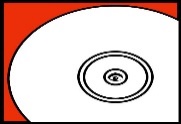
**République de Côte d’Ivoire**

**Union – Discipline – Travail**

**Ministère de l’Enseignement**

**Supérieur et de la Recherche Scientifique**





**CEFIVE**

**PROJET DE FIN DE CYCLE**

**Pour l’obtention du diplôme d’ingénieur des Techniques**

**En Informatique**

* ***Option : Système de Gestion des Bases de Données, Génie Logiciel et Applications Métiers***

***THEME :***

**AUTOMATISATION DE CERTAINES FONCTIONS DE LA GESTION ADMINISTRATIVE DU PERSONNEL : CAS DE LA DIRECTION DES MARCHES PUBLICS (DGMP)**

# 

**Maître de stage:**

**M. DJAN DJAN OLIVIER**

Réalisé par **COULIBALY LENIMAMAN IBRAHIMA**

**Promotion : 2017 - 2019**

**Directeur de projet:**

**M AFFIAN ROGER**

# DEDICACES

# DEDICACE

A ma famille

# EMERCIEMENTS

Je remercie le Directeur Général des Marchés Publics, Monsieur Youl Sansan François, pour le stage qu'il m'a offert ainsi que pour l'opportunité que j'ai pu avoir de réaliser cette formation en cours du soir alors qu'étant en exercice dans la structure qu'il dirige.

Je remercier également le Sous-Directeur des Etudes du Suivi et de l'Evaluation des Marchés Publics, Monsieur Kpagni Pierre pour son accompagnement, ses conseils avisés et surtout pour les facilités dont j'ai pu bénéficier relativement aux heures de descente afin de pouvoir suivre les cours.

Je ne manquerai pas de dire merci au Chef du Service des Moyens Généraux et du Personnel, Monsieur Yao Constant ainsi qu'à ses collaborateurs pour les précieux renseignements qu'ils ont pu me fournir ainsi que pour leur bonne collaboration à la réussite de ce projet.

Mes remerciements vont également à l'endroit du Sous-directeur des Systèmes d'Information et de Communication, Monsieur Bocoum Amadou, ainsi qu’à toute son équipe pour l'encadrement technique

Grand merci à mon maître de stage M. Djan Olivier pour le suivi et l'encadrement professionnel dont j'ai bénéficié de sa part.

Je remercie également le Directeur des Etudes du CEFIVE, M. AFFIAN Roger pour sa grande sollicitude ainsi que l'encadrement technique et pédagogique dont il a fait preuve à mon égard

Mes remerciements vont particulièrement à l'endroit de mon grand frère et indéfectible mentor, Soro Nimbéléfia Amadou pour la relecture, les corrections et les conseils d’usage. Je n’oublierai surtout pas de lui dire merci pour tous les bienfaits que je ne saurais finir de citer dans ce présent rapport.

J’adresse ma profonde gratitude à mon épouse pour son soutien affectif et moral, ainsi que pour ses encouragements.

Mes sincère remerciement vont surtout à l’adroit de mon Père et de ma Mère pour tous les sacrifices qu’ils n’ont jamais manqués de consentir à mon égard.

Je ne manquerai jamais l’occasion de remercier tous les membres de ma famille pour leur inconditionnel soutient à tous les niveaux.

Je remercie également toutes les personnes que je n’ai pas peu cité ici, qui de près ou de loin m’accorde leur précieux soutient.

# PRESENTATION DE L’ECOLE

L

’accélération des progrès technologiques a placé l’informatique et les télécommunications au premier rang des enjeux industriels. La technologie en générale et l’informatique en particulier sont au cœur des entreprises et se positionnent comme le levier incontournable de l’activité des entreprises.

Acteur majeur dans le secteur de la formation et du développement des capacités technologiques et informatique, Le CEFIVE (Centre d’Etudes, de Formation en Informatique & Visio Enseignement), établissement privé né dans les enceintes de l’IPNETP, s’est donné pour mission de dispenser aux étudiants et travailleurs les formations suivantes dans les domaines des technologies :

**Brevet de Technicien Supérieur (BTS) :**

* BTS Finance Comptabilité et Gestion d’Entreprise (FCGE) ;
* BTS Gestion Commerciale (GC) ;
* BTS Informatique et Développeur d’Application (IDA) ;
* BTS Réseau informatique et télécommunication (RIT).

**Cycle Ingénieur (BAC +4 ou BAC +5) :**

**•  Ingénieur des Techniques en Informatiques (ITI BAC+4) :**

Options :

* Système d’Information (SI) : Système de Gestion des Bases de Données, Génie Logiciel et Applications Métiers ;
* Réseaux et Télécoms (RT) : Etude, Maintenance et Administration d’Infrastructures, des Systèmes et des Réseaux (avec Certification Cisco Systems).

**•  Ingénieur de Conception Informatique (ICI BAC+5) :**

Options :

* Etude des Systèmes d’Information, Data Science et Technologie du WEB ;
* Systèmes Avancées, Réseaux Informatiques – Telecom et BIG DATA (Avec Certification Cisco Systems).

**Master de Spécialité (BAC +6) :**

Options :

* Marketing Digital et Data Analytics ;
* Technologies du Web Avancées et Cyber Sécurité (Cisco Security) ;
* Management des Projets Informatiques / Telecom et Innovation ;
* Data Management et Big Analytics.

# SOMMAIRE

[DEDICACES](#_Toc68766640)

[DEDICACE I](#_Toc68766641)

[EMERCIEMENTS II](#_Toc68766642)

[PRESENTATION DE L’ECOLE III](#_Toc68766643)

[SOMMAIRE 0](#_Toc68766644)

[RESUME 1](#_Toc68766645)

[INTRODUCTION 2](#_Toc68766646)

[PARTIE 1 : CADRE DE REFERENCE 3](#_Toc68766647)

[1. PRESENTATION DE LA STRUCTURE D'ACCUEIL 4](#_Toc68766648)

[2. ETUDE DE L’EXISTANT 6](#_Toc68766649)

[3. Critiques, Problématique et Solutions 11](#_Toc68766650)

[4. Cahier des charges et choix de la solution 14](#_Toc68766651)

[PARTIE II : ETUDE DE LA SOLUTION 17](#_Toc68766652)

[1. Méthodes d'analyse 17](#_Toc68766653)

[2. Etudes conceptuelles 25](#_Toc68766654)

[3. Etudes techniques 43](#_Toc68766655)

[4. Planning de Réalisation **Erreur ! Signet non défini.**](#_Toc68766656)

[PARTIE III : REALISATION DE LA SOLUTION RETENUE 58](#_Toc68766657)

[1. Réalisation Technique **Erreur ! Signet non défini.**](#_Toc68766658)

[2. Mise en exploitation 63](#_Toc68766659)

[3. Tests **Erreur ! Signet non défini.**](#_Toc68766660)

[4. COUT DE LA SOLUTION 64](#_Toc68766661)

[CONCLUSION 64](#_Toc68766662)

[REFERENCES BIBLIOGRAPHIQUES 65](#_Toc68766663)

[WEBBOGRAPHIE 65](#_Toc68766664)

[TABLE DES MATIERES 66](#_Toc68766665)

[Table des matières 66](#_Toc68766666)

# RESUME

La gestion du personnel est une fonction très importante dans le fonctionnement d’une administration. La mise en œuvre de cette fonction soulève d’importants défis à relever. Ces défis concernent aussi bien l’efficacité et la célérité dans le traitement des dossiers que la sécurité des données sur le personnel. Ainsi, sans nécessairement être l’activité de production d’une structure, la gestion du personnel peut se révéler très couteuse en temps, en ressources matérielles, financières et humaines si l’on ne dispose pas de moyens adéquats. De ce fait, pour être menée à bien, il s’avère nécessaire de disposer d’outils permettant d’une part d’économiser les ressources et d’autres parts d’assurer la sécurité et la qualité des informations manipulées.

Cette problématique demeure d’actualité pour la Direction Générale des Marchés Publics dont le système de gestion du personnel reste essentiellement basé sur un archivage physique (papiers) des documents.

Dans le souci de résoudre ce problème, nous avons opté, après analyse et étude de plusieurs solutions, pour la mise en place d’une application web, permettant d’automatiser la gestion administrative du personnel.

Cette solution utilise le Système de Gestion de Base de Données (SGBD) Oracle pour le stockage des données. Elle fonctionne dans une architecture à trois niveaux (client léger, server d’application, server de base de données). Le server d’application implémente une architecture Modèle – Vue - Contrôleur (MVC) et est développée à l’aide de la spécification JEE du langage JAVA en sa version 8. Nous avons, en outre utilisé, le Framework Spring (Spring Boot) pour assurer l’inversion de contrôle et l’injection des dépendances. Par ailleurs, Les vues du système sont basées sur le moteur de Template Thymeleaf et utilisent les librairies JQuery et Bootstrap pour la gestion des mises en forme et l’interactivité.

# INTRODUCTION

La gestion administrative du personnel peut vite devenir lourde et difficile à tenir sans un système adapté, efficace et performant. A l’ère du numérique, l’[utilisation d’un système automatisé de gestion du](https://www.appvizer.fr/magazine/ressources-humaines/systeme-dinformation-rh-sirh/guides/utilisation) personnel est en effet devenu incontournable pour un service en charge du personnel. De fait, Dans ce climat de globalisation accéléré que nous connaissons aujourd'hui, l'administration fait face à des défis de plus en plus croissants. En effet, elle doit pouvoir s'adapter aux mutations technologiques pour satisfaire au mieux les besoins grandissants des usagers et répondre aux attentes de ses travailleurs. Ainsi, la gestion administrative du personnel qui constitue un maillon essentiel pour la survie d'une administration, a un rôle capital à jouer. En substance, elle doit répondre efficacement au besoin de célérité dans le traitement courant des dossiers et aider à la prise de décision en mettant, en temps réel, des statistiques fiables à la disposition du manager. C’est pour apporter une réponse à ce problème que nous avons choisi de travailler sur le thème automatisation de la gestion administrative du personnel dans le cadre de notre projet de fin de cycle.

Pour ce faire, notre document sera structuré en trois parties. La première partie abordera le cadre de référence, la seconde portera sur l’étude de la solution retenue et la troisième concernera la réalisation de la solution retenue.

# **PARTIE 1 : CADRE DE REFERENCE**

## PRESENTATION DE LA STRUCTURE D'ACCUEIL

Créée en 1968, la Direction Générale des Marchés Publics est une structure du Ministère d’Etat auprès du Premier Ministre chargé du Budget et du Portefeuille de l’Etat, chargée d’assurer le contrôle de la passation, de l’approbation et le suivi de l’exécution des marchés publics. A cet effet, elle veille à l’application du code des marchés publics dans le cadre des commandes de travaux, fournitures et prestations passées par les services de l’Etat et des personnes morales de droit public ou privé soumises aux procédures de marchés publics.

Aux termes des dispositions de l’arrêté n° 465/MPMB/DGBF/DGMP du 23 juin 2015 portant attributions, organisation et fonctionnement de la Direction Générale des Marchés Publics, La Direction Générale des Marchés Publics comprend six sous-directions, une cellule spécialisée, des services rattachés et des services extérieurs. Les sous-directions et la cellule spécialisée sont les suivantes :

* La sous-direction de de la réglementation et des régimes particuliers ;
* la sous-direction des procédures et opérations ;
* la sous-direction des opérations sur financement extérieur ;
* la sous-direction du système d’information et de la communication ;
* la sous-direction de la formation et des appuis techniques ;
* la sous-direction des études, du suivi et de l’évaluation des marchés publics ;
* la cellule de gestion des opérations de passation de marchés publics relatifs au Contrat de Désendettement et de Développement (C2D).

L’organigramme de la Direction Générale des Marchés Publics est présenté à la page suivante.

**Directeur des Marchés Publics**

**Sous-direction de la Réglementation et des Régimes Particuliers**

**Sous-direction des Procédures et Opérations**

**Sous-direction des Opérations sur Financements Extérieurs**

**Sous-direction des Systèmes d’information et de Communication**

**Sous-direction de la Formation et des Appuis Techniques**

**Sous-direction des études, du Suivi et de l’Evaluation des Marchés Publics**

**Cellule C2D de la DMP**

**Secteur éducation, formation et emploi**

**Secteur santé**

**Secteur Agriculture, développe-ment rural et biodiversité rural et biodiversité**

**Secteur développement urbain, Eau et assainisse -ment**

**Secteur infrastruc - tures de transport**

**Secteur justice et état de droit**

**Service de la règlementation et des analyses juridiques**

**Service des régimes particuliers**

**Service du suivi de la préparation des accords de financement**

**Service du suivi des acquisitions**

**Service du Suivi de la Gestion des Projets Cofinancés**

**Service de la communication et des publications**

**Service de l’informati -que**

**Service des Archives et de la Documenta- tion**

**Service de la gestion du SIGMAP**

**Service de préparation à la procédure des marchés publics**

**Service de l’approbation des marchés publics**

**Service de la conception et de l’organisa- tion des sessions de formation**

**Service de la prospection et du suivi -évaluation de la formation**

**Service des appuis techniques**

**Service des études**

**Service du suivi et de l’évaluation des marchés publics**

**Services rattachés**

Service Administratif et Financier

D

Carburant

Assistants DMP

**Services extérieurs :**

**Directions régionales**

Lagunes, Abidjan Nord

Lagunes, Abidjan sud et Sud-Comoé

Vallée de Bandama

Lacs

Sassandra, Marahoué et Woroba Est

Bas Sassandra et Gôh-Djiboua

Comoé Nord et Zanzan

Montagnes Woroba Ouest et Dinguélé

Savanes

## ETUDE DE L’EXISTANT

### Description générale de l’existant

Les activités de gestion du personnel au sein de la Direction Générale des Marchés Publics sont assurées par le Service des Moyens Généraux et du Personnel. Ce service directement rattaché au Directeur Général des Marchés Publics est chapeauté par un chef de service nommé par décision de nomination signé par le Directeur Général des Marchés Publics. Celui-ci est suppléé dans sa tâche par des chargés d’études.

Pour mener à bien sa mission, le service dispose d’un bureau équipé d’une armoire pour le stockage des dossiers, de matériels de bureau standard (imprimante, perforeuse, photocopieuse…) et d’ordinateurs de bureau à raison d’un ordinateur par personne. Ces ordinateurs sont équipés de logiciels de bureau Word et Excel de la suite Microsoft Office. C’est d’ailleurs principalement à l’aide de ces deux logiciels que le service assure la plupart de ses tâches. En l’occurrence, le Chef de service tient sur son ordinateur une liste du personnel dans un classeur Excel qu’il met à jour selon les mouvements d’arrivée et de départ des agents au sein de la DGMP. Ses collaborateurs disposent également de plusieurs fichiers Word disséminés sur leurs ordinateurs. Ces fichiers servent de Templates de base à la rédaction des différents courriers qu’ils traitent de façon courante.

Par ailleurs on note une présence massive et quasi indispensable du service courrier dans le fonctionnement du Service des Moyens Généraux et du personnel. Cette prépondérance se justifie par la nécessité de pouvoir acheminer les dossiers traités vers des cibles distantes au sein de la DGMP comme à l’extérieur.

### Description de l’existant par activités / Infrastructures

#### Le recrutement

De façon général, le Ministère de la fonction Publique se charge de recruter tous les fonctionnaires et agents de l'Etat de Côte d'Ivoire qu'il reparti dans différents Ministères. Une fois au sein des ministères, les Directions en charge des Ressources humaines desdits ministères procèdent aux affectations de ces fonctionnaires et agents dans les Directions Générales sous leur tutelle. Chaque Direction Général s'occupe ensuite d'affecter ses fonctionnaires dans les Directions d'administration Centrale sous sa tutelle.

Le processus de recrutement à la Direction Générale des Marchés Publics peut être scindé en trois étapes à savoir, l'identification et l'expression des besoins, l'affectation puis la mutation.

#### L'identification et l'expression des besoins

A la Direction Générale des Marchés Publics, l'identification et l'expression des besoins en personnel est réalisée par l'ensemble Directeurs, des Sous-Directeurs, des Directeurs Régionaux ainsi que des chefs de services, sous la supervision du Directeur Général des Marchés Publics.

Un besoin de recrutement peut provenir de la survenance d'un poste vacant suite à un départ définitif ou de longue durée d'un agent. Il peut également émaner de la nécessité de combler un besoin en personnel dans une compétence spécifique.

De façon concrète, les Directeurs, Sous Directeurs, Directeurs Régionaux et Chefs de Services déterminent en fonction de leurs attributions et de leurs activités, les profils à recruter pour atteindre les objectifs qui leur sont assignés. Sur cette base, ils formulent leurs besoins dans un document sous forme de tableau. Ces demandes sont centralisées par le Sous-Directeur des Moyens Généraux et du Personnel qui les compile afin de les transmettre au Directeur Général des Marchés Publics pour amendement et/ou validation.

Une fois validée, la demande ainsi formulée est acheminée à la Direction des Affaires Administratives et Financières (DAAF) du Ministère du Budget afin que celle-ci puisse en tenir compte lors des prochaines affectations.

L’activité d’identification et d’expression des besoins est réalisée une fois dans l’année.

#### La mise à disposition (par la DAAF du Ministère du budget)

La Direction Générale des Marchés Publics étant une Direction Générale sous la tutelle du Ministère en charge du Budget, l'affectation des agents à la DGMP est assurée par la DAAF dudit Ministère.

De fait, la DAAF du Ministère en charge du Budget centralise toutes les demandes en personnel issues des Directions Générales sous sa tutelle, y compris celles de la DGMP. C'est en fonction de ces demandes et aussi de l'offre en personnel qu'elle reçoit de la part du Ministère de la Fonction Publique qu'elle effectue les affectations des agents au sein de ses différentes Directions Générales.

#### L’affectation

La Direction Générale des Marchés Publics reçoit tous les agents mis à sa disposition par la DAAF du Ministère en charge du Budget. Elle se charge, ensuite, de les repartir dans ses directions, sous directions, directions régionales et services. La répartition se fait selon les besoins en personnel préalablement exprimés et les profils des agents faisant l'objet de recrutement.

Pour ce faire, l’agent mis à la disposition de la DGMP se présente au Service des Moyens Généraux et du Personnel muni de son **arrêté d'admission** au concours, de son **arrêté de nomination** **ou de promotion**, de sa **décision d'attente** et de sa **note de service de la DAAF**. Les agents venants d’un autre ministère devront joindre leur certificat de **première prise de service** aux documents cités ci-dessus. Sur cette base, un dossier comprenant ces documents est constitué à l'actif du fonctionnaire et stocké dans le bureau du Chef de Service des Moyens Généraux et du Personnel dans une armoire dédiée à cet effet. Un certificat de prise de service, précisant la Direction, la Sous-Direction, le Service et la fonction de l'agent est ensuite rédigée et introduite auprès du Directeur Général des Marchés Publics pour visa. Une fois visé par le Directeur Général des Marchés Publics, ce certificat est transmis à la DAAF au Ministère en charge du Budget. Le recrutement de l’agent à la DGMP sera enfin entériné par le certificat de prise de service signé par le Directeur des Affaires Administratives et Financière du Ministère en charge du Budget.

#### Le traitement des départs

**Cas 1 : Départ n’impliquant pas une sortie de la fonction public**

Ce type de départ est généralement introduit à la demande de l’agent. Il peut s’agir d’une volonté pour l’agent de changer de Direction Centrale, de Direction Générale ou de Ministère. Dans ce cas, l’agent adresse, au Directeur Général des Marché Publics, un courrier motivant sa volonté de départ. Il transmet ce courrier dans un premier temps à son supérieur hiérarchique direct pour accord. Une fois accordé par le supérieur hiérarchique direct, le courrier est transmis au chef du service des moyens généraux et du personnel. Celui-ci rédige un courrier réponse à la demande de l’agent. Dans ce courrier il marque l’accord ou le refus du Directeur Général des Marchés Publics. Cette réponse est ensuite transmise au Directeur Général des Marchés Publics pour signature. Après signature du Directeur Général des Marchés Publics, la réponse est enfin transmise à l’agent qui pourra s’en servir dans la suite du processus auprès de la DAAF du Ministère en charge du Budget.

**Cas 2 : Départ impliquant une sortie de la fonction public**

Ces procédures sont à l’initiative de l’agent mais peuvent également être prononcées d’office par le Ministre de la fonction publique. Ces procédures sont en outre sanctionnées par des actes administratifs délivrés par le ministère de la fonction publique que l’agent devra présenter au service des moyens généraux et du personnel de la DGMP aux fins de leur prise en compte. Ces procédures sont les suivantes :

* **Le détachement**
* **La mise en disponibilité**
* **La démission**
* **Le licenciement**
* **La révocation**
* **La retraite**
* **Le décès** (Il se constate)
  + - 1. **Le traitement des demandes d'actes**

Dans l’exercice de ses fonctions, le service des moyens généraux et du personnel assure le traitement des demandes de certificat de travail et d’attestation de présence au bénéfice de tout agent demandeur.

Ces demandes sont adressées par courrier au service des moyens généraux et du personnel qui réalise la rédaction des actes administratifs correspondant (attestation de travail ou certificat de présence). Après rédaction, ces actes sont transmis au Directeur Général des Marchés Publics pour signature. Une fois signés, lesdits actes sont transmis aux agents demandeurs pour servir et valoir ce que de droit.

#### Le traitement des demandes de congé

Conformément au statut général de la fonction publique, tout agent travaillant à la Direction Générale des Marchés Publics bénéficie d’un droit aux congés.

Les demandes de congés sont toutes à l’initiative de l’agent. Pour en bénéficier, celui-ci adresse, au Directeur des Marché Publics, un courrier précisant la date de début et de fin du congé, le type de congé et éventuellement d’autres informations complémentaires. Il transmet ce courrier dans un premier temps à son supérieur hiérarchique direct pour accord. Une fois accordé par le supérieur hiérarchique direct, le courrier est transmis au chef du service des moyens généraux et du personnel qui s’assure de la conformité de la demande avant de la transmettre au Directeur Général des Marchés Publics pour signature. Après signature du Directeur Général des Marchés Publics, le courrier est transmis à la Direction des Ressources Humaines de la DGBF. Ensuite, ladite Direction des Ressources Humaines produit un avis de cessation de service pour congé sur la période indiquée dans la demande. Cet avis est enfin transmis à l’agent pour servir et valoir ce que de droit.

On note par ailleurs que Le statut général de la fonction public prévoit à cet effet trois types de congés. On distingue ainsi le **congé annuel,** le **congé maladie** et le **congé de maternité.**

* **Le congé annuel**

Conformément aux dispositions de l’article 64 du statut général de la fonction publique, tout agent travaillant à DGMP à droit à un congé annuel de 30 jours. La prise de congé annuel est régie par les règles de gestion suivantes :

* L’accord ou non de la prise de congé annuel par le supérieur hiérarchique direct se fait en prenant en compte les nécessités de service.
* Le congé de l’année n doit être pris en année n+1.
* L’agent à la possibilité d’échelonner son congé annuel en plusieurs prises, pourvu que la somme cumulée des jours de congé ne dépasse pas les 30 jours réglementaires.
* Si l’agent ne parvient pas à prendre son congé de l’année n en année n+1, il a la possibilité de le rattraper en année n+2. Dans ce cas, il pourra alors cumuler, en plusieurs prises, 60 jours de congés en année n+2.
* Dans tous les cas, il n’est pas permis à un agent de prendre en une seule fois, plus de 30 jours de congés d’affilés.
* **Le congé maladie**

Tout agent travaillant à la Direction Générale des Marchés Publics à droit à des congés de maladie d'une durée maximum de six (6) mois, pendant une période de douze (12) mois consécutifs.

Si, après la période de six (6) mois, l'état de santé du fonctionnaire exige la poursuite de soins, sur proposition du Conseil de Santé, il est mis au congé maladie de longue durée et perçoit l'intégralité de sa rémunération pendant six (6) mois.

Si au terme de trente-six (36) mois y compris les six (6) premiers mois de congé maladie, l'état de santé du fonctionnaire ne lui permet toujours pas de reprendre son service, il est déclaré invalide sur avis du Conseil de Santé et admis d'office à la retraite.

* **Le congé de maternité**

Comme partout ailleurs dans l’administration, toute femme en grossesse travaillant à la Direction des Marché Publique bénéficie au terme de sa grossesse d’un congé de maternité et des périodes de repos pour allaitement sur une période de 3 mois. Ce congé lui est accordé à sa demande, appuyée d'un certificat médical délivré par un médecin de l'Administration. Si, à l'expiration de ce congé, l'intéressée n'est pas en état de reprendre son service, elle est placée en congé de maladie après avis du Conseil de Santé.

#### Le traitement des demandes d’autorisation d'absences

Outre les congés, tout agent à la DGMP a droit à des autorisations spéciales d'absence et des permissions spéciales pour événements familiaux d’une durée maximum de trois (03) jours.

A l’instar des demandes de congés, les demandes d’autorisation d’absence sont à l’initiative de l’agent. Pour en bénéficier, celui-ci adresse, au Directeur des Marché Publics, un courrier précisant la date de début et de fin de l’autorisation d’absence, le motif de l’absence et éventuellement d’autres informations complémentaires. Il transmet ce courrier dans un premier temps à son supérieur hiérarchique direct pour accord. Une fois accordé par le supérieur hiérarchique direct, le courrier est transmis au chef du service des moyens généraux et du personnel qui s’assure de la conformité de la demande avant de la transmettre au Directeur Général des Marchés Publics pour signature. Après signature du Directeur Général des Marchés Publics, le courrier est enfin transmis à l’agent pour servir et valoir ce que de droit.

#### Le traitement des cas de promotion

Selon l’article 57 du statut général de la fonction publique, la promotion est le passage du fonctionnaire de son grade à un grade immédiatement supérieur. Elle est faite par voie de concours internes, sauf dérogations prévues par décret.

La promotion est donc une procédure dont le traitement relève entièrement du ministère de la fonction publique. Toutefois, l’aboutissement d’un processus de promotion est sanctionné par un arrêté de nomination ou une décision d’attente signé par le Ministre de la fonction publique et délivré à l’agent concerné. Ainsi, tout agent de la DGMP ayant fait l’objet d’une promotion est tenu d’en informer le Chef du Service des Moyen Généraux et du personnel avec son arrêté de nomination ou sa décision d’attente à l’appui, afin que cette information soit mise à mise jour.

#### Le traitement des cas de nomination

Le directeur des Marchés Publics est nommé par décret en Conseil des ministres, les sous-directeurs et Directeurs Régionaux sont nommés par arrêté et les chefs de service sont nommés par décision signée par le Directeur Général du Budget et des Finances.

Toutes les fois qu’il se produit une nomination au sein de la DGMP, tout le personnel en est informé par une note de service à laquelle est joint l’acte administratif (décret, arrêté ou décision de nomination) qui entérine cette nomination. Ainsi, le service des moyens Généraux et du Personnel procède à la mise à jour de ses informations

## Critiques, Problématique et Solutions



### Critique de l'existant

Au regard de ce qui précède, le système de gestion du personnel de la Direction Générale des Marchés Publics est essentiellement basé sur un archivage physique (papiers) des documents, une utilisation accrue des logiciels Word et Excel de la suite Microsoft Office et une présence quasi indispensable du service courrier pour la transmission de l’information vers des cibles divers.

Ce mode de fonctionnement a l’avantage de marcher car c’est ainsi qu’à toujours fonctionné le service depuis de longues années. Il présente également le mérite de toujours laisser des traces écrites de toutes les actions menées dans le cadre de la gestion du personnel. Notons en outre que l’utilisation de courrier modèle dans la rédaction des dossiers courants permet de faciliter le traitement de certains dossiers tout en évitant au chargé d’études de réinventer la roue à chaque fois qu’il a besoin de traiter un dossier. En outre, l’utilisation de classeur Excel pour la liste du personnel présente l’avantage de faciliter la consultation et la mise à jour des informations sur le personnel.

En revanche, s’il est vrai que le mode de gestion actuel du personnel présente quelques avantages, il n’en demeure pas moins que celui-ci présente d’importants handicapes qu’il s’avère nécessaire de corriger. Ces handicapes sont principalement les suivants :

#### L’inefficience dans le traitement des dossiers courants

En effet, la rédaction d’un courrier en s’appuyant sur un modèle prédéfini peut entrainer des erreurs d’accord, d’omission et même de mise à jour. Ces erreurs apparemment anodines suffisent à faire retourner un courrier, déjà transmis, pour prise en compte de corrections. Ce qui contribuerait à rallonger le temps de traitement du dossier. De plus, la transmission physique des courriers d’un service à un autre est souvent source de lourdeurs, de perte de temps et de mobilisation de ressources humaines qu’on pourrait affecter à d’autres tâches.

Outre cela, on note une difficulté relative pour les services distants comme les Directions Régionale, à réaliser certaines actions. Il est parfois nécessaire pour certains agents de ces services distants, d’effectuer de longs déplacements afin de récupérer des actes administratifs.

#### Le manque de fiabilité des données issues de la liste du personnel

L’utilisation d’un simple classeur Excel pour la gestion de la liste du personnel ne propose aucun mécanisme de contrôle des données saisies dans la liste. Cela signifie que l’utilisateur pourrait, par inadvertance, saisir des données erronées (Exemple : **lieu de naissance** en lieu et place de **date de naissance**). Il pourrait également faire des mises à jour sur les informations d’un agent en lieu et place d’un autre. A terme, la multiplication de ce genre d’erreurs finirait par rendre la base de données quasi inutilisable. Ou au mieux l’utilisation d’une telle base de données nécessiterait d’importants efforts d’apurement.

#### Le manque de sécurité du système de fichiers (Word et Excel) servant à la gestion du personnel

Le classeur Excel hébergeant les données sur le personnel ainsi que les fichiers Word servant de base à la rédaction des courriers sont disséminés sur les ordinateurs du service dans des dossiers non protégés. Cet état de fait expose ces fichiers à de graves risques de sécurités. En effet, des individus malveillants pourraient, à des moments d’inattention, accéder à ces fichiers et les utiliser à leur guise. Ils pourraient par exemple y récupérer des informations sensibles, ajouter des informations erronées, procéder à des modifications indues ou tout simplement supprimer les fichiers en questions.

#### La difficulté à produire des statistiques fiables et en temps réel

Le système actuel de gestion du personnel, permet de fournir des statistiques basiques comme le nombre d’agents, le nombre d’hommes, le nombre de femmes etc. Mais la question peut très vite se compliquer si l’on a besoin de faire croisements ou d’obtenir des statistique plus avancées (Exemple : liste du personnel allant à la retraite sur les cinq prochaines années, déterminer le nombre de jours de congés restant par agent, l’année en cours). De telles questions obligeraient le Service des Moyens Généraux et du personnel à faire des fouilles dans des piles de documents physiques afin d’obtenir les informations de chaque agent et les traiter.

### Problématique

La description et l’étude critique du système actuel de gestion du personnel de la DGMP laisse apparaître que ledit système n’est essentiellement pas automatisé (informatisé). Ce mode de fonctionnement est à la base d’importants problèmes auxquels il est impérieux d’apporter une solution. Ces problèmes sont précisément liés à l’efficience dans le traitement des dossiers courants, à la fiabilité des données issues du système, à la sécurité du système et la capacité pour le système à produire des statistiques en temps réel. La conjonction de ces problèmes impacte négativement la performance du système. C’est pourquoi, pour le…

Apporter une solution cette problématique reviendrait donc à répondre aux questions suivantes :

1. Comment rendre efficient le système de gestion du personnel de la DGMP ?
2. Comment rendre fiable les données issues du système de gestion du personnel de la DGMP ?
3. Comment sécuriser les données sur le personnel ?
4. Comment produire des statistiques fiables en temps réel sur le personnel ?

Pour répondre à ces questions, nous étudierons trois (3) propositions de solutions.

### Solutions

Trois principales solutions se présentent à nous pour répondre au problème posé.

#### Acheter un Logiciel ERP

L’ERP (Enterprise Resource Planning ou Programme de Gestion Intégrée en français)

est un [progiciel](https://fr.wikipedia.org/wiki/Progiciel) qui permet « de gérer l'ensemble des processus d'une entreprise en intégrant l'ensemble de ses fonctions, dont la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais aussi la vente, la distribution, l'approvisionnement et le commerce électronique ». Les ERP présentent de nombreux avantages parmi lesquels nous pouvons citer

* L’Intégration des [processus](https://fr.wikipedia.org/wiki/Processus_d%27affaires) de gestion, largement exploitée par les entreprises pour optimiser le suivi financier et le contrôle de gestion ;
* La Cohérence et homogénéité des informations (un seul fichier articles, un seul fichier clients, etc.) ;
* L’Intégrité et l’unicité du [Système d'information](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27information) ;
* Le Partage du même système d’information facilitant la communication interne et externe ;

Les ERP ne sont cependant pas exempts d'inconvénients :

* La mise en œuvre peut s'avérer complexe si le périmètre fonctionnel est mal déterminé ou trop mouvant ou le projet mal piloté ;
* coût élevé et pouvant rapidement augmenter en fonction de l'industrie et de la complexité du projet.
* Périmètre fonctionnel non adapté au besoin réel de l'organisation : le progiciel peut être surdimensionné et donc sous-utilisé s'il est plus large que les besoins effectifs de l'organisation, ou au contraire être sous-dimensionné s'il n'est pas capable de couvrir l'ensemble des besoins avérés[12](https://fr.wikipedia.org/wiki/Progiciel_de_gestion_int%C3%A9gr%C3%A9#cite_note-Les_ERP,_26_mars_2004,_M.Volle-12) ;
* captivité vis-à-vis de l'éditeur : le progiciel choisi ne peut pas toujours s'adapter à l'organisation qui s'équipe. Le choix d'une solution devient fortement structurant pour l'entreprise, et, au-delà d'un simple paramétrage, ce sera à l'organisation de s'adapter au progiciel (et non l'inverse). Par ailleurs, l'outil risque d'être extrêmement lourd et couteux à gérer notamment s'il nécessite une maintenance continuelle et s'il rend la moindre adaptation (voire son abandon pour un autre produit) très difficile.

#### S’abonner à un SaaS (Software as a Service ou Logiciel en tant que Service)

Le Software as a Service (SaaS), ou Logiciel en tant que Service en Français, est un modèle de distribution de logiciel au sein duquel un fournisseur tiers héberge les applications et les rend disponibles pour ses clients par l’intermédiaire d’internet.

Grâce à un logiciel SaaS, les entreprises n’ont plus besoin d’installer et de lancer des applications sur leurs propres ordinateurs ou sur leurs Data Centers. Le coût d’acquisition de matériel est ainsi éliminé, au même titre que les coûts d’approvisionnement et de maintenance, de licence de logiciel, d’installation et de support.

Cependant, le SaaS présente également des inconvénients potentiels. Les entreprises doivent compter sur des vendeurs externes pour fournir des logiciels, maintenir ces logiciels, proposer une facturation détaillée et garantir un environnement sécurisé pour les données de l’entreprise. En cas d’interruption du service, de changements de l’offre, de faille de sécurité ou tout problème, les conséquences pour le client peuvent être dramatiques.

#### Développement d’une application

Cette solution consiste à développer nous-même, une application permettant de répondre au problème posé. Les principaux avantages d’une telle solution sont les suivants :

* Couts de production réduits du fait de l’exploitation de ressources matérielles (Machine Serveur, serveur de base de données, infrastructure réseau,…) et humaines disponibles ;
* Solution typiquement adaptée au besoin de la DGMP ;
* Facilité de prise en compte de la communication avec des applications existantes (SIGOMAP, SIGMAP, BDPR) ;
* Gestion en interne des données sur le personnel ;
* Gestion en interne de la maintenance et de l’évolutivité du système ;
* Contribue à maintenir les aptitudes au développement des ingénieurs de la DGMP ;

Le principal inconvénient de cette solution reste le temps de livraison qui pourrait s’avérer long. On note également la mobilisation de ressources humaines et matérielles qui pourraient être utilisées à d’autres fins.

## Cahier des charges et choix de la solution



### Cahier des charges

#### Le thème

THEME : Automatisation de certaines fonctions de la gestion administrative du personnel : cas de la Direction Générale des Marchés Publics (Automatisation de la gestion administrative du personnel : cas de la DGMP)

#### Définition des concepts liés au thème

##### Automatisation

L'automatisation est une technique ou un ensemble de techniques ayant pour but de réduire ou de rendre inutile l'intervention d'opérateurs humains dans un processus où cette intervention était coutumière. Dans le cadre de ce projet, l'automatisation consistera à limiter l'intervention humaine à l'usage d'une application informatique qui exposera sur différentes interfaces graphiques les principales fonctionnalités de la gestion administrative du personnel.

##### Certaines fonctions

Les fonctions de la gestion administrative prises en compte par le projet sont les suivantes :

* la gestion des recrutements ;
* la gestion affectations
* la gestion des nominations
* la gestion des promotions
* la gestion des Départs ;
* la gestion des congés ;
* la gestion desabsences ;
* la gestion des demandes d'actes ;
* Tableau de bord statistique ;

**NB :**

* par gestion, il faut entendre : l'enregistrement, la visualisation, la modification et la suppression des informations selon des règles de gestion propre à la DGMP

Ce projet ne prend pas en compte la gestion de la paie qui est une fonction déjà assurée par les services de la Direction de la Solde.

##### Gestion administrative du personnel

La gestion administrative du personnel regroupe l'ensemble des activités relatives au traitement des dossiers et documents administratifs du personnel, depuis le recrutement jusqu'au départ du travailleur quel qu'en soit le motif.

##### Le personnel

Le personnel désigne l'ensemble des personnes travaillant à la DGMP.

##### Résultats

Les résultats attendus de ce projet sont les suivants :

* Mise à disposition d'une base de données fiable sur le personnel;
* mise à la disposition de la Direction Générale des Marchés Publics d'un applicatif de gestion du personnel intégrant les fonctionnalités suivantes :
* Tableau de bord statistique ;
* la gestion des recrutements ;
* la gestion des Départs;
* la gestion des congés ;
* la gestion des demandes d'actes;
* la gestion desabsences;
* la gestion des promotions;
* la gestion des nominations;

### Choix de la solution

Au regard des solutions précédemment décrites, nous pouvons dresser le tableau comparatif suivant :



Le tableau ci-dessus présente les scores attribués aux trois solutions proposées (**ERP, SAAS, Applicatif interne**) selon 6 critères pondérés en fonction de leurs importances relatives. La détermination des différents critères et de leur poids ou coefficients de pondération a été faite sur la base d’une enquête réalisée auprès des acteurs métiers (agents du service en charge du personnel), des développeurs de la DGMP et des futurs utilisateurs du système. Le questionnaire et les résultats de cette enquête sont disponibles en **annexe XXX.** Ces critères sont les suivants : le **coût**, la **durée de livraison**, la **facilité d’intégration** de la solution, la **détention exclusive des données** par la DGMP, la **consommation de ressources internes**, la **haute disponibilité**. Pour chacun de ces critères, un score est attribué à chaque solution sur une échelle allant de 1 à 5. Le score global de chaque solution est obtenu en faisant la somme des scores obtenus sur chaque critère pondéré par le poids du critère.

Ainsi, sur la base de ce principe, la solution « **applicatif interne** » apparait la plus opportune avec un score global de **390** contre 290 pour l’ERP et 250 pour le SAAS. De ce fait, pour répondre à la problématique posée, nous avons choisi de développer en interne, une application web qui exposera, à travers différentes interfaces graphiques, les principales fonctions de la gestion administrative du personnel. Cette application s’appuiera sur une base de données qui permettra de stocker et de sécuriser les données sur personnel. Cette application sera doté d’un module « Tableau de bord » qui présentera les statistiques relatives au personnel.

# PARTIE II : ETUDE DE LA SOLUTION

1. Méthodes d'analyse

Une méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client. Pour ce faire, on part d'un énoncé informel (le besoin tel qu'il est exprimé par le client), ainsi que de l'analyse de l'existant éventuel c'est-à-dire la manière dont les processus à traiter par le système se déroulent actuellement chez le client.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité, d'extensibilité, etc.

La phase de conception permet de décrire de manière plus précise, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

L'on distingue plusieurs types de méthode d'analyse et de conception. Dans ce rapport, nous donnerons un aperçu illustré d'exemples des principaux types.

* 1. Méthodes d'analyse systémique
     1. Définitions et caractéristiques

L'analyse systémique est une méthode d'analyse qui envisage les éléments d'une structure complexe, non pas isolément mais globalement, en tant que parties intégrante d'un ensemble dont les différents composants sont dans une relation de dépendance réciproque. L'analyse systémique « s’appuie sur une approche globale des problèmes ou des systèmes que l’on étudie et se concentre sur le jeu des interactions entre leurs éléments. Elle s'oppose au précepte cartésien de réductionnisme qui demande une décomposition en autant de parcelles qu’il est possible de le faire. Les objets d’études ne sont pas en effet des tous à diviser, mais des parties d’un plus grand tout. L’approche systémique a précisément pour rôle de traiter le global de façon rationnelle et rigoureuse. Dans cette approche, le système est perçu comme une boîte noire pouvant interagir à des événements extérieurs ou émettre des informations à destination de l'extérieur.

* + 1. Exemples de méthodes systémiques

**MERISE** La méthode Merise, souvent définie comme une Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise est née à la fin des années 1970 en France, avec pour objectif de définir une démarche de conception de Système d'Informations. Son principe de base repose sur la séparation des données et des traitements. C'est une démarche qui s'inscrit dans trois dimensions que sont : **la démarche ou cycle de vie**, **le raisonnement ou cycle d'abstraction**, **la maîtrise ou cycle de décision**.

* **Le cycle de vie** traduit le caractère « vivant » du système d'information, présentant une phase de conception, une phase de réalisation et une phase de maintenance.
* **Le raisonnement** s'effectue sur quatre niveaux d'abstraction permettant de définir une hiérarchisation des préoccupations. Ces niveaux d'abstraction sont :

• le niveau conceptuel;

• le niveau organisationnel;

• le niveau logique;

• le niveau physique.

Les deux premiers niveaux sont adaptés à la conception du système d'information organisationnel c'est-à-dire le système d'information tel qu'il fonctionne avant la réalisation du projet. Les deux derniers à la conception du système d'information informatisé. A chacun de ces niveaux sont associés un modèle de données et un modèle des traitements comme le montre la figure suivante.



À chaque niveau d'abstraction, pour chaque volet (données, traitements), le système d'information est représenté par un modèle. Chaque modèle est exprimé dans un formalisme utilisant des concepts adaptés.

Au niveau conceptuel, le modèle conceptuel des données (MCD) formalise la signification des informations sur lesquelles repose le système d'information, sans contrainte technique ni économique. Le modèle conceptuel des traitements (MCT) formalise l'activité du domaine abordé, sans préciser les ressources ni leur organisation.

Au niveau organisationnel, le modèle organisationnel des traitements (MOT) décrit le fonctionnement du domaine en précisant les ressources humaines et matérielles mobilisées, ainsi que l'organisation de ces ressources dans le temps et dans l'espace. Le modèle organisationnel des données (MOD) précise quelles sont parmi les données définies au niveau conceptuel (MCD) celles qui sont prises en compte par le futur système informatisé, où ces données sont localisées (répartition par site organisationnel), leur confidentialité pour chaque intervenant de l'entreprise.

Au niveau logique, le modèle logique de données (MLD) fournit une description des données tenant compte des moyens informatiques de mémorisation et de leurs conditions d'utilisation par les traitements. Le modèle logique de traitements (MLT) décrit comment les tâches informatisées définies dans les MOT précédents sont conçues en termes de logiciel.

Au niveau physique, le modèle physique de données (MPD) est une description de la ou des bases de données ou de l'ensemble des fichiers, exprimée dans la syntaxe du système de gestion de bases de données (SGBD) ou système de gestion de fichiers (SGF) adoptés. Enfin, le modèle physique des traitements (MPT) précise, pour la réalisation, les spécifications techniques des différents modules définis au niveau du MLT. Ces modules pourront être réalisés soit en langages de quatrième génération, soit de façon plus traditionnelle en langage de troisième génération (Cobol, C…).

* **La maîtrise ou cycle de décision** comprend une succession de choix permettant, d'une part, de contrôler la durée globale de la conception-réalisation, d'autre part, de définir un système en harmonie avec les objectifs généraux de l'entreprise. Notons qu'elle comprend également l'ensemble des décisions d'arbitrage relatives aux coûts, délai et niveau de gamme associés au projet.

**OSSAD** - **Office Support System Analysis and Design** - est le nom d’une méthode de modélisation graphique d’organisation d’entreprise créée en 1989. Mise en place dans le cadre d’un programme européen (ESPRIT), son objectif était de promouvoir en Europe l’organisation des processus tertiaires, transverses, dits aussi « administratifs » et l’utilisation des technologies de l’information. Il s’agit d’un ensemble de principes, d’outils conceptuels et graphiques, un guide de comportement qui structure et facilite les projets de changement dans l’entreprise et répond aux problèmes organisationnels. C’est avant tout un outil de communication pour rendre possible la participation de tous les acteurs internes, ainsi que celle de l’environnement de l’entreprise, à la compréhension et à la définition de leur travail, de leurs activités et de leurs processus. Il existe une réelle volonté d’analyser et de maîtriser la mise en œuvre de ces systèmes d’information et non plus de la subir. C’est autant une approche globale de conduite de changement organisationnel et d’informatisation qu’une méthode d’analyse et de conception de processus. Elle est utile à la conduite de projet et à la modélisation graphique des processus, notamment pour faciliter le dialogue entre maîtrise d’ouvrage (métiers) et maîtrise d’œuvre (informatique). Cette problématique est toujours d’actualité.

**Les principes de la méthode OSSAD**

Les principes qui guident l’action lors d’un projet OSSAD dans un organisme sont : l’adaptabilité à l’environnement du projet; la participation de tous les intéressés à l’analyse et à la conception des systèmes bureautiques comme principe d’éthique et d’efficacité ; le pragmatisme pour coller à la situation de l’organisme et détecter les opportunités de changement ; l’expérimentation permanente, garante que les solutions imaginées soient faisables et répondent aux attentes réelles du client ; l’itérativité des modèles proposés comme reconnaissance du droit à l’imperfection et à l’apprentissage continuel ; l’agrégation/décomposition qui permet toujours de situer le problème au niveau du détail pertinent par rapport à l’interlocuteur. Les trois niveaux et modèles de la méthode OSSAD propose trois niveaux fondamentaux selon une démarche Top Down d’analyse : le niveau stratégique (ce que veut faire l’entreprise), le niveau organisationnel (comment on déploie la stratégie et qui fait quoi), le niveau des modes opératoires. ƒ le modèle abstrait, de niveau stratégique s’intéresse aux objectifs de l’organisation, indépendamment des moyens et des ressources mis en œuvre pour les réaliser. Il offre plusieurs niveaux de visualisation en permettant de décomposer un processus en sous-processus et en activités par un effet de zoom. Il a donc un positionnement stratégique et permet de représenter à un niveau global les activités majeures de l’organisme, leur environnement et les flux d’information (données d’entrée et de sortie) qui circulent. Ce niveau est d’autant plus intéressant qu’il permet de stabiliser les objectifs fondamentaux, les missions de l’organisme quand l’environnement, les technologies, les métiers évoluent rapidement. Il fournit ainsi une base solide pour les personnes en charge de gérer ces évolutions et de promouvoir l’amélioration des systèmes. ƒ le modèle descriptif complète le premier et décrit les moyens humains et les ressources technologiques de l’organisation. Il permet une représentation transversale, interservices, des activités de l’entreprise. Il s’élabore en termes de procédures d’organisation et décrit les éléments fondamentaux du système d’organisation : opérations, rôle, outils et ressources nécessaires à leur accomplissement. ƒ le modèle prescriptif termine la description des activités au niveau des métiers de l’entreprise. Il développe comme d’autres approches (modes opératoires textuels, pictogrammes, schémas, photos…) le savoir-faire associé aux compétences des collaborateurs de l’organisme.

Le schéma suivant illustre pour chacun des trois modèles leurs éléments descriptifs ainsi que leur représentation.

En résumé, chacun des trois niveaux permet de répondre aux questions suivantes : modèle abstrait : Pourquoi et vers quoi ? ƒ modèle descriptif : Qui fait quoi ? Modèle prescriptif : Comment ?

* 1. Méthodes d'analyses cartésiennes ou fonctionnelles

1. Définitions et caractéristiques

Le système étudié est abordé par les fonctions qu'il doit assurer plutôt que par les données qu'il doit gérer. Il est surtout abordé par la décomposition systématique du domaine étudié en sous domaines, eux-mêmes décomposés en sous-domaines jusqu'à un niveau considéré élémentaire.

* + 1. Exemple de méthode cartésienne

**SADT** (**Structured-Analysis-Design-Technique**)

L'acronyme S.A.D.T signifie Structured Analysis and Design Technic. Cette méthode a été mise au point par la société Softech aux Etats Unis. La méthode SADT est une méthode d'analyse par niveaux successifs d'approche descriptive d'un ensemble quel qu'il soit. On peut appliquer le SADT à la gestion d'une entreprise tout comme à un système automatisé.

L'objectif de cette étude doit mener les intervenants (ingénieurs, techniciens, opérateurs) à un tout qui soit cohérent et homogène avec le système à étudier. Dans n'importe quel système automatisé, circulent un certain nombre de flux de données. Les flux les plus caractéristiques sont : les flux de pièces : flux qui caractérisent la valeur ajoutée à un produit. Les flux d'informations, Les flux énergétiques, les flux divers (Rebuts, déchets…). L'analyse SADT va permettre d'organiser ces flux de données pour donner une vision globale du système puis permettre de préciser de plus en plus finement le rôle de chacun des éléments du système à travers une analyse des niveaux successifs. La finesse de cette description dépendra directement des besoins des utilisateurs.

* 1. Méthodes d'analyse objet

1. Définitions et caractéristiques

Les méthodes objets ont été créées sous l'influence des langages de programmation basés sur les objets comme C++ et des systèmes de gestion de bases de données objet (SGBDO). Nées dans le début des années 80, ces méthodes ont proliféré dans les années 90. La plupart de ces méthodes aborde l’étude d’un problème suivant trois aspects :

* **Aspect statique** : identifie les propriétés des objets et leur liaison avec les autres objets ;
* **Aspect dynamique** définit le cycle de vie des objets : comportement des objets, les différents états par lesquels ils passent et les événements déclenchant ces changements d’états ;
* **Aspect fonction** : précise les fonctions réalisées par les objets par l’intermédiaire des méthodes.
  + 1. Exemples de méthodes objets

**OMT** (en [anglais](https://fr.wikipedia.org/wiki/Anglais) « **Object Modeling Technique** », c'est-à-dire « technique de modélisation objet ») a été inventée dans le centre de recherche et de développement de Général Electric à la fin des années 80. C’est une technique de modélisation destinée à la conception et la modélisation pour la [programmation orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet). Elle a été conçue en 1991 par [James Rumbaugh](https://fr.wikipedia.org/wiki/James_Rumbaugh), Michael Blaha, William Lorensen, Frederick Eddy et William Premerlani.

La notation des diagrammes de classe d'OMT était très populaire, et a ainsi été utilisée dans des ouvrages de référence publiés avant UML, comme « [Design Patterns](https://fr.wikipedia.org/wiki/Design_Patterns) ». Les concepts de base y étaient la [classe](https://fr.wikipedia.org/wiki/Classe_(informatique)), l'[instance](https://fr.wikipedia.org/wiki/Instance_(programmation)) et l'[objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet) ainsi que l'[attribut](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet#L'objet_(attribut_et_m%C3%A9thodes)) et la [méthode](https://fr.wikipedia.org/wiki/M%C3%A9thode_(informatique)).

La méthode Booch est une méthode de développement de logiciels pour la [programmation orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet). Elle a été conçue par [Grady Booch](https://fr.wikipedia.org/wiki/Grady_Booch) qui l'a publiée en 1992 puis révisée en 1994. Elle se compose d'un [langage de modélisation](https://fr.wikipedia.org/wiki/Langage_de_mod%C3%A9lisation) graphique, d'un processus itératif de développement, et d'un ensemble de pratiques recommandées.

OMT propose, pour sa mise en œuvre, **une démarche d'analyse et de conception** ainsi que **des outils de modélisation.**

* La démarche d'analyse et de conception d'OMT. Comme la plupart des méthodes orientées objet, OMT propose une démarche en trois phases.

**L'étape d'**[**analyse**](http://d.martg.pagesperso-orange.fr/laphase.htm)**:** elle concerne l'approche conceptuelle. Elle permet d'élaborer les trois modèles conceptuels (statique, dynamique et fonctionnel). Cette étape est une description de ce que le système doit faire et non du "comment le faire".

**L'étape de**[**conception**](http://d.martg.pagesperso-orange.fr/laphase1.htm)**:** elle permet de présenter l'architecture d'ensemble du système, par intégration des contraintes et des soucis de performance. La conception technique des objets donne une spécification détaillée de l'implémentation des objets, indépendante d'un environnement donné, mais compatible avec une technologie choisie. C'est le niveau logique, il définit les structures de données et les algorithmes d'implémentation des classes.

**L'implémentation du Système :** Il s'agit de l'écriture du code, en tant qu'extension du processus de conception .Elle varie en fonction des langages cibles et des systèmes de gestion de bases de données choisis, suivant qu'ils sont ou non orientés objet.

* OMT fournit trois modèles principaux pour décrire les aspects **statique**, **dynamique** et **fonctionnel** d'un système d'information.

[**Le modèle objet**](http://d.martg.pagesperso-orange.fr/le.htm) : il décrit la structure statique des objets, classes et associations. C'est une extension du modèle Entité-Relation vers les concepts d'objet (classe, agrégation, généralisation, héritage).

[**Le modèle dynamique**](http://d.martg.pagesperso-orange.fr/le1.htm)**:** il décrit les transformations d'un système dans le temps. Il est fondé sur des diagrammes d'états/transitions avec spécification des événements qui déclenchent les transitions, des opérations de transformation, des attributs concernés par ces transformations. Le modèle dynamique est employé pour spécifier les aspects "contrôle" d'un système.

[**Le modèle fonctionnel**](http://d.martg.pagesperso-orange.fr/le2.htm) : il décrit la transformation des valeurs des données dans le système. Il est fondé sur des diagrammes de flux de données classiques (DFD). Ces diagrammes sont utilisés pour spécifier des fonctions générales qui sont opérées sur les objets.

**OOD** : Grady Booch a établi dans plusieurs de ses articles et son ouvrage, les principales étapes conduisant à la décomposition d'un programme ainsi que des outils de modélisation selon la méthode de conception orientée-objets. Ces étapes ont été reprises dans leurs grandes lignes par toutes les méthodologies parues depuis.

* **Etapes de décomposition d'un programme**

**Identifier les objets et les classes**

Cette première étape vise à identifier les objets du monde réel que l'on voudra réaliser. Pour cela, on doit identifier les propriétés caractéristiques de l'objet. Cette étape est bien entendu celle qui demande le plus de talent et d'expérience personnelle. Un moyen relativement informel pour identifier les objets consiste à faire une description littéraire (en français) du problème. On pourra déduire les bons candidats des noms utilisés dans cette description, et leurs propriétés des adjectifs et autres qualifiants.

**Identifier la sémantique des objets et des classes**

On cherchera ensuite à identifier les actions que l'objet subit et provoque. Les verbes utilisés dans la description informelle de l'étape précédente fournissent de bons indices pour l'identification des opérations. C'est également à cette étape que l'on pourra définir les conditions d'ordonnancement temporel des opérations, si nécessaire.

**Identifier les relations entre les objets et les classes**

L'objet étant maintenant identifié par ses caractéristiques et ses opérations, on définira ses relations avec les autres objets. On établira quels objets le "voient" et quels objets "sont vus" par lui. Autrement dit, on insérera alors l'objet dans la topologie du projet. On définit également l'interface précise de l'objet avec le monde extérieur. Cette interface spécifie exactement quelles fonctionnalités seront accessibles et sous quelle forme.

**Implémenter les classes et les objets**

La dernière étape consiste, bien entendu à implémenter les objets en écrivant le code correspondant aux spécifications dans un langage de programmation. Lors de cette étape, on identifiera de nouveaux objets de plus bas niveau d'abstraction, ce qui provoquera l'itération de la méthode.

* **Outils de modélisation**

La méthode de G. Booch, très complète, offre des outils permettant de couvrir tous les besoins possibles en matière de conception de systèmes informatiques de tous types. Il n'est néanmoins pas toujours nécessaire d'avoir recours à la totalité de ces outils dans le cadre d'un cas précis. C'est pourquoi nous n'allons présenter ici que les concepts qui nous paraissent les plus importants.

**Modèles logiques et physiques**: Ils permettent de faire la distinction entre ce qui relève de l'analyse du problème (modèle logique) et ce qui relève de son implantation informatique (modèle physique). On utilisera pour cela quatre types de diagrammes : les diagrammes de classe, les diagrammes d'objets, les diagrammes de modules et les diagrammes de processus.

**Modèles statiques et dynamiques****:** Les quatre types de diagrammes précédents, décrivent essentiellement la structure statique du problème. Pour en décrire l'aspect dynamique, comment les objets sont créés puis détruits, comment ils s'échangent des messages, on utilise deux autres types de diagrammes : les diagrammes de transition d'états et les diagrammes de temps

**OOSE** (**Object Oriented Software Engineering**) est un langage de [modélisation objet](https://fr.wikipedia.org/wiki/Mod%C3%A9lisation_objet) créé par [Ivar Jacobson](https://fr.wikipedia.org/wiki/Ivar_Jacobson). OOSE est une méthode pour l’analyse initiale des usages de logiciels, basée sur les « cas d’utilisation » et le cycle de vie des logiciels. Ce langage fut fusionné avec [OMT](https://fr.wikipedia.org/wiki/Object_Modeling_Technique) de [James Rumbaugh](https://fr.wikipedia.org/wiki/James_Rumbaugh) et OOD de [Grady Booch](https://fr.wikipedia.org/wiki/Grady_Booch) pour créer l’[UML](https://fr.wikipedia.org/wiki/Unified_modeling_language) en 1995.

La méthode OOSE est basée sur cinq modèles et trois types d’objets fondés par Ivar Jacobson en 1992. Les cinq modèles sont :

* le modèle des besoins est un modèle qui se situe au niveau de l’interaction entre l’individu et l’environnement ;
* le modèle d’analyse permet de définir les différentes relations entre les objets, les acteurs et le système ;
* le modèle de conception permet de modéliser ces relations à travers différents diagrammes ;
* le modèle d’implémentation permet le passage au code ;
* le modèle de test basé sur les essais d’actions et d’interactions du système.

Les trois types d’objets sont :

* les entités représentant un objet matériel ou immatériel ;
* des contrôles permettant différentes vérifications ;
* des interfaces représentant des classes abstraites, sans attribut et dont les méthodes sont abstraites et publiques.

**UML**

UML est l’acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». Il est né de la fusion des trois méthodes qui ont influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. Il s’agit d’un compromis qui a été trouvé par une équipe d'experts : Grady Booch, James Rumbaugh et Ivar Jacobson. UML est à présent un standard défini par l'Object Management Group (OMG). De très nombreuses entreprises de renom ont adopté UML et participent encore aujourd’hui à son développement.

La notation UML est un **langage visuel** constitué d’un ensemble de schémas, appelés des **diagrammes**, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour **représenter** le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d’être effectuées par le logiciel, etc. UML est surtout utilisé lorsqu’on prévoit de développer des applications avec une démarche objet (développement en Java, en C++, en C# etc.).

À ce jour, il existe **13 diagrammes**« officiels ». Ces diagrammes sont tous réalisés **à partir du besoin des utilisateurs** et peuvent être regroupés selon les deux aspects suivants :

* **Les aspects fonctionnels** : Qui utilisera le logiciel et pour quoi faire ? Comment les actions devront-elles se dérouler ? Quelles informations seront utilisées pour cela ?
* **Les aspects liés à l’architecture** : Quels seront les différents composants logiciels à utiliser (base de données, librairies, interfaces, etc.) ? Sur quel matériel chacun des composants sera installé ?
  1. Choix de la méthode d'analyse

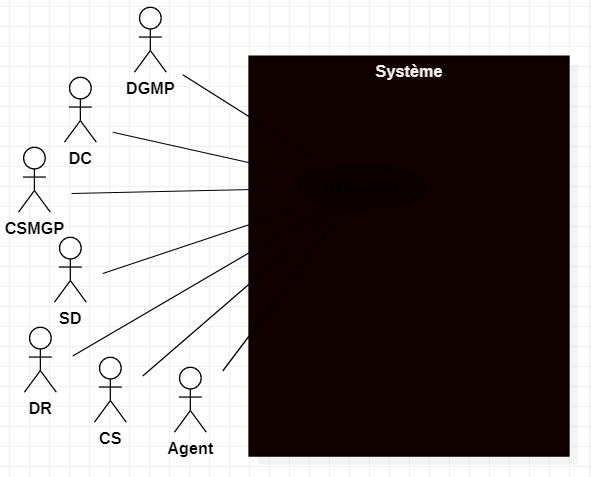
De ce qui précède, l’on note l’existence de trois principaux types de méthodes d’analyse et de conception (méthodes cartésiennes, méthodes systémiques et méthodes objets).

Si l'analyse systémique  s’appuie sur une approche globale des problèmes étudiés tout en séparant les données des traitements, l’analyse cartésienne ou fonctionnelle aborde le système par la décomposition systématique du domaine étudié en sous domaines tout en mettant en avant les fonctionnalités qu’il est sensé remplir. Les méthodes d’analyse objet sont quant à elles beaucoup plus adaptées aux langages de programmation et systèmes de gestion de base de données orientés objets. Elles ont surtout l’avantage d’aborder l’étude d’un problème suivant les aspects statiques, dynamiques puis fonctionnels. C’est ce type de méthode que nous choisirons dans le cadre de ce projet. En l’occurrence, nous utiliserons le Langage Unifié de Modélisation UML pour Unified Modeling Language en anglais.

1. Etudes conceptuelles
3. 1. Identification des acteurs du système

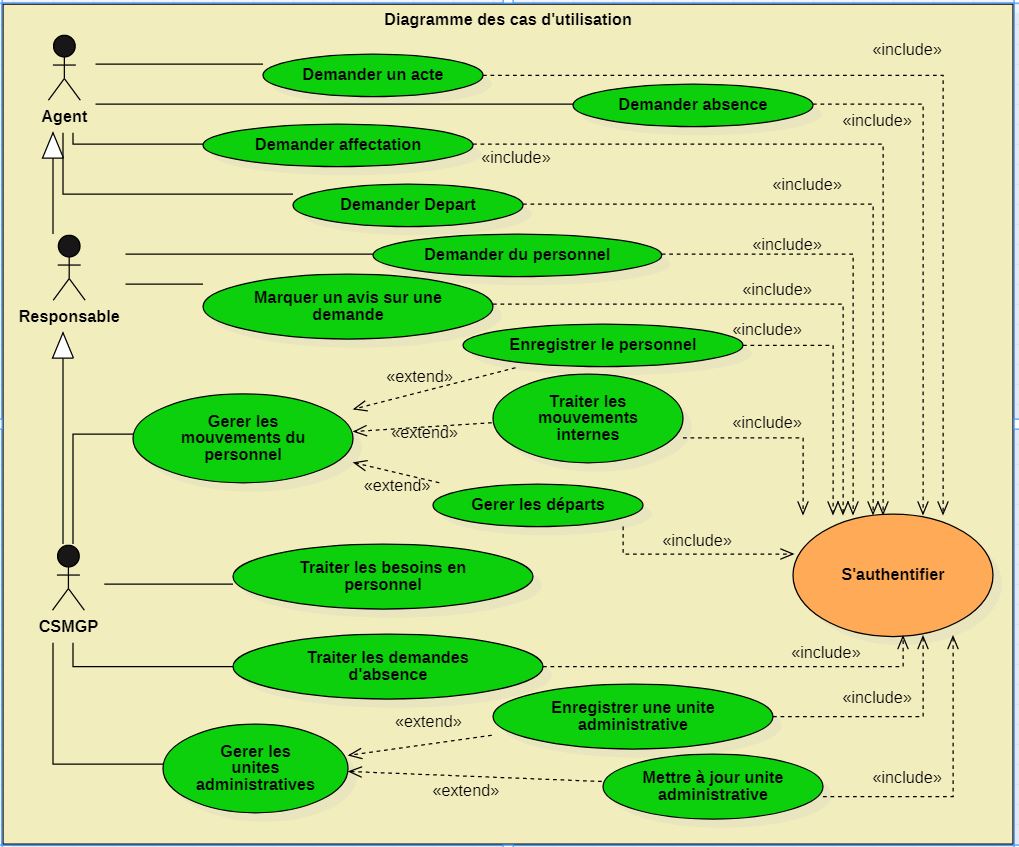
Les acteurs du système sont :

* Le Directeur Général des Marchés Publics (DGMP) :
* Le chef du Service des Moyens Généraux et du Personnel (CSMGP):
* Les Directeur Centraux (DC)
* Les Sous-Directeurs (SD)
* Les Directeurs Régionaux (DR)
* Les chefs de service (CS)
* Les agents

**Diagramme de contexte**

* 1. Diagrammes des cas d'utilisation

Les **diagrammes de cas d'utilisation** sont des [diagrammes UML](https://fr.wikipedia.org/wiki/Unified_Modeling_Language#Les_diagrammes) utilisés pour donner une vision globale du comportement fonctionnel d'un système. Un cas d'utilisation représente une unité d'interaction entre un utilisateur (humain ou machine) et un système. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases).

****

1. Description textuelle du cas d’utilisation « Gérer les mouvements du personnel »

|  |  |
| --- | --- |
| N° | 1 |
| Nom du cas d’utilisation | Gérer les mouvements du personnel |
| Acteur(s) | Chef de Service des Moyens Généraux et du Personnel (CSMGP) |
| Description | * Le CSMGP enregistre dans le système les informations concernant tout nouvel agent. * Le CSMGP Traite les mouvements internes (affectations) * Le CSMGP Gère les départs |
| Précondition | Le CSMGP doit être authentifié en tant que CSMGP. |

* + 1. Description textuelle du cas d’utilisation « Traiter les besoins en personnel »

|  |  |
| --- | --- |
| N° | 2 |
| Nom du cas d’utilisation | Traiter les besoins en personnel |
| Acteur(s) | Chef du Service des Moyens Généraux et du Personnel (CSMGP) |
| Description | Le CSMGP traite les demandes en personnel exprimée par les responsables en procédant à des affectations selon la disponibilité des agents et les profils demandés |
| Précondition | * Le CSMGP doit être authentifié en tant que CSMGP. * Ce traitement doit répondre à une demande en personnel exprimée par un responsable (SD, DR ou CS) |

* + 1. Description textuelle du Cas d’utilisation « Traiter les demandes d’absence »

|  |  |
| --- | --- |
| N° | 3 |
| Nom du cas d’utilisation | Traiter les demandes d’absence |
| Acteur(s) | Chef du Service des Moyens Généraux et du Personnel (CSMGP) |
| Description | Le Chef du Service des Moyens Généraux et du Personnel traite les demandes d’absence en la soumettant à la signature du Directeur puis en joignant par la suite la réponse à la demande. |
| Précondition | * Le CSMGP doit être authentifié en tant que CSMGP. * Ce traitement doit répondre à une demande préalablement exprimée par un agent. |

* + 1. Description textuelle du cas d’utilisation « Gérer les unités administratives »

|  |  |
| --- | --- |
| N° | 4 |
| Nom du cas d’utilisation | Gérer les unités administratives |
| Acteur(s) | Chef du Service des Moyens Généraux et du Personnel (CSMGP) |
| Description | Le CSMGP peut créer dans le système de nouvelles unités administratives (Services, Sous-Direction, Direction Régionale, Cellule,…) et les modifier. |
| Précondition | * Le CSMGP doit être authentifié en tant que CSMGP. |

* + 1. Description textuelle du cas d’utilisation « Marquer un avis »

|  |  |
| --- | --- |
| N° | 5 |
| Nom du cas d’utilisation | Marquer un avis |
| Acteur(s) | Responsable (CSR, SD, DR) |
| Description | Chaque responsable pourra marquer son accord ou refuser (avec motif à l’appui) les demandes de ses agents. |
| Précondition | * Le responsable doit être authentifié en tant que CSR, SD ou DR. * L’avis doit porter sur une demande préalablement formulée par un agent du responsable concerné. |
| Démarrage | Clique sur le lien «Marquer un avis» |

* + 1. Description textuelle du cas d’utilisation « Demander du personnel »

|  |  |
| --- | --- |
| N° | 6 |
| Nom du cas d’utilisation | Demander du personnel |
| Acteur(s) | Responsable (SD, CS, DR, D) |
| Description | Tout responsable peut exprimer un besoin en personnel en y précisant les profils dont il a besoin et pour chaque profil, le nombre d’agents nécessaires. |
| Précondition | * Le Responsable doit être authentifié en tant que SD, DR ou CSR. |
| Démarrage | Clique sur le lien « Demander du personnel » |

* + 1. Description textuelle du cas d’utilisation « Demander acte »

|  |  |
| --- | --- |
| N° | 7 |
| Nom du cas d’utilisation | Demander acte |
| Acteur(s) | Agent |
| Description | Tout agent peut, à partir du système, faire une demande d’acte administratif (Certificat de présence, attestation de travail,…) |
| Précondition | Etre authentifié |
| Démarrage | Clique sur le lien « Demander un acte administratif » |

* + 1. Description textuelle du cas d’utilisation « Demander une absence»

|  |  |
| --- | --- |
| N° | 8 |
| Nom du cas d’utilisation | Demander une absence |
| Acteur(s) | Agent |
| Description | Tout agent peut exprimer, à partir du système, une demande d’absence temporaire (autorisation d’absence, congé annuel, congé maladie, congé de maternité,…) |
| Précondition | * Etre authentifié. * Respecter les critères métiers requis pour chaque type d’absence temporaire exemple du congé annuel : |
| Démarrage | Clique sur le lien « Demander une absence temporaire » |

* + 1. Description textuelle du cas d’utilisation « Demander affectation »

|  |  |
| --- | --- |
| N° | 9 |
| Nom du cas d’utilisation | Demander affectation |
| Acteur(s) | Agent |
| Description | Tout agent a la possibilité d’effectuer, à travers le système, une demande de changement de service ou de sous-direction. |
| Précondition | Etre authentifié.  Avoir déjà été muté au sein d’un service. |
| Démarrage | Clique sur le lien « Demander une affectation » |

* 1. Diagramme de classes

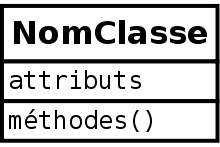
Une [classe](https://fr.wikipedia.org/wiki/Classe_(informatique)) est un ensemble de [fonctions](https://fr.wikipedia.org/wiki/Fonction_(informatique)) et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la [programmation orientée objet](https://fr.wikipedia.org/wiki/Programmation_objet). Elles permettent de modéliser un [programme](https://fr.wikipedia.org/wiki/Programme_informatique) et ainsi de découper une tâche complexe en plusieurs petites tâches simples.

Une [classe](https://fr.wikipedia.org/wiki/Classe_(informatique)) décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les [instances](https://fr.wikipedia.org/wiki/Instance_(programmation)) de la classe.

Le **diagramme de classes** est un schéma utilisé en [génie logiciel](https://fr.wikipedia.org/wiki/G%C3%A9nie_logiciel) pour présenter les [classes](https://fr.wikipedia.org/wiki/Classe_(informatique)) et les [interfaces](https://fr.wikipedia.org/wiki/Interface_(informatique)) des systèmes ainsi que les différentes relations entre celles-ci. Ce [diagramme](https://fr.wikipedia.org/wiki/Diagramme) fait partie de la partie [statique](https://fr.wikipedia.org/wiki/Statique) d'[UML](https://fr.wikipedia.org/wiki/Unified_modeling_language) car il fait abstraction des aspects temporels et [dynamiques](https://fr.wikipedia.org/wiki/Dynamique).

Les classes peuvent être liées entre elles grâce au mécanisme d'[héritage](https://fr.wikipedia.org/wiki/H%C3%A9ritage_(Informatique)) qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un symbole spécifique dans le diagramme de classes.

Une classe est représentée par un rectangle séparé en trois parties :



**Représentation d’une classe**

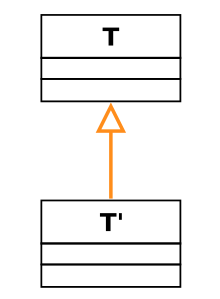
* la première partie contient le nom de la classe
* la seconde contient les attributs de la classe
* la dernière contient les méthodes de la classe

La seconde et la dernière représentent le **comportement**de la classe.

Dans un diagramme de classes, les classes sont liées entre elles par différents types de relation selon la situation à modéliser. On distingue les relations d’héritage (généralisation-spécialisation) et les relations d’association (Composition, agrégation et dépendance).

* L’héritage décrit une relation entre une classe générale (classe de base ou classe parent) et une classe spécialisée (sous-classe, classe dérivée ou classe fille). La classe spécialisée est intégralement cohérente avec la classe de base, mais comporte des informations supplémentaires (attributs, opérations, associations). Un objet de la classe spécialisée peut être utilisé partout où un objet de la classe de base est autorisé.

Le symbole utilisé pour la relation d'héritage est une flèche avec un trait plein dont la pointe est un triangle fermé.



**T’ Hérite de T**

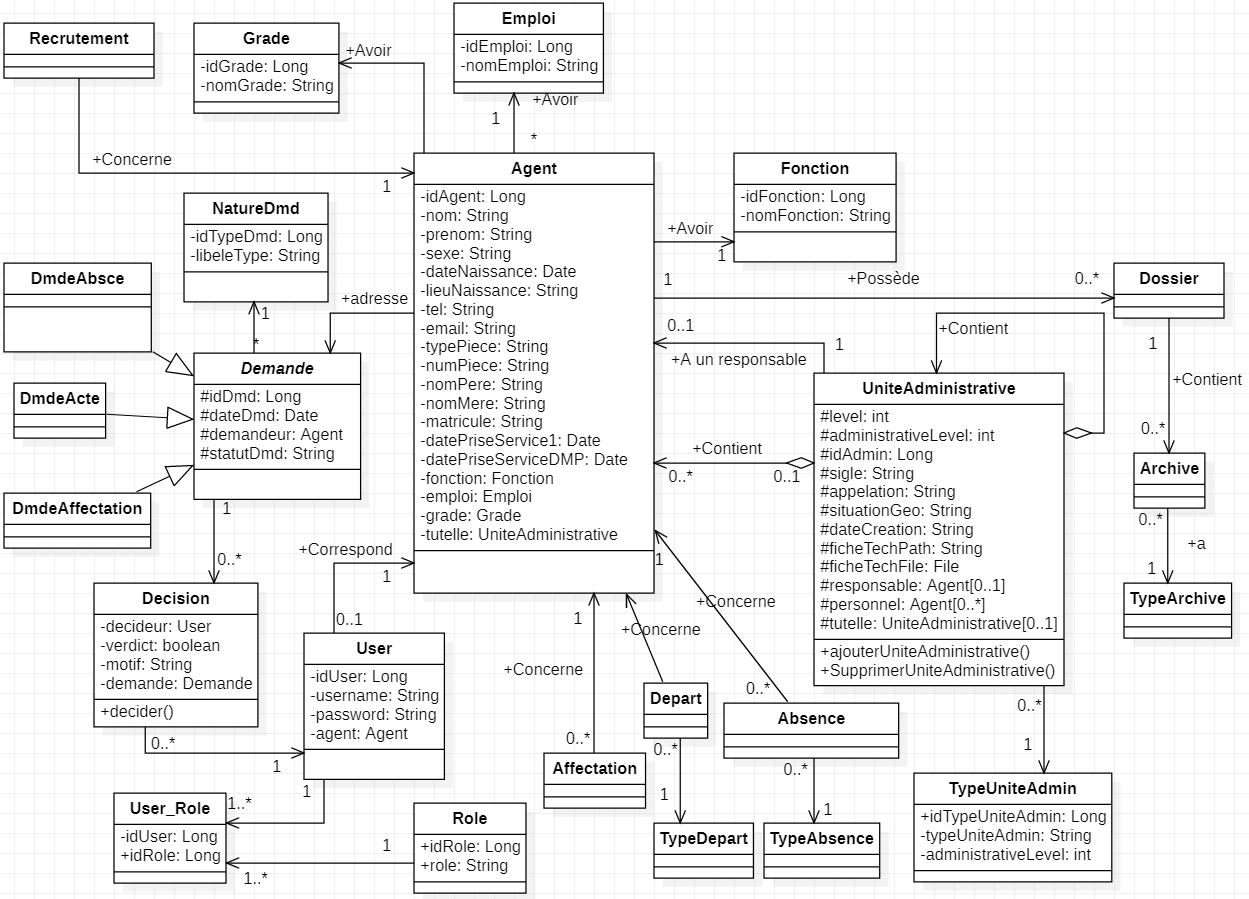
Les propriétés principales de l'héritage sont :

* la classe dérivée possède toutes les caractéristiques de la classe parent, mais elle ne peut accéder aux caractéristiques privées de cette dernière ;
* une classe dérivée peut redéfinir une ou plusieurs méthodes de la classe parent;
* toutes les associations de la classe parent s'appliquent aux classes dérivées ;
* une instance d'une classe peut être utilisée partout où une instance de sa classe parent est attendue;
* une classe peut avoir plusieurs parents dans certains langage, on parle alors d'héritage multiple. Le langage C++ est un des langages objet permettant son implémentation effective, le langage Java ne le permet pas.

En UML, la relation d'héritage n'est pas propre aux classes. Elle s'applique à d'autres éléments du langage comme les paquetages, les acteurs ou les cas.

* L'[association](https://fr.wikipedia.org/wiki/Association_(programmation)) est une connexion sémantique entre deux classes (relation logique). Une association peut être nommée. Elle peut être binaire, dans ce cas elle est représentée par un simple trait, ou n-aire. Les **multiplicités** dans une association sont comparables aux [cardinalités](https://fr.wikipedia.org/wiki/Cardinalit%C3%A9_(programmation)) du système [Merise](https://fr.wikipedia.org/wiki/Merise_(informatique)). Elles servent à compter le nombre minimum et maximum d'instances de chaque classe dans la relation liant 2 ou plusieurs classes.

Le schéma suivant présente le diagramme du système de gestion du personnel des marchés publics.

**Diagramme de classes du système** 

1. Transformation du diagramme de classe en modèle relationnel

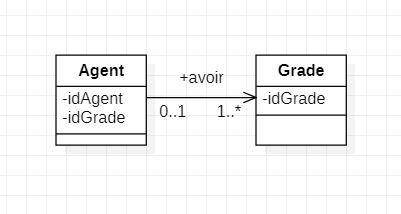
* **Règles de transformation**
* Chaque classe se transforme en une table ;
* Chaque attribut de classe se transforme en un champ de table.
* **Définition de la clé primaire**

La clé primaire est un champ ou un ensemble de champs d’une table dont la valeur identifie de façon unique chaque enregistrement et qui nous sert de référence pour établir des liens avec d'autres tables.

Pour définir la clé primaire, l’on peut choisir un attribut de la classe qui, pour chaque instance de la classe, est unique, non nul et dont la valeur n’est pas susceptible de changer avec le temps. Si la classe ne possède aucun attribut vérifiant ces conditions, il est alors nécessaire de générer un identifiant.

* + 1. Transformation des relations (association, agrégation, composition)
* **Cas 1 : Relation de type 1 à n**

L’identifiant de la classe de cardinalité n devient une clé étrangère de la classe de cardinalité 1. On crée ensuite pour chacune de ces classes, la table correspondante dans le modèle relationnel. Exemple : Un agent possède un grade, en revanche, un grade est possédé par plusieurs agents.



La clé primaire de la classe Grade devient clé étrangère dans la classe Agent

* **Cas 2 : Relation de type plusieurs à plusieurs**

On crée une classe intermédiaire, composée des identifiants des deux premières classes et d’éventuels autres attributs. L’identifiant de cette classe intermédiaire sera la concaténation des identifiants des deux autres. On obtient le modèle relationnel en créant pour chacune de ces classes ainsi constituées, la table correspondante. La table correspondant à la classe intermédiaire admettra donc les clés primaires des deux autres tables comme clés étrangères. Sa clé primaire sera ainsi la concaténation des clés primaires de ces deux tables.

* + 1. Transformation des relations d'héritage

La transformation de la relation d’héritage en modèle relationnel se fait selon trois stratégies :

* **Stratégie 1 : une table par hiérarchie de classe (Single table)**

Dans cette stratégie, le modèle relationnel est fait d'une seule table pour toute la hiérarchie de classes (Classe mère et classes filles). Cette table contient tous les attributs de la classe mère ainsi que ceux des classes filles. A ces colonnes, vient s'ajouter une colonne technique appelée discriminant. Cette colonne technique permet de distinguer, pour chaque enregistrement, le sous-type duquel il est issu.

Cette méthode à l’avantage de factoriser en une seule table toutes les données de plusieurs sous-types d’une classe mère. Elle permet ainsi d’éviter de créer plusieurs tables répétant quasiment les mêmes attributs.

Elle a cependant l’inconvénient de favoriser l’existence, dans la base de données, de plusieurs champs vides (sans objets) selon le sous-type de l’enregistrement.

L’utilisation de cette stratégie se justifie donc lorsque les classes filles possèdent plusieurs attributs communs et peu d’attributs spécifiques.

* **Stratégie** **2 :** [**Une table par classe concrète**](https://viseo.developpez.com/tutoriels/java/hibernate/strategies-heritage/#LI.3) **(Table per class)**

Cette stratégie consiste à créer une table pour chaque sous-type. Chacune de ces tables comprend des attributs génériques (ceux de la classe mère) et des attributs spécifiques au sous-type concerné.

Cette méthode résout le problème des champs vides de la méthode « Single table » mais présente l’inconvénient de générer plusieurs tables possédant des champs communs.

L’utilisation de cette méthode se justifie donc lorsque les sous types possèdent peu d’attributs commun et plusieurs attributs spécifiques.

* **Stratégie 3 :** [**Une table et une jointure par classe**](https://viseo.developpez.com/tutoriels/java/hibernate/strategies-heritage/#LI.4) **(Joined table)**

Cette stratégie consiste à créer une table pour la classe mère et une table pour chaque classe fille. La table de la classe mère comprend les attributs génériques (communs à tous les sous-types) les tables des classes filles contiennent les attributs spécifiques. Autrement dit, les informations concernant une instance d’une classe fille sont réparties entre la classe fille et la classe mère. La seule colonne commune entre les tables est la colonne ID qui permet de faire les jointures entre table mère et table fille.

Cette stratégie résout les problèmes posés par les deux premières méthodes mais a l’inconvénient d’être coûteuse. En effet, Pour enregistrer un simple objet, il faut faire plusieurs écritures (au niveau de la table mère et au niveau de la table fille). Dès qu'il s'agit de faire une lecture, il faut parcourir plusieurs tables et effectuer des jointures. Dans le cas d'une requête sur une table mère, il faut faire des jointures sur l’ensemble des tables filles. Une recherche sur une table fille réduit le nombre de jointures mais ne permet pas de s'en affranchir complètement car il faut malgré tout charger les informations stockées dans la table mère.

* 1. Diagrammes de séquence

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs. Ils sont une solution populaire de modélisation dynamique en langage UML, car ils se concentrent plus précisément sur les lignes de vie*,* les processus et les objets qui vivent simultanément, et les messages qu'ils échangent entre eux pour exercer une fonction avant la fin de la ligne de vie.

* Les objets au cœur d’un système interagissent en s’échangent des messages.
* Les acteurs interagissent avec le système au moyen d’IHM (Interfaces Homme-Machine).

Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Ces diagrammes sont utilisés à la fois par les développeurs logiciels et les managers d'entreprises pour analyser les besoins d'un nouveau système ou documenter un processus existant.

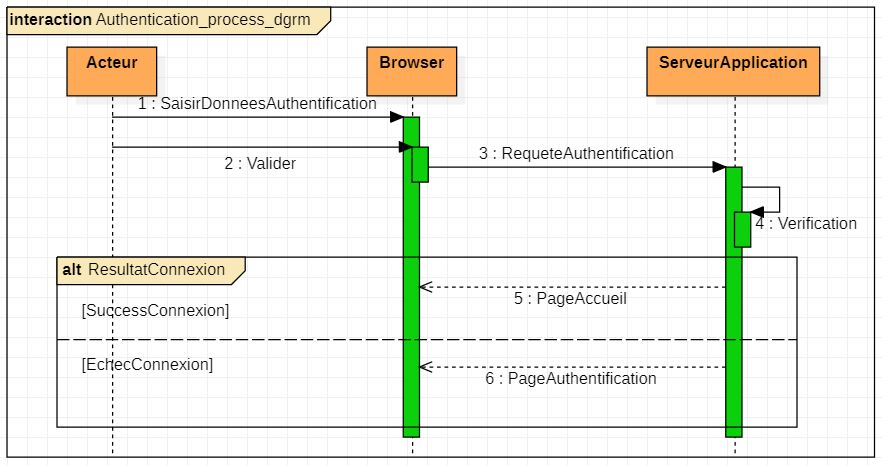
Les diagrammes de séquence peuvent constituer des références utiles pour les entreprises et d'autres organisations. Les diagrammes de séquence sont utiles pour :

* Représenter les détails d'un cas d'utilisation UML ;
* Modéliser le déroulement logique d'une procédure, fonction ou opération complexe ;
* Voir comment les objets et les composants interagissent entre eux pour effectuer un processus ;
* Schématiser et comprendre le fonctionnement détaillé d'un scénario existant ou à venir.

Dans le cadre de ce projet, nous utiliserons des diagrammes de séquence afin de décrire certains processus métiers de la gestion administrative du personnel de la Direction Générale des Marchés Publics. C’est processus sont les suivants :

* Le recrutement ;
* Le traitement des demandes d’acte ;
* Le traitement des demandes d’absence temporaire ;
* Le traitement des demandes d’affectation.

Nous illustreront également à l’aide d’un diagramme de séquence le cas d’utilisation « authentification » qui est un cas d’utilisation inclus à tous les autres cas d’utilisation du système.



1. Diagramme de séquence du cas d’utilisation « Authentification »

**Diagramme de séquence du cas d’utilisation « Authentification »**

* **Commentaires**

**1** : L’acteur saisi ses données d’authentification dans le formulaire d’authentification à partir d’un navigateur web.

**2** : Il clique sur le bouton valider du formulaire, ce qui déclenche la soumission du formulaire.

**3**: La soumission du formulaire déclenche une requête http vers le serveur d’application

**4** : Le serveur d’application procède à la vérification du login et du mot de passe saisie par l’acteur.

**5** : Si les données d’authentification sont correctes, le serveur d’application retourne la page d’accueil à l’écran du navigateur.

**6** : Si les données d’authentification ne pas correctes, le serveur d’application retourne la page d’authentification au navigateur.

Le processus d’authentification ainsi décrit, sera requis pour toutes les actions à réaliser à travers le système. Cela dit, dans les diagrammes de séquence que nous présenterons par la suite, nous nous limiterons à faire référence à ce diagramme d’authentification, sans toutefois le représenter entièrement, afin de ne pas surcharger la présentation. Toutefois, toujours dans le souci de ne pas surcharger les diagrammes, nous omettrons sciemment, dans certains cas, de faire référence à ce diagramme d’authentification.

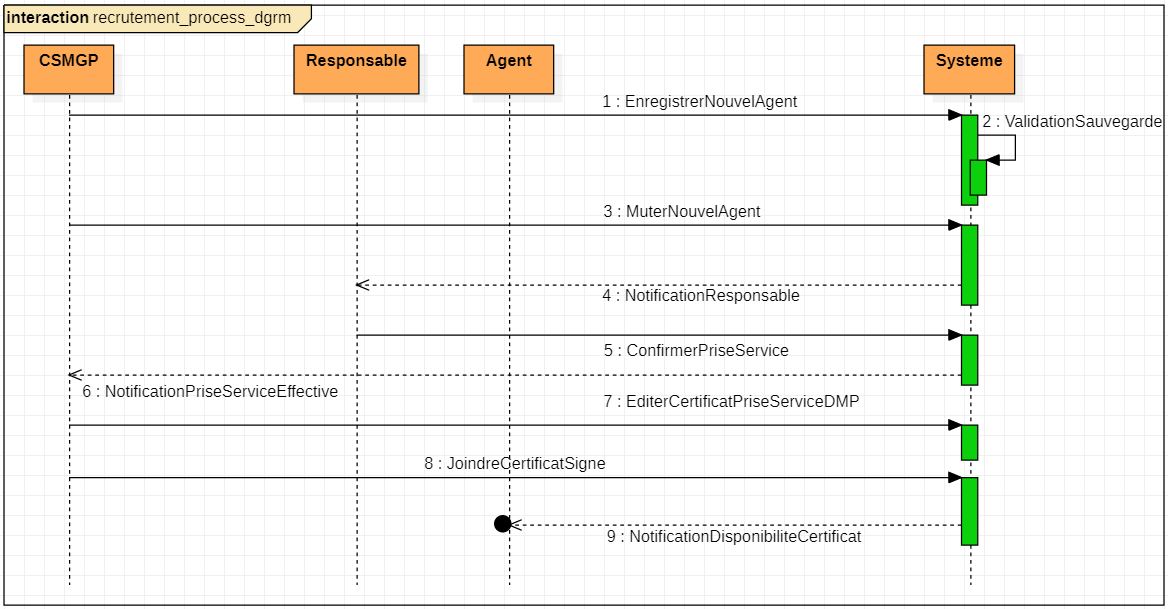
* + 1. Processus de recrutement
* **Commentaires :**

**1 :** Le CSMGP enregistre les données d’un nouvel agent dans le système

**2 :** Le système vérifie et valide les informations saisies avant de les sauvegarder

**3 :** Le CSMGP procède à la mutation du nouvel agent au sein d’un service de la DGMP

**4 :** Le système notifie le responsable de ce service de l’affectation d’un nouvel agent au sein de l’un de ses services



**Diagramme de séquence du processus de recrutement**

**5 :** Le responsable du service confirme la prise de service du nouvel agent (après que l’agent ait effectivement pris service)

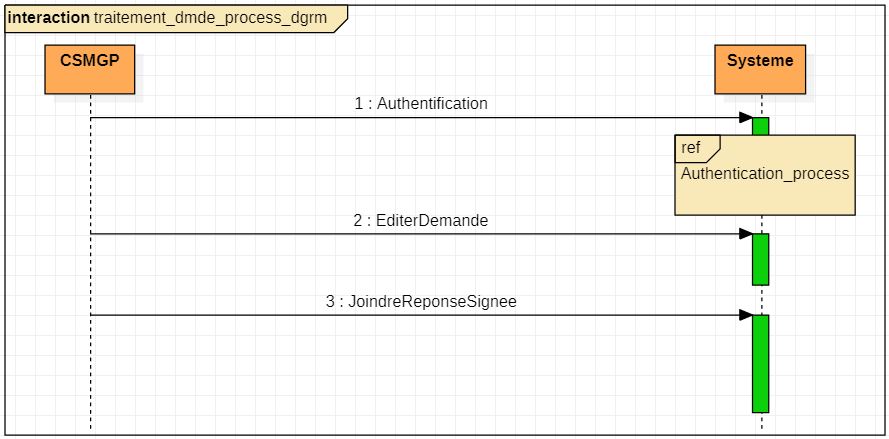
**6 :** Le système notifie le CSMGP de la prise de service effective du nouvel agent

**7 :** Le CSMGP édite le certificat de prise de service du nouvel agent

**8 :** Le CSMGP joint le certificat de prise de service signé par le DGMP au dossier de l’agent

**9 :** Le système notifie l’agent de la disponibilité de son certificat de prise de service.

* + 1. Traitement des demandes



**Diagramme de séquence du processus de traitement des demandes**

* **Commentaires :**

**1 :** Le CSMGP s’authentifie

**2 :** Le CSMGP édite la demande adressée par l’agent

**3 :** Le CSMGP joint la réponse signée par l’autorité compétente

* + 1. Traitement des demandes de départ
* **Commentaires :**

**1 :** L’agent s’authentifie

**2 :** L’agent formule sa demande de départ.

**3 :** Le système notifie le responsable (supérieur hiérarchique) de l’agent de la demande.

**4 :** Le supérieur hiérarchique s’authentifie.

**5 :** Le supérieur hiérarchique marque un avis sur la demande. En cas de refus, le supérieur hiérarchique peut préciser le motif

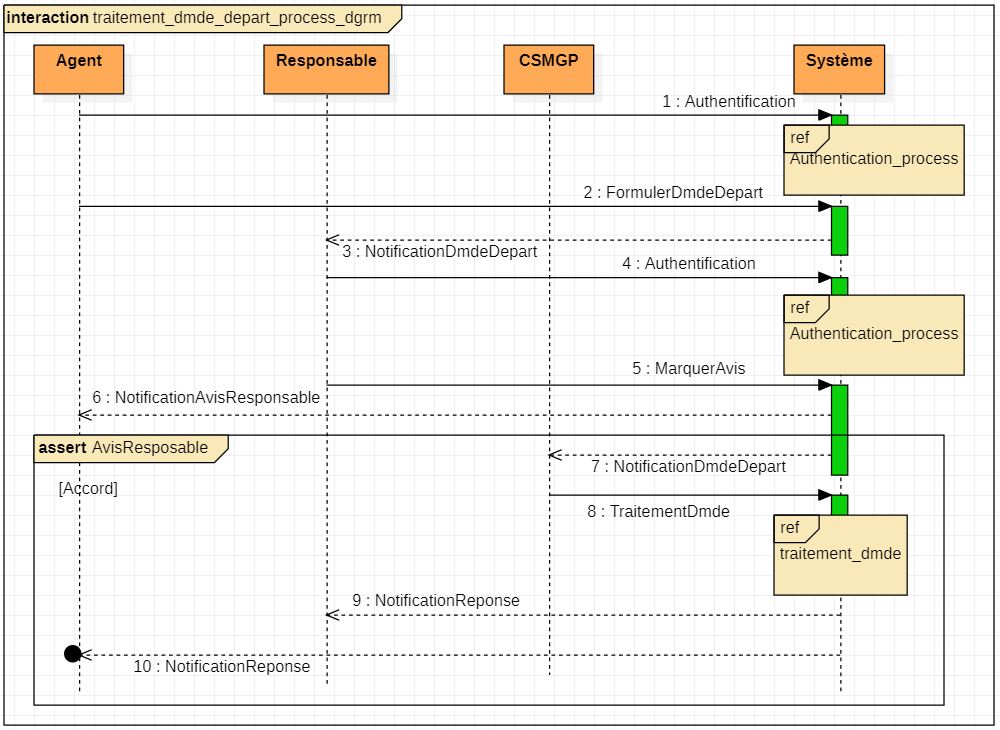
**6 :** Le système notifie l’agent de l’avis (accord ou pas) de son supérieur hiérarchique. En cas de refus, le processus prend fin à ce stade.

**7 :** Si l’avis du responsable est un accord, le système notifie le CSMGP de la demande.

**8 :** Le CSMGP traite la demande.

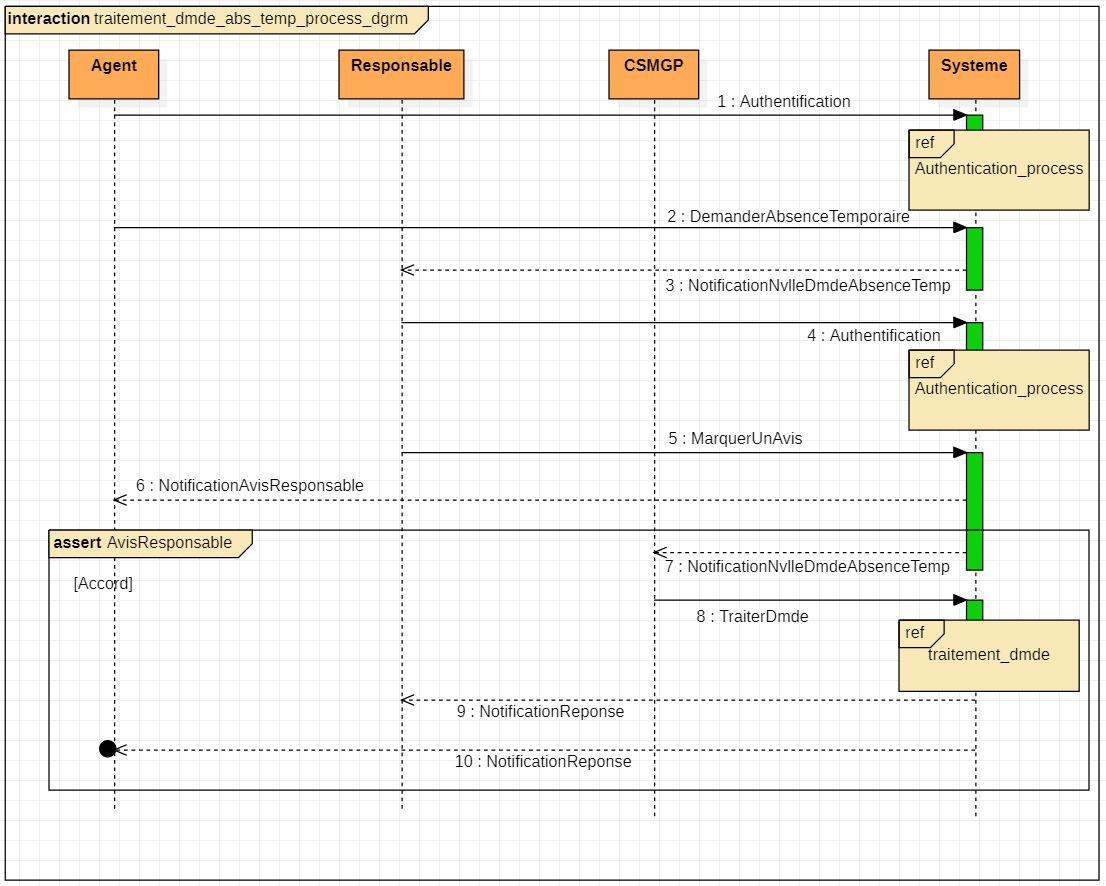
**9 :** Le système notifie le responsable de la réponse à la demande de départ de son agent.

**10 :** Le système notifie l’agent de la réponse à sa demande de départ.



**Processus de traitement des demandes de départ**

* + 1. Traitement des demandes d’absences (Congé, absences)



**Processus de traitement des demandes d’absences temporaires**

* **Commentaires :**

**1 :** L’agent s’authentifie

**2 :** L’agent formule sa demande d’absence temporaire (Congé ou permission d’absence).

**3 :** Le système notifie le responsable (supérieur hiérarchique) de l’agent de la demande.

**4 :** Le supérieur hiérarchique s’authentifie.

**5 :** Le supérieur hiérarchique marque un avis sur la demande. En cas de refus, le supérieur hiérarchique peut préciser le motif

**6 :** Le système notifie l’agent de l’avis (accord ou pas) de son supérieur hiérarchique. En cas de refus, le processus prend fin à ce stade.

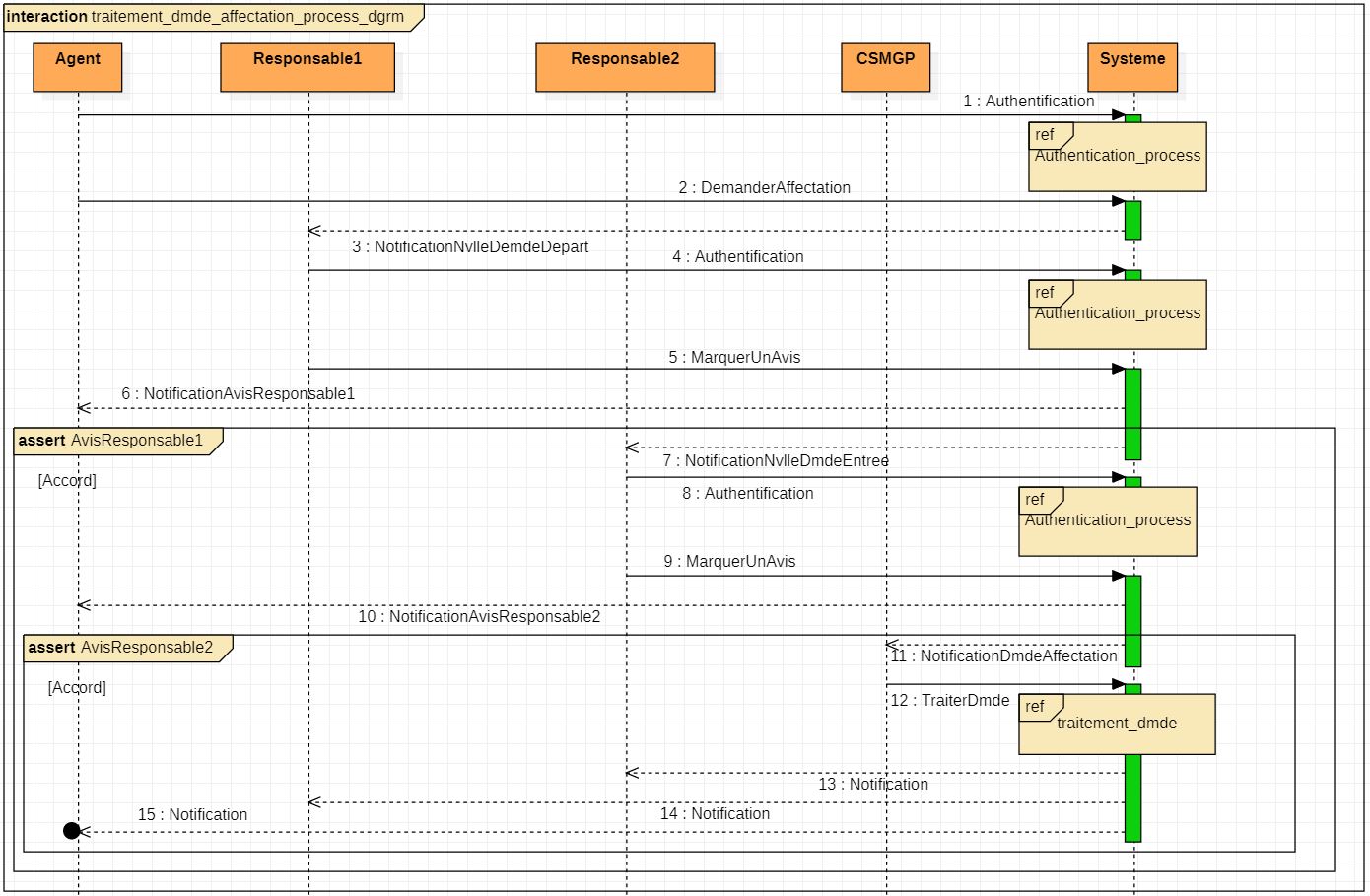
**7 :** Si l’avis du responsable est un accord, le système notifie le CSMGP de la demande.

**8 :** Le CSMGP traite la demande de l’agent.

**9 :** Le système notifie le responsable de la réponse à la demande de son agent.

**10 :** Le système notifie l’agent de la réponse à sa demande.

* + 1. Traitement des demandes d’affectation interne (au sein de la DGMP)



**Processus de traitement des demandes d’affectations internes**

* **Commentaires :**

**1 :** L’agent s’authentifie

**2 :** L’agent formule sa demande d’affectation (dans un autre service de la DGMP).

**3 :** Le système notifie le responsable du service de départ d’une nouvelle demande de départ de l’un de ses agents.

**4 :** Le responsable du service de départ s’authentifie.

**5 :** Le responsable du service de départ marque un avis sur la demande. En cas de refus, le responsable peut préciser le motif.

**6 :** Le système notifie l’agent de l’avis (accord ou pas) de son supérieur hiérarchique. En cas de refus, le processus prend fin à ce stade.

**7 :** Si l’avis du responsable du service de départ est un accord, le système notifie le responsable du service de destination.

**8 :** Le responsable du service de destination s’authentifie.

**9 :** Le responsable du service de destination marque un avis sur la demande.

**10 :** Le système notifie l’agent de l’avis (accord ou pas) du responsable du service de destination. En cas de refus, le processus prend fin à ce stade.

**11 :** Si l’avis du responsable du service de destination est un accord, le système notifie le CSMGP d’une nouvelle demande d’affectation en interne.

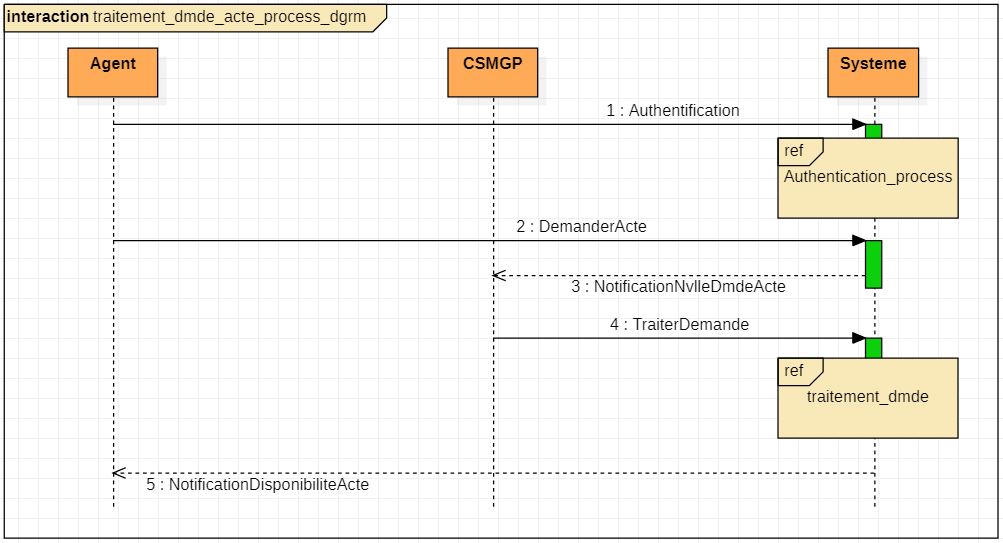
**12 :** Le CSMGP traite la demande (il édite la demande, il joint la réponse signée par le DGMP).

**13 :** Le système notifie le responsable du service de destination de la réponse du DGMP.

**14 :** Le système notifie le responsable du service de départ de la réponse du DGMP.

**15 :** Le système notifie l’agent de la réponse du DGMP.

* + 1. Traitement des demandes d’actes administratifs



* **Commentaires :**

**1 :** L’agent s’authentifie

**2 :** L’agent formule sa demande d’acte administratif.

**3 :** Le système notifie le CSMGP d’une nouvelle demande d’acte administratif.

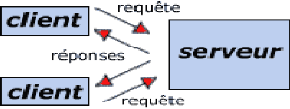
**4 :** Le CSMGP traite la demande (Il s’authentifie, il édite demande, puis il joint la réponse signée par l’autorité compétente).

**5 :** Le système notifie l’agent de la disponibilité de l’acte demandé.

1. Etudes techniques
2. 1. Architectures

De nombreuses applications fonctionnent selon un modèle de communication client/serveur, cela signifie que des machines clientes (machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client FTP, client de messagerie, de client http..., selon la nature des messages échangés entre le client et le serveur.



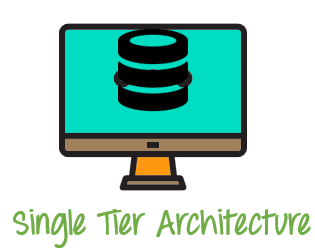
**Schéma de fonctionnement d'un système client/serveur**

· Le client émet une requête vers le serveur grâce à son adresse et le port, qui désigne un service particulier du serveur

· Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port.

* + 1. Architecture un tiers

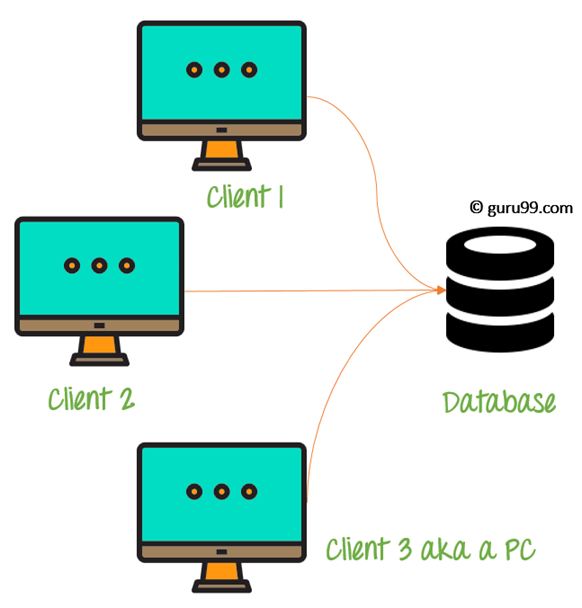
L'architecture d’application la plus simple est à 1 niveau. C’est une architecture dans laquelle le client, le serveur et la base de données résident tous sur la même machine. Dans ce cas, on ne peut pas parler d'architecture client-serveur mais d'informatique centralisée. Une telle architecture ne permet pas le partage de l’information sur plusieurs machines distantes et est rarement utilisée en environnement professionnel de production.



Architecture 1-tiers

* + 1. Architecture deux tiers

L'architecture à deux niveaux (aussi appelée architecture 2-tiers) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service. Dans cette architecture, La couche de présentation s'exécute sur un client (PC, mobile, tablette, etc.) et les données sont stockées sur un serveur. L'architecture à 2 niveaux offre une sécurité supplémentaire au SGBD car il n'est pas exposé directement à l'utilisateur final. Cependant elle offre moins de possibilités en termes de réalisation de traitements métiers.



Architectures 2-tiers

* + 1. Architecture trois tiers

Le schéma à 3 niveaux est une extension de l'architecture à 2 niveaux. L'architecture à 3 niveaux a les couches suivantes :

* Couche de présentation (PC, tablette, mobile, etc.)
* Couche d'application (serveur)
* Serveur de base de données

Cette architecture de SGBD contient une couche d'application entre l'utilisateur et le SGBD, qui est chargée de communiquer la demande de l'utilisateur au système de SGBD et d'envoyer la réponse du SGBD à l'utilisateur.

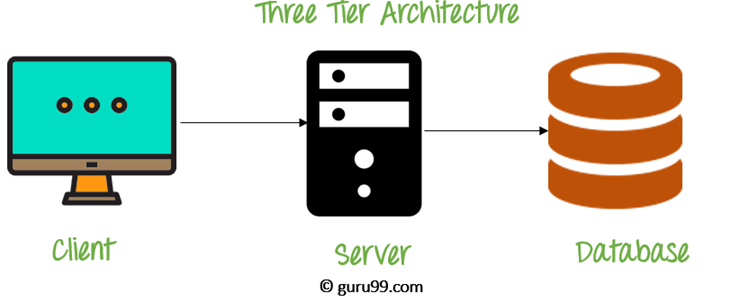
La couche application (couche logique métier) traite également la logique fonctionnelle, les contraintes et règles métiers avant de transmettre des données à l'utilisateur ou vers le SGBD. L'architecture à trois niveaux est l'architecture d’application la plus populaire.

L'architecture à trois niveaux a pour objectif:

De séparer les applications clientes de la base de données physique

Assurer l’indépendance des données du programme

Prendre en charge de plusieurs sources des données



Architectures 3-tiers

* + 1. Architecture Microservice
* ***Définition et caractéristiques***

Les microservices, également connus sous le nom d'architecture microservices, sont un style architectural qui structure une application comme un ensemble de services relativement petit. Ces services sont développés et déployés indépendamment mais communiquent les uns avec les autres pour faire fonctionner l’application mère. Chacun de ces services, appelé microservice, ne gère qu’une seule fonctionnalité de l’application et possède, éventuellement, une base de données dédiée.

**Dans le cadre de ce projet, c’est cette architecture (microservice) que nous avons choisi de mettre en œuvre pour répondre à la problématique qui nous a été soumise**. Ce choix se justifie par la multitude des avantages dont bénéficie l’architecture microservice au regard de ses inconvénients qui restent des problèmes dont les solutions techniques sont à porter de main.

* **Avantages**

Les microservices présentent de nombreux avantages aussi au niveau développement, de l’administration système qu’en termes de performances.

Pour ce qui relève du développement, les microservices sont **faciles à maintenir**, **facile à faire évoluer** et **facile à tester**. Concernant l’administration système, ils sont **hautement scalables**, ils se prêtent aisément aux méthodes de l’**intégration et du déploiement continue**, leurs **temps de livraison réduits**, ils peuvent être **reconfigurés à chaud** (sans interruption) et leurs **temps de redémarrage sont très faibles** (de l’ordre de la milliseconde). Enfin, les performances des microservices se caractérisent par des temps de réponse relativement faibles.

* **Inconvénients**

Cependant les microservices n’ont pas que des avantages. Ils ont aussi des inconvénients dont les plus importants sont le **défi de la communication entre microservices**, La nécessité de gérer des **transactions distribuées**, le défis d’un **découpage optimal du domaine fonctionnel en microservices** et enfin le risque d’un **déploiement périlleux** avec une multitude de microservices à gérer.

Fort heureusement, il existe bien des moyens de contourner la plupart de ces inconvénients. Le tableau ci-dessous présente les inconvénients liés aux architectures microservices et les solutions pour y remédier.

|  |  |
| --- | --- |
| **Problèmes** | **Solutions** |
| Défi de la communication entre microservices | * 1. Implémenter un modèle de communication Rest entre les microservices et/ou   2. Implémenter un modèle communication asynchrone via un broker (Kafka, Rabbit MQ, …) |
| Transactions distribuées | Implémenter le design pattern saga |
| Déploiement périlleux | Mettre en œuvre un pipeline DevOps permettant d’assurer l’automatisation du déploiement. |

* 1. Technologies de développement

Techniquement, le développement des applications web consiste en l’écriture de plusieurs lignes de code qui sont ensuite interprétées par le navigateur afin de produire les éléments d’affichage graphique (images, couleurs, tableaux, formulaires, …). Cela dit, les activités de développement sont assez fastidieuses et couteuses en temps.

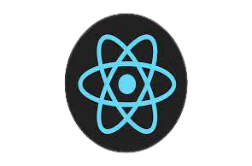
De ce fait, il s’avère indispensable d’utiliser des assistants permettant de réduire la quantité de code requise et optimiser le processus de développement. Ces assistants peuvent être des Framework, des bibliothèques ou des librairies. Dans le cadre de ce projet, les technologies que nous utiliseront peuvent être classées en quatre catégories que sont les technologies **frontend**, les technologies **backend**, les technologies **DevOps** et les technologies de **base** **de** **données**. La figure ci-dessous présente l’ensemble de ces technologies.

****

* + 1. Technologies frontend

Le frontend désigne la partie visuelle des applications. Dans les applications web cette partie est affichée dans des navigateurs (Chrome, Firefox, IE, Edge, Safari, …). L’architecture frontend des applications web repose sur les trois principales technologies que sont : HTML, CSS et JavaScript. Au tour de ces trois principales technologies, gravitent un ensemble de librairies, bibliothèques et Framework permettant d’optimiser le processus de développement par la réduction du temps de développement et des coûts de production.

**React.JS**



Le tableau ci-dessous décrit certaines de ces technologies que nous utiliseront dans le cadre de ce projet.

|  |  |
| --- | --- |
| **Technologies** | **Description** |
| **HTML** | **HTML** signifie « HyperText Markup Language ». C’est un langage créé en 1989 par l’informaticien Tim Berners-Lee.Il est utilisé afin de présenter et créer le contenu des pages web et leurs structures. L’un des principaux avantages de HTML, est son universalité. Les pages écrites à base du HTML peuvent être consultées sur tous les navigateurs web indépendamment du système d’exploitation. Avec le HTML, l’on peut :   * publier des informations en ligne sous forme de titres, texte, tables, listes, etc. ; * amener vers des informations en ligne via des liens hypertextes avec un clic ; * concevoir des formulaires de saisie ; * inclure des feuilles de calculs, des vidéos ou des photos et d'autres [applications](https://www.futura-sciences.com/sciences/definitions/mathematiques-application-13200/) directement dans les documents, etc. |
| **CSS** | Le CSS pour Cascading Style Sheets, est un langage descriptif utilisé pour faire la mise en forme de pages [HTML](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/). On le traduit en français par feuilles de style en cascade.  Apparu dans les années 1990, le CSS se présente comme une alternative à la mise en forme via des balises, notamment HTML.  Le but de CSS est séparer la structure d'un document HTML et sa présentation. En effet, avec HTML, on peut définir à la fois la structure et la présentation. Mais cela pose des problèmes d’efficience et de maintenabilité. Avec le couple HTML/CSS, on peut créer des pages web où la structure du document se trouve dans des fichiers HTML tandis que la présentation se situe dans un fichier CSS.  Ainsi, on peut, définir dans un fichier CSS, un ensemble de règles stylistiques communes à toutes les pages d'un site internet.  Cette pratique permet un gain de temps considérable et facilite la modification de la présentation du site en entier.  Outre cela, CSS permet d’assurer le caractère responsive des pages web en définissant des règles différentes pour chaque support d'affichage (navigateur classique, télévision, tablette, support mobile, etc.) |
| **Sass** | Sass (**S**yntactically **a**wesome **s**tyle **s**heets) est une extension du CSS. Il utilise une syntaxe compatible au CSS, mais offre des fonctionnalités supplémentaires telles que l’ajout de variables et des règles d’imbrication qui le rendent plus efficace et facile à maintenir.  Un des inconvénients du CSS standard est qu’il ne supporte pas [les variables](https://techlib.fr/definition/variable.html). Par exemple, si vous avez plusieurs styles de la même couleur, vous devez définir la couleur séparément pour chaque style. Si vous décidez de changer la couleur, vous devez la changer pour chaque instance du document CSS. Avec Sass, vous pouvez définir la couleur en tant que variable et l'affecter à chaque style qui l'utilise. Si vous décidez de changer la couleur, il vous suffit de la changer une seule fois, là où elle a été initialement définie dans le document. |
| **JavaScript**  **JavaScript** | [JavaScript](https://developer.mozilla.org/fr/docs/Glossary/JavaScript) est un langage de programmation informatique dynamique.  Il est léger et le plus souvent utilisé en complément de HTML et CSS dans la mise en œuvre des pages Web. C'est un langage de programmation interprété pouvant implémenter le paradigme orienté objet.  Entre autres, il permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, de réagir aux actions de l’utilisateur (au clique, au survol, à la saisie, …).  Le langage JavaScript est principalement employé pour améliorer l'ergonomie des pages web. Son principal intérêt réside dans son mode de fonctionnement : le langage JavaScript offre en effet la possibilité d'exécuter un code sans être contraint de recharger une page web. En cela, il joue un rôle dans l'amélioration de la rapidité de chargement des pages. |
| **jQuery** | JQuery, est une bibliothèque [JavaScript](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/) gratuite, libre et multiplateforme. C’est une bibliothèque rapide, petite et riche en fonctionnalités. Compatible avec l'ensemble des navigateurs Web (Internet Explorer, Safari, Chrome, Firefox, etc.), elle a été conçue et développée en 2006 pour faciliter l'écriture du JavaScript qui est connu pour être assez verbeux. jQuery est la bibliothèque JavaScript la plus connue et la plus utilisée. Elle permet la manipulation de documents HTML, la gestion des événements, l'animation et l’Ajax avec une API facile à utiliser. JQuery s'exécute essentiellement côté client. |
| **Bootstrap**  **Bootstrap** | Bootstrap est une bibliothèque développée par l'équipe du réseau social Twitter. cette bibliothèque utilise les langages [HTML](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/), CSS et JavaScript tout en s’appuyant sur la bibliothèque JQuery. Elle fournit aux développeurs des outils pour créer un site facilement. Cette bibliothèque est pensée pour développer des pages avec un design responsive, qui s'adapte à tout type d'écran (smartphone, tablette, ordinateur, télévision, …). Bootstrap fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. |
| **NPM** | npm est le gestionnaire de paquets officiel de la plateforme JavaScript Node.js. npm fonctionne avec un terminal et gère les dépendances pour les applications aussi bien front-end que back-end. Il pourra être utilisé pour installer dans le projet les dépendances Bootstrap, JQuery, React.JS, Next.JS et bien d’autres |
| **React.JS**  **React**.**JS** | React.JS est une bibliothèque JavaScript [open-source](https://ibracilinks.com/blog/quest-ce-quun-logiciel-open-source-et-pourquoi-est-ce-important) développée par [Facebook](https://fr.wikipedia.org/wiki/Facebook) depuis [2013](https://fr.wikipedia.org/wiki/2013_en_informatique) qui est utilisée pour construire des interfaces utilisateur s’exécutant sur une seule page. Il est utilisé pour gérer la couche d'affichage des applications considérée comme la vue dans le modèle [MVC](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur). React nous permet également de créer des composants d'interface utilisateur réutilisables. |
| **Next**.**JS** | Next.js est un [Framework](https://fr.wikipedia.org/wiki/Framework) gratuit et [open source](https://fr.wikipedia.org/wiki/Open_source) s'appuyant sur la librairie [React](https://fr.wikipedia.org/wiki/React).JS et sur la technologie [Node.js](https://fr.wikipedia.org/wiki/Node.js). Next.JS permet de créer des applications web prenant en charge les techniques de rendu des pages web [côté serveur](https://fr.wikipedia.org/wiki/Server-side) (SSR : Server Side Rendering) et de rendu statique des pages web (SSG: Static Site Generation).  L'utilisation de ce Framework présente de nombreux avantages comme la rapidité d'affichage et de chargement des pages pour les utilisateurs qui les consultent. Ils permettent un meilleur référencement de l'application par les moteurs de recherches car leur contenu est lisible par ces derniers moyennant moins de ressources techniques |

* + 1. Technologies Back-end

Dans les architectures logiciel, le terme « backend » désigne la partie des applications invisible à l’utilisateur et s’exécutant côté serveur. Cette partie est considérée comme l’épine dorsale des applications car c’est elle qui implémente la logique métier sur laquelle la partie frontend s’appui pour afficher les informations à l’utilisateur. Pour mettre en place le backend des applications, les développeurs sont amenés à faire des choix entre différentes technologies concurrentes. Les plus connues de ces technologies sont les suivantes : PHP, JAVA, C#, Python, Node.JS, C++, …

Le choix d’une technologie peut être motivé par différentes raisons (connaissances techniques des développeurs, facilité de prise en main, existence d’une importante communauté, les particularités techniques de la technologie, le coût de mise en œuvre, …). Dans une architecture distribuée, notamment microservice, chaque microservice peut utiliser une technologie différente selon ses spécificités.

Dans le cadre de ce projet, nous avons choisi d’utiliser la technologie java car c’est la technologie que nous maîtrisons le mieux. Outre cet aspect, la technologie java présente l’avantage d’être robuste, de disposer d’une importante communauté active et de d’avoir d’innombrables librairies et Framework permettant d’accélérer et d’optimiser le processus de développement.

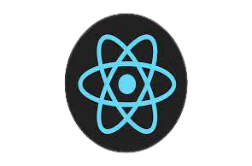
Par ailleurs, pour la gestion des notifications instantanées, nous utiliserons la technologie Node.JS qui y est spécifiquement adaptée.

Le tableau ci-dessous présente l’ensemble des technologies (langage, Framework, librairies) que nous choisies d’utiliser pour le développement du backend de l’application.

|  |  |
| --- | --- |
| **Technologies** | **Descriptions** |
|  | **Java** est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) [orienté objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet). Reprenant en grande partie la syntaxe du langage C++, Java permet de développer des applications de Desktop, des applications mobiles, des applications web, des applications pour composant embarqués et bien d’autres.  Une particularité de Java est son excellente portabilité. Les logiciels écrits dans ce langage sont [compilés](https://fr.wikipedia.org/wiki/Compilateur_Java) vers une [représentation binaire](https://fr.wikipedia.org/wiki/Syst%C3%A8me_binaire) intermédiaire qui peut être exécutée dans une [machine virtuelle Java](https://fr.wikipedia.org/wiki/Machine_virtuelle_Java) (JVM) en faisant abstraction du [système d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) |
| **Spring Boot**  **D:\INFORMATIQUE\Cours\CEFIVE\Projet-Prof\CEFIVE-Rapport-ProjetProf\Rapport stage\Rédaction\icons\spring-boot-removebg-preview.png** | Spring Boot  est un Framework open source permettant de développer des applications Java.  Spring Boot a la particularité de faciliter le développement d’applications basées sur des architectures micro-service. C’est un outil à la fois robuste, performant, léger et facile à prendre  en main. |
| **Spring Security** | Spring Security est un Framework qui se concentre sur la fourniture à la fois d'authentification et d'autorisation aux applications Java. C'est la norme de facto pour sécuriser les applications basées sur Spring. Comme tous les projets Spring, la véritable puissance de Spring Security réside dans la facilité avec laquelle elle peut être étendue pour répondre aux exigences personnalisées.  **D:\INFORMATIQUE\Cours\CEFIVE\Projet-Prof\CEFIVE-Rapport-ProjetProf\Rapport stage\Rédaction\icons\spring-security-removebg-preview.pngSpring Security** |
| **Spring Data** | La mission de Spring Data est de fournir un modèle de programmation familier et cohérent basé sur Spring pour l'accès aux données tout en conservant les caractéristiques spéciales du système de gestion de données sous-jacent.  Il facilite l'utilisation des technologies d'accès aux données aussi bien pour les bases de données relationnelles que non relationnelles. Il s'agit d'un grand projet qui contient de nombreux sous-projets spécifiques à des SGBD différents. Les projets sont développés en collaboration avec de nombreuses entreprises et développeurs qui sont à l'origine de ces SGBD. |
| Spring Cloud | Spring Cloud fournit des outils aux développeurs afin de créer rapidement certains des modèles courants dans les systèmes distribués (gestion de configuration, découverte de services, circuit-breaker, load balancing, …). La coordination des systèmes distribués conduit à des modèles standards, et en utilisant Spring Cloud, les développeurs peuvent rapidement mettre en place des services et des applications qui implémentent ces modèles. Ils fonctionneront bien dans n'importe quel environnement distribué, y compris le propre ordinateur portable du développeur et les plates-formes gérées telles que Google Cloud Platform (GCP), Azur, ... |
|  | Apache est le serveur Web HTTP le plus répandu sur Internet. Le serveur Tomcat est un conteneur d'applications Web basé sur Java qui est utilisé pour exécuter des applications Web basées sur les Servlet java. Tomcat est devenu l'implémentation de référence pour les spécifications Java Servlets et Java Server Pages. C’est le server d’application utilisé de facto pour développer des microservices basés dur Spring Boot. |
|  | Hibernate est un ORM (Object Relational Mapping) c’est à dire un outil de mappage objet-relationnel open source qui fournit un cadre permettant de gérer la persistance des objets en base de données relationnelle. Il apporte une solution aux problèmes d’adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau.  La principale caractéristique d'Hibernate est le mappage des classes Java vers les tables de base de données et le mappage des types de données Java vers les types de données SQL. Hibernate fournit également des fonctionnalités de requête et de récupération de données. Il génère des appels SQL et soulage le développeur de la gestion manuelle et de la conversion d'objets du jeu de résultats. |
|  | **Apache Kafka** est un broker ou [agent de messages](https://fr.wikipedia.org/wiki/Agent_de_messages). C’est un projet open source de développé par l'[Apache Software Foundation](https://fr.wikipedia.org/wiki/Apache_Software_Foundation) écrit en [Scala](https://fr.wikipedia.org/wiki/Scala_(langage)). Le projet vise à fournir un système unifié, [en temps réel](https://fr.wikipedia.org/wiki/Syst%C3%A8me_temps_r%C3%A9el) à [latence](https://fr.wikipedia.org/wiki/Latence_(informatique)) faible pour la manipulation de flux de données entre systèmes distribués.  Kafka est principalement utilisé pour la mise en place de « data pipeline » temps réel. Mais il est aussi de plus en plus utilisé dans les architectures micro services comme système d’échange de message (événements / données) entre les différents microservices. Plus particulièrement, il peut être utilisé dans les architectures microservices pour résoudre le problème des transactions distribuées ou mettre en place système event sourcing. |
| **NodeJS** | Node.js (Node.JS) est une plate-forme de développement open source pour l'exécution de code JavaScript côté serveur. Node.JS est utile pour développer des applications qui nécessitent une connexion persistante du navigateur au serveur et est souvent utilisé pour des applications en temps réel telles que le chat, les flux d'actualités et les notifications Web push. Dans le cadre de ce projet, nous utiliserons Node.JS pour la gestion de la notification instantanée. |

* + 1. Technologies DevOps

**React.JS**



Le DevOps est une approche de l’ingénierie logiciel dont le but est de concilier les métiers de développeurs (Dev) et d’administrateurs système (Ops pour opérationnels). En effet, Ces deux métiers, indispensables à la production et à la mise en exploitation des logiciels, sont historiquement connus pour avoir des objectifs antagonistes. De fait, les Dev ont pour mission de créer et faire évoluer les applications tandis que les Ops se chargent de leur mise en production en se portant garant de leur disponibilité et de leurs performances.

Le problème vient du fait que les Dev responsables de l’évolution des applications ont tendance à toujours proposer de nouvelles versions des logiciels en vue de leurs améliorations. Par contre, les Ops, responsables du bon fonctionnement des applications dans leurs environnements, sont plutôt enclin à ne pas aimer les évolutions. Car toute amélioration vient éventuellement avec son lot de configurations et de dépendances logiciels à installer. Or l’installation de nouvelles dépendances et la prise en compte de nouvelles configurations dans l’environnement d’un logiciel pourraient impacter le fonctionnement de plusieurs autres applications fonctionnant sur la même infrastructure.

L’approche DevOps vise donc à rompre le mur existant entre ces deux mondes en prenant en compte les contraintes de déploiement dès la phase de programmation. Pour ce faire, le DevOps propose différents types d’outils, qui une fois mise en œuvre de façon coordonnée, permettent de faire le pont entre développeurs et administrateurs système. Ces différents types d’outils sont :

* Les outils de collaboration ;
* Les outils de gestion de projet ;
* Les outils de gestion de code source ;
* Les outils d’[intégration continue](https://www.padok.fr/blog/devops-integration-continue) et de déploiement continu, ou CI/CD ;
* Les outils d'automatisation et configuration ;
* Les outils de conteneurisation ;
* Les outils de monitoring ;
* Les providers.

Le tableau ci-dessous présente l’ensemble des outils DevOps que nous avons dans le cadre de ce projet.

|  |  |  |
| --- | --- | --- |
| **Type** | **Technologies** | **Descriptions** |
| **Gestion de code source** |  | Apache Maven (couramment appelé Maven) est un outil de [gestion et d'automatisation de production des projets logiciels](https://fr.wikipedia.org/wiki/Moteur_de_production) [Java](https://fr.wikipedia.org/wiki/Java_(langage)) en général et [Java EE](https://fr.wikipedia.org/wiki/Java_EE) en particulier. Il est utilisé pour automatiser l'[intégration continue](https://fr.wikipedia.org/wiki/Int%C3%A9gration_continue) lors d'un développement de logiciel. Maven est géré par l'organisation [Apache Software Foundation](https://fr.wikipedia.org/wiki/Apache_Software_Foundation). L'objectif recherché est de produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.  Maven utilise un paradigme connu sous le nom de Project Object Model (POM) afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa production. |
|  | JUnit est un Framework open source pour le développement et l'exécution de tests unitaires automatisables. Le principal intérêt est de s'assurer que le code répond toujours aux besoins même après d'éventuelles modifications. Plus généralement, ce type de tests est appelé tests unitaires de non-régression. JUnit propose un cadre pour le développement, l’exécution et l’affichage des résultats des tests unitaires reposant sur des assertions qui testent des résultats attendus. |
|  | Git est un [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) [décentralisé](https://fr.wikipedia.org/wiki/Gestion_de_versions#Gestion_de_versions_d.C3.A9centralis.C3.A9e). C'est un [logiciel libre](https://fr.wikipedia.org/wiki/Logiciel_libre) créé par [Linus Torvalds](https://fr.wikipedia.org/wiki/Linus_Torvalds), auteur du [noyau Linux](https://fr.wikipedia.org/wiki/Noyau_Linux), et distribué selon les termes de la [licence publique générale GNU](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU) version 2. |
|  | GitHub est un service web d'[hébergement](https://fr.wikipedia.org/wiki/H%C3%A9bergeur_web) et de gestion de développement de logiciels, utilisant le [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) [Git](https://fr.wikipedia.org/wiki/Git). Ce site est développé en [Ruby on Rails](https://fr.wikipedia.org/wiki/Ruby_on_Rails) et [Erlang](https://fr.wikipedia.org/wiki/Erlang_(langage)) par Chris Wanstrath, PJ Hyett et Tom Preston-Werner. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de [logiciels libres](https://fr.wikipedia.org/wiki/Logiciels_libres). Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet. |
| **CI/CD** |  | Jenkins est un outil logiciel d’intégration continu. Il s’agit d’un logiciel open source, développé à l’aide du langage de programmation Java. Il permet de tester et de rapporter les changements effectués sur une large base de code en temps réel. En utilisant ce logiciel, les développeurs peuvent rapidement détecter et résoudre les problèmes dans une base de code. Ainsi les tests de nouveaux builds peuvent être automatisés, ce qui permet d’intégrer plus facilement des changements à un projet, de façon continue. L’objectif de Jenkins est en effet d’accélérer le développement de logiciels par le biais de l’automatisation. Jenkins permet l’intégration de toutes les étapes du cycle de développement. L’intégration continue est assurée par le biais de plugins. Ces plugins permettent l’intégration de diverses étapes dites DevOps. Pour intégrer un outil particulier, il est nécessaire d’installer les plugins correspondant à cet outil (Git, Maven, …). |
| **Automatisation et configuration** |  | Ansible est une [plate-forme logicielle libre](https://fr.wikipedia.org/wiki/Logiciel_libre) pour la configuration et la gestion des ordinateurs. Elle combine le [déploiement de logiciels](https://fr.wikipedia.org/w/index.php?title=D%C3%A9ploiement_de_logiciels&action=edit&redlink=1) multi-nœuds, l'exécution des [tâches](https://fr.wikipedia.org/wiki/T%C3%A2che_(informatique)) ad-hoc, et la [gestion de configuration](https://fr.wikipedia.org/wiki/Gestion_de_configuration). Elle gère les différents nœuds à travers [SSH](https://fr.wikipedia.org/wiki/Secure_Shell) et ne nécessite l'installation d'aucun logiciel supplémentaire sur ceux-ci. Les modules communiquent via la sortie standard en notation [JSON](https://fr.wikipedia.org/wiki/JavaScript_Object_Notation) et peuvent être écrits dans n'importe quel langage de programmation. Le système utilise [YAML](https://fr.wikipedia.org/wiki/YAML) pour exprimer des descriptions réutilisables de systèmes, appelées playbook. |
| **Conteneurisation** |  | Docker une technologie qui permet d'embarquer une application dans un ou plusieurs containers logiciels qui pourra s'exécuter sur n'importe quel serveur, qu'il soit physique ou virtuel. C'est une technologie qui a pour but de faciliter les déploiements d'application, et la gestion du dimensionnement de l'infrastructure sous-jacente. |
|  | Kubernetes est une plate-forme open-source extensible et portable pour la gestion de charges de travail (workloads) et de services conteneurisés. Elle favorise à la fois l'écriture de configuration déclarative (declarative configuration) et l'automatisation.  Kubernetes a un certain nombre de fonctionnalités. Il peut être considéré comme:   * une plate-forme de conteneurs * une plate-forme de microservices * une plate-forme cloud portable et beaucoup plus.   Kubernetes fournit un environnement de gestion **focalisé sur le conteneur** (container-centric). Il orchestre les ressources machines (computing), la mise en réseau et l’infrastructure de stockage sur les workloads des utilisateurs.  **Source : https://kubernetes.io/** |
| **Monitoring** |  | "ELK" est un acronyme pour trois projets en open source : Elasticsearch, Logstash et Kibana. Elasticsearch est un moteur de recherche et d'analyse. Logstash est un pipeline côté serveur, destiné au traitement des données (type ETL). Sa mission ? Ingérer simultanément des données provenant de multitude de sources, puis les transformer et les envoyer vers un système de stockage comme Elasticsearch. Kibana permet aux utilisateurs de visualiser des données avec des tableaux et des graphes dans Elasticsearch.  **Source : https://www.elastic.co/** |

* + 1. Système de Gestion de Base de Données (SGBD)

Un Système de Gestion de Base de Données (SGBD) est un logiciel utilisé pour structurer et gérer les données. Il existe sur le marché, différents types de SGBD. Chaque type de SGBD présente des caractéristiques différentes et chacun est adapté à des contextes particuliers. Les principaux types de SGBD sont :

* SGBD relationnel : les données sont représentées dans différents tableaux, appelés tables, pouvant être liés entre eux.
* SGBD NoSQL (clé-valeur, orienté graphe, orienté document...) : les données ne sont pas structurées en tables mais sur des structures différentes :
  + clé-valeur : par exemple un dictionnaire qui à chaque mot (clé) associe une définition (valeur)
  + orienté graphe : associe à chaque élément les éléments liés (ex. : les amis d'une personne)

Les SGBD relationnels (**SGBD-R**) sont, de loin, le type de SGBD le plus couramment utilisé. Ils sont basés sur un modèle relationnel.

Pour interagir avec un SGBD-R on utilise un langage appelé **SQL** (Structured Query Language). Ce langage permet d'ajouter, modifier ou supprimer des données mais aussi d'interroger la base de données selon certains critères et faire des recoupements d'information en suivant les relations entre les tables. C’est ce type de Base de Données que nous utiliserons dans le cadre de ce projet.

Dans le monde des bases de données relationnelles on distingue :

* Oracle Database, PostgreSQL, MS SQL Server, MySQL, …

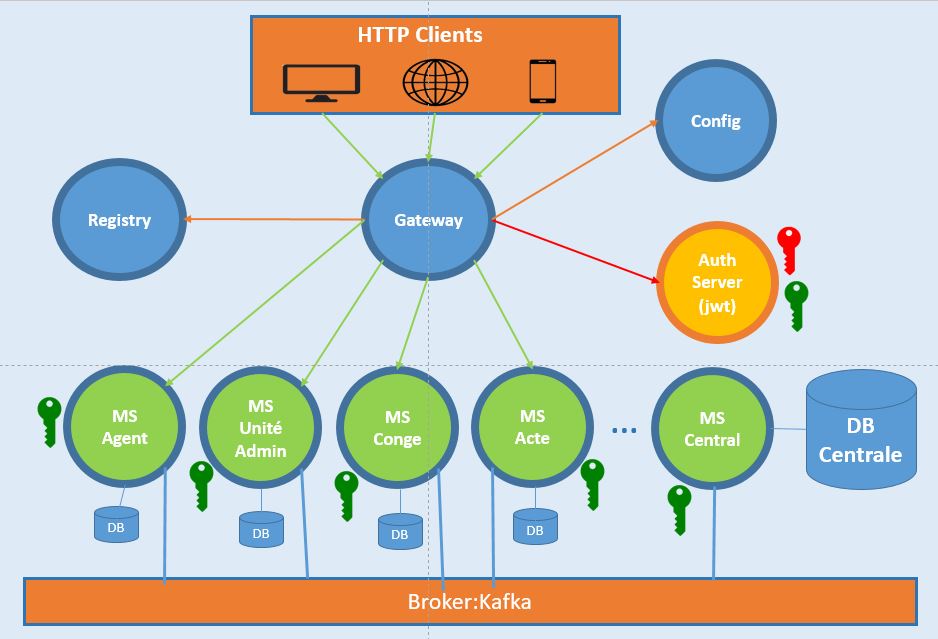
Tous ces SGBD-R ont leurs caractéristiques, leurs avantages et leurs inconvénients.

* Oracle Database est certes payant, mais taillé pour les grandes entreprises, il peut gérer efficacement d'énormes bases de données.
* PostgreSQL est gratuit, libre, de qualité professionnelle, il permet de gérer des bases de données, même assez importantes.
* MySQL est un SGBDR également performant et fiable. Il présente l’avantage d’une prise en main facile et est surtout réputé pour sa popularité. Il demeure cependant de moindre capacité comparativement aux deux premiers.

Pour la réalisation de ce projet, nous utiliserons le Système de Gestion de Bases de Données Relationnel [PostgreSQL](https://www.postgresql.org/).

|  |  |
| --- | --- |
| **Technologies** | **Descriptions** |
| **D:\INFORMATIQUE\Cours\CEFIVE\Projet-Prof\CEFIVE-Rapport-ProjetProf\Rapport stage\Rédaction\icons\PostgreSQL-removebg-preview.png** | PostgreSQL est un puissant système de base de données relationnelle objet open source qui utilise et étend le langage SQL combiné à de nombreuses fonctionnalités qui stockent et adaptent en toute sécurité les charges de travail de données les plus complexes. Les origines de PostgreSQL remontent à 1986 dans le cadre du projet [**POSTGRES**](https://www.postgresql.org/docs/current/history.html) à l'Université de Californie à Berkeley et compte plus de 30 ans de développement actif sur la plate-forme principale.  PostgreSQL a acquis une solide réputation pour son architecture éprouvée, sa fiabilité, l'intégrité des données, son ensemble de fonctionnalités robustes, son extensibilité et le dévouement de la communauté open source derrière le logiciel pour fournir constamment des solutions performantes et innovantes. PostgreSQL fonctionne sur [**tous les principaux systèmes d'exploitation**](https://www.postgresql.org/download/) , est conforme à la norme [**ACID**](https://en.wikipedia.org/wiki/ACID) depuis 2001 et dispose de puissants modules complémentaires tels que l’extension de base de données géo spatiale [**PostGIS**](https://postgis.net/) .  **Source :** https://www.postgresql.org/ |
| **H2** | H2 est un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) [relationnelles](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle) écrit en [Java](https://fr.wikipedia.org/wiki/Java_(langage)). Il peut être intégré à une application Java ou bien fonctionner en mode [client-serveur](https://fr.wikipedia.org/wiki/Client-serveur). Son fichier [jar](https://fr.wikipedia.org/wiki/JAR_(format_de_fichier)) est de petite taille : environ 1 Mo.  Les tables peuvent être créées en mémoire ou persistante. Dans ce projet, nous utiliserons H2 comme base de données de développement. |

**Architecture globale**



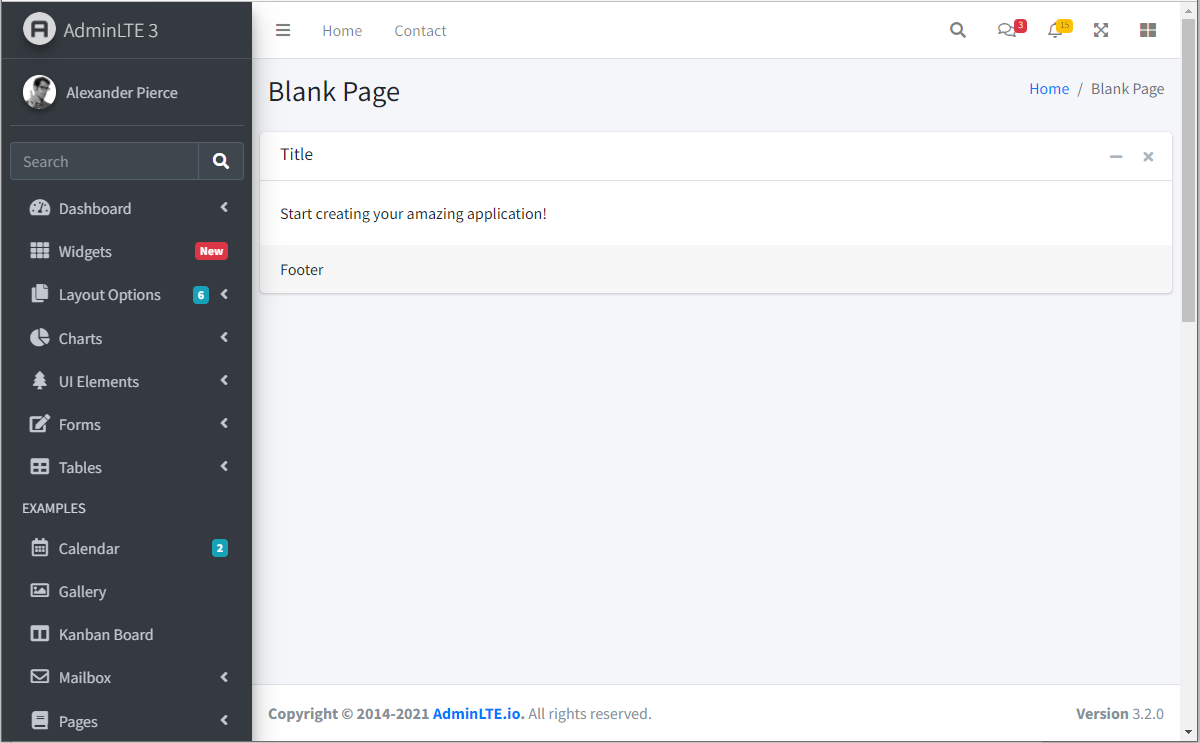
**Architecture interne des microservices**

# PARTIE III : REALISATION DE LA SOLUTION RETENUE

1. Développement
   1. Développement frontend
   2. Choix d’un Template

Un template est un **modèle de site "préfabriqué**", dont le gabarit est déjà défini. Il est conçu pour pouvoir s’adapter et s’intégrer facilement à différents cas de figure. Pour ce faire, on installe le template et on remplace simplement les images, les textes d'origine par les siens. En outre, dans le cas de la mise en place de sites dynamiques, une partie du contenu statique du template devra être remplacé par des données issues d’un système de stockage ou de plusieurs API distantes. Dans ce cas, il est indispensable que le template choisi intègre des technologies pouvant assurer une interaction facile avec les API et les SGBD.

Dans le cadre de ce projet, nous avons choisi d’utiliser le template **AdminLTE v3.2.0** disponible en téléchargement gratuit sur le site <https://www.adminlte.io>. C’est un template d’administration construit à partir de Bootstrap 4 qui fournit une gamme variée de composants responsives et réutilisables tels que des fenêtres modales, des tables stylisées, des panneaux d’affichage de contenu (texte, images, multimédias), des timelines ainsi que les composants standards de formulaire (champ de texte, listes déroulantes, sélecteurs de date, boutons poussoirs, case à cocher, …).



* 1. Identification des modules et pages

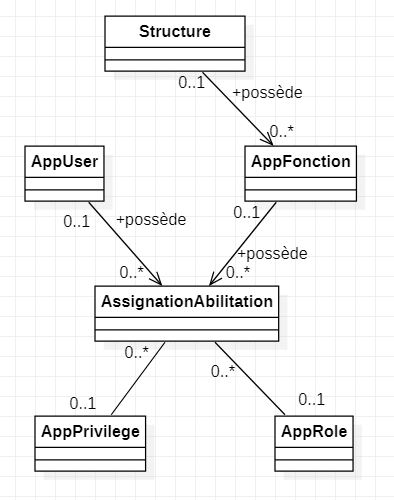
|  |  |  |
| --- | --- | --- |
| Modules | Pages | Fonctionalités |
| Gestion des comptes et droits d’accès | Login | * + S’authentifier |
| Gestion des comptes utilisateurs | * + Consulter la liste des utilisateurs (table)   + Créer un compte utilisateur (fenêtre modale)   + Modifier un utilisateur (fenêtre modale)   + Consulter le détail d’un utilisateur (fenêtre modale) ;   + Assigner / révoquer un privilège à un utilisateur ;   + Assigner / révoquer un rôle à un utilisateur ;   + Assigner / révoquer une fonction à un utilisateur ;   + Activer ou désactiver un compte utilisateur ; |
| Activation de nouveaux comptes | * + Activer son compte à partir un lien d’activation reçu par mail |
| Gestion des droits d’accès | * + Consulter la liste des fonctions, un rôle ou privilèges   + Créer une fonction, un rôle ou un privilège ;   + Modifier une fonction, un rôle ou un privilège ;   + Assigner/révoquer un privilège à un rôle ;   + Assigner/révoquer un rôle à une fonction |
| Gestion des agents | Liste des agents | * + Consulter la liste des agents ;   + Enregistrer un nouvel agent ;   + Modifier un agent   + Affecter un agent à une structure ;   + Nommer un agent à un poste de responsabilité ;   + Enregistrer une promotion pour un agent ; |
| Details d’un agent | * + Consulter le détail d’un agent ;   + Modifier un agent ;   + Affecter un agent à une structure ;   + Nommer un agent à un poste de responsabilité ;   + Enregistrer une promotion pour un agent ;   + Consulter la liste des demandes d’un agent ;   + Consulter l’historique des événements sur un agent (mouvements, promotion, nomination, absences, …) ; |
| Gestion des structures | Liste des structure | * + Consulter la liste des structures ;   + Créer une nouvelle structure ;   + Modifier une structure ;   + Affecter un agent à une structure ; |
| Détails structure | * + Consulter le détail d’une structure ;   + Voir la liste des agents appartenant à une structure ;   + Voir la liste des structures sous tutelle ;   + Voir les informations sur le responsable d’une structure ;   + Modifier une structure ;   + Affecter un ou plusieurs agents à une structure ; |
| Gestion des mouvements | Liste mouvements | * + Voir la liste des mouvements selon le type (Arrivée, Affectation, retraites, décès, départs volontaires, …) ;   + Enregistrer un nouveau mouvement ;   + Modifier un mouvement ;   + Rechercher les mouvements selon différents critères |
| Gestion des demandes | Liste des demandes | * + Consulter la liste des demandes ;   + Voir le détail d’une demande ;   + Faire une nouvelle demande ;   + Modifier une demande ;   + Annuler une demande ;   + Traiter une demande.   + Rechercher les demandes selon différents critères |
| Gestion des promotions | Liste des promotions | * + Consulter la liste des promotions ;   + Enregistrer une nouvelle promotion ;   + Modifier une promotion ;   + Rechercher les promotions selon différents critère |
| Gestion des nominations | Liste des nominations | * + Consulter la liste des nominations ;   + Enregistrer une nouvelle nomination ;   + Modifier une nomination ;   + Rechercher les nominations selon différents critère |
| Gestion des absences | Liste des absences | * + Consulter la liste des absences selon le type (congé annuel, autorisation d’absence, repos maladie, absence injustifiée, …)   + Enregistrer une nouvelle absence ;   + Modifier une absence ;   + Rechercher les absences selon différents critères |
| Gestion des actes et archives | Gestion des documents | * + Editer un acte ;   + Archiver un document ;   + Consulter une archive ;   + Rechercher une archive selon différents critères |
| Gestion des types | Gestion des types | * + Consulter la liste des types ;   + Créer de nouveaux types ;   + Modifier un type ;   + Rechercher les types selon différents critères |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Maquettage de quelques pages
* **Page de d’authentification (login)**
* **Liste des agents**
* **Formulaire d’enregistrement des agents**
* **Liste des structures**
* **Formulaire d’enregistrement des structures**
  1. Développement back-end
     1. Identification des MS

L’architecture micro service ayant été adoptée comme architecture du système, le back-end de l’application sera constitué d’un ensemble de microservices pouvant échanger des données les uns avec les autres. Le tableau ci-dessous présente les différents microservices que nous avons identifiés en rapport avec le domaine fonctionnel étudié. Nous les avons classés en trois groupes selon leur type.

|  |  |  |
| --- | --- | --- |
| **Type** | **Nom** | **Rôle** |
| Techniques | EUREKA-SERVER | Service d’enregistrement de tous les microservices |
| GATEWAY | Service de loadbalancing, permettant d’assurer l’aiguillage des requêtes vers les microservices et de faire l’équilibrage des charges. |
| CONFIG-SERVICE | Service permettant de centraliser la configuration des microservices. |
| Sécurité | AUTH-SERVER | C’est le service d’authentification et d’autorisation. Ce service sera chargé de générer les token (JWT) permettant d’authentifier chaque requête et d’autoriser l’accès à des ressources protégées. |
| Métiers | AGENT-SERVICE | Service de gestion des agents (Enregistrement, modification, recherche) |
| STRUCTURE-SERVICE | Service de gestion des structures (Enregistrement, modification, recherche) |
| MOUVEMENT-SERVICE | Service de gestion des mouvements (affections, arrivées, départs, retraites, décès, démissions, mise à disposition, disponibilité, licenciement, …) |
| DEMANDE-SERVICE | Service de gestion des demandes (demande congé, demande d’autorisation d’absence, demande d’actes administratifs, demande d’affectation, demande de départ, …). Ce service sera juste chargé de créer les demandes et éventuellement de les modifier. Par contre le traitement de ces demandes sera assuré par d’autres microservices. |
| PROMOTION-SERVICE | Service de gestion des promotions |
| ABSENCE-SERVICE | Service de gestion des absences |
| ACTE-SERVICE | Service de production des actes administratifs |
| ARCHIVAGE-SERVICE | Service de gestion de l’archivage |
| TYPE-SERVICE | Service de gestion de tous les types du système. C’est ce service qui va permettre de paramétrer le système |
| NOTIFICATION-SERVICE | Service d’envoie des notifications (notification interne à l’application et notification par e-mail) |

* + 1. Etude détaillée des microservices
       1. Auth-server
* **Diagramme** **de** **classe**

****

**Diagramme de classes**

**Une structure possède zéro ou plusieurs fonctions. Une fonction appartient à zéro ou une structure**

**Un utilisateur possède zéro ou plusieurs fonctions, une fonction peut appartenir à zéro ou plusieurs utilisateurs.**

Un utilisateur possède zéro ou plusieurs rôles et un rôle peut appartenir à zéro plusieurs utilisateurs.

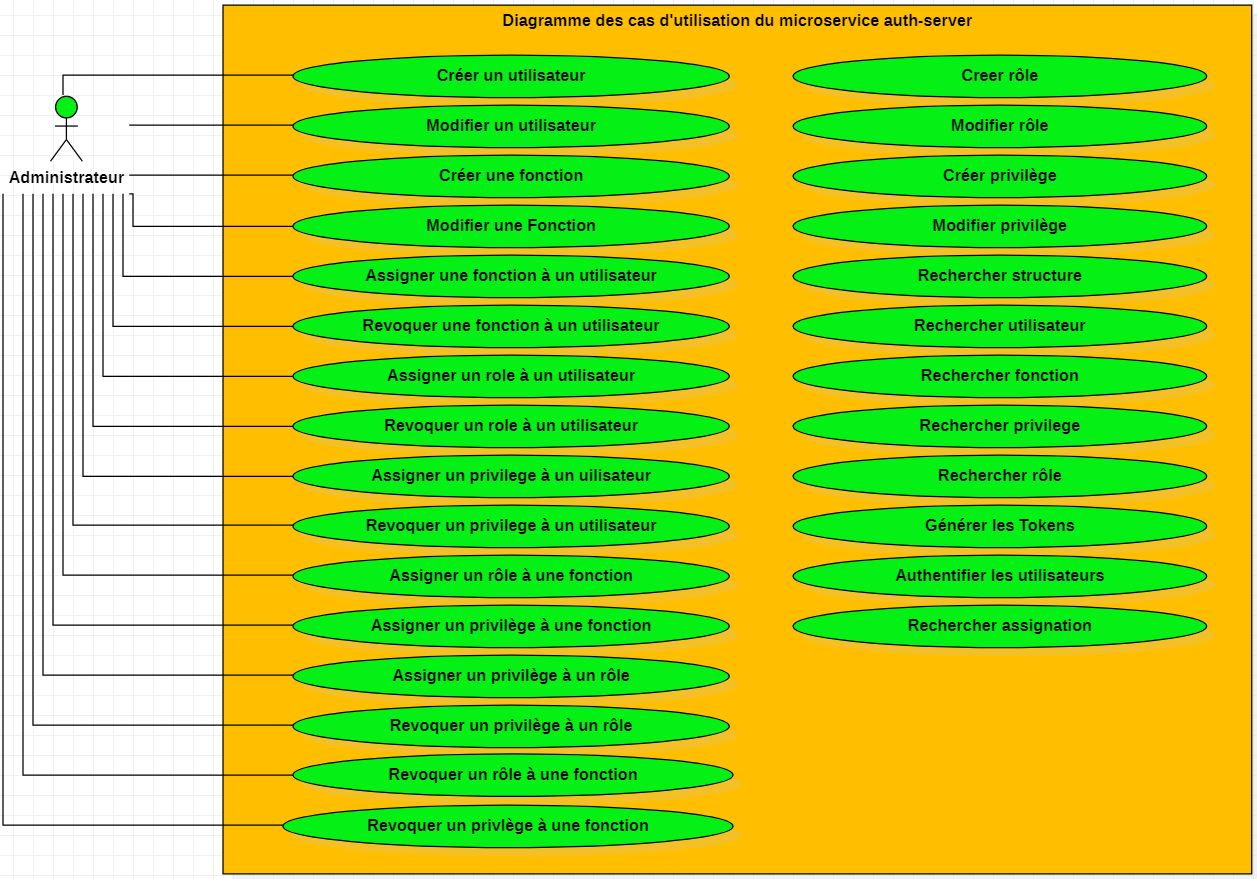
Un utilisateur possède zéro ou plusieurs privilèges, un privilège peut appartenir à zéro ou plusieurs utilisateurs.

**Une fonction possède zéro ou plusieurs rôles et un rôle peut appartenir à zéro ou plusieurs fonctions.**

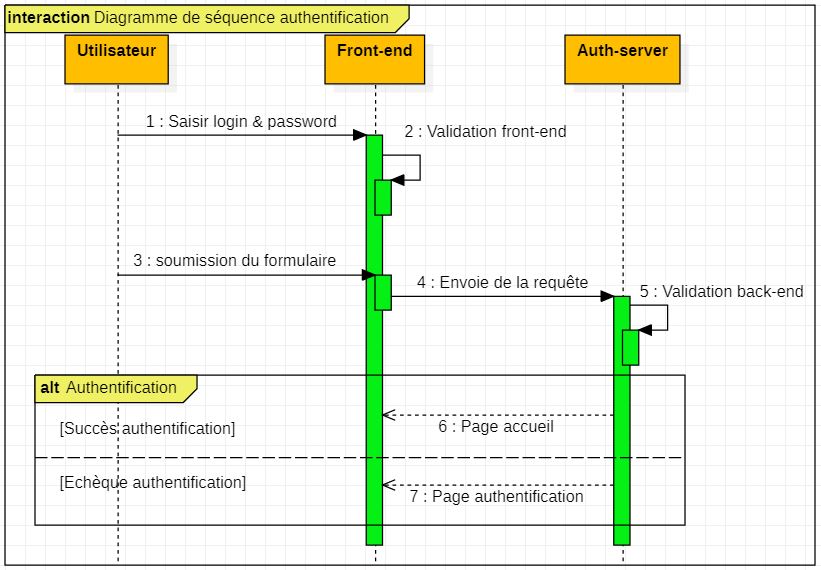
Une fonction possède zéro ou plusieurs privilèges et un privilège peut appartenir à zéro ou plusieurs fonctions.

**Un rôle possède zéro ou plusieurs privilèges et un privilège peut appartenir à zéro ou plusieurs rôles.**

* **Diagramme des cas d’utilisation**

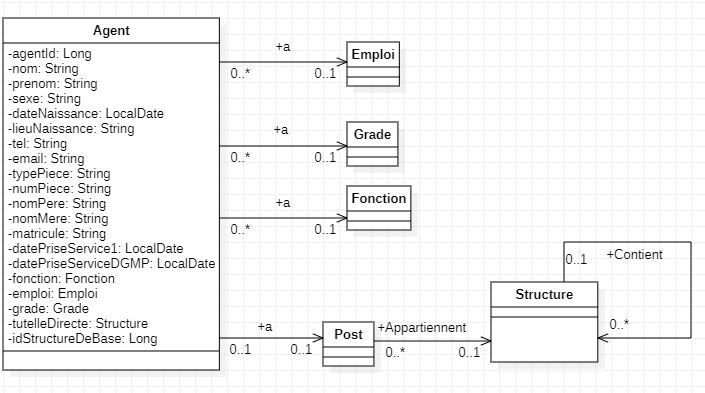


* **Diagramme de séquence**



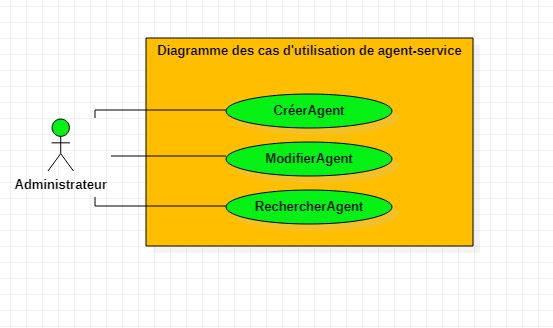
**Diagramme de séquence authentification**

* + - 1. Agent-service
* **Diagramme de classes**



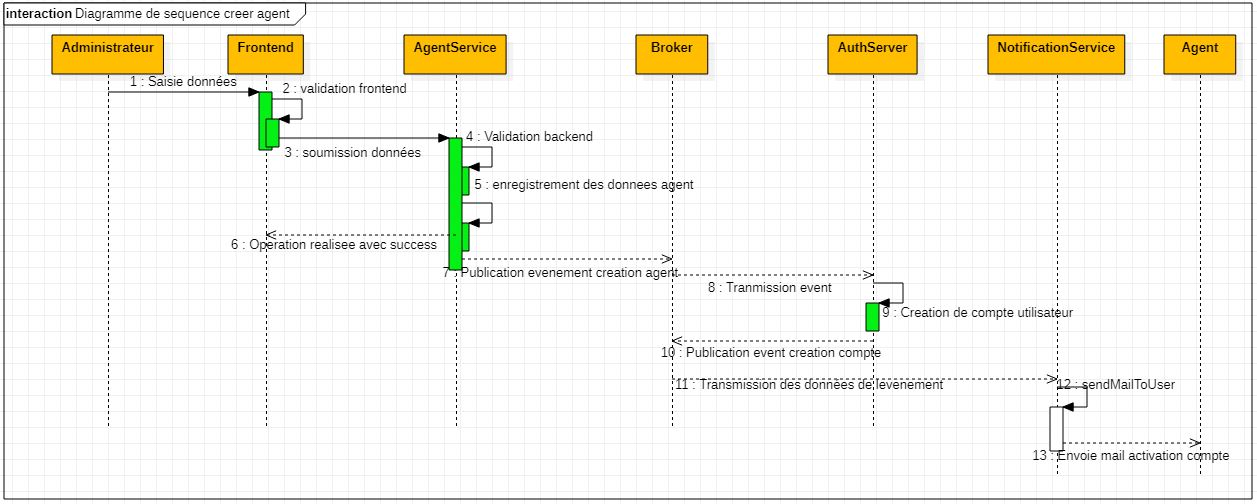
**Diagramme de classe du micro service Agent-service**

* **Diagramme des cas d’utilisation**



**Diagramme des cas d’utilisation du micro service Agent-Service**

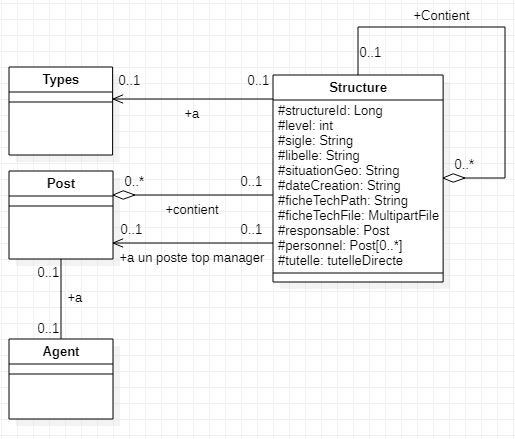
* **Diagramme de séquence**



**Diagramme de séquence du cas d’utilisation créer un agent**

Commentaires :

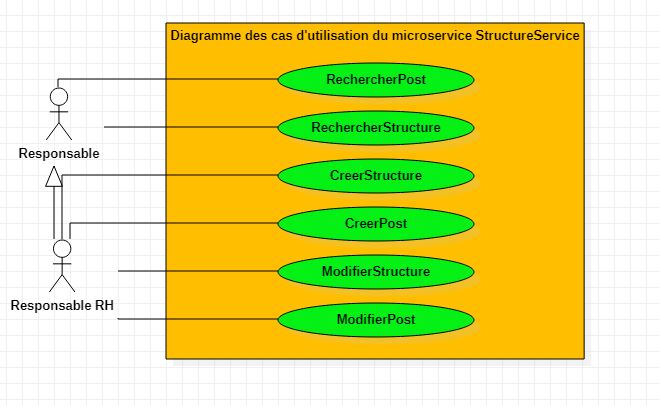
* + 1. L’utilisateur (administrateur) saisit les données relatives à l’agent qu’il souhaite enregistrer ainsi que les paramètres de compte utilisateur de cet agent ;
    2. L’application frontend valide les données saisies au fur et à mesure de la saisie ;
    3. L’administrateur soumet les données saisies vers le backend (micro service agent-service) ;
    4. Le backend valide les données reçues du frontend ;
    5. Le micro service agent-service enregistre les données de l’agent dans sa base de données ;
    6. Le micro service agent-service retourne un message de succès de l’opérateur à l’utilisateur ;
    7. Le micro service agent-service publie l’événement de création de l’agent sur une file d’attente du broker ;
    8. Le broker pousse les données de l’événement vers le micro service d’authentification (auth-server) ;
    9. Le micro service d’authentification crée un compte utilisateur pour le nouvel agent à partir des données de l’événement ;
    10. Le micro service auth-server publie l’événement de création du compte sur une file d’attente du broker ;
    11. Le broker pousse l’événement vers le micro service notification-service
    12. Le micro service notification-service appelle sa méthode sendMailToUser en lui passant en paramètre l’adresse mail de l’agent contenue dans les données de l’événement ;
    13. Un mail de d’activation de compte est envoyé à l’agent.
        1. Structure-service
* **Diagramme de classes**



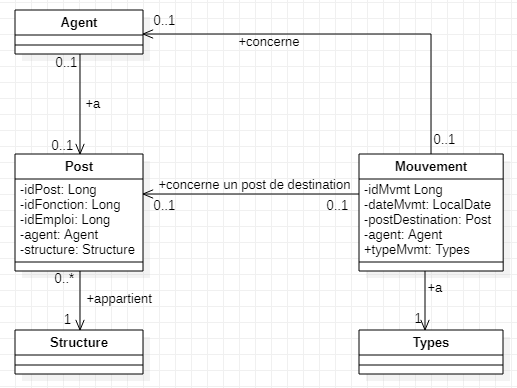
**Diagramme de classes du micro service Structure-service**

Commentaires :

* Une structure contient plusieurs structures sous sa tutelle. Elle a cependant zéro ou une structure parente ;
* Une structure a un type ;
* Une structure contient plusieurs post. Dont un post pour le responsable (top manager) ;
* Chaque post est occupé par un agent.
* **Diagramme des cas d’utilisation**



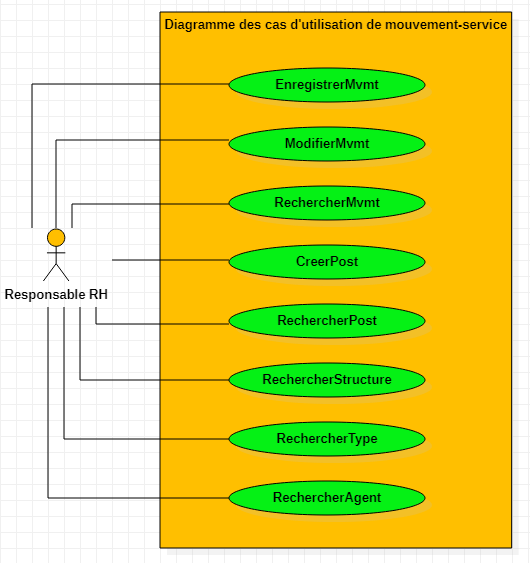
* **Diagramme de séquence**
  + - 1. Mouvement-service
* **Diagramme de classes**



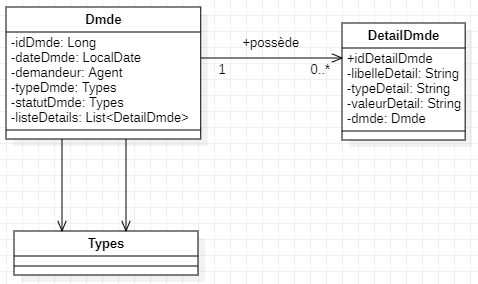
**Diagramme de classes du micro service Mouvement-service**

**Commentaires :**

* Un mouvement concerne un agent et post de destination ;
* Un mouvement a un type ;
* Un agent occupe un post ;
* Un post appartient à une structure ;
* **Diagramme des cas d’utilisation**

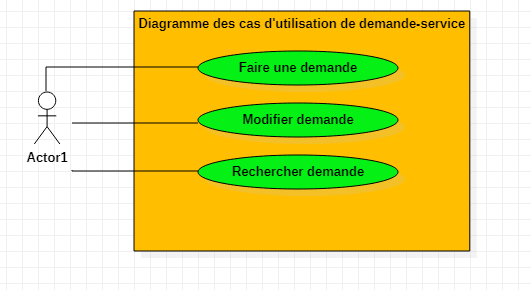


* + - 1. Demande-service
* **Diagramme de classes**



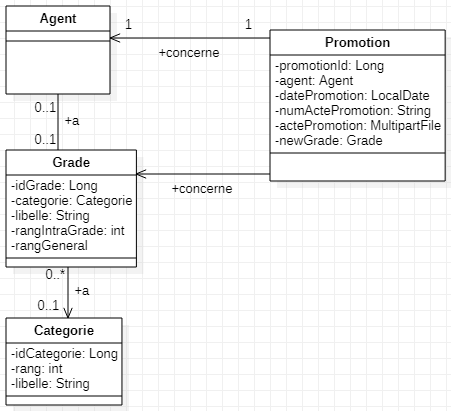
**Diagramme de classe du micro service demande-service**

* **Diagramme des cas d’utilisation**



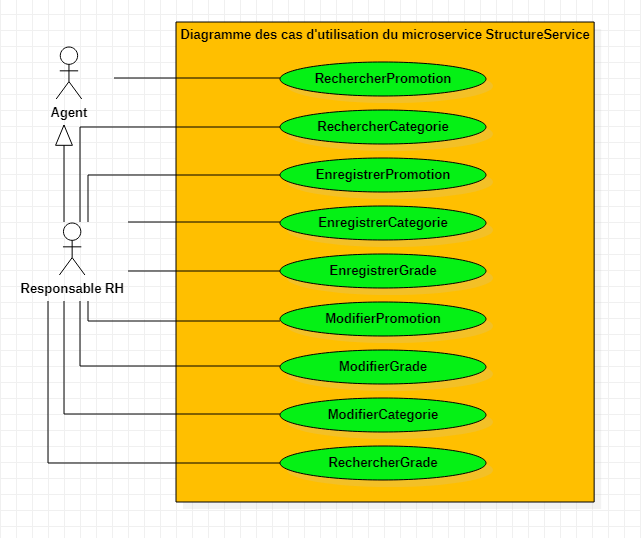
**Diagramme des cas d’utilisation du micro service demande-service**

* + - 1. Promotion-service
* **Diagramme de classes**

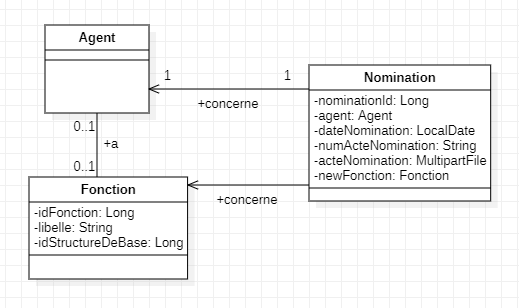


**Diagramme de classes du micro service promotion service**

* **Diagramme des cas d’utilisation**

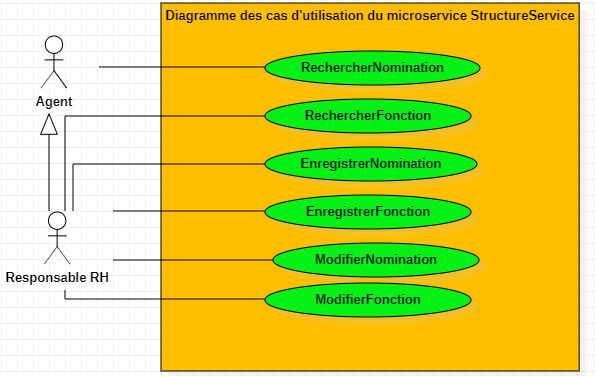


* + - 1. Nomination-service
* **Diagramme de classes**

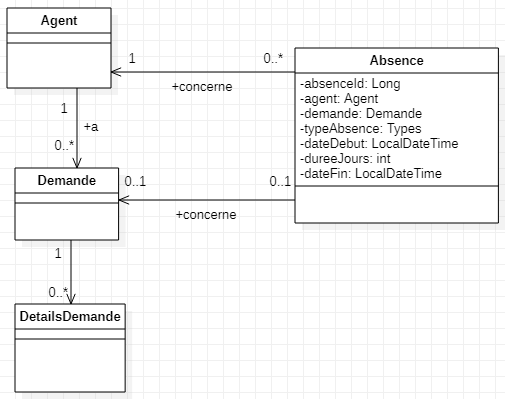


**Diagramme de classe du micro service nomination-service**

* **Diagramme des cas d’utilisation**

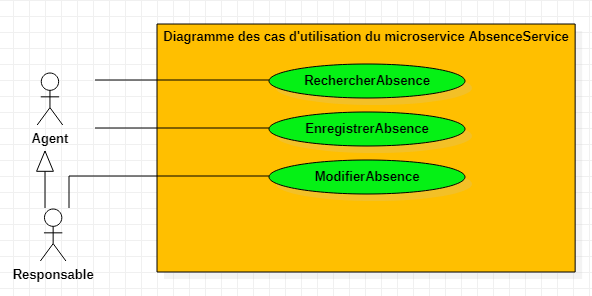


* + - 1. Absence-service
* **Diagramme de classes**

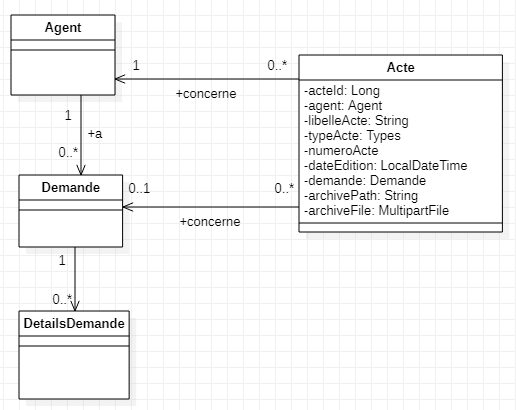


**Diagramme de classe du micro service Absence-service**

* **Diagramme des cas d’utilisation**

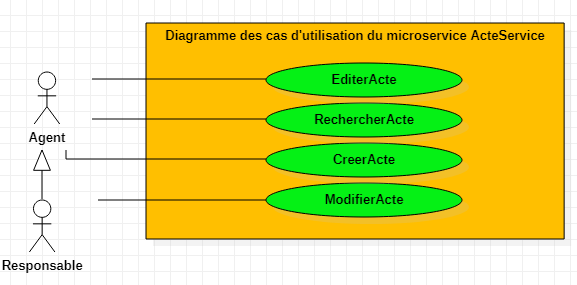


* + - 1. Actes-service
* **Diagramme de classes**

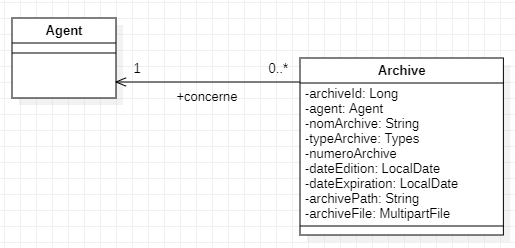


**Diagramme de classes du micro service acte-service**

* **Diagramme des cas d’utilisation**

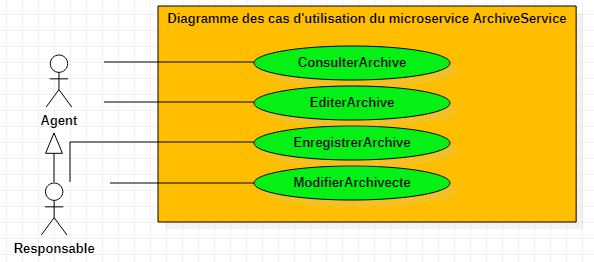


* **Diagramme de séquence**
  + - 1. Archivage-service
* **Diagramme de classes**

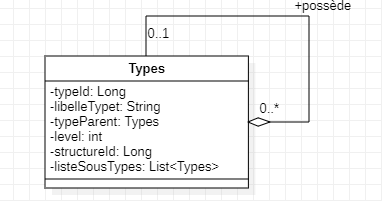


**Diagramme de classes du micro service archive-service**

* **Diagramme des cas d’utilisation**

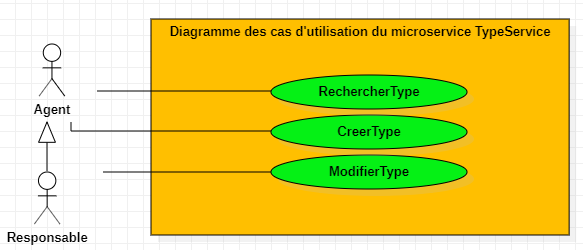


* + - 1. Type-service
* **Diagramme de classes**



**Diagramme de classes du micro service type-service**

* **Diagramme des cas d’utilisation**



* + - 1. Notification-service

Mise en exploitation

La mise en exploitation du système fait référence à un processus prenant en compte un ensemble de tâches dont la réalisation vise à mettre à disposition le système dans les conditions optimale d’utilisation. Ce processus comprend plusieurs phases pouvant varier selon les besoins du projet. Dans le cadre de ce projet, la mise en exploitation va consister en :

* La réalisation des tests unitaires et d’intégration ;
* Le packaging (génération des archives jar et de l’application frontend) ;
* Le Déploiement sur les environnement intermédiaires (environnement de développement et de test) ;
* La réalisation des tests fonctionnels ;
* Le déploiement en environnement de production.

**2.1 Test unitaires et d’intégration**

Les tests unitaires et d’intégration sont des scripts écrits par le développeur pendant la phase de développement de l’application. Ce sont des tests automatiques faciles à exécuter mais dont la pertinence dépend fortement de la qualité des scripts écrits par le développeur.

Les tests unitaires sont de très bas niveau, directement liés au code source de l’application. Ils consistent à tester isolément les méthodes des classes, des composants ou des modules développés au sein de l’application. Les tests unitaires sont généralement peu couteux car une fois développés, ils peuvent être exécutés autant de fois qu’on le souhaite. En outre, leur temps d’exécution est assez court (de l’ordre de la seconde). Cependant, il s’avère parfois nécessaire de développer un nombre impressionnant de tests unitaires si l’on veut …

Les tests d'intégration quant à eux visent à vérifier que les fonctions, services ou modules développés isolément, fonctionnent bien et de façon cohérente ensemble. Ils assurent que les différents blocs du système s’intègrent les uns avec les autres. Pour exemple, un test d’intégration peut servir à tester la communication avec une base de données ou s'assurer que différents microservices fonctionnent ensemble comme prévu. Ces types de tests sont plus coûteux à exécuter, car ils nécessitent que plusieurs parties de l'application soient fonctionnelles.

Dans le cadre de ce projet, il sera développé, pour chaque microservice un ensemble de tests unitaires afin de tester la quasi-totalité des fonctions s’y trouvant. Par ailleurs, afin de s’assurer que la communication entre microservices est toujours fonctionnelle, nous développeront des tests d’intégration pour chaque fonctionnalité au sein d’un microservice faisant appel à d’autres microservices.

Une fois développés, les tests unitaires et d’intégration peuvent être exécuté par le moyen de ligne de commande ou par l’intermédiaire d’un IDE adapté. Pour les projets utilisant Maven comme gestionnaire de dépendances, c’est commande **mvn test** qui est utilisée à cet effet.

**2.2 Packaging**

Le packaging des applications est un processus qui consiste à compiler le code source du projet en bytecode, à construire l’application et à en générer une archive dans un format permettant de faciliter sa distribution et son déploiement. Selon la technologie de développement utilisée pour le projet, différents outils peuvent être employer pour packager l’application.

En général, les projets backend développés en java utilisent Graddle ou Maven tandis que les projet frontend écrit en Javascript utilise Yarn ou NPM pour le packaging. Dans le cadre de ce projet, nous utiliserons Maven pour les microservices (backend) écrits en java et NPM pour l’application frontend. Pour ce faire, nous exécuterons la commande **mvn package** pour chaque microservice et la commande **npm run build** pourl’application frontend.

2.3 Déploiement

COUT DE LA SOLUTION

La réalisation de ce projet a engendré des coûts. Les coûts liés à ce projet sont résumés dans le tableau suivant :

|  |  |
| --- | --- |
| **Coûts de la solution** | |
| **Libellé** | **Montant** |
| Achat d’un ordinateur portable | 500 000 FCFA |
| Achat de forfait internet | 200 000 FCFA |
| Paiement de cours en ligne | 300 000 FCFA |
| Temps consacré à la recherche et au développement : 12 mois (10 mois de recherche, 2 mois de développement) | 1 000 000 FCFA |
| Total | 2 000 000 FCFA |

# CONCLUSION

Ce projet qui s’inscrit dans le cadre de la mise en place d’une plateforme de gestion du personnel pour le compte de la Direction Générale des Marchés Publics avait pour objectif premier d’automatiser les principales tâches de la gestion administrative du personnel. Ce projet avait également pour objectif de mettre à disposition une base de données fiable afin de fluidifier et de sécuriser les accès aux données sur personnel. Il a été surtout question d’élaborer un tableau de bord permettant de visualiser en temps réel les statistiques sur le personnel.

Pour répondre à cette problématique, nous avons choisi de développer une application web permettant de gérer les mouvements du personnel, les demandes d’actes administratifs, les demandes d’absences et de congés ainsi que la création la modification d’unités administratives. Cette application dispose par ailleurs d’un module statistique.

~~La conception de cette application a été réalisée à l’aide du Langage de Modélisation Unifié (UML). Nous avons utilisé le SGBDR Oracle Database pour le stockage des données. L’implémentation de la solution a été réalisée avec la spécification JEE du langage de programmation JAVA (JAVA 8). Nous avons en outre utilisé le Framework Spring Boot pour l’inversion de contrôle et l’injection des dépendances. Les vues de l’application ont été généré coté serveur avec le moteur de Template Thymeleaf. Les librairies JQuery et Bootstrap nous ont permis de faire les mises en forme (CSS) et gérer l’interactivité (JavaScript) des pages.~~

# REFERENCES BIBLIOGRAPHIQUES

[**"Thinking in Java"**](http://www.eckelobjects.com/javabook.html)Bruce Eckel (Prentice-Hall 1998, ISBN: 0-13-659723-80)

# WEBBOGRAPHIE

https://openclassrooms.com/

<https://www.alphorm.com/>

<https://www.youtube.com/user/mohamedYoussfi/playlists>

<https://www.youtube.com/channel/UCzuaB4F2znrMggxcwUuVhAw>

<https://www.youtube.com/channel/UC0FIKAE7xORm6Y91kWbg4cA>

<https://www.youtube.com/channel/UCj_iGliGCkLcHSZ8eqVNPDQ>

<https://www.youtube.com/channel/UCORuRdpN2QTCKnsuEaeK-kQ>

<https://www.youtube.com/channel/UCcHGNIgYpYOheI7oPWYd9Mw>

# TABLE DES MATIERES

[DEDICACES](#_Toc68766668)

[DEDICACE I](#_Toc68766669)

[EMERCIEMENTS II](#_Toc68766670)

[PRESENTATION DE L’ECOLE III](#_Toc68766671)

[SOMMAIRE 0](#_Toc68766672)

[RESUME 1](#_Toc68766673)

[INTRODUCTION 2](#_Toc68766674)

[PARTIE 1 : CADRE DE REFERENCE 3](#_Toc68766675)

[1. PRESENTATION DE LA STRUCTURE D'ACCUEIL 4](#_Toc68766676)

[2. ETUDE DE L’EXISTANT 6](#_Toc68766677)

[2.1. Description général de l’existant 6](#_Toc68766678)

[2.2. Description de l’existant par activités 6](#_Toc68766679)

[3. Critiques, Problématique et Solutions 11](#_Toc68766680)

[3.1. Critique de l'existant 11](#_Toc68766682)

[3.2. Problématique 12](#_Toc68766683)

[3.3. Solutions 13](#_Toc68766684)

[4. Cahier des charges et choix de la solution 14](#_Toc68766685)

[4.1. Cahier des charges 14](#_Toc68766687)

[4.2. Choix de la solution 15](#_Toc68766688)

[PARTIE II : ETUDE DE LA SOLUTION 17](#_Toc68766689)

[1. Méthodes d'analyse 17](#_Toc68766690)

[1.1. Méthodes d'analyse systémique 17](#_Toc68766691)

[1.2. Méthodes d'analyses cartésiennes ou fonctionnelles 21](#_Toc68766692)

[1.3. Méthodes d'analyse objet 21](#_Toc68766693)

[1.4. Choix de la méthode d'analyse 25](#_Toc68766694)

[2. Etudes conceptuelles 25](#_Toc68766695)

[2.1. Identification des acteurs du système 25](#_Toc68766698)

[2.2. Diagrammes des cas d'utilisation 26](#_Toc68766699)

[2.3. Diagramme de classes 29](#_Toc68766700)

[2.4. Diagrammes de séquence 33](#_Toc68766701)

[3. Etudes techniques 43](#_Toc68766702)

[3.1. Choix du Système de Gestion de Base de Données (SGBD) **Erreur ! Signet non défini.**](#_Toc68766704)

[3.2. Architectures de déploiement 43](#_Toc68766705)

[3.3. Choix d'une architecture **Erreur ! Signet non défini.**](#_Toc68766706)

[4. Planning de Réalisation **Erreur ! Signet non défini.**](#_Toc68766707)

[PARTIE III : REALISATION DE LA SOLUTION RETENUE 58](#_Toc68766708)

[1. Réalisation Technique **Erreur ! Signet non défini.**](#_Toc68766709)

[1.1. La couche de sécurité **Erreur ! Signet non défini.**](#_Toc68766710)

[1.2. La couche Contrôleur (Web) **Erreur ! Signet non défini.**](#_Toc68766711)

[1.3. La couche Métier (Services) **Erreur ! Signet non défini.**](#_Toc68766712)

[1.4. La couche d’Accès aux données (DAO) **Erreur ! Signet non défini.**](#_Toc68766713)

[2. Mise en exploitation 63](#_Toc68766714)

[3. Tests **Erreur ! Signet non défini.**](#_Toc68766715)

[**2.** **Erreur ! Signet non défini.**](#_Toc68766716)

[3.1. Formulaire d’authentification **Erreur ! Signet non défini.**](#_Toc68766718)

[3.2. Liste du personnel **Erreur ! Signet non défini.**](#_Toc68766719)

[3.3. Formulaire d’enregistrement d’un agent **Erreur ! Signet non défini.**](#_Toc68766720)

[4. COUT DE LA SOLUTION 64](#_Toc68766721)

[CONCLUSION 64](#_Toc68766722)

[REFERENCES BIBLIOGRAPHIQUES 65](#_Toc68766723)

[WEBBOGRAPHIE 65](#_Toc68766724)

[TABLE DES MATIERES 66](#_Toc68766725)