centraliser les données dans un seul endroit (base de données unique) qui sera partagée par tous les modules de l'application. Donc la donnée sera saisie une seule fois et accessible pour tous les services de lunité. Cette application permet aussi d’assurer la sécurité des données et leur fiabilité.

**3.2. Apport sur le plan fonctionnel**

Sur le plan fonctionnel, ce projet apporte deux avantages principaux. Le premier c’est le gain de temps relatif au traitement des données, et le second c’est la simplification de la tâche du responsable.

**Conclusion**

Dans ce chapitre nous avons présenté le cadre général de notre projet en déterminant la problématique et en proposant une solution envisagée pour faire face à la situation courante. Nous avons dévoilé le langage et la méthodologie de conception qui seront utilisés dans les prochains chapitres de ce rapport et nous avons argumenté notre choix.

# CHAPITRE II: SPECIFICATIONS ET ANALYSE DES BESOINS

**Introduction**

Les travaux réalisés dans cette période de temps conduit à construire une bonne vision du produit, identifier les rôles des utilisateurs et dégager les fonctionnalités principales afin de produire le backlog initial ainsi qu'une première planification des sprints.

Cette phase fera donc l’objet de ce chapitre où nous commençons par la capture des différents besoins, identifier les rôles des utilisateurs et préparer un diagramme de cas d’utilisation qui va définir les besoins du systéme.

**I. Capture des besoins**

**I.1. Identification des acteurs**

**a. Acteurs**

« *Un acteur représente l’abstraction d’un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.*»

Tous simplement un acteur est une entité physique (personne) ou abstraite (logiciel) capable d’utilisée le système afin de répondre à un besoin bien définit. Les acteurs de notre application sont :

**Le chef de corps** : c’est la personne possédant le privilège de plus haut niveau (level =2, on va voir cette notion par la suite). Cet acteur est capable de manipuler toutes les fonctionnalités proposées par l’application notamment l’ajout des personnel, l’approbation des demandes de congé, etc.

**Le chef de service :** Il a des autorisations moins que celles de chef de corps

**Le personnel :** Ils n’ont pas beacoup d’autorisations pour exploiter l’application

Les personnels manipulent quelques fonctionnalités basiques notamment la consultation de profil et la mise à jour de leurs informations

**b. Diagramme de contexte statique**

Ce diagramme d’UML permet simplement de montrer la relation des différents acteurs avec le système. Il spécifie le nombre d’instances de chaque acteur relié au système à un moment donné.

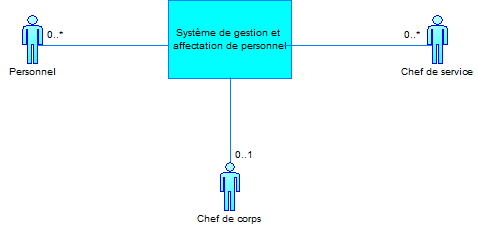


Figure 2 : Diagramme de contexte statique

Pour expliquer le diagramme ci-dessus, nous pouvons dire qu’à un instant *t* nous pouvons avoir 0 ou un chef de corps qui manipule l’application, et 0 ou plusieurs chef de service et personnel qui sont en train d’utiliser l’application.

## I.2. Les besoins fonctionnels

Les besoins fonctionnels ou les cas d’utilisations en terme d’UML peuvent être définis comme suit : « *Un cas d’utilisation (use case) représente un ensemble de séquences d’actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier.* »

Un cas d’utilisation est une suite d’actions effectuées par le système afin de répondre à une demande d’un utilisateur (acteur). Il nous a été confiés dans le cadre de ce projet, de faciliter la mission de la gestion du flux de personnel au sein de toute structure militaire. Devant être facile d’utilisation et accessible à tout type de personnes, même novice en informatique, notre projet doit aboutir au développement d’une application bureau ergonomique capable d’être intégrée au portail des unités militaires, et mise à la disposition de tous les personnels.

Dans ce qui suit, nous décrivons les différents besoins fonctionnels de notre système :

* Enregistrer les nouveaux militaires de l’unité.
* Faciliter les mises à jour nécessaires sur les militiares.
* Suivre l’activité et l'évolution de chaque militaire.
* Gérer les demandes de congé.
* Gérer les absences.
* Gérer les formations.
* Gérer les punitions.
* Pouvoir envoyer des notifications par SMS (ie : Avertir des absences).
* La recherche facile d’une fiche de renseignement.
* Lister tous les personnels appartenant à une sous-unité (section, compagnie...).
* Gestion d’affectation du problème selon des critères ayant des coefficients (La Recherche operationnelle).
* D’autres fonctionnalités.

**I.3. Besoins non fonctionnels**

Les besoins non fonctionnels sont des besoins qui ont un aspect visible pour l’utilisateur, mais qui ne sont pas reliés directement au comportement du système. Les besoins non fonctionnels de notre système se décrivent comme suit :

**Besoins de disponibilité :** notre application constitue le cœur de l’activité de la base, il est indispensable que cette dernière soit disponible à tout moment.

**Besoins de sécurité :** vu que cette application contient des données confidentielles, tous les accès aux différents espaces (administrateur, personnel, etc.) doivent être protégés par un mot de passe et un privilège d’accès.

**Besoins de performance :** il s’agit d’optimiser le temps de chargements des pages par l’utilisation des bonnes pratiques du développement.

**Besoins de portabilité et de compatibilité :** notre application doit être portable sur tous les environnements logiciels (Windows, Mac OS, Linux) et compatible avec ces systèmes d’exploitation.

**Besoins d’utilisation :** Tous les standards d’ergonomies doivent être présents : interface utilisateur bien claire et simple dans l’utilisation.

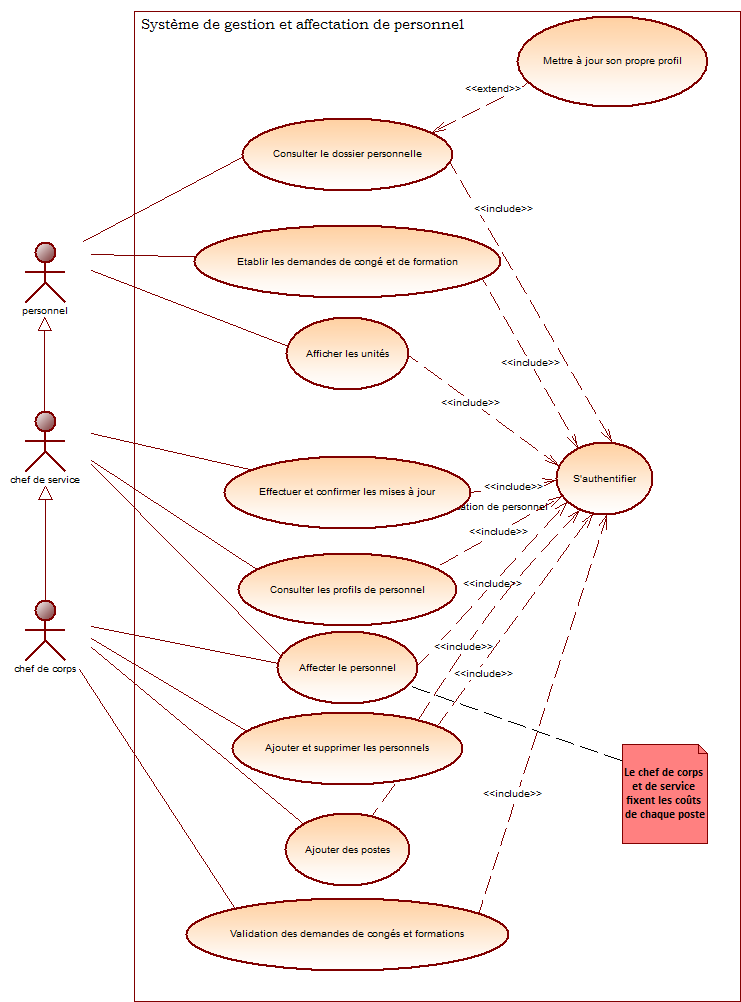
**I.4. Scenario**

A cette étape, on recapitule les fonctionnalités présentant des privilèges particulieres de chaque acteur par rapport à un autre.

|  |  |
| --- | --- |
| Personnel | * Consulter son dossier personnel. * Afficher les listes du personnel appartenant à une unité. * Saisir les demandes de congés. * Demander de participer à une formation. * Consulter ses sanctions, formations, etc. * Mettre à jour de son propre profil (Confirmation est effectuée par un supérieur hiérarchique) |
| Chef de service | * Consuler tous les dossiers des personnels * Mise à jour des profils * Confirmer les mises à jour de personnel * Affecter les personnels dans les postes appartenant à son service |
| Chef de corps | * Ajouter et supprimer un personnel (profil d’utilisateur) * Valider des demandes de congés et de formations * Affecter les personnels dans n’importe quel poste. * Ajouter des postes |

**II. Diagramme de cas d’utilisation global**

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. Pour le cas d’utilisation que l’on a conclu à partir des sections précédentes, (à savoir les acteurs, besoins fonctionnels, etc.), le diagramme doit répondre de manière plus claire à la question *« que va  faire le système et qui l’utilisera? »*.  
Et à partir de ce diagramme, on dégagera les processus principaux et étudier chacun à l’aide des diagrammes UML.

Figure 3 : Diagramme des cas d'utilisation global

**III. Problème d’Affectation :**

On entend souvent parler d’affectation, de prise de décision, de faire un choix dans n’importe quel domaine de la vie.il s’agit en effet d’une grande branche de science, c’est la recherche opérationnelle.

Pour le grand public, c’est une technique récente, qui date de la 2eme guerre mondiale .la première fois ou on l’a appliqué ce sont les opérations militaires

C’est pourquoi l’on peut légitimement s’interroger sur les raisons de l’apparition si tardive de la recherche opérationnelle

Il est bon d’énumérer quelque unes des principales raisons de cette naissance laborieuse :

Le recours aux mathématiques était fatal car la science est parvenue à une certaine maturité. A notre époque les problèmes sont devenus irrémédiablement complexes en raison de la taille croissante des firmes et des liens qui les unissent.

A ce propos, il peut être utile de remarquer que la recherche opérationnelle est une simple exécution du travail intellectuel, donc c’est l’auxiliaire de la décision humaine. Jadis on pensait que les décisions à propos de différents phénomènes étaient du ressort du seul bon sens. Néanmoins, cette recherche constitue la méthode adéquate pour ramener l’attention de l’homme aux domaines ou sa raison individuelle peut s’exercer efficacement c'est-à-dire au niveau des choix.

Pour s’en convaincre, il suffit de constater que l’énumération de 20 ! Solutions (nombre des affectations de vingt personnes a vingt postes) a la vitesse de microseconde, prendrait 77096ans.ou bien d’imaginer un instant qu’il est possible d’énumérer les fonctions différentes de 9 variables binaires

Chaque décideur est astreint de prendre une décision, de faire un choix et affecter telle personne à tel poste. Mais vu l’ampleur de ces problèmes une méthode de résolution s’impose pour lui faciliter la tâche.

Pour tirer au clair cette problématique, une étude qui passe par une formulation du problème, enchainée par une modélisation pour clore avec une résolution de ce probleme.

**III.1 Formulation du problème**Le problème d’affectation peut être représenté sous forme d’un graphe biparti de la manière  
suivant :

L’ensemble des militaires

L’ensemble des postes

Figure: la représentation du problème de d’affectation.

**III.2 Modélisation du problème d’affectation standard:**Le problème d’affectation peut être modélisé comme suit :

Min z =   
Sujet à :

Ou : C = (cij)🡺 Matrice des couts d’affectation

1 si le ième personne est affecté à la jème tache

Xij

0 sinon

Remarque :

Le problème d’affectation est dit non standard, si on a m personnes et n tâches avec m ≠ n. Mais on peut transformer ce problème non standard à un problème standard de la manière suivante :

Si m > n alors nous créons m-n tâches fictives.

Si n > m alors nous créons n-m tâches fictives.

Le cout d’affectation de ces élément fictive est égale à M(ou M>0).

Remarque :

Le problème standard d’affectation, peut être résolût comme un problème de transport ou ai = bj= 1, et les Cij représente le cout unitaire de transport.

Remarque :

Le problème d’affectation peut être défini comme un problème de maximisation.

Ou la fonction objective devient :

Max z =

Ainsi, le nouveau problème consistera à affecter les n personnes aux n taches de telle sorte à maximiser la rentabilité totale sous les mêmes contraintes du problème de minimisation.

Propriété :

Supposons Cij ≥ 0. L’affectation optimale n’est pas modifiée si on ajoute ou on soustrait une constante à tous les éléments d’une ligne ou d’une colonne quelconque de la matrice du cout du problème standard d’affectation.

Preuve :

Supposons, sans perte de généralité, que l’on ajoute une constante K à tous les éléments de la première ligne de la matrice cout, autrement dit, on remplace C1j par C1j + K, alors le nouveau problème d’affectation aurait comme fonction objectif :

On sait que z =

On ajoute la constant K à la 1ere ligne on aura :

Z’ =

=

= (car)

Donc Z’ = Z+K.

**III.3 Modélisation du problème d’affectation standard:**

**III.3.1 Méthode hongroise**

L'algorithme hongrois ou méthode hongroise parfois appelé aussi algorithme de Kuhn-Munkres est un algorithme d'[optimisation combinatoire](http://fr.wikipedia.org/wiki/Optimisation_combinatoire), qui résout le [problème d'affectation](http://fr.wikipedia.org/wiki/Probl%C3%A8me_d%27affectation) en [temps polynomial](http://fr.wikipedia.org/wiki/Th%C3%A9orie_de_la_complexit%C3%A9_des_algorithmes#Classe_P). Il a été proposé en 1955 par le mathématicien américain [Harold Kuhn](http://fr.wikipedia.org/wiki/Harold_W._Kuhn)[6], qui l'a baptisé « méthode hongroise » parce qu'il s'appuyait sur des travaux antérieurs de deux mathématiciens hongrois : [Dénes Kőnig](http://fr.wikipedia.org/wiki/D%C3%A9nes_K%C5%91nig) et [Jenő Egerváry](http://fr.wikipedia.org/w/index.php?title=Jen%C5%91_Egerv%C3%A1ry&action=edit&redlink=1).

[James Munkres](http://fr.wikipedia.org/w/index.php?title=James_Munkres&action=edit&redlink=1) a revu l'algorithme en 1957 et a prouvé qu'il s'exécutait en temps polynomial.

Cet algorithme, sert à résoudre les problèmes d'affectation, problèmes qu'on peut résumer de la manière suivante : considérant une matrice (appelée tableau de coûts), il faut choisir un seul élément par ligne et par colonne de façon à rendre la somme optimale (minimal ou maximal).

Exemple :



Nous exposons l’algorithme angrois sous forme de succession d’étapes :

**ETAPE 0 : REDUCTION DU TABLEAU INITIAL**

On soustrait à chaque ligne du tableau initial, le plus petit élément de la ligne

On fait de même avec les colonnes.



**ETAPE 1 : ENCADRER ET BARRER DES ZEROS**

On cherche la ligne comportant le moins de zéros non barrés (en cas d'égalité, choisir arbitrairement la plus haute).

On encadre un des zéros de cette ligne (arbitrairement le plus à gauche).

On barre tous les zéros se trouvant sur la même ligne ou sur la même colonne que le zéro encadré.

On recommence l'opération jusqu'à ce qu'on ne puisse plus encadrer, ni barrer de zéros :



Exemple :

Si l'on a encadré un zéro par ligne et par colonne, c'est terminé, on a la solution optimale.

Sinon, on passe à l'étape 2.

**ETAPE 2 : MARQUER ET BARRER DES LIGNES ET DES COLONNES**



a) On marque d'une croix toutes les lignes ne contenant aucun zéro encadré.

b) On marque toute colonne ayant un zéro barré sur une ligne marquée.

c) On marque toute ligne ayant un zéro encadré dans une colonne marquée.

On répète alternativement les opérations b) et c) jusqu'à ne plus pouvoir marquer de rangée.

On trace alors un trait sur toute ligne non marquée et sur toute colonne marquée.

Exemple :

**ETAPE 3 : MODIFICATION DU TABLEAU**

Les cases non traversées par un trait constituent un tableau partiel**.**

On retranche à toutes les cases de ce tableau partiel le plus petit élément de celui-ci

On ajoute ce même élément à toutes les cases du tableau initial barrées deux fois

On obtient alors un nouveau tableau sur lequel on pourra répéter la succession des étapes 1 à 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 8 | 5 | **0** | *0* | 3 |
| 7 | 6 | 1 | **0** | 1 |
| 7 | 9 | 9 | 10 | **0** |
| **0** | 3 | 10 | 17 | 9 |
| 5 | **0** | *0* | 8 | 9 |

Exemple :

Ainsi, dans l'exemple, la valeur de l'affectation minimale est 9 + 5 + 5 + 4 + 9 = 3 2

Remarquons, pour finir, que la méthode hongroise, telle qu'elle est décrite, permet de résoudre les problèmes d'affectation minimale (on considère le tableau initial comme un tableau de coûts).

Si l'on veut résoudre un problème d'affectation maximale (c'est à dire en considérant les éléments du tableau comme des indices de satisfaction), il faudra transformer le tableau initial **en retranchant tous les éléments du tableau au plus élevé d'entre eux.**

Création de la matrice des couts d’affectation

REDUCTION DU TABLEAU INITIAL

Existe-il une colonne qui ne contient pas un zéro ?

Pour chaque colonne sans zéro, on soustrait le plus petit cout de toute les éléments de la cette colonne.

Oui

Détermination d’un couplage correspondant aux zéros obtenus.

Non

Détermination d’un support minimal correspondant au couplage maximal.

Modification du tableau

Affectation admissible ?

Non

Oui

*Figure  : l’algorithme de Hongroise.*

**Exemple du problème d’affectation : rotation du personnel militaire :**

A Intervalles réguliers, le chef de corps organisent la rotation d’une partie de son personnel technique entre les différentes postes au de la base militaire.

Plusieurs raisons régissent ces décisions : permettre l'acquisition d'une expérience de travail diversifié, donner l'occasion de suivre des cours aux demandes de mutation vers des postes ou le climat est plus favorable, récompenser certains ou punir d'autre supposons, à titre d’exemple, qu'une liste comportant les noms de 10 spécialistes dans la mécanique des chars d'assaut tous de grade caporal-chef ont ancienneté égale. Souhaitons-les réaffecter chacun au poste de l'un de ses 9 collègues. Certains de ces militaires sont célibataires, d'autres sont mariés mais n'ont pas d’enfants, d'autres encore sont maries et ont des enfants ...

L’autorité hiérarchique a évalue pour chacun les coûts de mutation à chaque poste. L’objectif est d'assurer au moindre cout que chaque sergent occupe un nouveau poste et que tous les postes soient comblés dénotons les S/chef par les lettres A, B,.., H M et N .Et désignons par i le poste occupé présentement par le S/chef 1 : le S/chef A occupe présentement le poste a, et ainsi de suite. Le tableau suivant présent la matrice des couts de mutation de chaque sergent a chacun des postes

MATRICE DES COUTS

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C/Chef | POSTES | | | | | | | | | |
| a | b | C | d | e | f | G | h | m | n |
| A | \* | 12 | 15 | 11 | 17 | 15 | 11 | 12 | 10 | 10 |
| B | 6 | \* | 14 | 12 | 16 | 11 | 17 | 18 | 18 | 16 |
| C | 8 | 17 | \* | 21 | 17 | 16 | 14 | 12 | 10 | 15 |
| D | 7 | 16 | 9 | \* | 12 | 18 | 18 | 14 | 11 | 14 |
| E | 7 | 13 | 8 | 12 | \* | 22 | 19 | 12 | 13 | 12 |
| F | 8 | 8 | 11 | 14 | 12 | \* | 12 | 17 | 9 | 18 |
| G | 6 | 9 | 13 | 9 | 11 | 16 | \* | 14 | 13 | 16 |
| H | 7 | 14 | 16 | 11 | 16 | 22 | 15 | \* | 14 | 18 |
| M | 11 | 16 | 17 | 15 | 17 | 18 | 21 | 22 | \* | 11 |
| N | 8 | 9 | 8 | 13 | 9 | 7 | 8 | 9 | 8 | \* |

Comme il n’est pas permis qu’un caporal-chef conserve le même poste (poste qu’il occupe présentement), on remplace chaque astérisque de la diagonale par un montant M largement supérieur à ceux qui sont en jeu pour la mutation envisagée, en marquant ainsi l’impossibilité de maintenir un sergent dans son poste actuel.

Posons par exemple : M = 500.

**1-VARIABLE DE DECISION :**

On définit les variables de décision binaires suivantes :

Vij = 1 si le sergent I est muté du poste i au poste j.

La fonction-objectif s’écrit comme suit :

Min Z = 500 VAa + 12 VAb+ 15 VAc + …+9VNh+8VNm+500VNm

Les contraintes indiquent :

Qu’à chaque sergent on doit attribuer un poste ;

Que chaque poste doit être comblé.

**2-MODELE :**

* Â titre d’exemple, la contrainte associée au sergent A est :
* VAa + VAb + VAc + … + VAn = 1
* Et celle associée au poste a :
* VAa + VBa + VCa + … + VNa = 1
* Une solution optimale de

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **C/chef** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **M** | **N** |
| **Poste** | **m** | **f** | **h** | **e** | **c** | **b** | **d** | **a** | **n** | **g** |

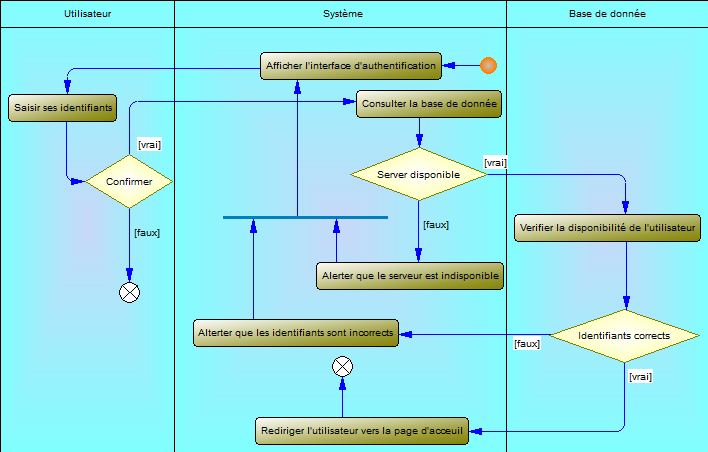
**IV. Modélisation du point de vue utilisateur**

Nous allons maintenant reprendre chacun des processus identifiés et conclus à partir du Ch2.II et les étudier à l’aide de diagrammes UML d’activités. Cette étape permettra par la suite une vue complète, répondant de manière plus détaillée à la question que va faire le système et qui l’utilisera.

En théorie, tous les mécanismes dynamiques pourraient être décrits par les differents diagrammes d’UML, mais seuls les mécanismes complexes ou intéressants méritent d'être représentés c’est pourcela qu’on va représenter limiter les processus identifiés :

**IV.1. Processus d’authentification :**

Pour accéder à notre application, l’utilisateur, que ce soit chef de corps, chef de service ou personnel, doit s’authentifier en entrant son login qui est le CIN et son mot de passe, après vérification des champs saisis l’application interroge la base de données pour charger le groupe et les privilèges accordés à l’utilisateur. L’utilisateur accédera automatiquement à son interface d’accueil personnel. Le processus d’authentification peut être résumé dans le diagramme d’activité suivant :

  
Figure 4: Digramme d’activités authentifications

**IV.2. Processus de modification :**

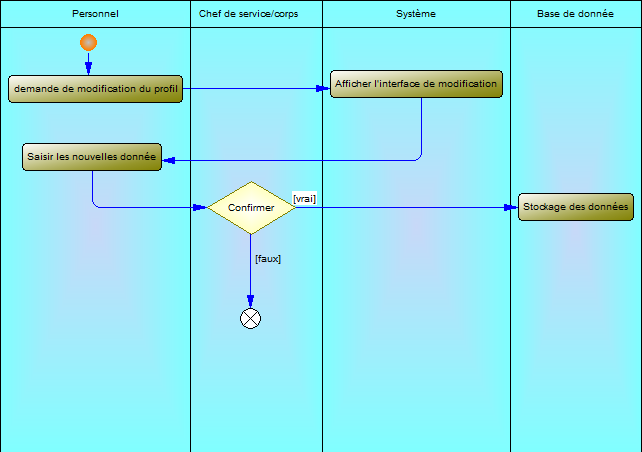


Figure 4: Digramme d’activités authentifications

**Conclusion**

Dans ce chapitre une combinaison de différents types de diagrammes UML (diagramme de cas d’utilisation, diagramme d’activité diagramme de séquence..) a offert une vue complète des aspects statiques et dynamiques de notre système, nous avons aussi pu modéliser le problème de l’affectation et choisir la solution offerte par la méthode hongroise, dans l’implémentation on va essayer d’appliquer cet algorithme par le language Java.

**CHAPITRE III:  
ANALYSE ET CONCEPTION**

**Introduction**

Dans la section précédente, nous avons modélisé ce que le système doit faire, sans pour autant aborder « la question du comment il le fera ? ». Le but de cette partie du travail est de se pencher sur l’aspect logique du futur système. Encore une fois, nous allons continuer sur les pratiques du processus unifié de Rational (RUP), dont la discipline Anlyse et Conception (Analysis and Design)a pour but de décrire comment le système sera implémenté. Parmi les multiples tâches de cette discipline, sont l’analyse architecturale et la conception de la base de données.

**I. Considérations préalables**

**I.1 Outil de modélisation : PowerDesigner**  
 AMCDesignor (c'est son nom d'origine) a été conçu à la base pour palier au manque d'outil de modélisation graphique pour Oracle (d'où le or final). Il était basé sur la méthodologie Merise. Lors de son rachat par Powersoft, l'application a été traduite en anglais et le module modèle conceptuel de données a été refondu pour la méthodologie IE (Information engineering), plus exactement selon la méthode Yourdon-DeMarco.   
Deux versions apparurent donc parallèlement:

* PowerAMC, méthodologie Merise, application en français
* PowerDesigner, méthodologie IE, application en anglais

Sybase, lors de son rachat de Powersoft, essaya de supprimer la version francophone, mais face à la levée de bouclier des analystes merisiens, la tentative fut un échec. Depuis leur version 7, PowerDesigner et PowerAMC ne se différencient plus que par la langue, hormis la spécificité des traitements Merise.  
L'application est multi méthodes depuis sa refonte (version 7) et gère tous les diagrammes UML 2.0 dans ses versions Studio.

Alors, on va utiliser ce logiciel de modélisation puisqu’il va nous permettre de mettre  
on œuvre le standard UML avec ses fameux diagrammes ainsi que la méthode  
Merise en facilitant la création de la base de donnée et de développement d’application.

**I.2. Architecture du futur système**

L’architecture décrit les composantes du futur système d’information. Dans le cadre de ce travail, le futur système sera implémenté sur une architecture en trois couches:

1. Une couche de présentation, qui sert d’interface entre le système et ses utilisateurs : C’est l’application proprement dite.
2. Une couche de logique métier, qui implémente les règles de gestion et effectue les calculs : des pages programmées en JAVA.
3. Une couche d’accès aux données, qui communique avec la base de données à l’aide d’un langage de requêtes : Création de la base de données avec MySQL

**II. Conception de la base de données**

Cette section traite de la conception de la base de données (BD).

La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels nous nous intéressons.  
  
**II.1. Présentation de la méthode MERISE**

MERISE est une méthode française née dans les années 70, développée initialement par Hubert Tardieu. Elle fut ensuite mise en avant dans les années 80, à la demande du Ministère de l'Industrie qui souhaitait une méthode de conception des SI.

MERISE est donc une méthode d'analyse et de conception des SI basée sur le principe de la séparation des données et des traitements. Elle possède un certain nombre de modèles (ou schémas) qui sont répartis sur 3 niveaux :

* Le niveau **conceptuel**,
* Le niveau **logique** ou **organisationnel**,
* Le niveau **physique**.

Merise est encore tout à fait valable pour:

* la modélisation des données en vue de la construction d'une base de données relationnelle,
* la modélisation des processus métiers d'un SI automatisé en partie par du logiciel,
* la formalisation des besoins utilisateur dans le cadre de cahier des charge utilisateur, en vue de la conception d'un logiciel adapté.

Donc, MERISE reste la meilleure méthode pour la conception de notre base de données, ainsi qu’elle permet de se faire aisément comprendre.

**N.B**: Merise et UML sont techniquement **complémentaires** dans notre analyse.  
L'élaboration de notre MCD passe par les étapes suivantes :

* La mise en place de **règles de gestion** (déjà préparé dans la phase d’analyse des besoins),
* L'élaboration du **dictionnaire des données**,
* La recherche des **dépendances fonctionnelles** entre ces données,
* L'élaboration du MCD (création des **entités** puis des **associations** puis ajout des **cardinalités**).

**II.2. L’élaboration du dictionnaire des données**

Les données qui figurent dans le MCD (et donc dans le dictionnaire des données) doivent être, dans la plupart des cas, **élémentaires**, alors on doit éliminer les propriétés calculées et décomposer les propriétés concaténée :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom abrégé** | **Nom détaillé** | **Nature** | **Type** | **Taille** | **Remarques** |
| cin per | CIN du prsonnel | Elémentaire | AlphaNumerique | 10 | Clé primaire |
| nom\_per | Nom du personnel | Elémentaire | Alphabetique | 40 | -- |
| sexe\_per | Sexe du personnel | Elémentaire | Alphabetique | 40 | -- |
| grade | Grade du personnel | Elémentaire | Alphabetique | 20 | -- |
| lieu\_ns | Lieu de naissance | Concaténée | AlphaNumerique | 180 | -- |
| date\_ns | Date de naissance | Concaténée | Date | -- | Pas nécessaire de décomposer |
| sit\_fam | Situation familiale | Elémentaire | Alphabetique | 20 | -- |
| tel\_per | Numero de telephone | Elémentaire | Numérique | -- | -- |
| photo\_per | Photo personnelle | Elémentaire | Image (Pending) | -- | -- |
| nbr\_enf | Nombre d’enfants | Elémentaire | Numérique | -- | -- |
| ctr\_conge | Contour de congé | Elémentaire | Numérique | -- | -- |
| date\_engag | Date d’engagement | Concaténée | Date | -- | Pas la peine de décomposer |
| note | Note du discipline | Elémentaire | Numérique | -- | -- |
| password | Password du login | Elémentaire | AlphaNumérique | 30 | -- |
| level | Level d’administration | Elémentaire | Numérique | -- | -- |
| email | Email | Elémentaire | AlphaNumerique | 70 | -- |
| id\_service | Identifiant du service | Elémentaire | Numérique | -- | Clé primaire |
| libelle\_service | Nom du service | Elémentaire | AlphaNumerique | 60 | -- |
| date\_effectuation | Date effectuation du service | Date | Date | -- | Pas nécessaire de décomposer |
| id\_conge | Identifiant du congé | Elémentaire | Numérique | -- | Clé primaire |
| datedeb\_cong | Datedeb\_congé | Date | Date | -- | Pas nécessaire de décomposer |
| datefin\_cong | Datefin\_congé | Date | Date | -- | Pas nécessaire de décomposer |
| type\_cong | Type de congé | Elémentaire | Alphabétique | 20 | -- |
| etat\_cong | Status de congé | Elémentaire | Alphabetique | 20 | -- |
| comment\_chef | Chef commentaire du chef | Elémentaire | Alphabétique | 80 | -- |
| etat\_cong | Status de congé | Elémentaire | Alphabetique | 20 | -- |
| id\_abs | Identifiant d’absence | Elémentaire | Numérique | -- | Clé primaire |
| date\_abs | Date d’absence | Date | Date | -- | Pas nécessaire de décomposer |
| duree\_abs | Duree d’absence | Elémentaire | Numérique | -- | -- |
| motif\_abs | Motif d’absence | Elémentaire | AlphaNumerique | 70 | -- |
| id\_sanction | Identifiant de sanction | Elémentaire | Numérique | -- | Clé primaire |
| dateDebut\_sanction | Date debut de sanction | Date | Date | -- | Pas nécessaire de décomposer |
| dateFin\_sanction | Date fin de sanction | Date | Date | -- | Pas nécessaire de décomposer |
| nature\_sanction | Nature de sanction | Elémentaire | AlphaNumerique | 40 | -- |
| motif\_sanction | Motif de sanction | Elémentaire | AlphaNumerique | 70 | -- |
| id\_unite | Identifiant d’unité/sous-unité | Elémentaire | Numérique | -- | Clé primaire |
| libelle\_unite | Nom de (sous-) unité | Elémentaire | AlphaNumerique | 40 | -- |
| id\_fonction | Identifiant de fonction | Elémentaire |  | -- | Clé primaire |
| libelle\_fonction | Nom de fonction | Elémentaire | AlphaNumerique | 60 | -- |
| id\_diplo | Identifiant du diplôme | Elémentaire | Numérique | -- | Clé primaire |
| libelle\_diplo | Libelle du diplôme | Elémentaire | AlphaNumerique | 60 | -- |
| date\_ob\_dip | Date obtention diplôme | Date | Date | -- | Pas nécessaire de décomposer |
| mil\_mat | Matricule du militaire | Elémentaire | AlphaNumerique | 15 | -- |
| civ\_mat | Matricule du civil | Elémentaire | AlphaNumerique | 15 | -- |
| id\_notif | Identifiant de notification | Elémentaire | Numérique | -- | Clé primaire |
| obj\_notif | Obje de notification | Elémentaire | AlphaNumerique | 70 | -- |
| contenu\_notif | Contenu de notification | Elémentaire |  | Long | -- |
| id\_form | Identifiant de formation | Elémentaire | Numérique | -- | Clé primaire |
| datedeb\_form | Date début de formation | Date | Date | -- | Pas nécessaire de décomposer |
| datefin\_form | Date fin de formation | Date | Date | -- | Pas nécessaire de décomposer |
| obje\_form | Object de formation | Elémentaire | AlphaNumerique | 70 | -- |
| theme\_form | Theme de formation | Elémentaire | AlphaNumerique | 50 | -- |
| pays\_form | Pays de formation | Elémentaire | AlphaNumerique | 40 | -- |
| lieu\_form | Lieu de formation | Elémentaire | AlphaNumerique | 30 | -- |
| id\_adresse | Identifiant d’adresse | Elémentaire | Numérique | -- | Clé primaire |
| ville\_adresse | Ville d’habitation | Elémentaire | AlphaNumerique | 30 | -- |
| lieu\_adresse | Lieu d’habitation | Elémentaire | AlphaNumerique | 150 | -- |
| id\_poste | Identifiant de poste | Elémentaire | Numérique | -- | Clé primaire |
| nom\_poste | Nom du poste | Elémentaire | AlphaNumerique | 80 | -- |
| id\_aff | Identifiant affectation | Elémentaire | Numérique | -- | -- |
| cout\_nbr\_enf | Cout nombre d’enfants | Elémentaire | Numérique | -- | -- |
| cout\_sit\_fam | Cout situation familiale | Elémentaire | Numérique | -- | -- |
| cout\_note | Cout note | Elémentaire | Numérique | -- | -- |
| cout\_anci | Cout ancienneté | Elémentaire | Numérique | -- | -- |
| cout\_formation | Cout formation | Elémentaire | Numérique | -- | -- |
| cout\_abs | Cout absence | Elémentaire | Numérique | -- | Son coefficent sera négatif |
| cout\_final | Cout final | Calculée | Numérique | -- | calculée (à supprimer) |
| id\_param | Identifiant paramètre | Elémentaire | Numérique | -- | -- |
| coef\_enf | Coefficient nombre enfants | Elémentaire | Numérique | -- | -- |
| coef\_sitfam | Coefficient situation familiale | Elémentaire | Numérique | -- | -- |
| coef\_note | Coeffiecient note | Elémentaire | Numérique | -- | -- |
| coef\_anci | Coeffiecient ancienneté | Elémentaire | Numérique | -- | -- |
| coef\_formation | Coeffiecient formation | Elémentaire | Numérique | -- | -- |
| coef\_abs | Coeffiecient absence | Elémentaire | Numérique | -- | -- |
| date\_debut | Date debut mission civile | Elémentaire | Date | -- | -- |
| date\_fin | Date fin mission civile | Elémentaire | Date | -- | -- |
| etat | Etat du personnel civil | Elémentaire | Alphabétique | -- | -- |
|  |  |  |  |  |  |

**II.3. Dépendances fonctionnelles**

Soit deux propriétés (ou données) P1 et P2. On dit que P1 et P2 sont reliées par une **dépendance fonctionnelle** (DF) si et seulement si une**occurrence** (ou valeur) de P1 permet de connaître une et une seule occurrence de P2.

Une DF doit être :

* **élémentaire** : C'est l'intégralité de la source qui doit déterminer le but d'une DF. Par exemple si P1 → P3 alors P1,P2 → P3 n'est pas élémentaire.
* **directe** : La DF ne doit pas être obtenue par transitivité. Par exemple, si P1 → P2 et P2 → P3 alors P1 → P3 a été obtenue par transitivité et n'est donc pas directe.

En reprenant les données du dictionnaire précédent, on peut établir les DF suivantes :

Cin\_per nom\_per, prenom\_per, sexe\_per, grade, lieu\_ns, date\_ns, adresse,…

Id\_conge datedeb\_cong, datefin\_cong, type\_cong, etat\_cong

Id\_abs date\_abs, duree\_abs, motif\_abs

Id\_sanction datedebut\_sanction, date\_fin sanction, nature\_sanction, motif\_sanction

Id\_form datedeb\_form, datefin\_form, obje\_form, theme\_form, lieu\_form

Id\_adresse ville\_adresse, lieu\_adresse

Id\_poste nom\_poste

Id\_service libelle\_service

Id\_notif obj\_notif, contenu\_notif

Id\_diplo libelle\_diplo, date\_ob\_dip

Id\_aff cout\_nbr\_enf, cout\_sit\_fam, cout\_note, cout\_anci, cout\_ formation, …

Id\_param coef\_enf, coef\_sitfam, coef\_note, coef\_anci, coef\_formation, coef\_abs

Id\_fonction libelle\_fonction

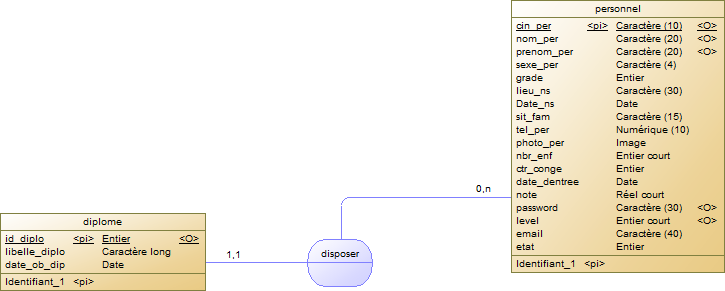
Id\_unite libelle\_unite

**II.4. Le Modèle conceptuel de données**

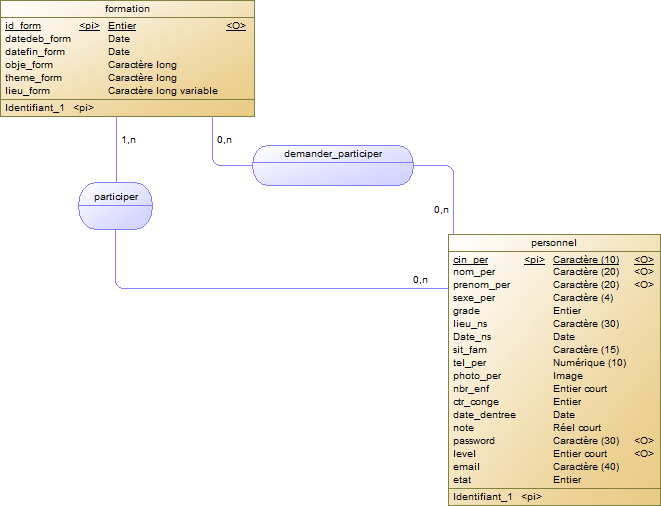
Le modèle conceptuel de données (MCD) a pour but d'écrire de façon formelle les données qui sont utilisées par le système d'information. Comme le modèle est de grande taille, on le sépara pour l’étude.

Les figures suivantes montrent le modèle conceptuel de données que nous avons développé pour notre application :

Les différentes entités y apparaissent avec leurs principaux attributs, dont les clés primaires soulignées.

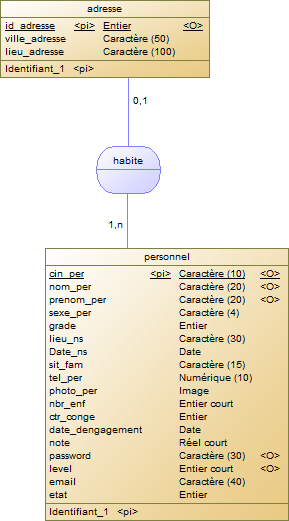


Un personnel peut disposer d’un ou aucun diplôme alors qu’un diplôme peut être disposé par un seul militaire.

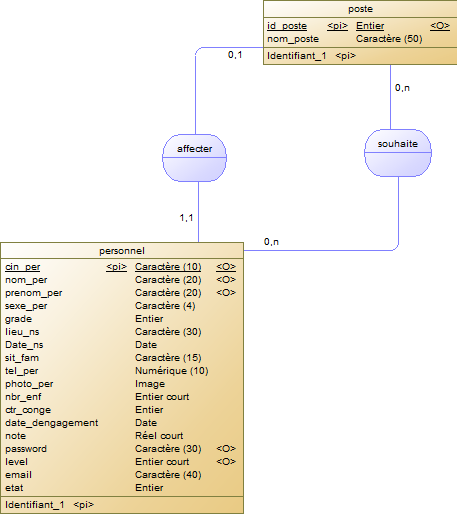
****

Ici les deux entités (personnel et formation) sont reliées par deux associations, parce qu’on a deux relations avec deux sens differents.  
Pour un utilisateur, on veut sauvegarder ses formations dans une table. Donc, Un personnel peut participer à plusieurs formations et une formation peut avoir au moins ou plusieurs stagiaires.

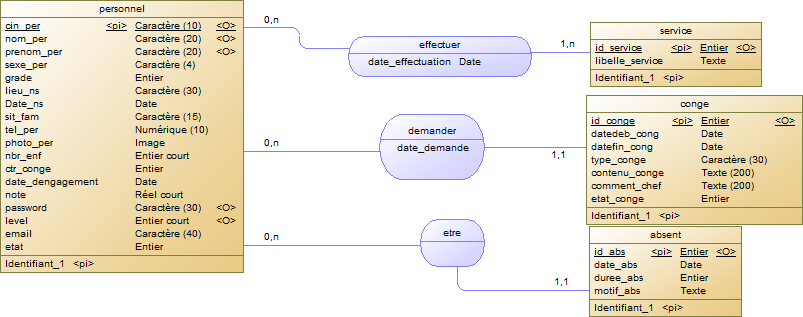
D’autre part, un personnel peut demander la participation dans plusieurs formations, et une formation peut avoir plusieurs demandes, toutfois on doit tenir compte qu’un personnel ne peut participer qu’à une seule formation courante.



On souhaite gérer mieux les adresses des personnel, c’est pourcela on va créer une table ‘adresse’ qui contient tous les informations de la résidence du personnel.  
Un personnel peut avoir au minimum une adresse (Une addresse obligatoire, et une autre qui sera facultative), et une adresse peut, en termes de l’unicité, être affectée à une seule personne, si on veut changer l’adresse d’un personnel, il suffit de modifier la table adresse dont l’ID du personnel va migrer vers la table d‘adresse.



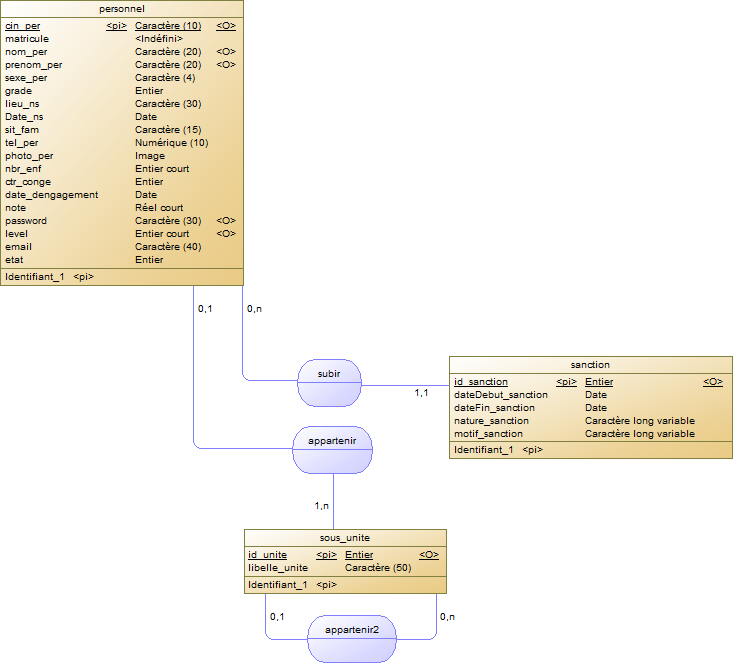
Ici les deux entités (personnel et poste) sont reliées par deux associations, parce qu’on a deux relations avec deux sens differents.  
Le premier sens indique que le personnel peut choisir plusieurs postes à la fois, et un poste peut être demandé par plusieur personnel.  
Le deuxième sens indique qu’un personnel peut occuper un seul poste, et un poste peut être occupé au maximum par un personnel ou aucun.



On souhaite créer un enregistrement pour les services effectués par une personne, pour suivre ses activités, ainsi le chef peut donner la note méritée. Un personnel peut effectuer plusieurs services, et un service peut être effectué par plusieurs personnes (service de garde, patrouille..).

Un personnel peut soumettre une demande de congé qui sera enregistrée en attente d’être soit acceptée ou refusée. En effet le chef de corps est le responsable de valider ou refuser son congé. Le personnel peut saisir plusieurs demandes de congé, et chaque congé sera affecté relativement à une seule personne.

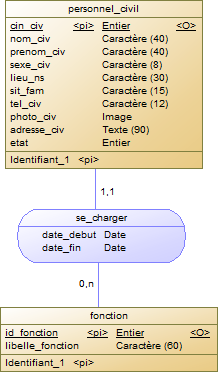
On souhaite bien enregistrer toutes les absences des personnes, et les gérer dans une seule table où chaque enregistrement jointe la personne en question. Chaque personnel peut s’absenter plusieurs fois, et une absence peut être lui attribuée.



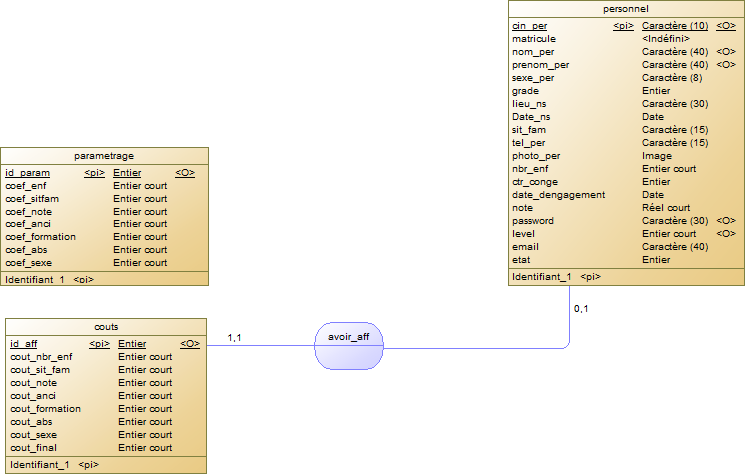
Pour les sanctions, on souhaite sauvegarder les enregistrements dans une table pour maintenir l’intégrité des données dans un format accessible. Un personnel peut avoir plusieurs enregistrements, et un enregistrement peut référencer à une et une seule personne.

Même chose pour les unités, chaque personnel ne peut appartenir qu’à une seule unité, et une unité contient un nombre de personnel.

En ce qui concerne l’association ‘appartenir2’, elle signifie qu’une sous-unité peut appartenir à une unité supérieure, et cette dernière peut avoir plusieurs sous-unités. Exemple : Une compagnie de service et de commandement est constituée de 3 sections et une autre pour le commandement.

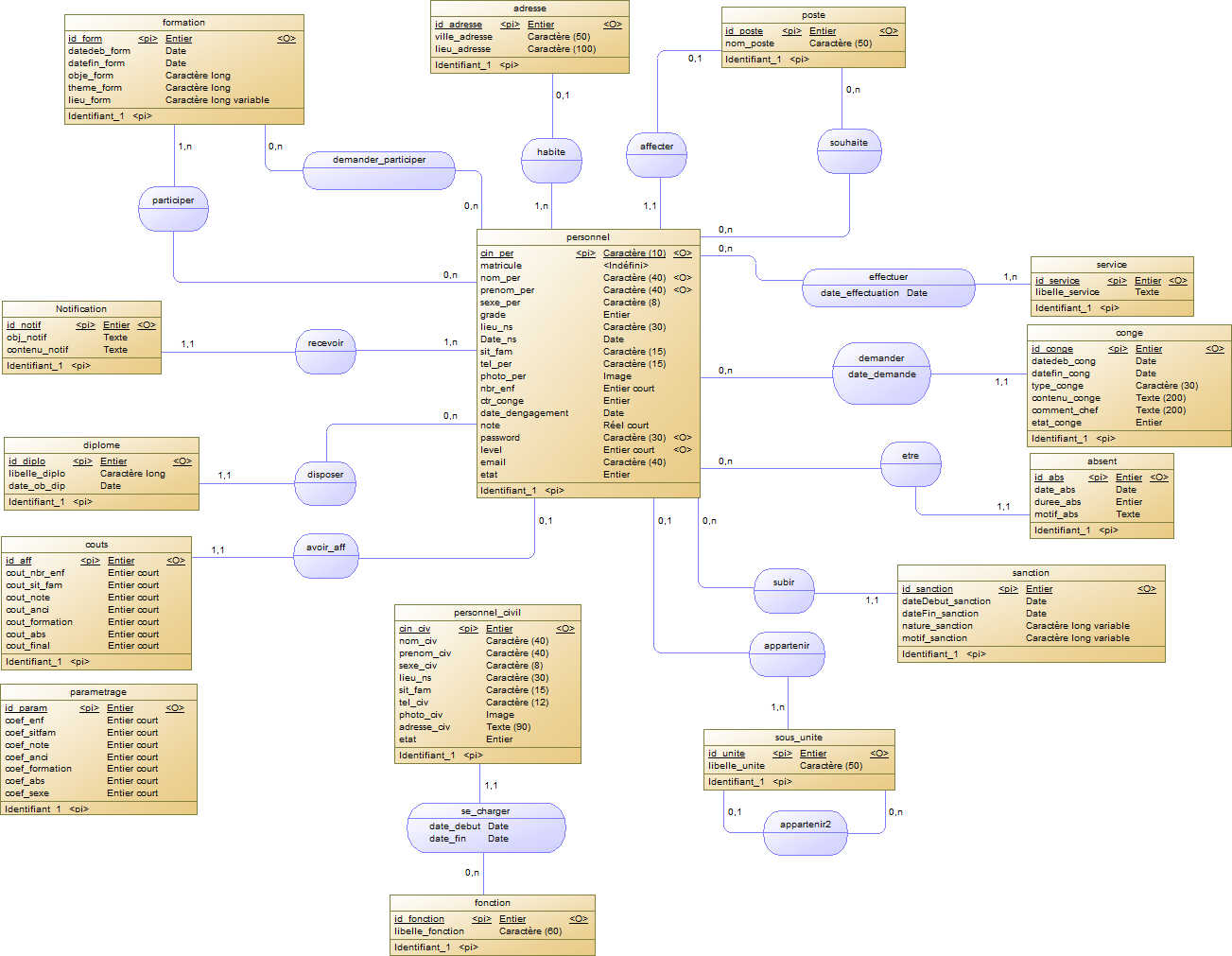


Nous gardons à l’esprit qu’il se peut, dans une unité militaire, de trouver des personnes civiles qui prennet des tâches spécifiques (Le cas des enseignants civils à l'Académie Royale Militaire) comme on a vu dans la partie **Chapitre II.2.2**. Nous ne nous intéressons pas à la gestion du personnel détaché, car cela n’est pas notre sujet. Toutefois, il est préférable pour des raisons administratives de permettre au chef de corps de suivre les activités du personnel civil, il suffit d’avoir une table qui permet de stocker tout le personnel civil, ainsi que les fonctions qui leurs sont attribuées.



On souhaite aussi de stocker les coûts pour les differentes affectation de l’unité, pour qu’il soit facile à réutiliser des coûts qui sont déjà saisis. Par exemple, le chef supérieur de tel service a sous son commandement 100 postes, il souhaite effectuer une rotation du personnel pour une deuxième fois. Il est difficile de saisir tous les critères à nouveau, d’où la nécessité d’une table pour stocker ces coefficients et qu’elle sera accessible à chaque tentative de rotation du personnel. On a ici la table ’couts’ est lié au personnel en question, les enregistrements dans cette table signifient les coûts obtenus (coefficient \* parametre).

**MCD global :**



**CHAPITRE IV : IMPLEMENTATION DU PROTOTYPE**

**Introduction**

Après avoir entamé les phases d’analyse et de conception, un prototype du système sera implementé. Nous pouvons maintenant nous lancer les travaux nécessaires pour produire le prototype. Nous allors aussi consacrer ce chapitre pour la présentation des langages et outils de programmation utilisés pour la réalisation de notre application.

**I. Environnement de développement**

L’environnement de développement est un terme qui désigne l’ensemble d’outils et de langage utilisé pour l’implémentation d’une solution informatique. Nous commençons par l’environnement matériel.

**I.1. Environnement matériel**

Pour le développement de notre application, nous avons passé par un seul environnement

#### **I.1.1 Environnement de développement et du test**

* Ordinateur : HP Notebook
* Processeur : Intel I3
* Mémoire vive : 4GO
* Disque dur : 500GO
* Version JDK : 8 (1.8.0.77)
* Version Apache : 2.4.0
* Version MySQL : 5.7.11
* Système d’exploitation : Windows

**I.2. Environnement logiciel**

Les logiciels utilisés pour l’implémentation de notre solution sont les suivants :

* **EasyPHP**

EasyPHP est un environnement de développement pour les développeurs PHP. Il inclut des outils de développement et d'autres fonctionnalités utiles comme une base de données MySQL et un serveur Apache.

EasyPHP est un excellent outil pour ceux qui ne sont pas à l'aise pour coder en PHP. Pour les codeurs plus expérimentés, il peut encore représenter un gain de temps séduisant. Le package combine efficacement MySQL, Apache, Xdebug et PHPMyadmin avec PHP, plus des outils de développement, pour créer une installation autonome.



Il est également préconfiguré pour accepter les modules plug-in, permettant à sa fonctionnalité d'être étendue pour prendre en charge les systèmes de gestion de contenu comme WordPress, Spip, phpBB, Drupal et Joomla.

**Points forts:**Simple à apprendre, avec de nombreuses fonctionnalités.

**Points faibles:** Un peu limité, aide faible.

* **Eclipse**

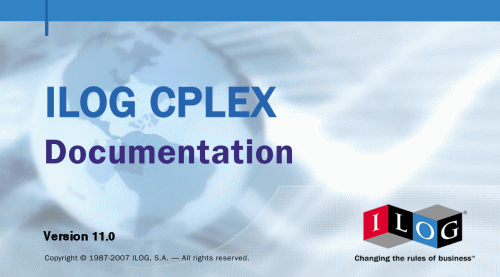
Eclipse est un projet de la fondation Eclipse visant à développer tout un environnement de développement libre, extensible, universel et polyvalent.



Son objectif est de produire et fournir divers outils gravitant autour de la réalisation du logiciel, englobant les activités de codage logiciel proprement dites (avec notamment un environnement de développement intégré) mais aussi de modélisation, de conception, de test, etc. Son environnement de développement notamment vise à la généricité pour lui permettre de supporter n’importe quel language de programmation.

* **Cplex :**

ILOG CPLEX (plus communément appelé "CPLEX") est un outil informatique d'[optimisation](http://fr.wikipedia.org/wiki/Optimisation_%28math%C3%A9matiques%29). Son nom fait référence au [langage C](http://fr.wikipedia.org/wiki/Langage_C) et à l'[algorithme du simplexe](http://fr.wikipedia.org/wiki/Algorithme_du_simplexe)



Il est composé d'un exécutable (CPLEX Interactif) et d'une [bibliothèque](http://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) de fonctions pouvant s'interfacer avec différents [langages de programmation](http://fr.wikipedia.org/wiki/Langages_de_programmation) (CPLEX Callable Library) : [C](http://fr.wikipedia.org/wiki/Langage_C), [C++](http://fr.wikipedia.org/wiki/C%2B%2B), [C#](http://fr.wikipedia.org/wiki/C_sharp), [Java](http://fr.wikipedia.org/wiki/Java_%28langage%29) et [Python](http://fr.wikipedia.org/wiki/Python_%28langage%29).

## I.3. Langages de programmation

## Les langages de programmations utilisées pour le développement de notre application sont les suivants :

**Java** est à la fois un langage de programmation informatique orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu’Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

Java permet de développer des applications autonomes mais aussi, et surtout, des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.



Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java, dont le nom technique est JRE (Java RuntimeEnvironment - Environnement d'exécution Java). Cette dernière est constituée d'une JVM (Java Virtual Machine - Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes.

* **JAVAFX**

JavaFX est une technologie créée par [Sun Microsystems](https://fr.wikipedia.org/wiki/Sun_Microsystems) qui appartient désormais à [Oracle](https://fr.wikipedia.org/wiki/Oracle_(entreprise)), à la suite du rachat de [Sun Microsystems](https://fr.wikipedia.org/wiki/Sun_Microsystems) par [Oracle](https://fr.wikipedia.org/wiki/Oracle_(entreprise)) le 20 avril 2009.

Avec l'apparition de Java 8 en mars 2014, JavaFX devient l'outil de création d'interface graphique ('GUI toolkit') officiel du langage Java, pour toutes les sortes d'application (applications mobiles, applications sur poste de travail, applications Web...), le développement de son prédécesseur [Swing](https://fr.wikipedia.org/wiki/Swing_(Java)) étant abandonné (sauf pour les corrections de bogues).



JavaFX est désormais une pure API Java (le langage de script spécifique qui a été un temps associé à JavaFX est maintenant abandonné).

JavaFX contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc.

Le SDK de JavaFX étant désormais intégré au JDK standard [Java SE](https://fr.wikipedia.org/wiki/Java_SE), il n'y a pas besoin de réaliser d'installation spécifique pour JavaFX.

**II. Réalisation**

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

En attendant la fin de l’application

**IV. Conclusion**

Ce chapitre nous a permis de mettre en œuvre notre conception. Nous avons commencé par présenté l’environnement matériel et logiciel de travail. Pour la suit, nous avons présenté quelques interfaces de l’application afin de donner une meilleure idée sur le travail réalisé.

**Conclusion**

Dans le cadre de notre projet de fin d’étude, nous avons conçu et développé une application de gestion et d’affectation du personnel utilisable dans chaque unité militaire. Le présent manuscrit détaille toutes les étapes par lesquelles nous sommes passées pour arriver au résultat attendu. Nous avons essayé tout au long de notre travail de construire notre application incrément par incrément en utilisant la méthodologie RUP.

Nous avons commencé dans un premier lieu par comprendre le contexte général de notre application et identifier les différentes exigences de notre futur système. Nous avons préparé par la suite notre planning de travail en respectant les priorités de nos besoins suite à une discussion avec le personnel militaire de l’Académie Royale Militaire.

Malgré toutes les difficultés rencontrées au niveau du processus de l’ARM et les contraintes de temps et de la formation militaire, nous avons réussi à réaliser la totalité de notre application tout en respectant l’aspect sécuritaire et en préparant la documentation nécessaire.

Ce travail était très intéressant puisqu’il nous a permis de découvrir un nouveau domaine de travail et de s’éloigner des projets traditionnels de type site web. Il nous a permis d’approfondir nos connaissances dans les bonnes pratiques de programmation notamment la sécurisation des données et la documentation des codes sources, etc.

Finalement, notre travail ne s’arrête pas à ce niveau, en effet plusieurs fonctionnalités peuvent être ajoutées à notre application notamment un système pour la gestion des emplois du temps et qui se base principalement sur la gestion des personnels qui sont déjà déployés dans notre application.