

# Rapport Projet JAVA

---

SERVICE ADMINISTRATIVE

Réaliser par :

Amzil ihssane  
Belmabkhout Rabab  
Boussabi Mina

Encadrer par :

**Mr. Hafidi Imad**

## Table des matières

<b>I. INTRODUCTION</b>	2
<b>II. Analyse des besoins</b>	3
1. Spécification des acteurs :	3
2. Description des cas d'utilisation :	4
<b>III. Spécification</b>	6
1. Diagramme de cas d'utilisation :	6
3. Diagramme de navigation :	8
4. Diagramme de classe métier :	10
5. Planning de développement :	10
<b>IV. Conception et architecture générale :</b>	11
• La communication entre le BackOffice et le FrontOffice :	13
1. Diagramme de classe détaillé :	13
• BackOffice :	13
• FrontOffice :	16
<b>V. Réalisation :</b>	18
1. Division des tâches entre les membres du groupe :	18
2. Démonstration du projet :	18
3. Problèmes rencontrés lors de la réalisation :	29
<b>VI. Conclusion</b>	29

# I. INTRODUCTION

Notre projet consiste à réaliser une application pour un service administrative, afin de permettre aux clients de poser leurs demandes et les suivre jusqu'à la fin de leur traitement. Et au personnel du service de gérer les demandes de leurs clients afin de satisfaire ces derniers.

En premier temp, nous avons commencé par la conception du projet en s'inspirant du modèle de développement logiciel en cascade que nous avons vu en UML et que vous nous avez recommander. Cette méthode nous avons permis d'avoir une présentation structurée de notre projet.

Dans ce premier rapport, nous vous propose notre modèle de conception afin de pouvoir passer à l'étape d'implémentation. Pour cela, nous allons détailler en premier lieu les diagrammes de cas d'utilisation puis nous allons présenter les descriptions textuelles, ainsi nous arriverons à constituer les interfaces hommes machines et les diagrammes de navigation et ensuite le diagramme de classe métier. En deuxième lieu nous présenterons l'architecture que nous adapterons dans la phase de l'implémentation ainsi que le plan des tests de notre projet.

## II. Analyse des besoins

Afin de commencer le travail sur notre projet, et après des lectures répétitives de cahier de charge nous avons eu une longue discussion afin de bien discuter les différentes fonctionnalités de ce projet. Ce qui nous a permis à définir ce que le système doit faire, ainsi qu'exprimer les interactions entre les utilisateurs et le système comme nous expliquons dans cette partie.

### 1. Spécification des acteurs :

#### → **Coté administrateur :**

- Responsable de la gestion des employés et les chefs de services.
- La gestion des procédures offerte au client, ainsi leurs étapes.
- Pouvoir affecter les étapes et les procédures au différents employé est chefs de services
- Consulter les différents statistiques du service

#### → **Coté chef de division :**

- Pouvoir accepter une demande en générant un jeton qui permet au client de suivre une demande.
- La possibilité de voir l'avancement des demandes du service ainsi archiver une demande.

#### → **Coté employé :**

- Le traitement d'une étape par prise de décision (accepter, demande de mise a jour, refus, ou rejet) en rédigeant un rapport a la fin.
- Consulter les étapes des demandes qui lui sont affecter.

#### → **Coté client :**

- Pouvoir poser une demande en choisissant une procédure bien définit.
- Il a le droit de suivre le progrès de sa demande.

Selon le cahier de charge, on a qu'un chef de division est aussi un employé, d'où existe une relation d'héritage entre les deux acteurs. Nous présentons ceci dans la figure suivante :

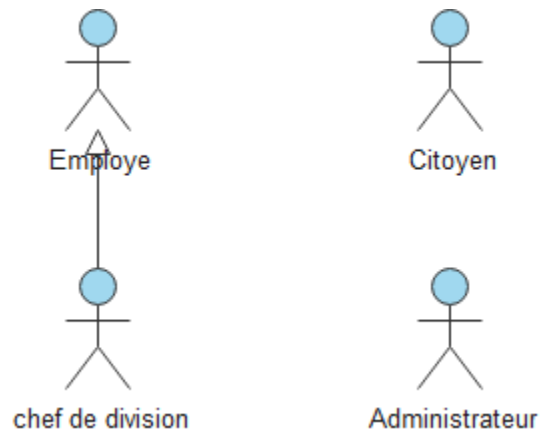


Figure 1: la relation d'héritage entre les acteurs

## 2. Description des cas d'utilisation :

→ Back Office :

Cas d'utilisation	Description	Acteurs Principaux
Authentification	L'acteur peut se connecter à l'application grâce à un CIN et un mot de passe, afin d'accéder aux fonctionnalités de l'application	Administrateur
Gestion des employés et chefs de division	L'acteur peut gérer l'ensemble des employés et chefs de division (Ajout, modification, Edition, archivage)	
Gestion des procédures et ses documents	L'acteur peut ajouter, modifier, éditer, et archiver les procédures ainsi que leurs documents	
Gestion des étapes	L'acteur peut gérer les étapes des procédures (Ajout, modification, Edition)	
Affectation des étapes aux employés	L'acteur peut affecter les étapes aux employés	

Tableau 1: Description des cas d'utilisation du back office

→ Front Office :

Cas d'utilisation	Description	Acteurs Principaux
Authentification	L'Employé et le chef de division peuvent se connecter à l'application grâce à un CIN et un mot de passe, or le citoyen juste par son CIN, afin d'accéder aux fonctionnalités de l'application.	Employé & Chefs de division & Citoyen

Consulter demande	L'acteur peut consulter les demandes qui lui sont affectés.	Employé
Gestion d'une étape	L'acteur peut gérer les étapes des demandes qui lui sont affectés (Accepter, Mise à jour, refus, rejet).	Employé
Accepter une demande	L'acteur peut accepter une demande de citoyen.	Chef de division
Générer un jeton	L'acteur peut générer un jeton après l'acceptation de demande afin de permettre au citoyen de suivre sa demande.	Chef de division
Archiver demande	L'acteur peut archiver les demandes terminées ou rejetés.	Chef de division
Consulter l'avancement des demandes	L'acteur peut consulter l'avancement de des demandes de procédure de son service.	Chef de division
Déposer demande	L'acteur peut déposer une demande en choisissant une procédure.	Citoyen
Déposer les documents	L'acteur peut ajouter des documents avec la demande	Citoyen
Mise à jour les documents	L'acteur peut modifier les documents déposer avec la demande	Citoyen
Suivi de la demande	L'acteur peut suivre sa demande en utilisant le jeton générer par le chef de division	Citoyen

Tableau 2:Description des cas d'utilisation du front office

→ Reporting :

Cas d'utilisation	Description	Acteurs Principaux
Authentification	L'acteur peut se connecter à l'application grâce à un CIN et un mot de passe, afin d'accéder aux fonctionnalités de l'application.	Administrateur
Consulter le nombre de demandes traitées	L'acteur peut consulter le nombre de demandes traitées pour chaque procédure.	
Consulter le taux d'acceptation des demandes	L'acteur peut consulter le taux d'acceptation des demandes.	
Consulter les taux de validation d'étapes par employé	L'acteur peut consulter les taux de validation d'étapes par employé.	

Consulter la moyenne de la durée des traitements	L'acteur peut consulter la moyenne de la durée des traitements par procédure, par étape, ou par employé.
--	--

Tableau 3: Description des cas d'utilisation de Reporting

## III. Spécification

### 1. Diagramme de cas d'utilisation :

→ Back office :

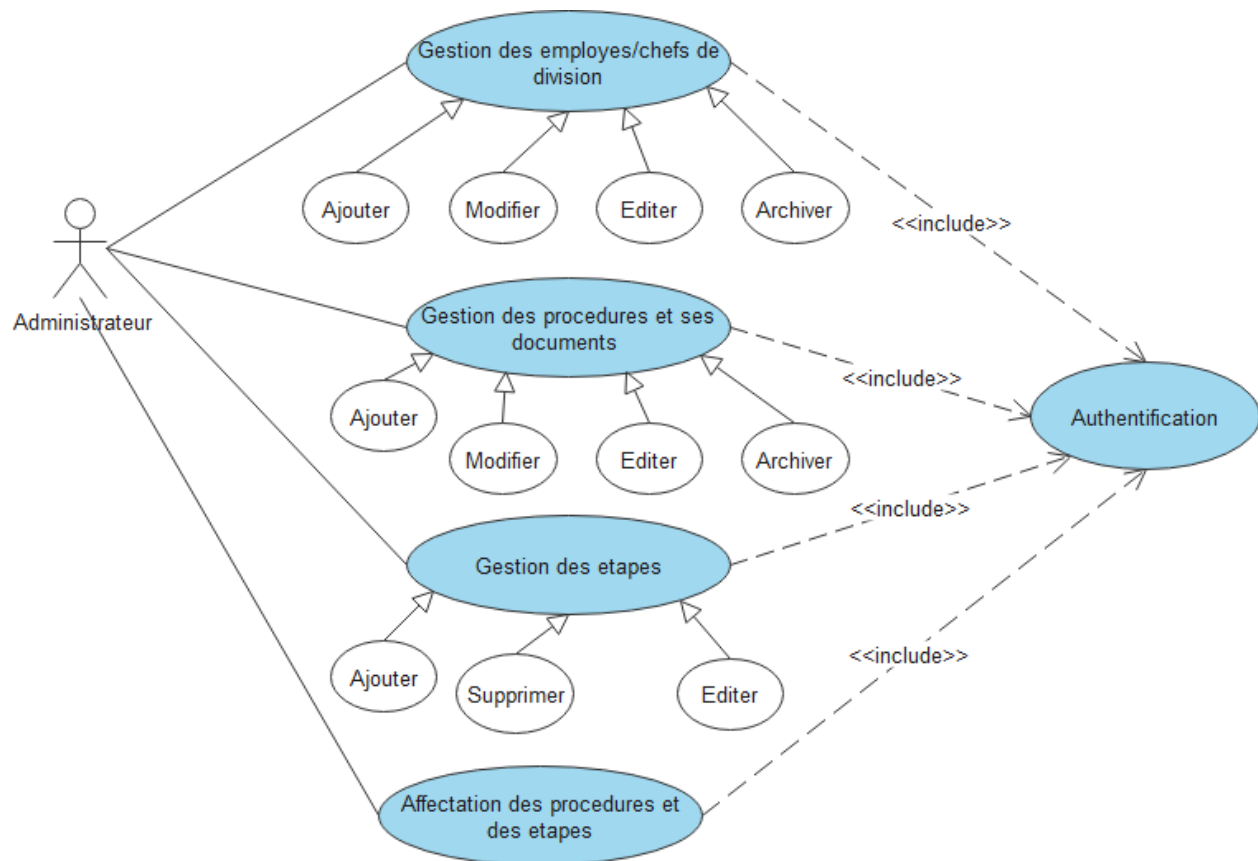


Figure 2: Diagramme cas d'utilisation du back office

```

graph TD
    Employe --> UC1[Consulter les donnees]
    Employe --> UC2[Gestion d'une etape]
    UC1 -.-> UC2
    UC2 -.-> UC3[Authentification]
    UC2 --> UC4[Refus]
    UC2 --> UC5[Rejet]
    UC2 --> UC6[Mise a jour]
    UC2 --> UC7[Accepter]
    UC4 -.-> UC8[Rediger un rapport]
    UC5 -.-> UC8
    UC6 -.-> UC8
    UC7 -.-> UC8
    UC8 --> UC9[Generer jeton]
    UC9 --> UC10[Authentification]
    UC10 --> UC11[Authentification(Citoyen)]
    UC10 --> UC12[Accepter demande]
    UC10 --> UC13[Archiver demande]
    UC10 --> UC14[Consulter l'avancement des demandes]
    UC12 --> UC11
    UC13 --> UC11
    UC14 --> UC11
    UC15[Citoyen] --> UC16[Depositer une demande]
    UC15 --> UC17[Suivi de la demande]
    UC16 --> UC18[Depositer les documents]
    UC18 --> UC11
    UC17 --> UC19[Mise a jour des documents]
    UC19 -.-> UC17
    UC16 -.-> UC17
    UC17 -.-> UC11
    
```

Figure 3: Diagramme cas d'utilisation du frontoffice

→ Reporting :



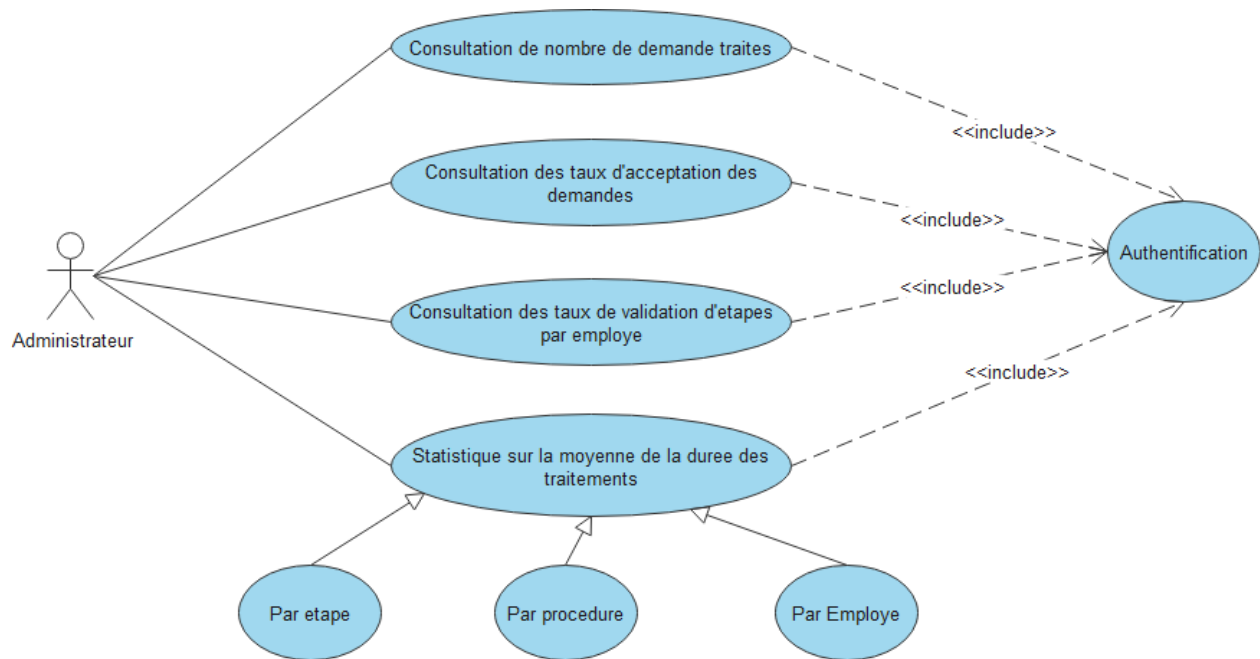


Figure 4:Diagramme cas d'utilisation du reporting

### 3. Diagramme de navigation :

→ Back office :

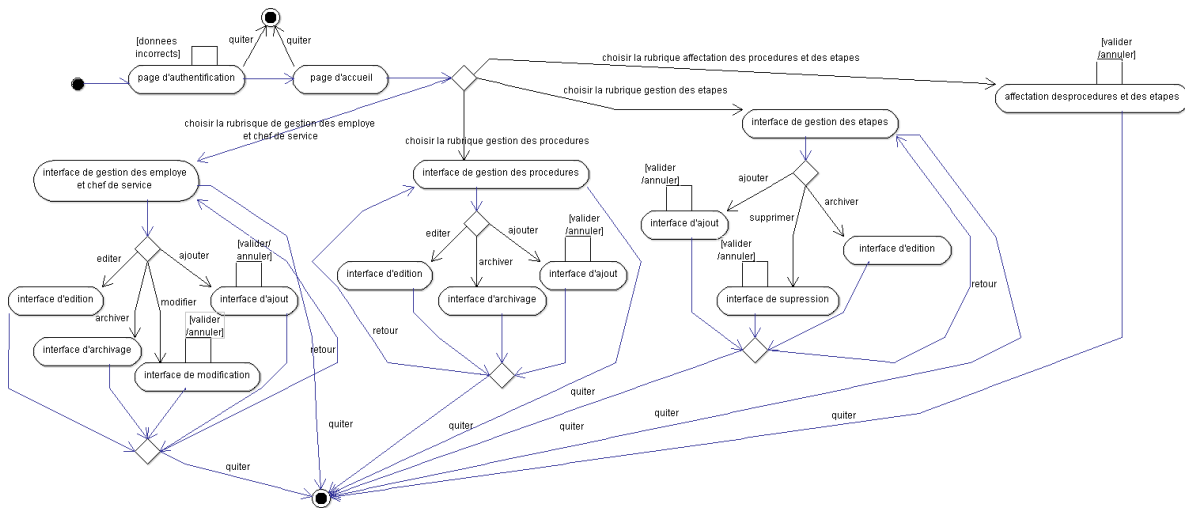


Figure 5:Diagramme de navigation du back office

→ Front office :

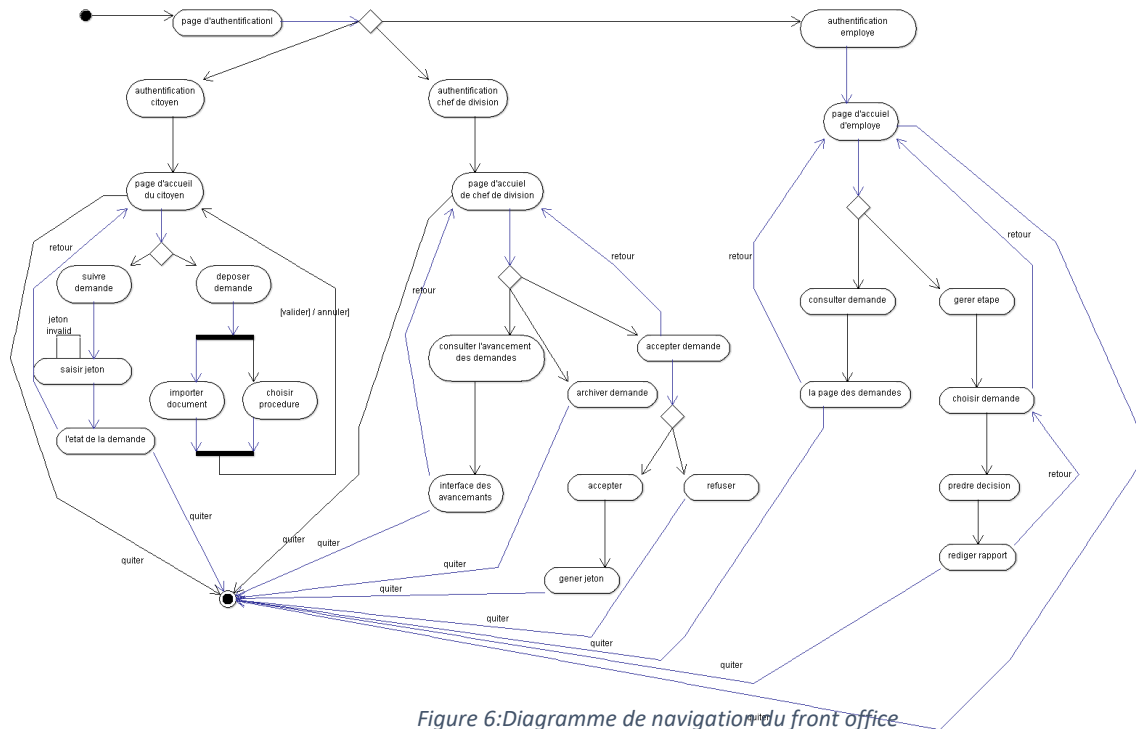


Figure 6: Diagramme de navigation du front office

## → Reporting :

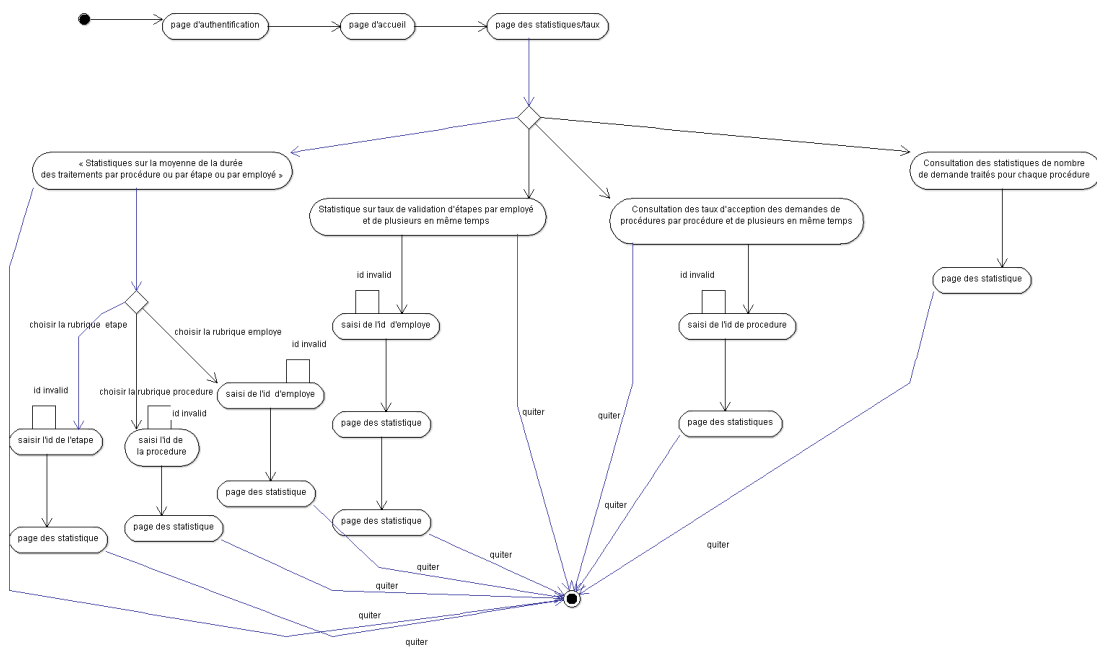


Figure 7: Diagramme de navigation du reporting

## 4. Diagramme de classe métier :

Suivant le cahier de charge, nous avons pu concevoir le diagramme de classe métier suivant :

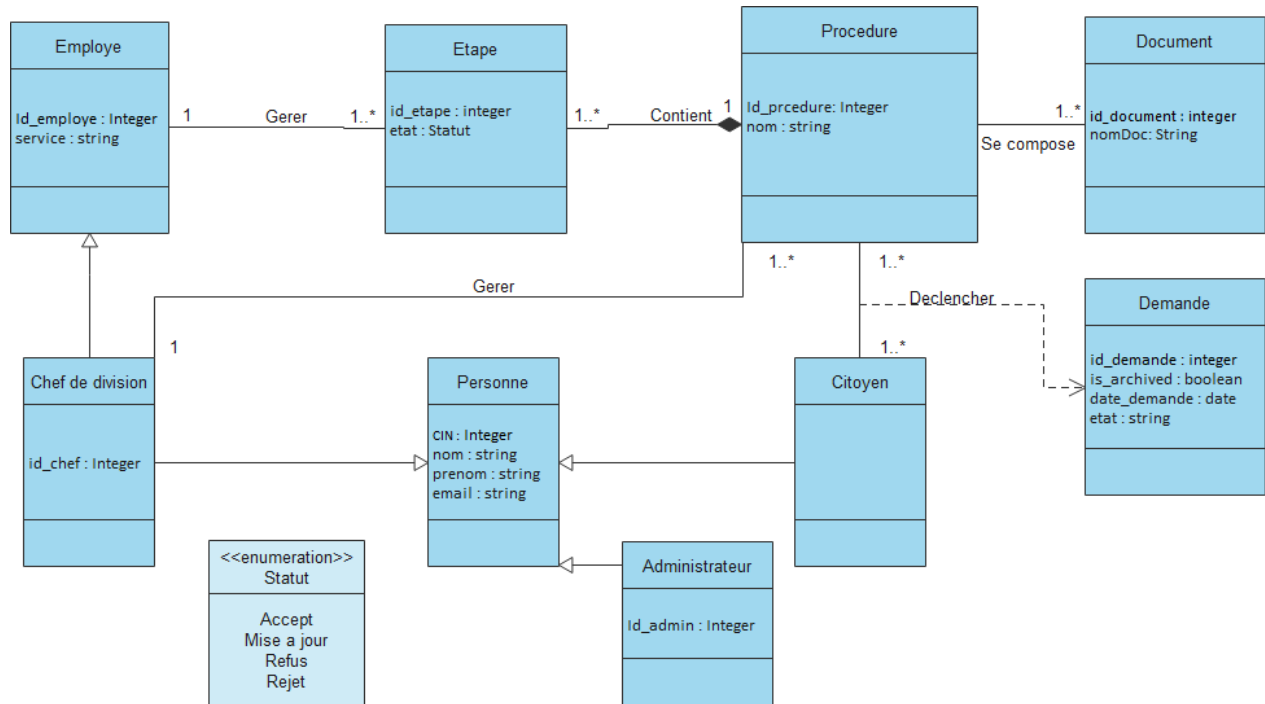
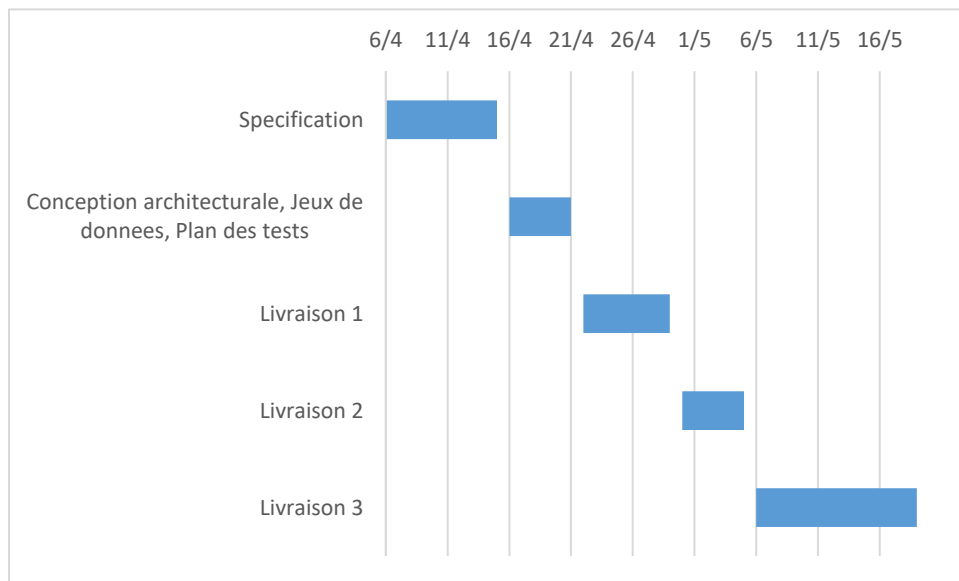


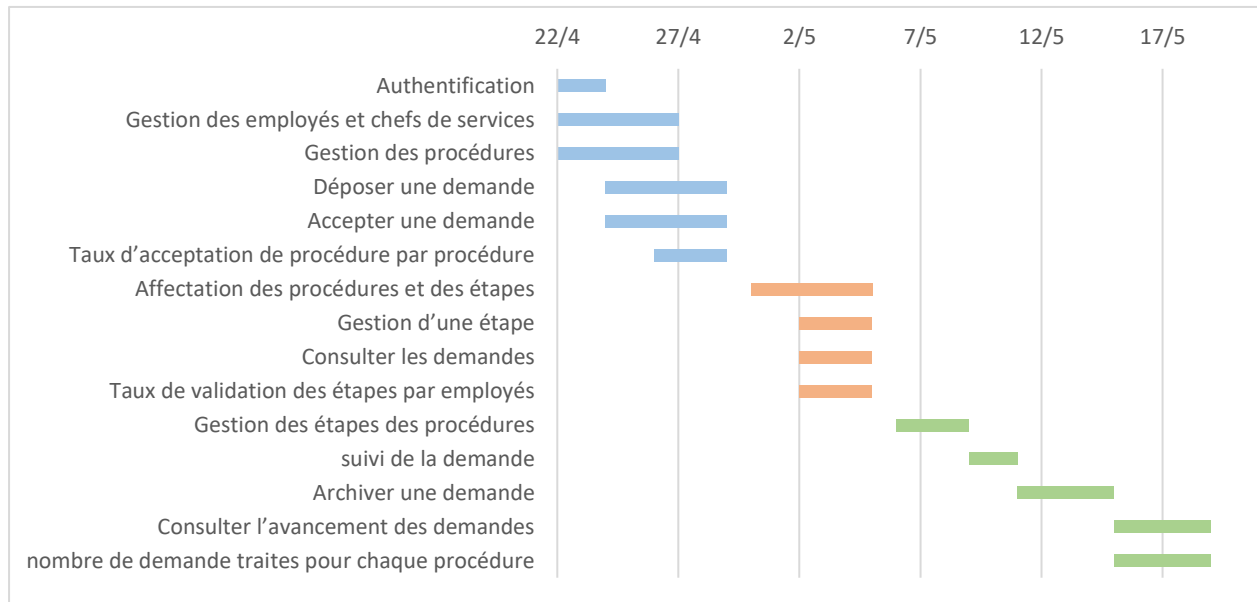
Tableau 4: Diagramme de classe metier

## 5. Planning de développement :

Voici un diagramme de Gantt qui représente les phases de réalisation selon le cahier de charge.



Ainsi selon votre demande, voici le planning de développement que nous proposons selon les cas d'utilisation de notre projet :



## IV. Conception et architecture générale :

Chaque système d'information nécessite la réalisation de trois groupes de fonctions : Le stockage des données, la logique applicative et la présentation. Ces trois parties sont indépendantes les unes des autres. D'où le choix de l'architecture 3-tiers pour les applications. L'architecture 3-tiers est une architecture partagée entre :

- Les hôtes clients (ou administrateurs).
- Le serveur d'application.
- Le serveur de base de données.

En effet, cette architecture est un modèle logique d'architecture applicative qui vise à séparer, très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et à présenter cette application comme un empilement de trois couches, qui sont les suivantes :

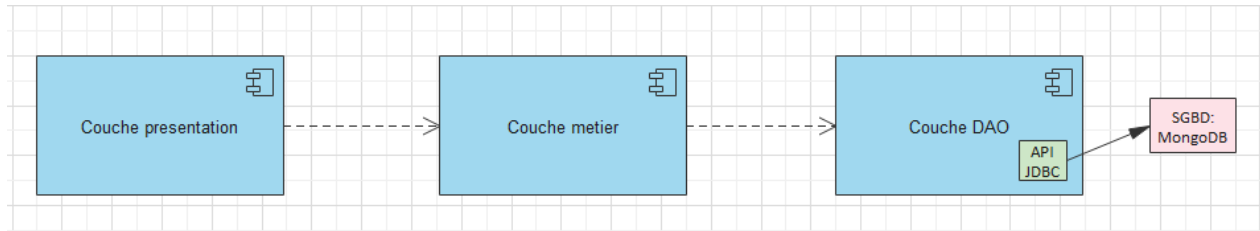
- **Couche Présentation** : Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'interface homme machine. La couche présentation relaie les requêtes de l'utilisateur à destination de la couche métier, et lui présente en retour les informations renvoyées par les traitements de cette couche. Il s'agit donc d'un assemblage de services métiers et applicatifs offerts par la couche inférieure.
- **Couche Métier** : Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la « logique », et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs effectués au travers de la couche présentation.
- **Couche Accès aux données** : C'est la partie qui gère l'accès aux données du système. Ces données peuvent être propres au système, ou gérées par un autre système.

D'après les exigences techniques présentés dans le cahier de charges :

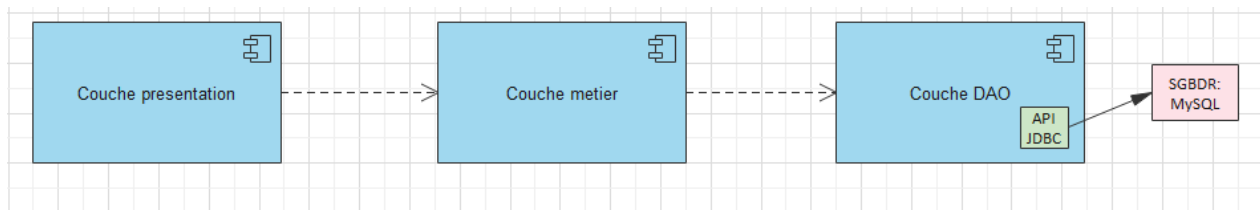
- La base de données du back office en BDR (MySQL)
- La base de données du front office en BD Non SQL (MongoDB)
- Le front Office, le back office et le reporting seront réalisés par des applications bureau (SWING, JAVA FX)

Nous avons construit l'architecture de chaque partie :

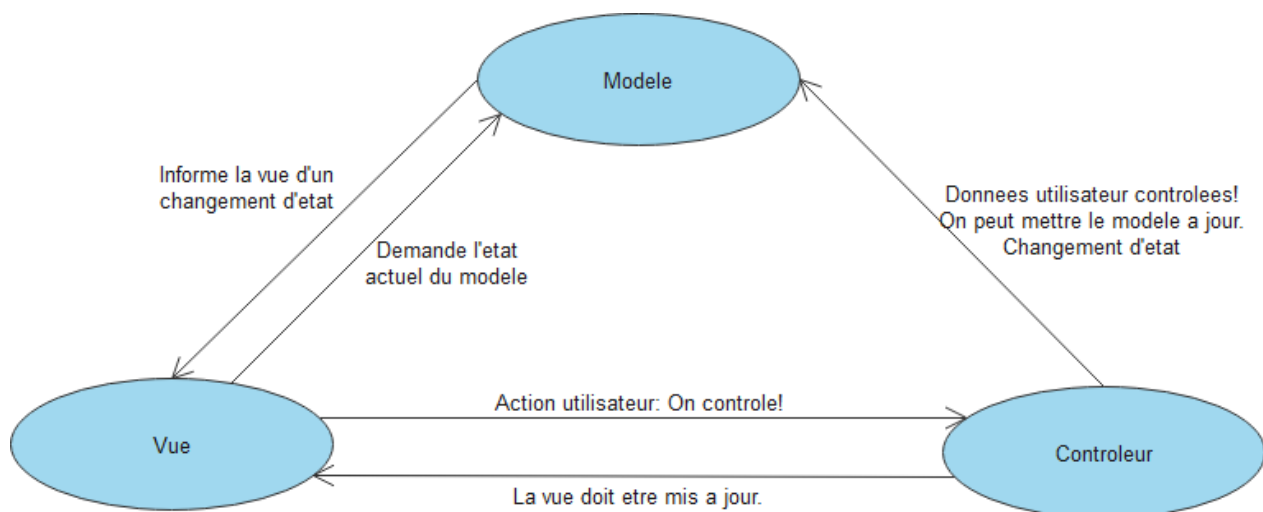
- FrontOffice:



- BackOffice & Reporting:



Les applications respectent une architecture MVC qui nous apprend à découper nos codes en trois parties (Modèle - Vue - Contrôleur).



- ✓ La vue :

Ce que l'on nomme « la vue » c'est en fait une IHM. Elle représente ce que l'utilisateur a sous les yeux. La vue peut être donc : une application graphique Swing. Une page web ...

✓ Le modèle :

Le modèle peut être divers et varié. C'est là que se trouvent les données. Il s'agit en général d'un ou plusieurs objets JAVA. Ces objets s'apparentent généralement à ce qu'on appelle souvent « La couche métier » de l'application et effectuent des traitements absolument transparents pour l'utilisateur => Le cœur du programme.

✓ Le contrôleur :

Cet objet permet de faire le lien entre la vue et le modèle lorsqu'une action utilisateur est intervenue sur la vue. C'est cet objet qui aura pour rôle de contrôler les données.

Le traitement d'une demande d'un client se déroule selon les étapes suivantes :

1. Le client fait une demande au contrôleur. Ce contrôleur va passer toutes les demandes des clients. C'est la porte d'entrée de l'application.
2. Le contrôleur traite cette demande. Pour ce faire, il peut avoir besoin de l'aide de la couche métier, ce qu'on appelle le modèle dans la structure MVC.
3. Le contrôleur reçoit une réponse de la couche métier. La demande du client a été traitée. Celle-ci peut appeler plusieurs réponses possibles. Un exemple classique est :
  - Une page d'erreurs si la demande n'a pu être traitée correctement
  - Une page de confirmation sinon
4. Le contrôleur choisit la réponse (= vue) à envoyer au client. Celle-ci est le plus souvent une page contenant des éléments dynamiques. Le contrôleur fournit ceux-ci à la vue.
5. La vue est envoyée au client.

- **La communication entre le BackOffice et le FrontOffice :**

Vue que notre projet impose l'utilisation de différentes bases de données, MySQL pour la partie BackOffice et MongoDB pour le front office. Nous avons décidé de lier la partie FrontOffice avec la base de données MySQL afin qu'on puisse récupérer les informations nécessaires traitées dans la partie BackOffice dans la base de données principale de la partie FrontOffice.

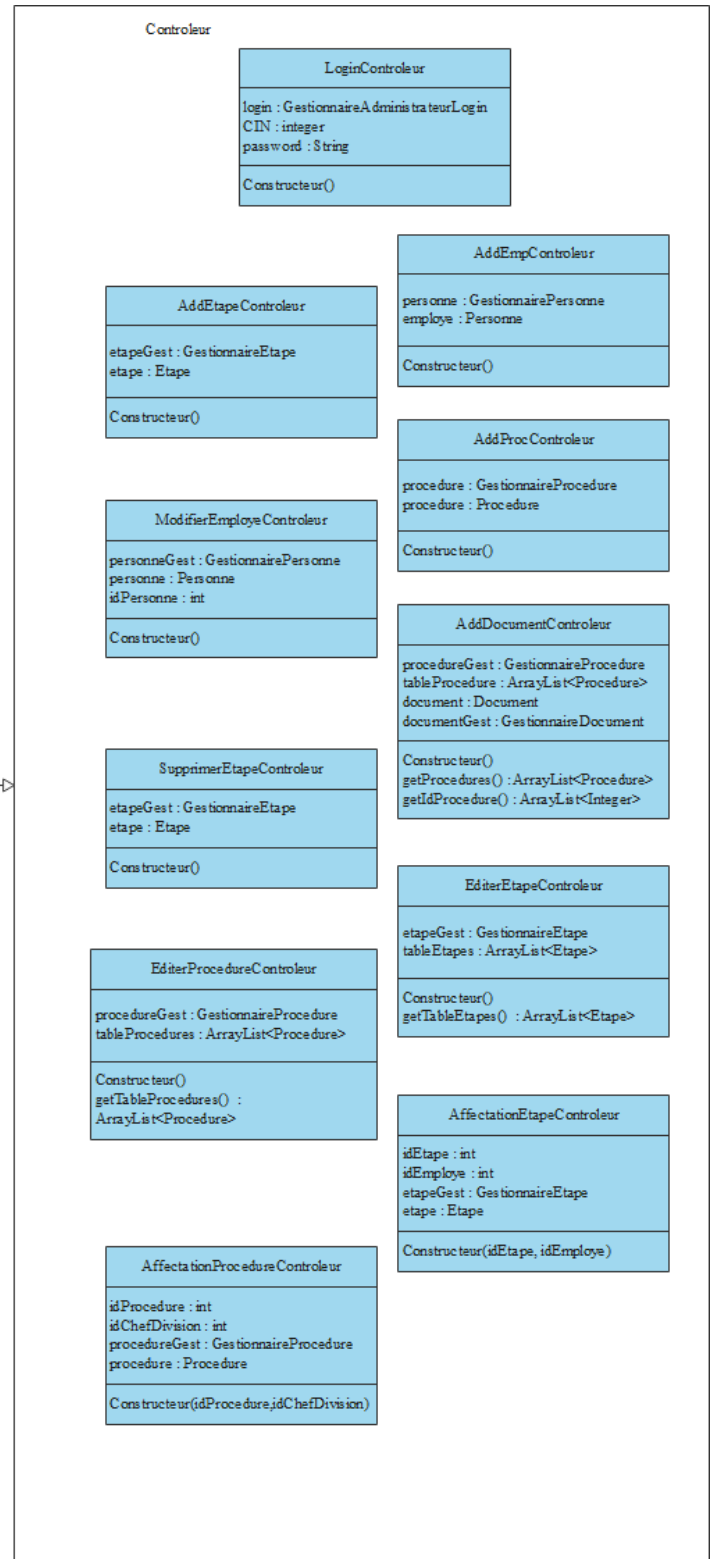
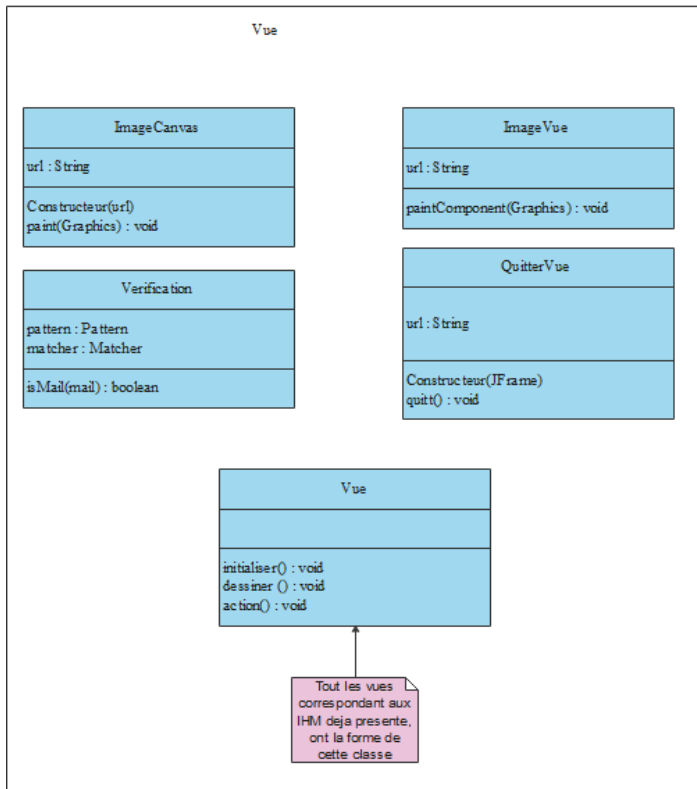
### 1. Diagramme de classe détaillé :

Nous pouvons maintenant construire nos diagrammes de classes détaillés pour notre solution en ajoutant les différents éléments (classes, associations, attributs, etc.) déduits à partir des activités précédentes.

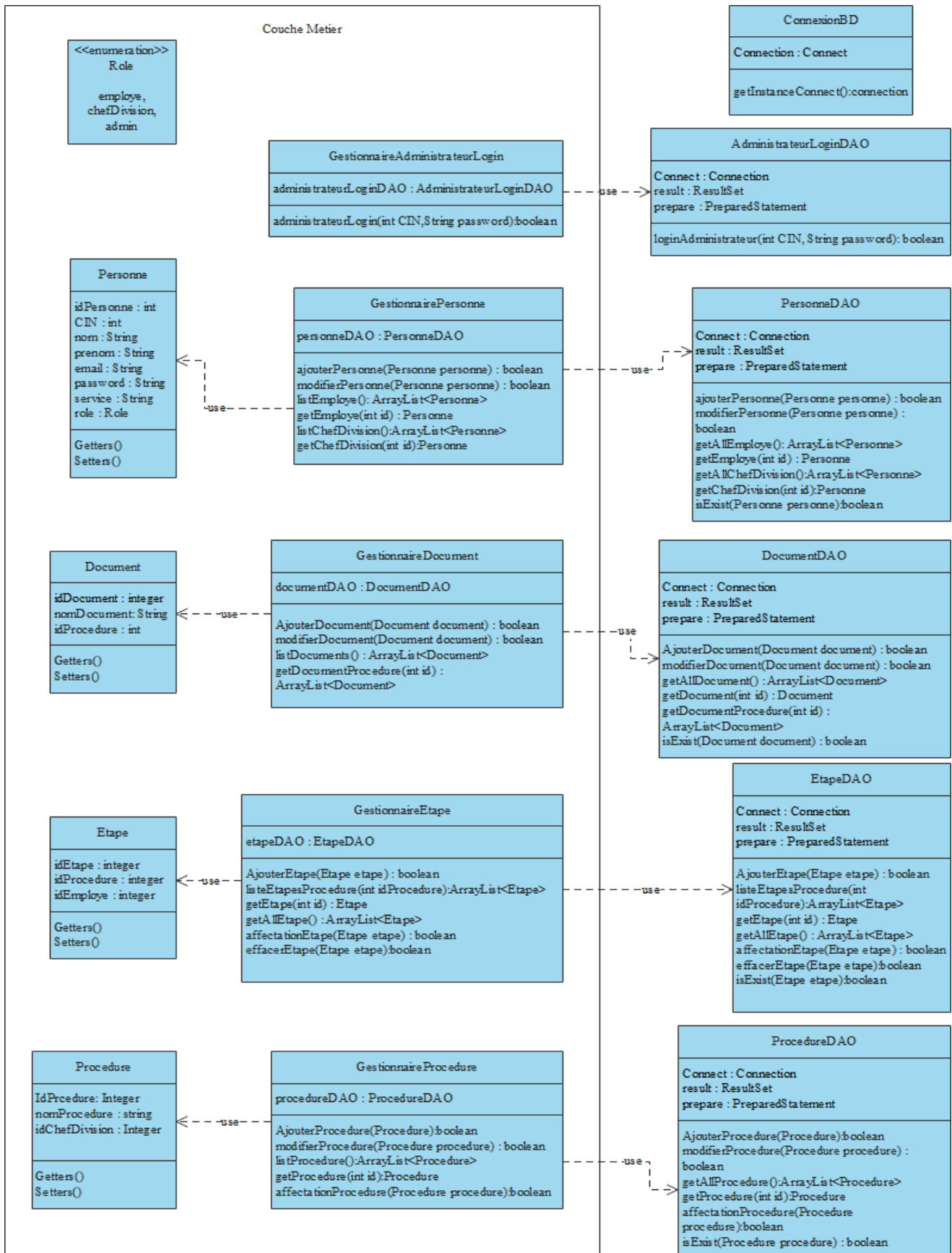
- **BackOffice :**

Pour cette partie on va construire notre 1<sup>er</sup> projet qui va contenir tous les cas d'utilisation qui concernent cette partie. Maintenant, nous présentons notre diagramme de classe en respectant l'architecture présentée ci-dessus.

→ Couche présentation (Contrôleur & Vue) :



→ Couche Metier & DAO :

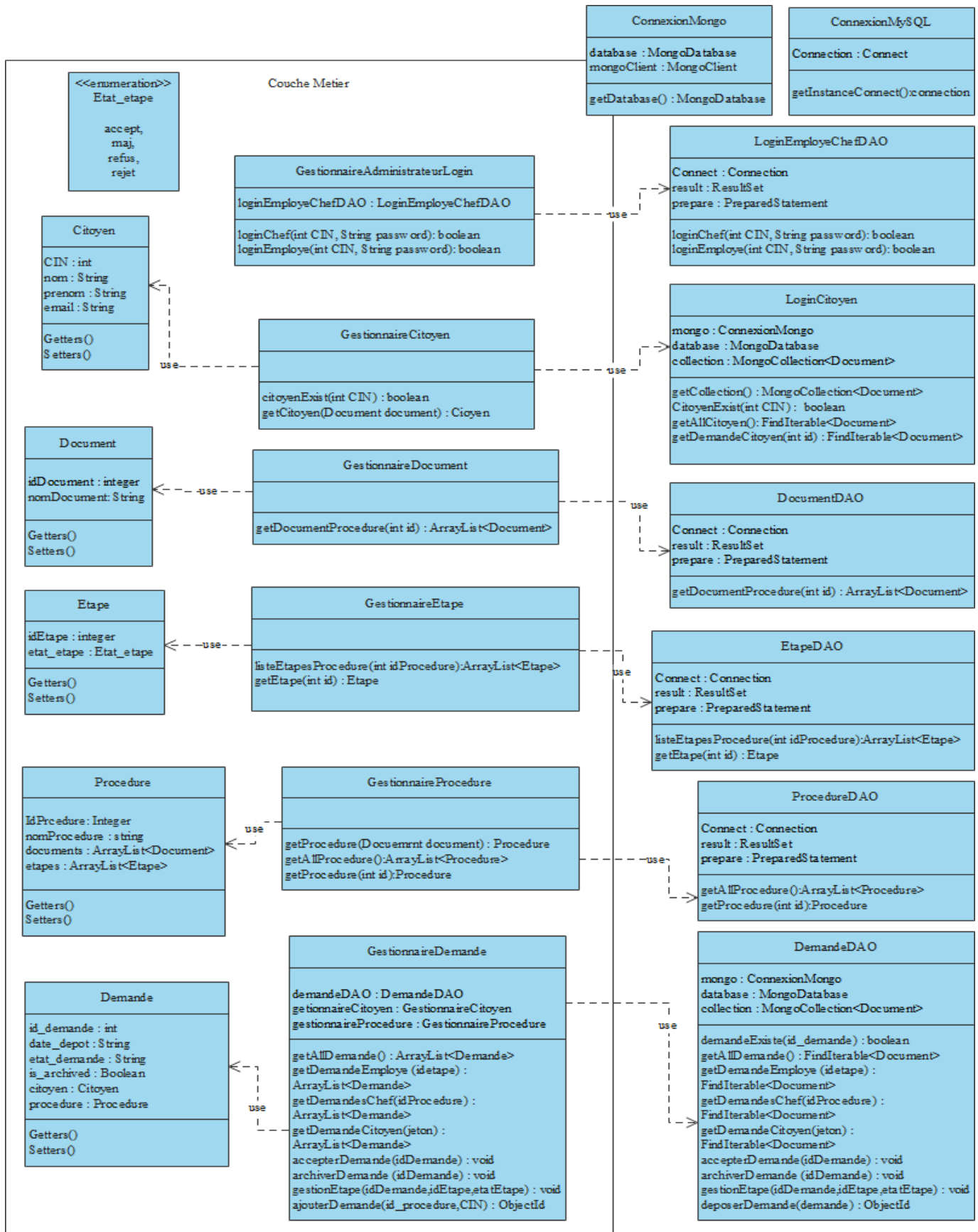




- **FrontOffice :**

Pour la réalisation de la partie FrontOffice, nous devons concevoir un 2eme projet contenant les cas d'utilisations qui concerne cette partie. D'où nous présentant notre diagramme de classe en respectant l'architecture présenter précédemment.

→ Couche Metier & DAO :



→ Couche présentation (Contrôleur & Vue)

Vue que cette partie est une interface client, donc nous avons décidé de réaliser cette partie avec JEE. Donc nos classes pour ces couches seront des servlets qui sont des classes avec deux fonctions (doGet() & doPost()) et des JSP qui sont responsable de l'interface afficher.

## V. Réalisation :

### 1. Division des taches entre les membres du groupe :

Afin de bien avancer dans la réalisation du projet, et puisque nous somme 3 personnes dans le groupe, nous avons décidé que chaqu'un de nous va être responsable de réaliser une couche (présentation, métier, et DAO) pour la partie BackOffice.

Or pour le FrontOffice, nous avons décidé qu'une personne travail sur les deux couches Métier et DAO vu que l'implémentation de la base de données MongoDB c'est un nouveau concept pour nous, c'est mieux qu'une personne se concentre sur ce concept et réalise les 2 couches puisqu'elles sont liées entre eux. Et les autres 2 membres ont eu la tâche de réaliser la couche présentation avec le nouveau concept JEE, qu'on a vu dans les dernières parties du cours JAVA.

### 2. Démonstration du projet :

- Création des collections MongoDB :

Pour créer une collection sous MongoDB c'est simple, mais ces collections peuvent avoir des données totalement différentes. D'où nous avons trouve une solution pour cela, c'est d'écrire un schéma exact de nos collections pour avoir une forme de données uni dans notre collection.

Pour cela nous devons avoir une méthode pour passer du MCD a une collection mongo, donc nous avons adapter une méthode traiter dans le cours [Maîtrisez les bases de données NoSQL](#).

Voici les schémas de nos collections MongoDB :

→ Citoyen :

```
db.createCollection("citoyen",
{
  validator:{
    $jsonSchema:{
      bsonType:"object",
      required:["CIN","nom","prenom","email"],
      properties:{
        CIN : {
          bsonType:"int"
        },
        nom : {
          bsonType:"string"
        },
        prenom : {
          bsonType:"string"
        },
        email : {
          bsonType:"string"
        }
      }
    }
  }
})
```

```

    }
  }
})

```

→ Demande :

```

db.createCollection("demande",
{
  validator:{
    $jsonSchema:{
      bsonType:"object",
      required:["id_demande","date_depot","etat_demande","is_archived","citoyen","procedure"],
      properties:{
        id_demande:{
          bsonType : "int"
        },
        date_depot:{
          bsonType : "date"
        },
        etat_demande :{
          enum : ["accept","rejet",null]
        },
        is_archived :{
          bsonType : "bool"
        },
        citoyen :{
          bsonType:"object",
          required:["CIN"],
          properties:{
            CIN : {
              bsonType:"int"
            }
          }
        },
        procedure:{
          bsonType:"object",
          required:["id_procedure","nom_procedure","document","etape"],
          properties:{
            id_procedure:{
              bsonType:"int"
            },
            nom_procedure:{
              bsonType:"string"
            },
            document:{
              bsonType:"array",
              properties:{
                id_document:{
                  bsonType:"int"
                },
                nom_document:{
                  bsonType:"string"
                }
              }
            }
          }
        }
      }
    }
  }
})

```

```

    },
    etape:{
      bsonType:"array",
      properties:{
        id_etape:{
          bsonType:"int"
        },
        etat_etape:{
          enum:["accept","maj","refus","rejet"]
        }
      }
    }
  }
}
})

```

- Présentation de la réalisation du projet :

→ BackOffice :

Commençons d'abord l'authentification de l'administrateur :

**SE CONNECTER :**

Nom d'utilisateur: 123

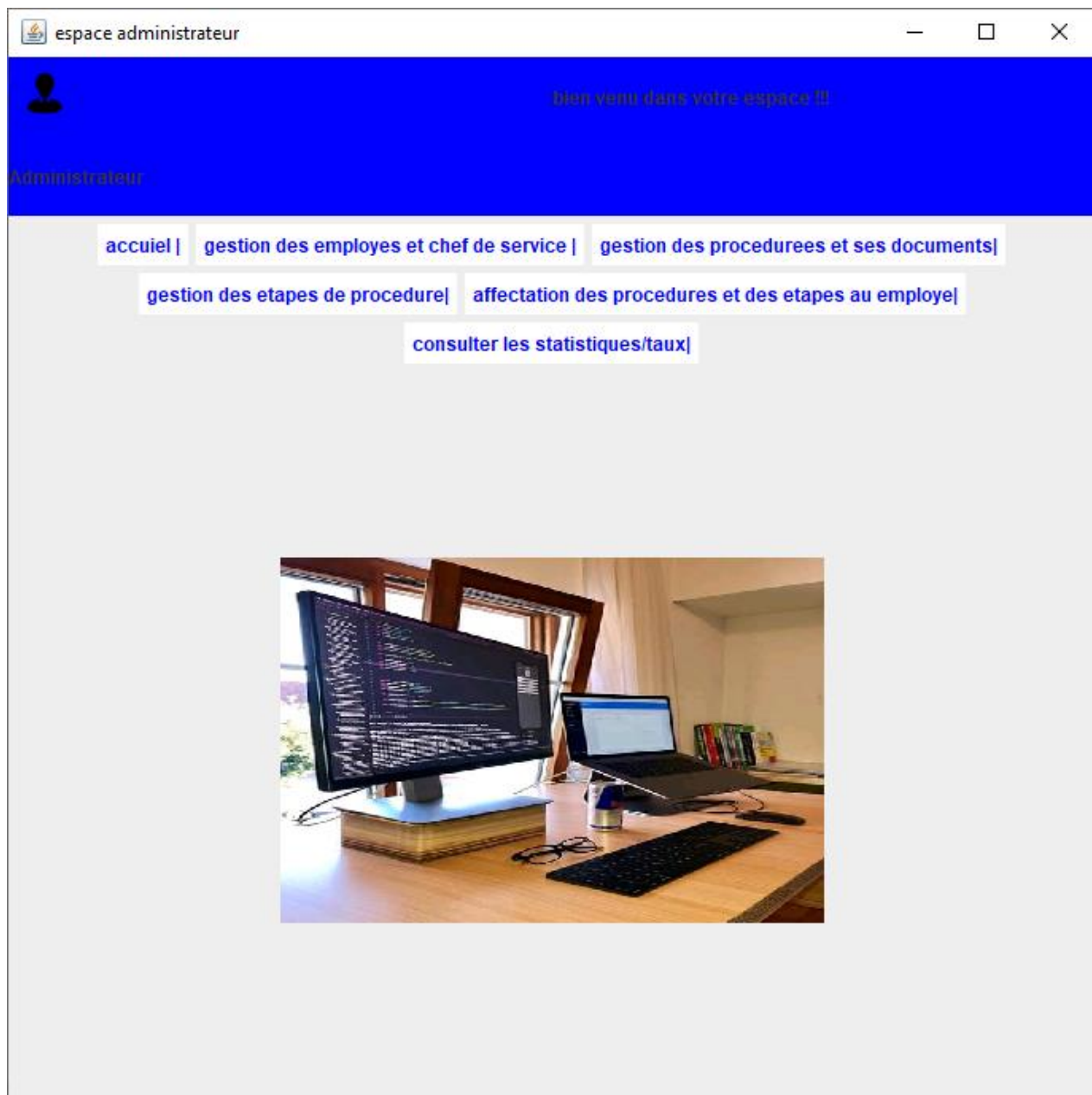
votre role: admin

mot de passe:

**se connecter**

pour le nom d'utilisateur veuillez entrer votre CIN

Ce qui nous dirige vers l'accueil :



Prenons par exemple le cas où l'administrateur veut ajouter un employé :

L'administrateur remplit les informations nécessaires et clique sur ajoute.

espace administrateur

bien venu dans votre espace !!!

Administrateur


accueil | gestion des employes et chef de service | gestion des procedures et ses documents |  
gestion des etapes de procedure | affectation des procedures et des etapes au employe |  
consulter les statistiques/taux |

Gestion des employe et chef de service      ajouter

CIN :	<input type="text"/>	Email :	<input type="text"/>
ID :	<input type="text"/>	password	<input type="text"/>
Nom :	<input type="text"/>	Fonction :	chef de service ▼
Prenom :	<input type="text"/>	Service :	RH ▼

ajouter

On prend un autre exemple, c'est l'edition des etapes :



espace administrateur



bien venu dans votre espace !!!

Administrateur

accueil |

gestion des employes et chef de service |

gestion des procedures et ses documents|

gestion des etapes de procedure|

affectation des procedures et des etapes au employe|

consulter les statistiques/taux|

Gestion des Procedures

Editer

id etape	id procedure	id Personne
2	2	1
3	2	34
4	2	1
6	4	34
7	4	1

→ FrontOffice :



La page initiale pour un citoyen est comme suit :

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/JAVA_PROJET_FRONT/connexionVerification`. The page has a dark blue space-themed background with a nebula and a comet. In the center, there is a white box with a pink border containing the following text:

**Connexion**

Vous pouvez vous connecter via ce formulaire.

CIN \*

vous devez remplir les champs requis

**Connexion**

At the bottom of the page, there is a dark blue footer with white text:

**realisé par:**

- AMZIL IHSSANE
- BELMABKHOUT RABAB
- BOUSSABI MINA

**Encadré par:**

- M.HAFIDI

Après l'authentification, le citoyen peut soit déposer ou bien suivre une demande, le dépôt d'une demande est comme suit :

Choisir une action:  
[Retour](#) [Suivre demande](#)

## DEPOSER DEMANDE

CIN: \*

EMAIL:

Choisir une procedure

1

valider pour afficher les documents necessaires

Emplacement du fichier \*

Voici votre jeton:

Or le suivie d'une demande :

http://localhost:8080/JAVA\_PROJET\_FRONT/suiviCitoyen

### suivre demande

Inserer le code de votre jeton: \*

L'etat de votre demande est:

les document a mettre a jour:

Emplacement du fichier:

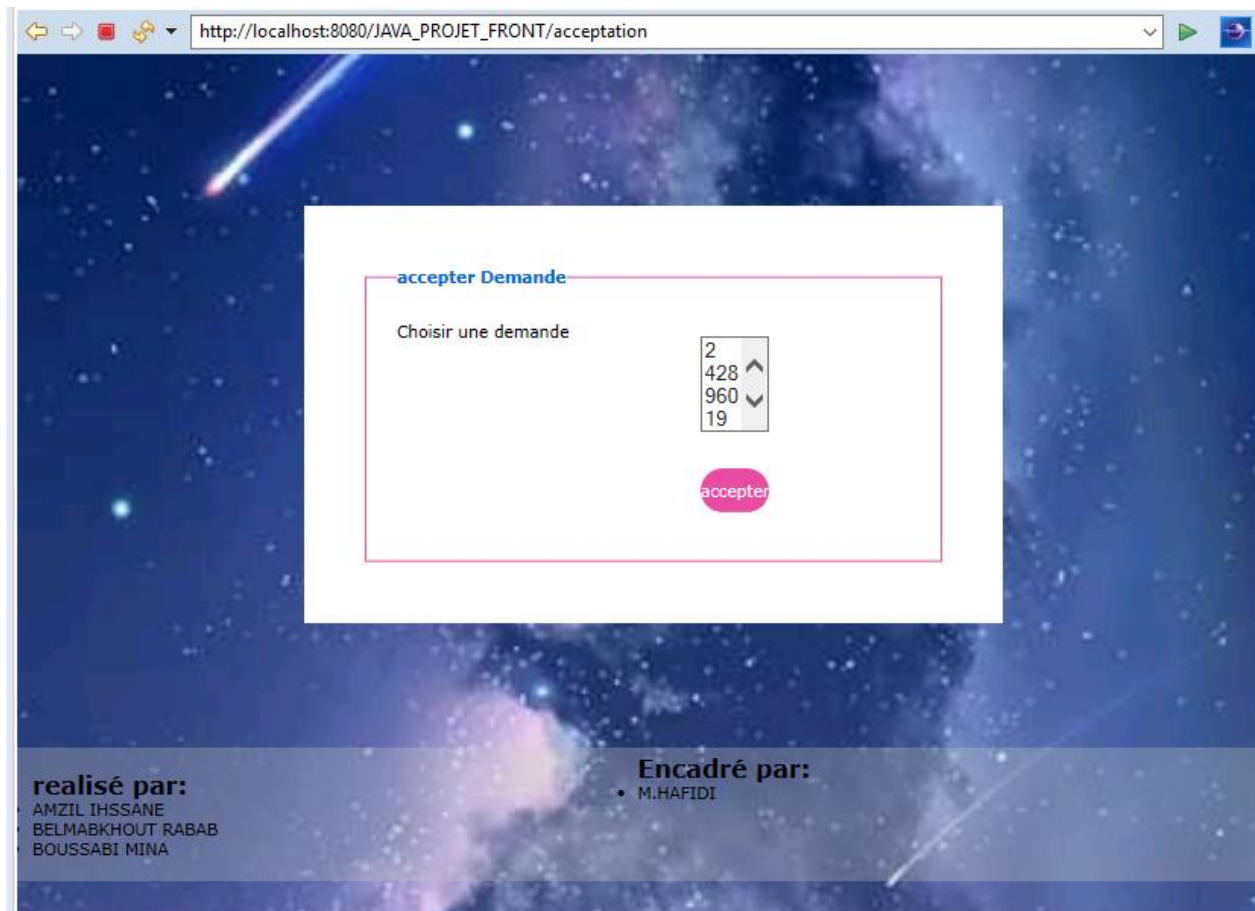
**realisé par:**  
 • AMZIL IHSSANE  
 • BELMABKHOUT RABAB  
 • BOUSSABI MINA

**Encadré par:**  
 • M.HAFIDI

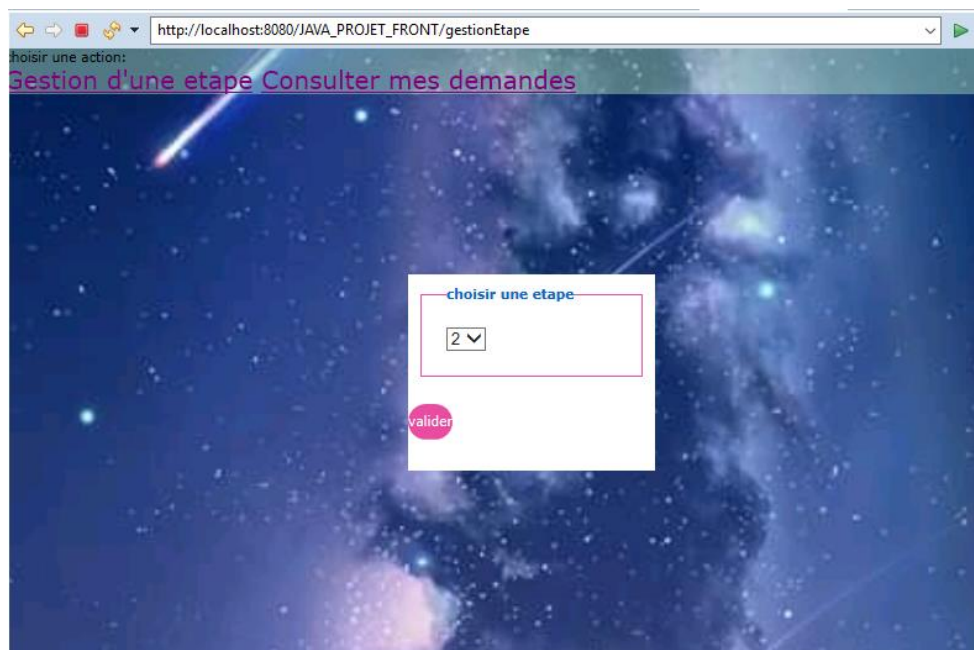
Maintenant pour un employe, ou un chef de division :



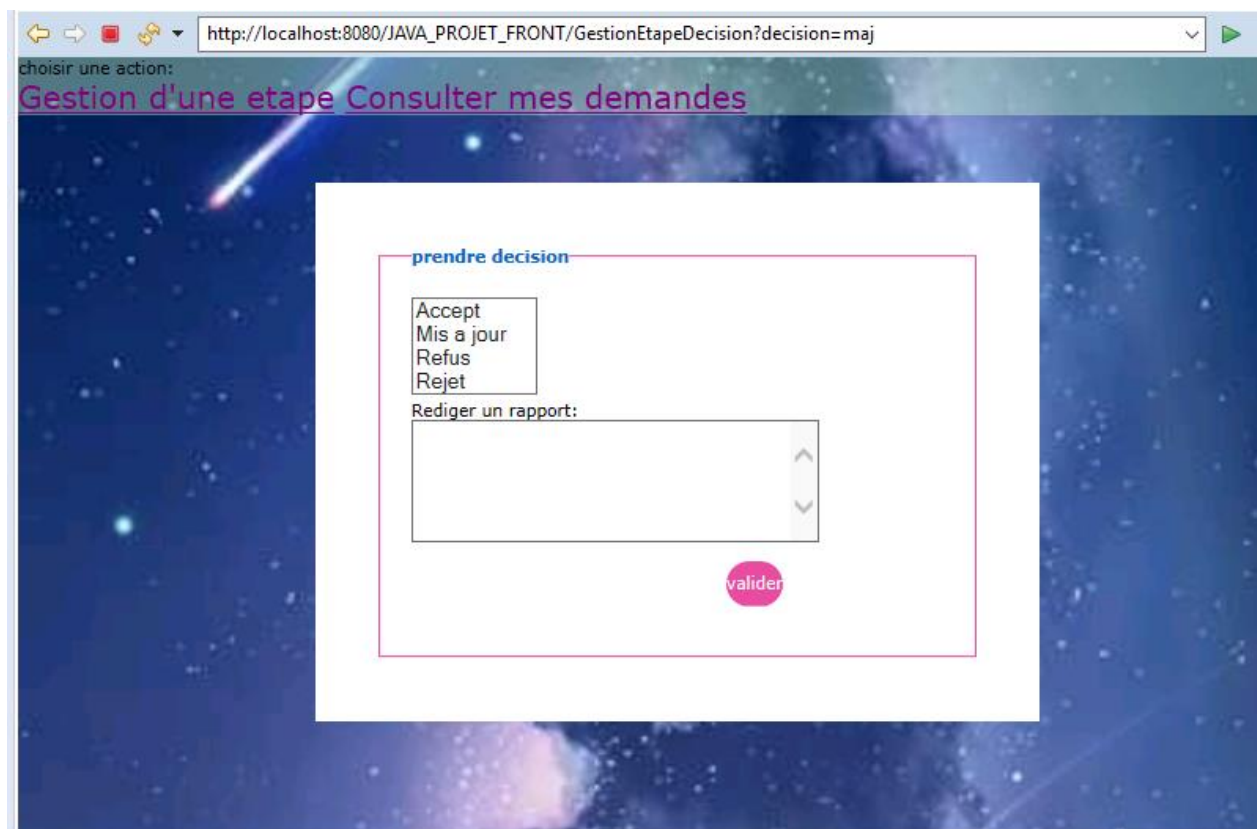
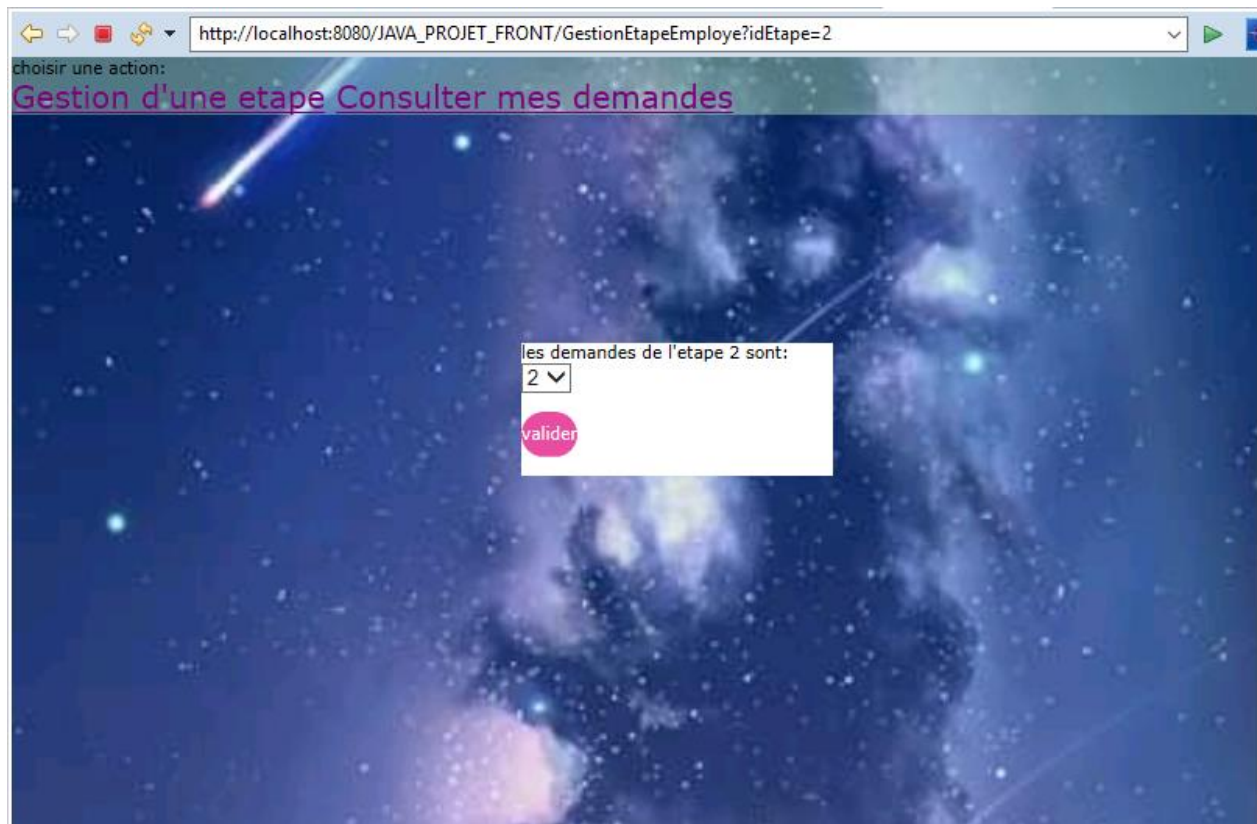
Prenons exemple d'acceptation d'une demande par le chef de division :



Et comme exemple pour un employe, on prend la gestion d'une etape :







### 3. Problèmes rencontrés lors de la réalisation :

- Synchronisation des données entre les bases de données SQL et NonSQL.
- Implémentation de la base de données MongoDB : création des collections avec un schéma exacte, les fonctions Java traitant les collections Mongo (souci des versions)

## VI. Conclusion

La réalisation de ce projet été une expérience riche, car chaqu'un de nous a eu la chance de travailler avec un nouveau outil qui peut être ajouter a notre bagage de connaissance, même si nous avons des difficultés a tous faire ou même de bien faire ce qu'on a pu faire, ca reste un projet avec lequel nous avons appris des choses importantes et nouvelle qui nécessite de nous plus de pratique pour bien les maitriser.