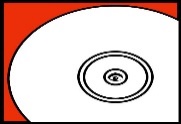
**République de Côte d’Ivoire**

**Union – Discipline – Travail**

**Ministère de l’Enseignement**

**Supérieur et de la Recherche Scientifique**





**CEFIVE**

**MEMOIR DE FIN DE CYCLE**

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

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

**ETUDE ET MISE EN PLACE D’UNE PLATEFORME POUR LA GESTION ADMINISTRATIVE DU PERSONNEL DE LA DIRECTION GENERALE DES MARCHES PUBLICS (DGMP)**

**Maître de stage:**

**M. DJAN DJAN OLIVIER**

Réalisé par : **COULIBALY LENIMAMAN IBRAHIMA, Elève Ingénieur des Techniques Informatiques au CEFIVE**

**Maître de stage:**

**M. DJAN DJAN OLIVIER**

# DEDICACES

**Directeur de projet :**

**M. AFFIAN ROGER, Directeur des Etudes au CEFIVE**

**Maître de stage :**

**M. DJAN OLIVIER, Chef de Service à la DGMP**

A ma famille

# REMERCIEMENTS

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 mémoire.

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 ainsi que pour les bénédictions avec lesquelles ils m’accompagnent dans toutes mes entreprises.

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

Table des matières

[DEDICACES 1](#_Toc108892750)

[REMERCIEMENTS 3](#_Toc108892751)

[PRESENTATION DE L’ECOLE 4](#_Toc108892752)

[SOMMAIRE 6](#_Toc108892753)

[RESUME 8](#_Toc108892754)

[INTRODUCTION 9](#_Toc108892755)

[PARTI I : CADRE DE REFERENCE 10](#_Toc108892756)

[1.1 Présentation de la structure d’accueil 11](#_Toc108892757)

[1.2 Etude de l’existant 13](#_Toc108892758)

[1.2.1 Description générale de l’existant 13](#_Toc108892759)

[1.2.2 Description de l’existant par activités 13](#_Toc108892760)

[1.3 Critiques, Problématique et Solutions 18](#_Toc108892761)

[1.3.1 Critique de l'existant 18](#_Toc108892763)

[1.3.2 Problématique 20](#_Toc108892764)

[1.3.3 Solutions 20](#_Toc108892765)

[1.4 Cahier des charges et choix de la solution 22](#_Toc108892766)

[1.4.1 Cahier des charges 22](#_Toc108892768)

[1.4.2 Choix de la solution 23](#_Toc108892769)

[PARTIE II : ETUDE DE LA SOLUTION 25](#_Toc108892770)

[2.1 Méthodes d'analyse 25](#_Toc108892773)

[2.1.1 MERISE 25](#_Toc108892774)

[2.1.2 UML 28](#_Toc108892775)

[2.1.3 Choix de la méthode d'analyse 29](#_Toc108892776)

[2.2 Etude conceptuelle 29](#_Toc108892777)

[2.2.1 Diagramme de contexte / Identification des acteurs du système 29](#_Toc108892778)

[2.2.2 Diagrammes des cas d'utilisation 30](#_Toc108892779)

[2.2.3 Description textuelle de certains cas d’utilisation 30](#_Toc108892780)

[2.2.4 Diagrammes des classes 34](#_Toc108892781)

[2.2.5 Diagrammes de séquence 39](#_Toc108892782)

[3.1.1 Traitement des demandes d’affectation interne (au sein de la DGMP) 44](#_Toc108892783)

[3.1.2 Traitement des demandes d’actes administratifs 45](#_Toc108892784)

[3.2 Etude technique 46](#_Toc108892785)

[3.2.1 Architectures 46](#_Toc108892786)

[3.2.2 Stratégie de communication entre micro services 49](#_Toc108892787)

[3.2.3 Technologies de développement 50](#_Toc108892788)

[PARTIE III : REALISATION DE LA SOLUTION 58](#_Toc108892789)

[3.1 Développement 59](#_Toc108892790)

[3.1.1 Développement frontend 59](#_Toc108892791)

[3.1.2 Développement backend 63](#_Toc108892792)

[3.2 Mise en exploitation 81](#_Toc108892793)

[3.2.1 Environnements de déploiement 81](#_Toc108892794)

[3.2.2 Cycle de déploiement 82](#_Toc108892795)

[1. COUT DE LA SOLUTION 83](#_Toc108892796)

[CONCLUSION 84](#_Toc108892797)

[REFERENCES BIBLIOGRAPHIQUES 85](#_Toc108892798)

[WEBOGRAPHIE 85](#_Toc108892799)

[TABLE DES MATIERES 86](#_Toc108892800)

[3.2.1 Test unitaires et d’intégration 87](#_Toc108892801)

[3.2.2 Packaging 87](#_Toc108892802)

[3.2.3 Déploiement en environnements intermédiaires (Test & Dev) 88](#_Toc108892803)

# SUMARY

Personnel management is a very important function in the functioning of an administration. The implementation of this function raises important challenges to be met. These challenges concern both efficiency and speed in the processing of files and the security of personnel data. Thus, without necessarily being the production activity of a structure, personnel management can prove to be very costly in terms of time, material, financial and human resources if we do not have adequate means. Therefore, to be carried out successfully, it is necessary to have tools allowing on the one hand to save resources and on the other hand to ensure the security and quality of the information handled.

This issue still available for the General Direction of the Public Markets (DGMP), whose personnel management system is essentially based on physical (paper) archiving of documents.

This thesis aims to provide a solution to this problem faced by the DGMP. To do this, we opted, after analysing and studying several solutions, for the implementation of a web application, to automate the management of personnel.

The implementation of this solution has two main aspects which are the frontend development which concerns the graphical interface and the backend development which concerns the functionalities at work in the background. The technologies used are respectively the React JS library for the front part and the Spring Boot Framework for the back part.

The architecture of the system is an event-driven microservice architecture (Event Driven Microservices).

The database management system used is PostGres SQL.

In addition to the deployment, we opted for the establishment of three environments which are the development, test and production environments.

Finally, the cost of the project calculated with the cocomo method is estimated at 40 million CFA francs.

# 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 Etude et mise en place d’une plateforme de gestion du personnel dans le cadre de notre mémoire 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.

# PARTI I : CADRE DE REFERENCE

## Présentation 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.

## 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 question.

#### 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

We found out three possible solutions to solve this problem:

* ERP for (Enterprise Resource Planning);
* Login to a SaaS (Service as a Software);
* Develop our own software;

## Cahier des charges et choix de la solution



### Cahier des charges

#### The theme

THEME: Etude et mise en place d’une plateforme automatisée de gestion du personnel : cas de la Direction Générale des Marchés Publics

#### Spécifications fonctionnelles

Les fonctionnalités de la gestion du personnel prises en compte par le projet sont les suivantes :

* **La gestion des agents** (Informations sur le personnel)
* **La gestion des mouvements** prenant en compte les entrées, les mouvements internes (affectations), les départs volontaires, la disponibilité, les retraites, les décès.
* **La gestion des Carrières** prenant en compte les promotions, les nominations, les avancements et la mobilité professionnelle ;
* **La gestion des absences** prenant en compte les autorisations d’absence, les congés annuels, les repos maladies, les congés de maternité et les absences injustifiées ;
* **La gestion des archives** prenant en compte les actes de nomination, de promotion, d’avancement, les pièces d’identités, les actes de naissance, de mariage, de décès, … ;
* **La gestion des demandes** qui prend en compte le suivi et le traitement des demandes d’acte (Certificat de travail, Attestation de présence au post), les demande d’absence, les demandes de mouvements et les besoins en personnel ;
* **Un tableau de bord statistique** donnant une vue d’ensemble sur les fonctions précitées.

NB : This project does not implement the payment functionality which is a feature already managed by the Payroll Directorate (Direction de la Solde)

### Choice of the solution

The following bord presents the score given to each of the three solutions listed above. To do that, we picked out six (6) criteria weighted according to their relative importance. The determination of these criteria and their weighting coefficients was made on the basis of a survey realised among the business actors (personnel of the service in charge of personnel management), the developers and the future users of the system. These criteria are as follows: cost, delivery time, ease of integration of the solution, exclusive holding of data by the DGMP, consumption of internal resources, high availability. For each of these criteria, a score is assigned to each solution on a scale ranging from 1 to 5. The overall score for each solution is obtained by summing the scores obtained on each criterion weighted by the weight of the criterion.

Thus, based on this principle, the “internal application” solution appears to be the most appropriate with an overall score of 390 against 290 for ERP and 250 for SAAS. Therefore, to respond to the problem posed, we have chosen to develop in-house a web application which will expose, through various graphical interfaces, the main functions of the administrative management of the personnel. This application will be based on a database that will store and secure personnel data. This application will have a "Dashboard" module which will present the statistics relating to the personnel.



# PART II: STUDY OF THE SOLUTION

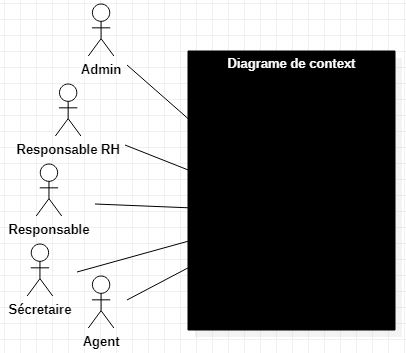


## Etude conceptuelle

### Context Diagram / Identification of the actors

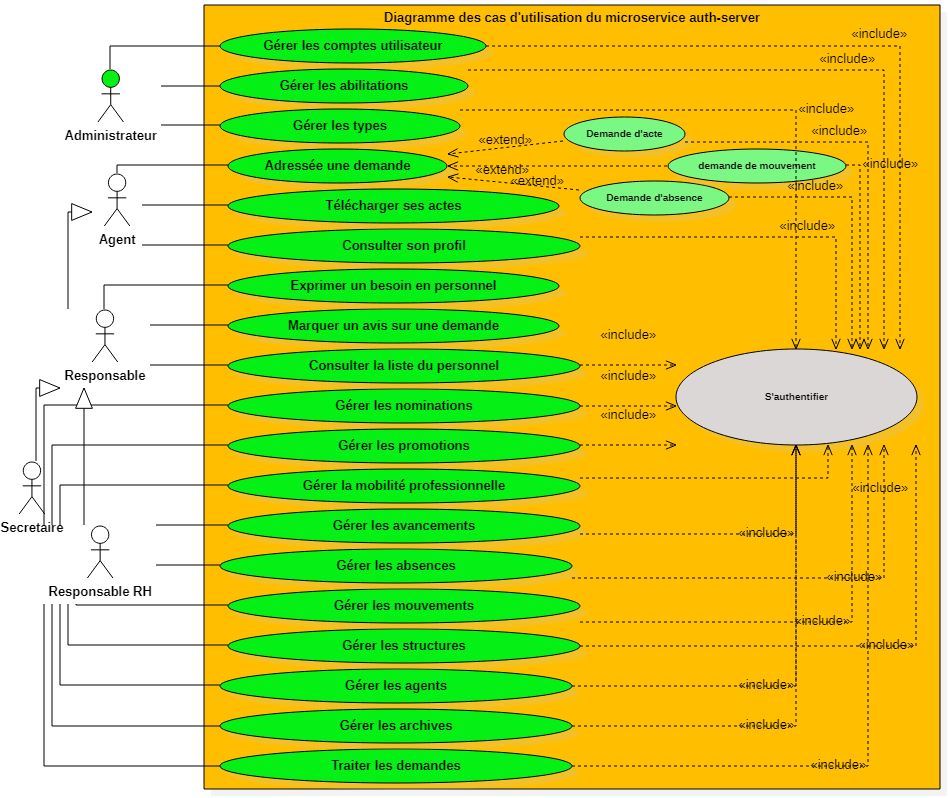
The actors of the system are :

* Administrator;
* Manager (Head of Department, Deputy Director, Regional Director, Director, General Manager, etc.):
* Secretary;
* HR Manager;
* Agent (In charge of studies, support agents, etc.);



### Use cases Diagram

Les **diagrammes de cas d'utilisation** sont des [diagrammes UML](https://fr.wikipedia.org/wiki/Unified_Modeling_Language" \l "Les_diagrammes" \o "Unified Modeling Language) 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).



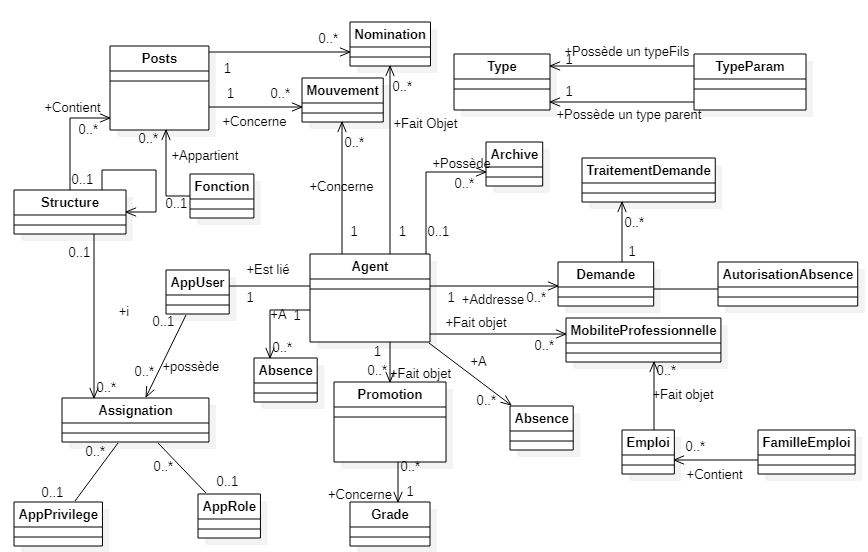
### Diagrammes des classes

Une [classe](https://fr.wikipedia.org/wiki/Classe_(informatique)" \o "Classe (informatique)) est un ensemble de [fonctions](https://fr.wikipedia.org/wiki/Fonction_(informatique)" \o "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" \o "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)" \o "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" \o "Génie 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" \o "Diagramme) fait partie de la partie [statique](https://fr.wikipedia.org/wiki/Statique" \o "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" \o "Dynamique).

#### Diagramme de classe du Système de Gestion du Personnel de la DGMP

**Diagramme de classes du système** 

**NB : Dans ce diagramme, nous mettons l’accent sur les liaisons sémantiques entre les classes. Ainsi, nous avons, expressément omis les attributs des classes en vue de rendre facile la lecture du diagramme.**

Ce diagramme est constitué de 23 classes liées entre elles par des relations d’association. Parmi elles, 19 sont les représentations des objets métiers du domaines étudiés tandis que 4 sont des objets techniques utilisés pour l’implémentation de la sécurité interne du système (Gestion de l’authentification, des autorisations et des niveaux d’accès). Les classes dites métiers ont pour objectifs de concourir à la mise en œuvre d’au moins une des fonctionnalités de la gestion administrative du personnel tel qu’indiqué dans les spécifications fonctionnelles du projet. Le tableau suivant présente la correspondance entre les classes du diagramme et les fonctionnalités du système dans lesquelles elles interviennent.

|  |  |  |
| --- | --- | --- |
| **Fonctionnalité** | **Type** | **Classes concernées** |
| Gestion des mouvements | Métier | Agent, Structure, Post, Mouvement, Type, TypeParam |
| Gestion des carrières | Métier | Agent, Promotion, Nomination, MobiliteProfessionnelle, Avancement, Post, Type, TypeParam |
| Gestion des absences | Métier | Agent, Absence, Archive, Type, TypeParam |
| Gestion des archives | Métier | Agent, Archive, Type, TypeParam |
| Gestion des demandes | Métier | Agent, Demande, TraitementDemande, Type, TypeParam |
| Tableau de bord | Métier | Toutes à l’exception de User, Rôle, Privilège et Assignation qui sont des classes techniques |
| Sécurité interne | Technique | User, Rôle, Privilège, Assignation, Structure |

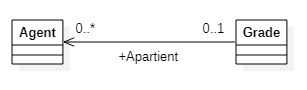
Ce diagramme de classes est le reflet de la future base de données sur laquelle tournera le système. En effet, les classes sont transformées en table et leurs attributs sont les colonnes des dites tables. Les relations d’association (agrégation et composition) sont matérialisées par le jeu de migration des clés primaires.

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 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.

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

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



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.

* **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 table de la classe fille et celle de 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 du point de vue de la performance. 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.

### 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’échangeant 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 qui 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.

#### D:\INFORMATIQUE\Cours\CEFIVE\Projet-Prof\CEFIVE-Rapport-ProjetProf\Rapport stage\models\staffadmin_authentication_sequence_dgrm2.JPGDiagramme de séquence « 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

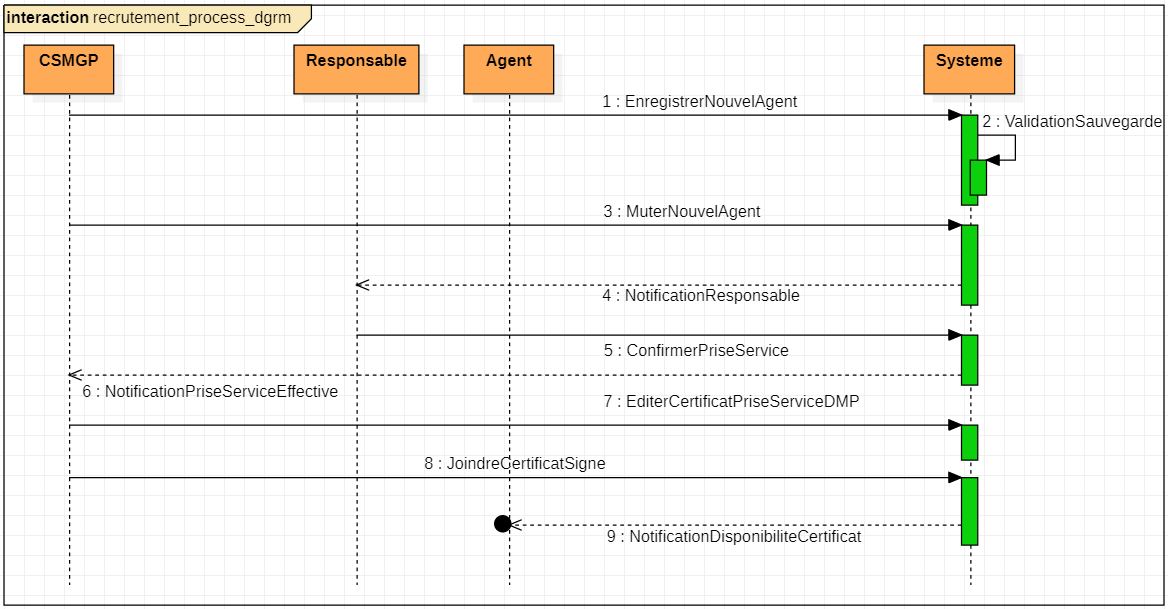
**4** : Le serveur d’application procède à la vérification du login et du mot de passe saisie par l’acteur.

**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.

#### Diagramme de séquence « Recrutement »



**Diagramme de séquence du processus de recrutement**

* **Commentaires :**

**1 :** Le Responsable RH 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 Responsable RH 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

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

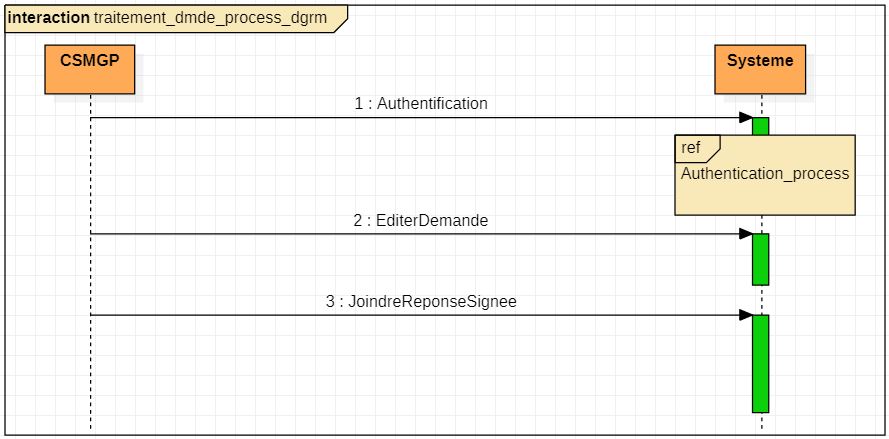
**6 :** Le système notifie le Responsable RH de la prise de service effective du nouvel agent

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

**8 :** Le Responsable RH 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.

#### Diagramme de séquence « traitement demande »



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

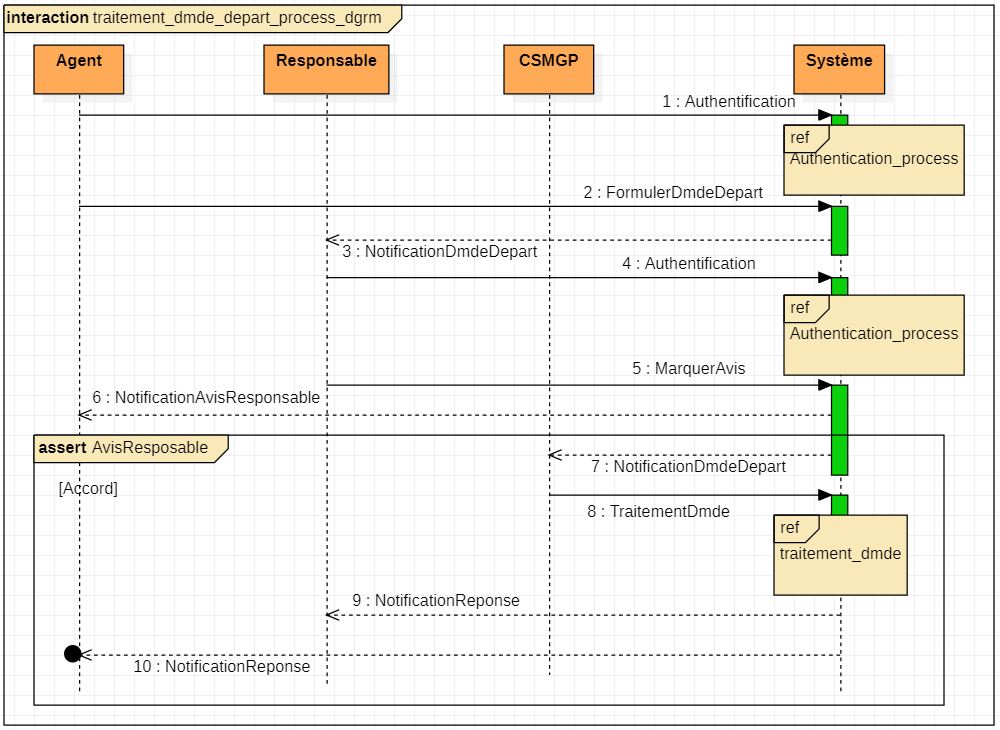
* **Commentaires :**

**1 :** Le Responsable RH s’authentifie

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

1. **:** Le Responsable RH joint la réponse signée par l’autorité compétente

#### Diagramme de séquence « traitement demande »



**Processus de 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 direct) 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.

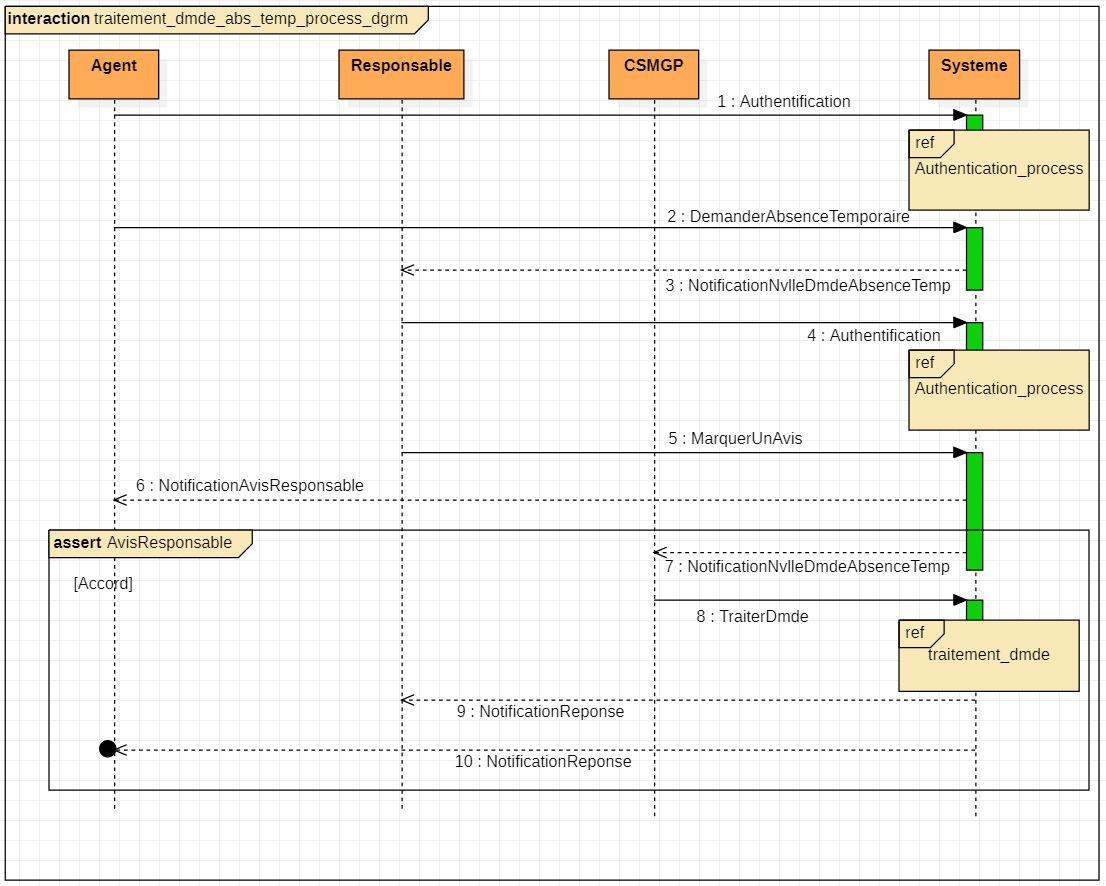
**7 :** Si l’avis du responsable est un accord, le système notifie le Responsable RH de la demande.

**8 :** Le Responsable RH 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.

#### 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.

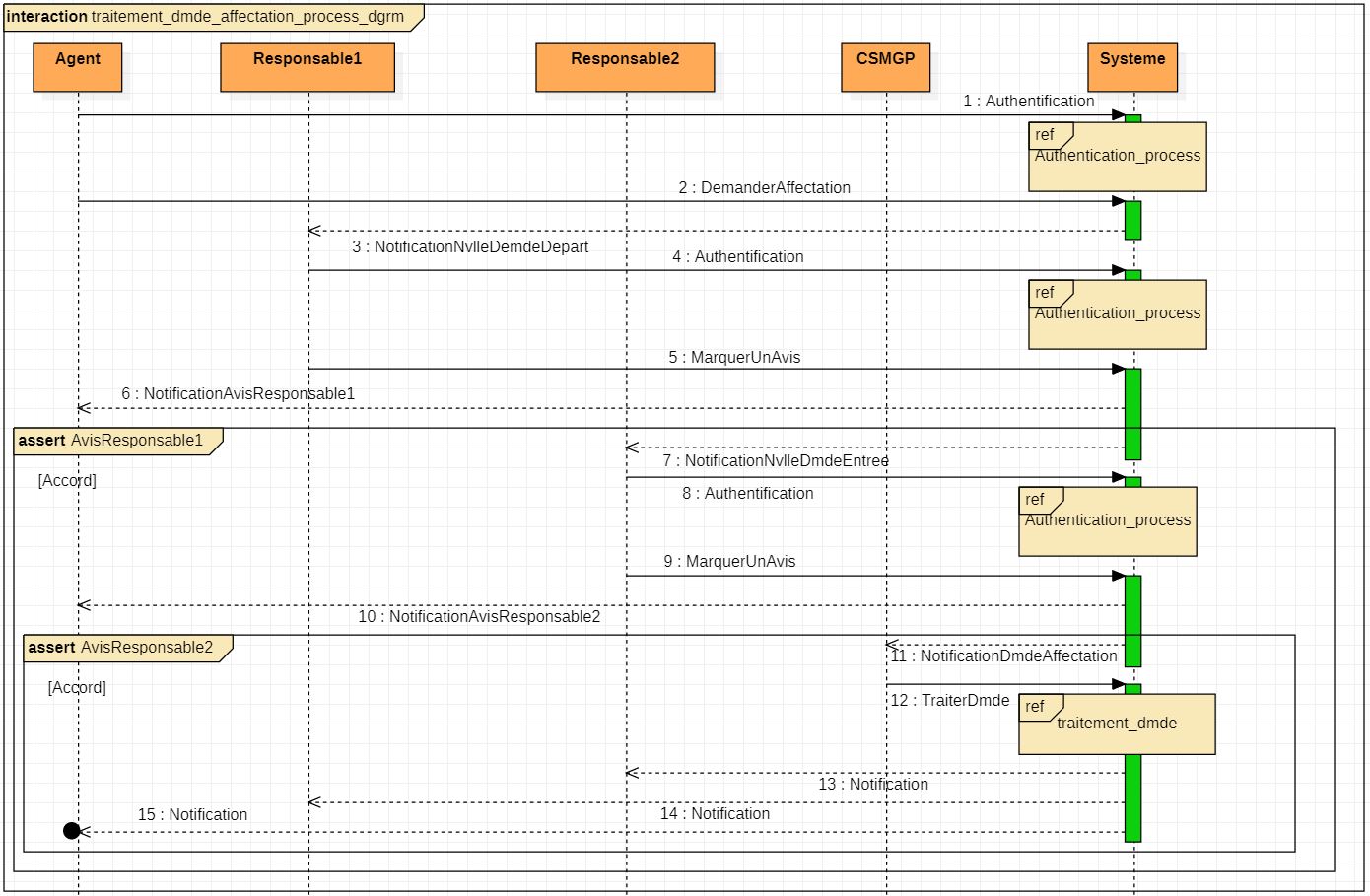
**7 :** Si l’avis du responsable est un accord, le système notifie le CSMGP de la demande.

**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.

### 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).

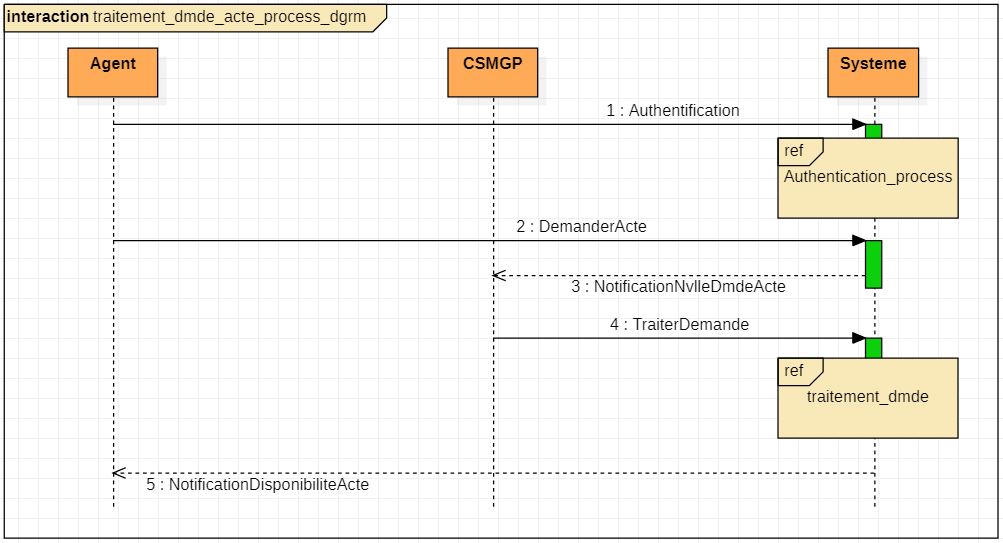
**13 :** Le système notifie le responsable du service de destination de la réponse du DGMP.

**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.

### 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).

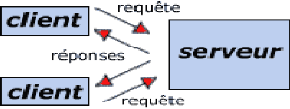
**5 :** Le système notifie l’agent de la disponibilité de l’acte demandé.

## Etude technique

### 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.

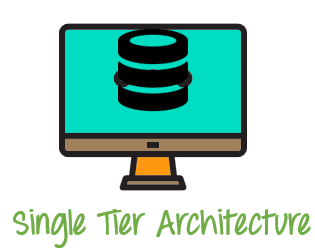


**Schéma de fonctionnement d'un système client/serveur**

* Le client émet une requête vers le serveur en contactant son adresse et le port sur lequel le service s’exécute ;
* Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port ;

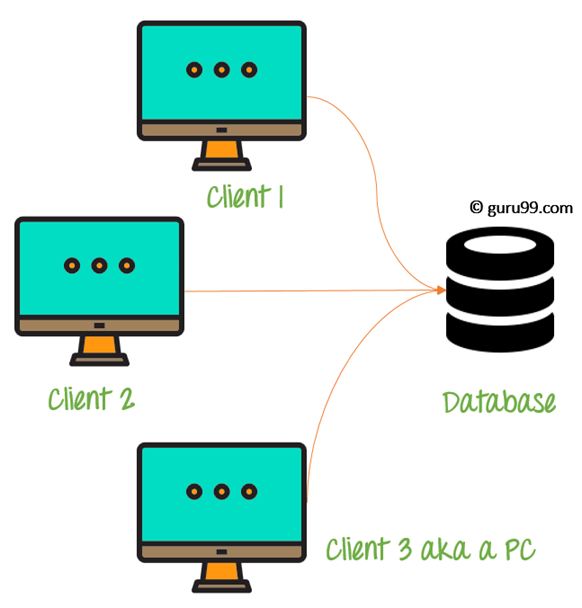
#### Architecture un tiers

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**

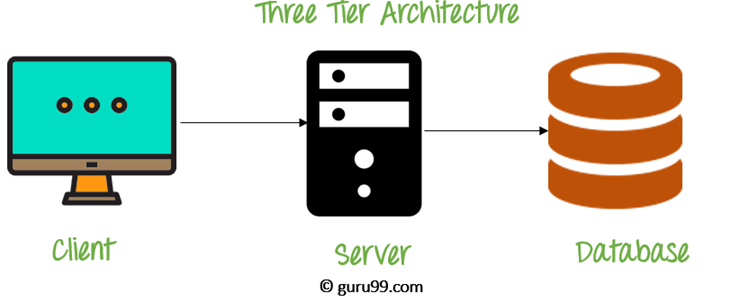
#### Architecture deux tiers



**Architectures 2-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.

#### Architecture trois tiers



**Architectures 3-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 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 ;

#### Architecture Micro service

Les micro services, également connus sous le nom d'architecture micro services, 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 micro service, 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 (micro service) 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 micro service au regard de ses inconvénients qui restent des problèmes dont les solutions techniques sont à portée de main.

* **Avantages**

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

Pour ce qui relève du développement, les micro services 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 micro services se caractérisent par des temps de réponse relativement faibles.

* **Inconvénients**

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

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

### Stratégie de communication entre micro services

Dans une architecture microservice, un ensemble de services relativement nombreux exécutent leurs fonctions de manière indépendante et interagissent les uns avec les autres de pour faire fonctionner l’application mère. La bonne marche de l’application mère implique donc la nécessité d’une communication cohérente et en temps voulu, entre les différents services. Ce problème de communication inexistant ou caché dans les architectures monolithiques, revêt une importance toute particulière dans les architectures microservices. En effet, c’est de la capacité à échanger des données entre les services que va dépendre l’agilité et l’évolutivité du système. Cela dit, la mise en place d’une communication efficiente au sein d’une architecture microservice impose un certain nombre de défis à relever tel que **La résilience et la tolérance aux pannes, L'équilibrage de charge** **ou load-balancing**, les **transactions distribuées**, la **découverte de services** **ou service discovery**.

En général, il existe deux principales manières de faire communiquer les micro services : La **communication synchrone**, et la **communication asynchrone**.

Dans une architecture microservice basé sur un système de communication synchrone, un service fait appel à une API exposée par un autre service en utilisant un protocole comme Rest ou gRPC (Remote Procedure Call). La communication est dite synchrone car le service appelant reste bloqué dans l’attendant de la réponse du service sollicité.

Par contre dans l’approche asynchrone, chaque service émet des messages sans attendre de réponses. Ainsi un ou plusieurs services souscripteurs peuvent consommer des messages émis sans avoir besoin d’en connaître les émetteurs.

Dans le cadre de ce projet, nous utiliserons un mode de communication asynchrone basé sur la diffusion d’événements pour faire communiquer les microservices tandis que la communication synchrone basée sur le protocole Rest nous permettra de faire la liaison entre l’application frontend et le backend constitué des microservices.

### Technologies de développement

Techniquement, le développement des applications web consiste en l’écriture de plusieurs lignes de code qui sont exécutés par un plusieurs serveurs en vue de générer des pages web. Ces pages sont ensuite transmises à des navigateurs qui les interprètent afin de produire les éléments d’affichage graphique (images, couleurs, tableaux, formulaires, …).

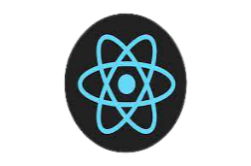
Cela dit, les activités de développement sont assez fastidieuses et couteuses en temps. De ce fait, il s’avère indispensable d’utiliser des assistants permettant de réduire la quantité de code requise et optimiser le processus de développement. Ces assistants peuvent être des Framework, des bibliothèques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

#### 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 utiliserons 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. |

#### Technologies Backend

Dans les architectures logicielles, 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 la mise en œuvre, …). Dans une architecture distribuée, notamment micro service, chaque micro service 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" \o "Langage de programmation) [orienté objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet" \o "Programmation orientée 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" \o "Compilateur Java) vers une [représentation binaire](https://fr.wikipedia.org/wiki/Syst%C3%A8me_binaire" \o "Système 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" \o "Système d'exploitation) |
| **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**  **D:\INFORMATIQUE\Cours\CEFIVE\Projet-Prof\CEFIVE-Rapport-ProjetProf\Rapport stage\Rédaction\icons\spring-security-removebg-preview.pngSpring 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. |
| **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. |
|  | 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 applications 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 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" \o "Système temps réel) à [latence](https://fr.wikipedia.org/wiki/Latence_(informatique)" \o "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 micro services. Plus particulièrement, il peut être utilisé dans les architectures micro services pour résoudre le problème des transactions distribuées. Dans ce projet, nous utiliserons kafka pour mettre en place système event sourcing en vue d’historiser et retracer toutes les actions d’écritures (Insertion, Modification, suppression) dans le système. |
| **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. |

#### 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 Data base, PostgreSQL, MS SQL Server, MySQL, …

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

* Oracle Data base 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/" \t "_blank)** . |
| **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" \o "Système de gestion de base de données) [relationnelles](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle" \o "Base de données 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. |

# PARTIE III : REALISATION DE LA SOLUTION

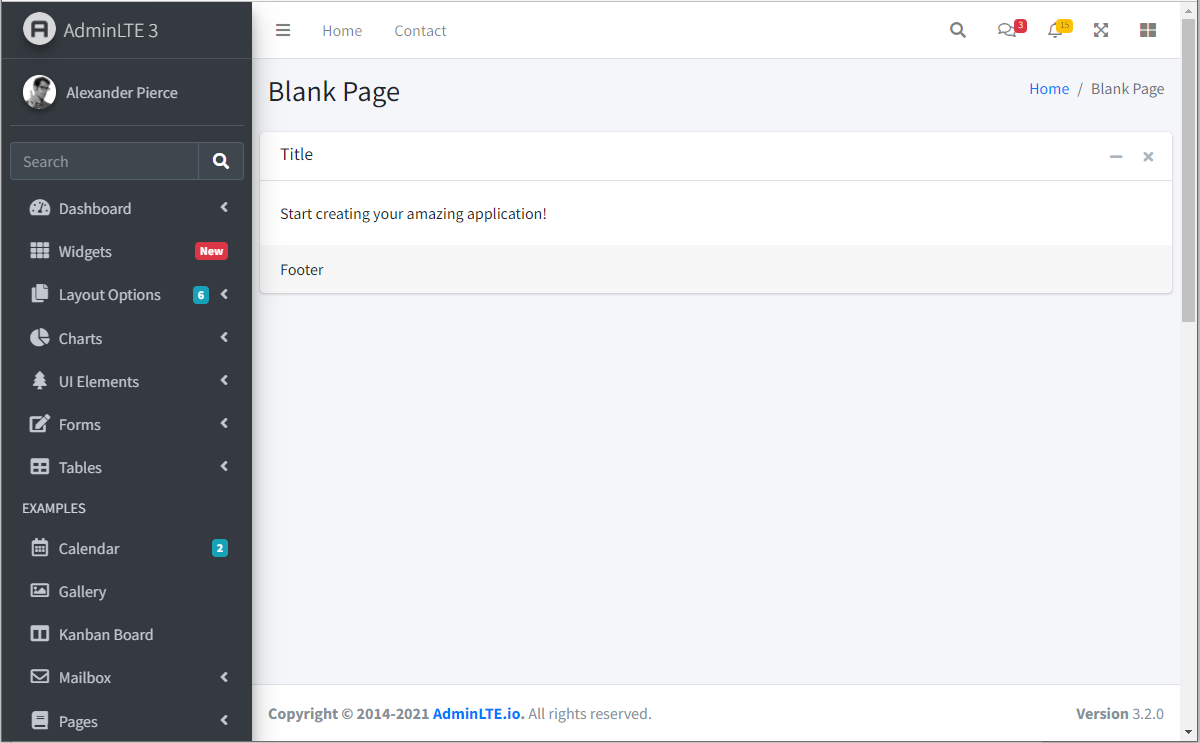
## Développement

### Développement frontend

#### 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, …).



#### 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 ;   + 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 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 |

### Développement backend

#### Mode de fonctionnement

Chaque microservice possède une base de données dédiée. La base de données d’un microservice est constituée de deux types de table. On distingue les tables principales et les tables secondaires. Les tables principales d’un microservice sont ses tables propres. C’est-à-dire des tables sur lesquelles ledit microservice possède les droits de lecture et d’écriture.

Les tables dites secondaires sont des copies de tables appartenant à d’autres microservices. Un microservice possédant des tables secondaires, maintient celles-ci à jour en consommant les événements d’écriture sur ses tables émis par leurs microservices propriétaires. Plus précisément, lorsqu’un microservice réalise une action d’écriture sur l’une de ses tables principales, il émet un événement d’écriture sur une file d’attente. Ainsi les microservices ayant, en leur sein une copie de cette table, pourront consommer l’événement et mettre à jour leur copie.

#### Identification des modules

Le backend de l’application sera constitué d’un ensemble de micro services pouvant échanger des données les uns avec les autres. Le tableau ci-dessous présente les différents micro services 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 et de découverte de tous les micro services |
| GATEWAY | Service de loadbalancing, permettant d’assurer l’aiguillage des requêtes vers les micro services et de faire l’équilibrage des charges. |
| CONFIG-SERVICE | Service permettant de centraliser la configuration des micro services. |
| Sécurité | Gestion des accès | 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) |

## 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 optimales d’utilisation.

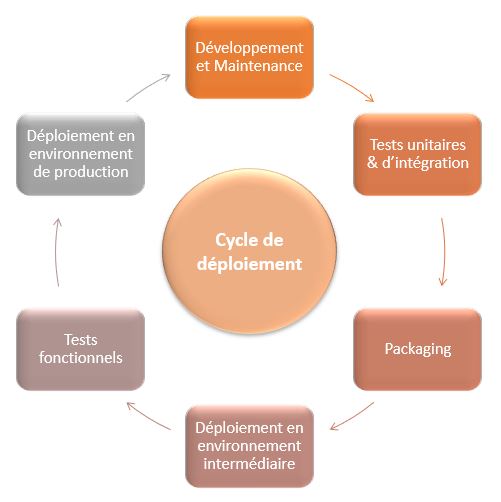
### Environnements de déploiement

Pour une meilleure mise en exploitation, différents environnements de déploiement peuvent être prévu en vue de prendre en compte, de façon optimale, les contributions des parties prenantes et de réduire au maximum les bugs pouvant survenir lors de l’utilisation du système. Dans une telle approche, les équipes peuvent créer, tester et déployer des produits en différentes phases dans l’optique de garantir un système de haute qualité aux utilisateurs. Ainsi, les développeurs peuvent expérimenter dans un environnement dédié, de nouvelles fonctionnalités, mises à jour et améliorations sans affecter ce que voient les utilisateurs sur leurs écrans. De même, une équipe de testeurs peut de tester une version quasi finale du produit pour garantir un bon fonctionnement et une bonne expérience utilisateur avant le déploiement final du produit ou de la mise à jour.

Dans le cadre de ce projet, nous prévoyons mettre en place trois environnements d’exploitation :

* Un environnement de développement
* Un environnement de test
* Un environnement de production

### Cycle de déploiement



**Cycle de déploiement**

1. COUT DE LA SOLUTION

# CONCLUSION

# REFERENCES BIBLIOGRAPHIQUES

# WEBOGRAPHIE

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

<https://www.youtube.com/c/LaurentiuSpilca>

https://www.baeldung.com/

https://www.youtube.com/c/amigoscode

<https://www.youtube.com/c/JavaBrainsChannel/playlists>

https://www.youtube.com/c/DailyCodeBuffer

https://openclassrooms.com/

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

<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

Annex xx :

### 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 micro services 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 micro service 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 micro services est toujours fonctionnelle, nous développeront des tests d’intégration pour chaque fonctionnalité au sein d’un micro service faisant appel à d’autres micro services.

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.

### 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 micro services (backend) écrits en java et NPM pour l’application frontend. Pour ce faire, nous exécuterons la commande **mvn package** pour chaque micro service et la commande **npm run build** pourl’application frontend.

### Déploiement en environnements intermédiaires (Test & Dev)