

# **ATELIER 1: Application de Gestion Générique des Processus Administratives**

ENCADRE PAR :

- Pr. Imad Hafidi

RÉLISÉ PAR:

-YOUNESS KEROUANI

# Sommaire

1. Conception et Architecture Générale des applications:
  - 1.1. Architecture Technique de la partie Front Office :
  - 1.2. Architecture Technique de la partie Back Office et Reporting :
2. Conception détaillée :
  - 2.1. Diagrammes des cas d'utilisation :
    - B. Partie Back Office :
    - C. Partie Reporting :
  - 2.2. Diagrammes de séquence :
  - 2.3. Diagramme de classe détaillé :

## 1. Conception et Architecture Générale des applications:

Tout 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, étages, ou niveaux qui sont les suivantes :

**Couche Présentation (premier niveau) :** 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 des services métiers et applicatifs offerts par la couche inférieure.

**Couche Métier / Business (second niveau) :** 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 (troisième niveau) :** 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 donnée du back office en BDR (MySQL)

La base de donnée du front office en BD Non SQL (MongoDB)

Le front Office, le back office et le reporting seront réalisé par des applications bureau (SWING, JAVA FX)

Nous avons construit l'architecture de chaque partie :

### 1.1. Architecture Technique de la partie Front Office :

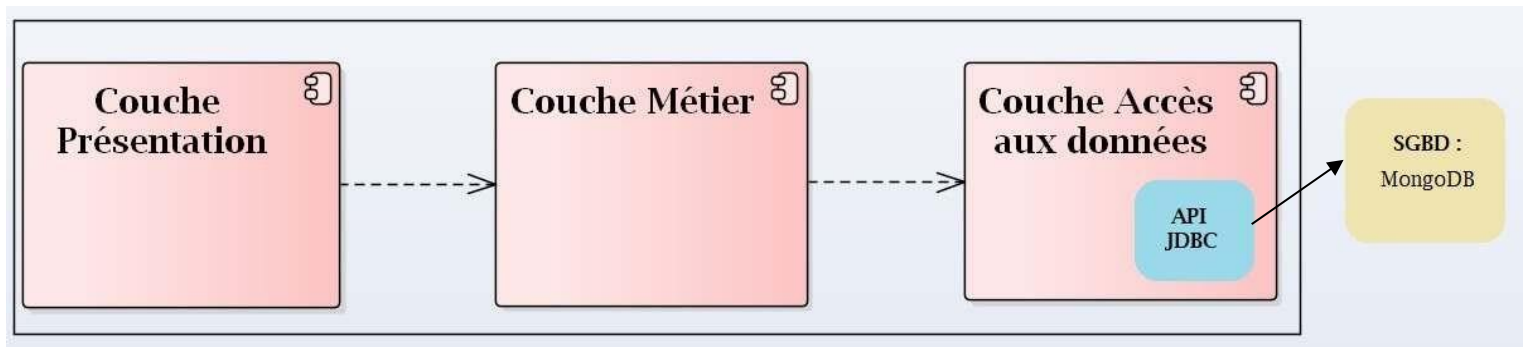


Figure 1: Architecture Technique de la partie Front Office

### 1.2. Architecture Technique de la partie Back Office et Reporting :

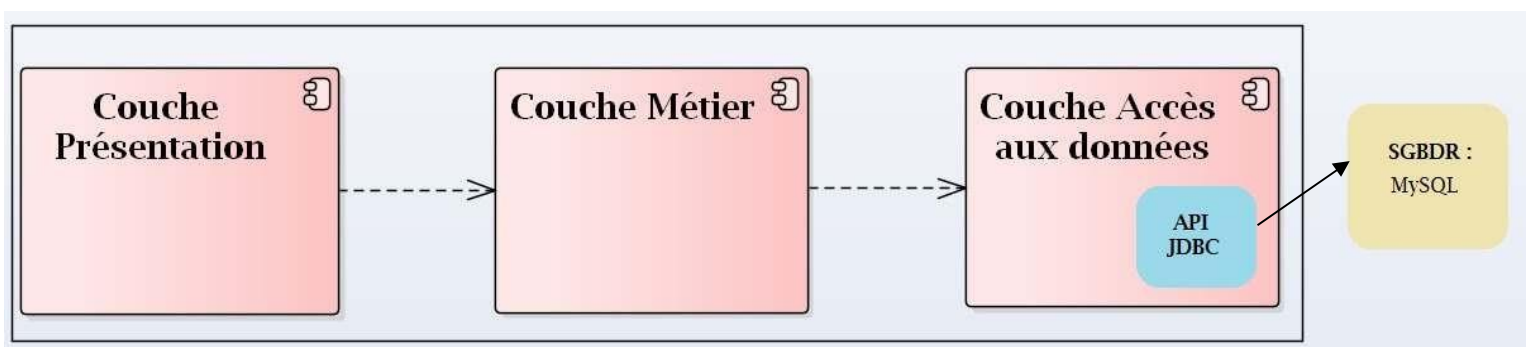


Figure 2: Architecture Technique des parties Back Office et Reporting

Les applications respectent une architecture MVC (Modèle - Vue - Contrôleur). Si nous reprenons le schéma en couches ci-dessus, l'architecture MVC s'y intègre de la façon suivante:

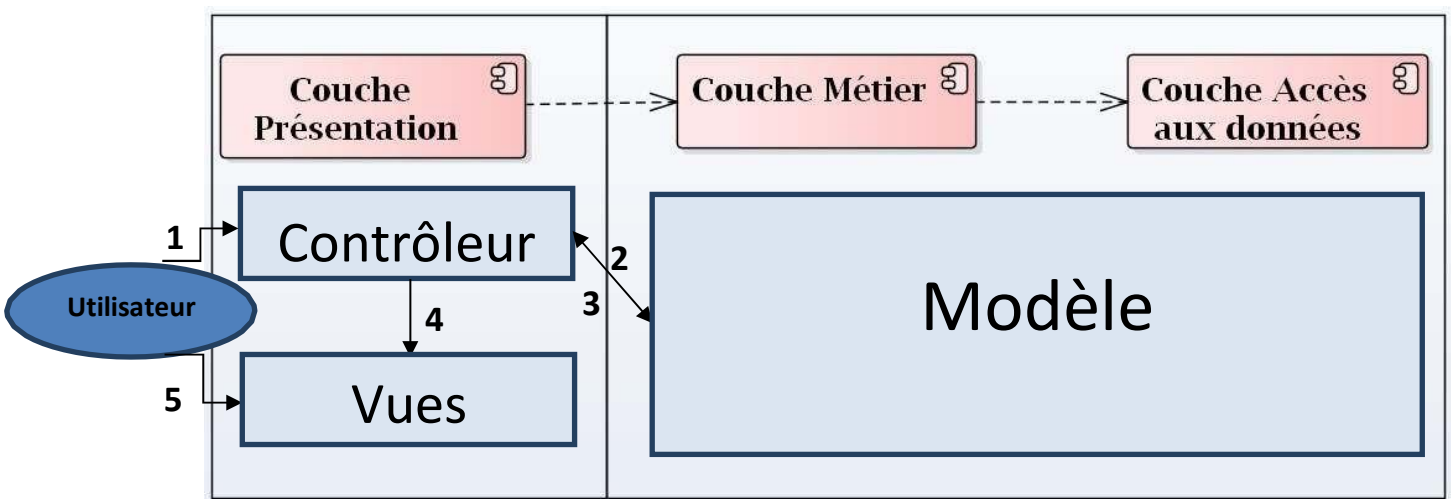


Figure 3: Architecture MVC intégrée en trois couches

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

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.

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.

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

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.

La vue est envoyée au client.

*Pour faciliter l'implémentation de la couche Présentation nous avons présenté cette figure :*

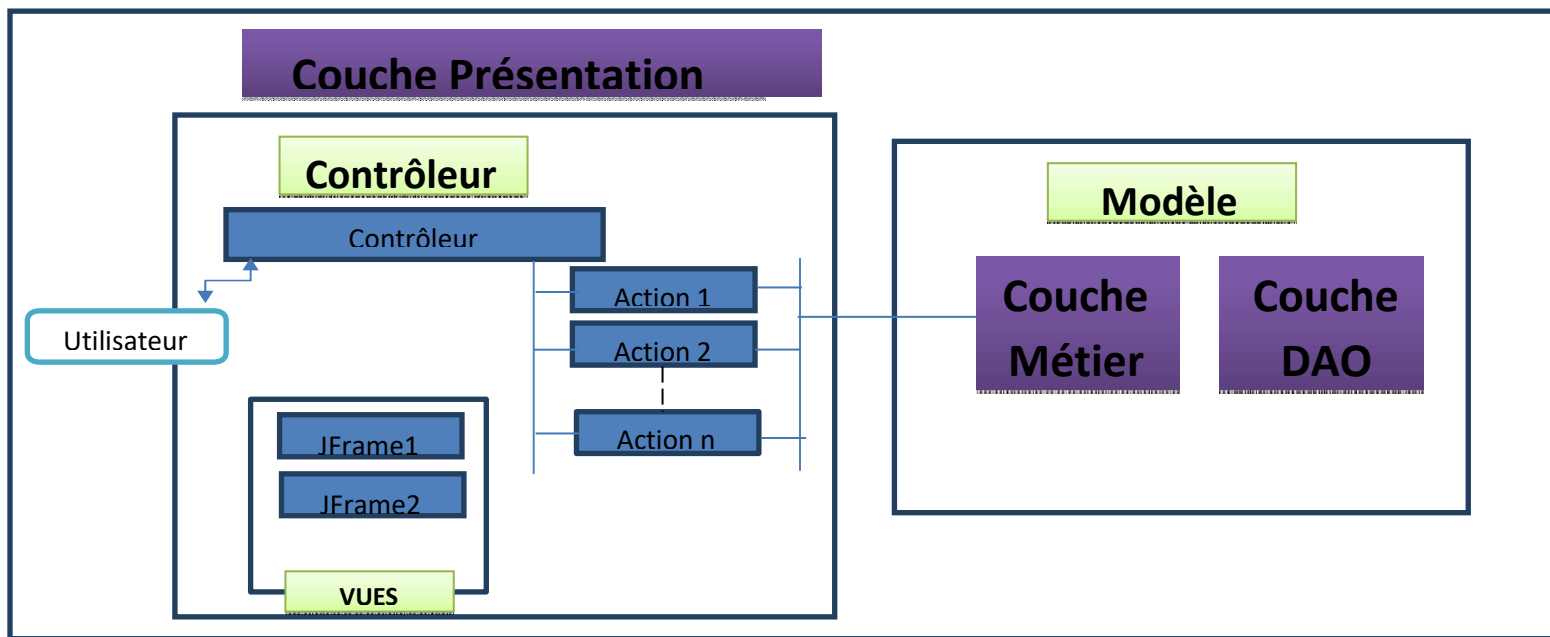


Figure 4: Architecture Détaillé pour l'implémentation de couche présentation

## 2. Conception détaillée :

Nous effectuons dans ce qui suit les diagrammes des cas d'utilisations détaillées de chaque partie, les diagrammes de séquence pour quelques cas d'utilisation et un diagramme de classe détaillé.

## 2.1. Diagrammes des cas d'utilisation :

Dans les Figures suivantes nous illustrons le diagramme des cas d'utilisation raffinés pour chaque partie.

### A. Partie Front Office :

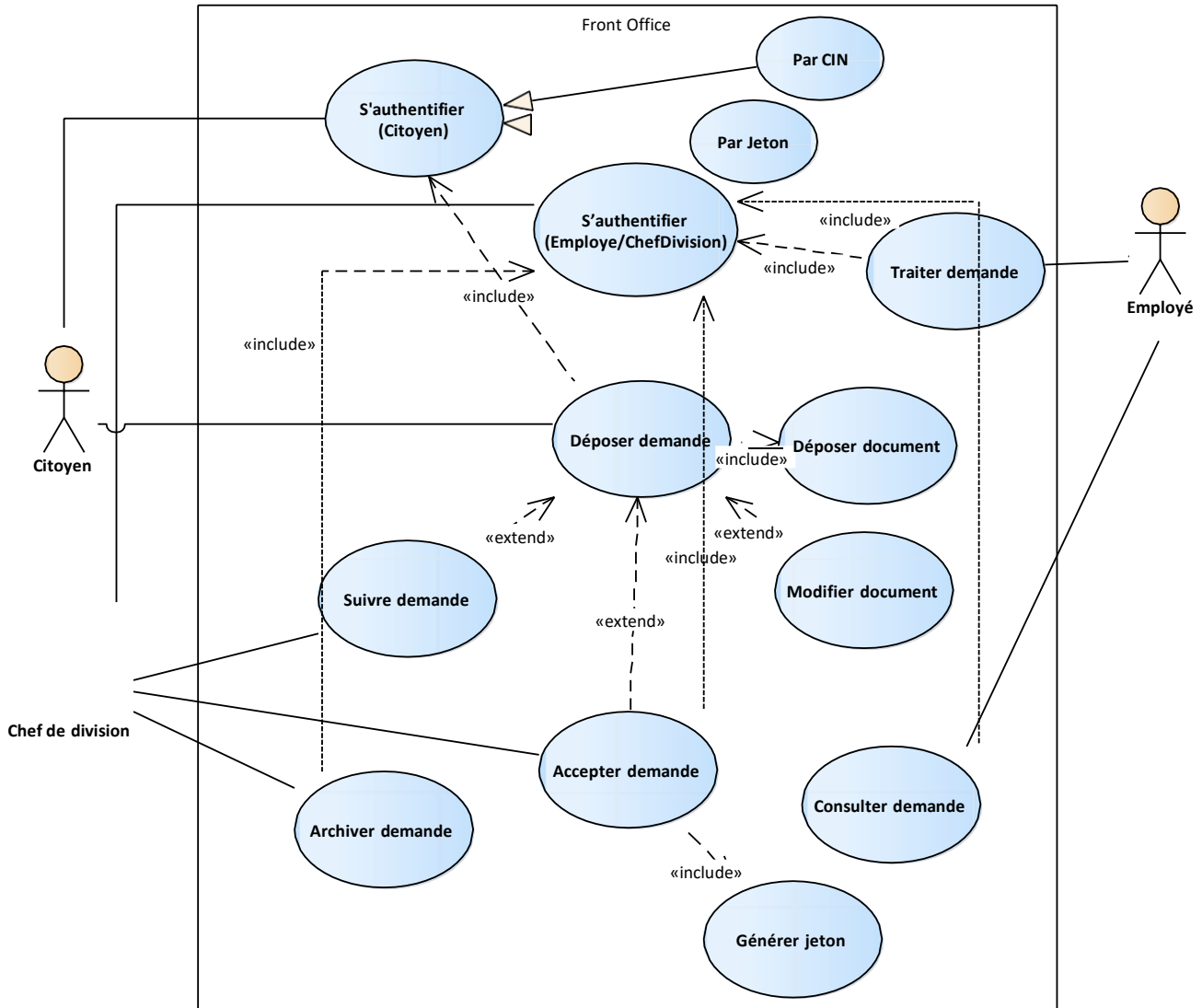


Figure 5 : Diagramme de cas d'utilisation détaillé de Front Office

## B. Partie Back Office :

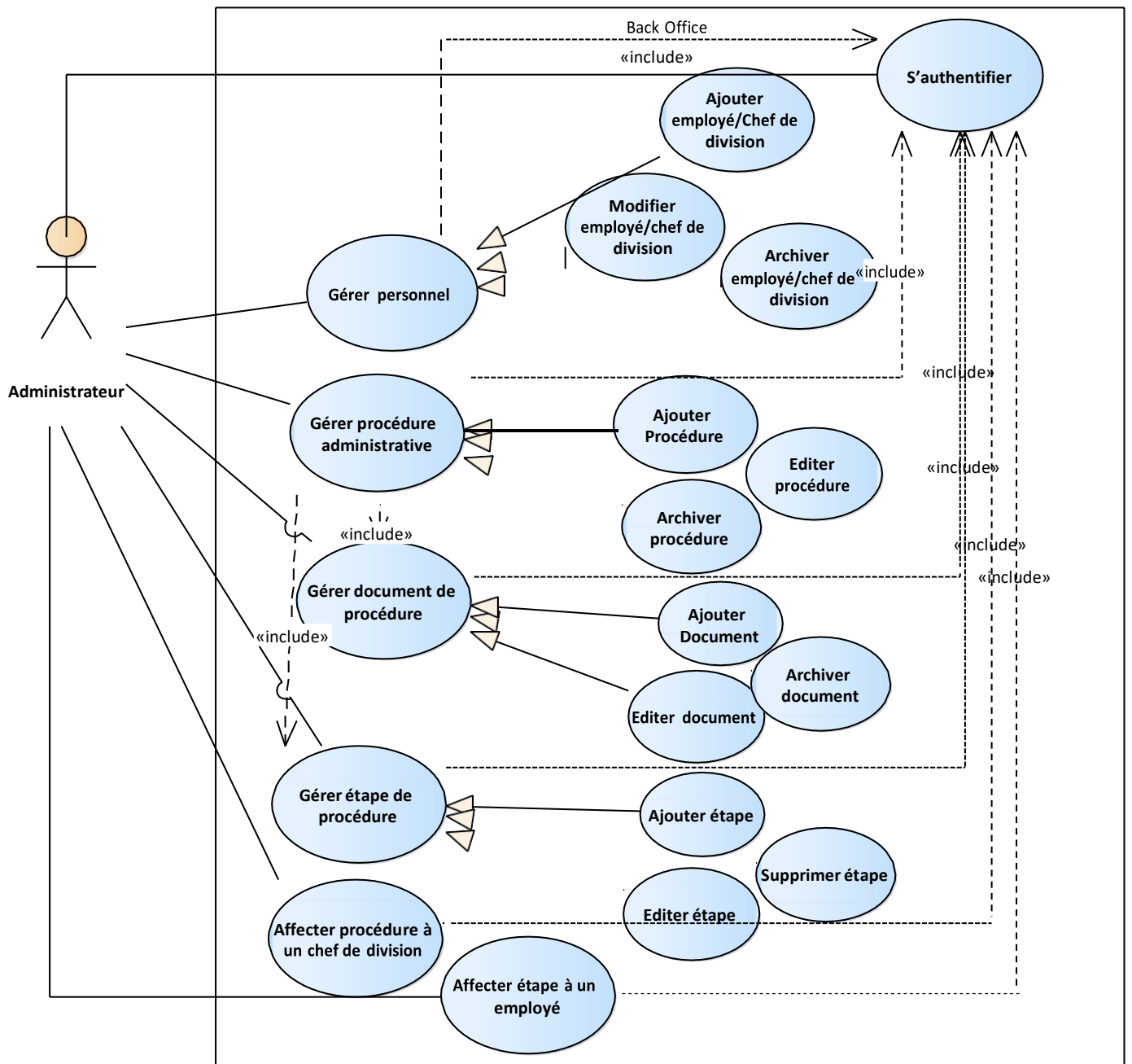


Figure 6: Diagramme de cas d'utilisation détaillé de Back Office

### C. Partie Reporting :

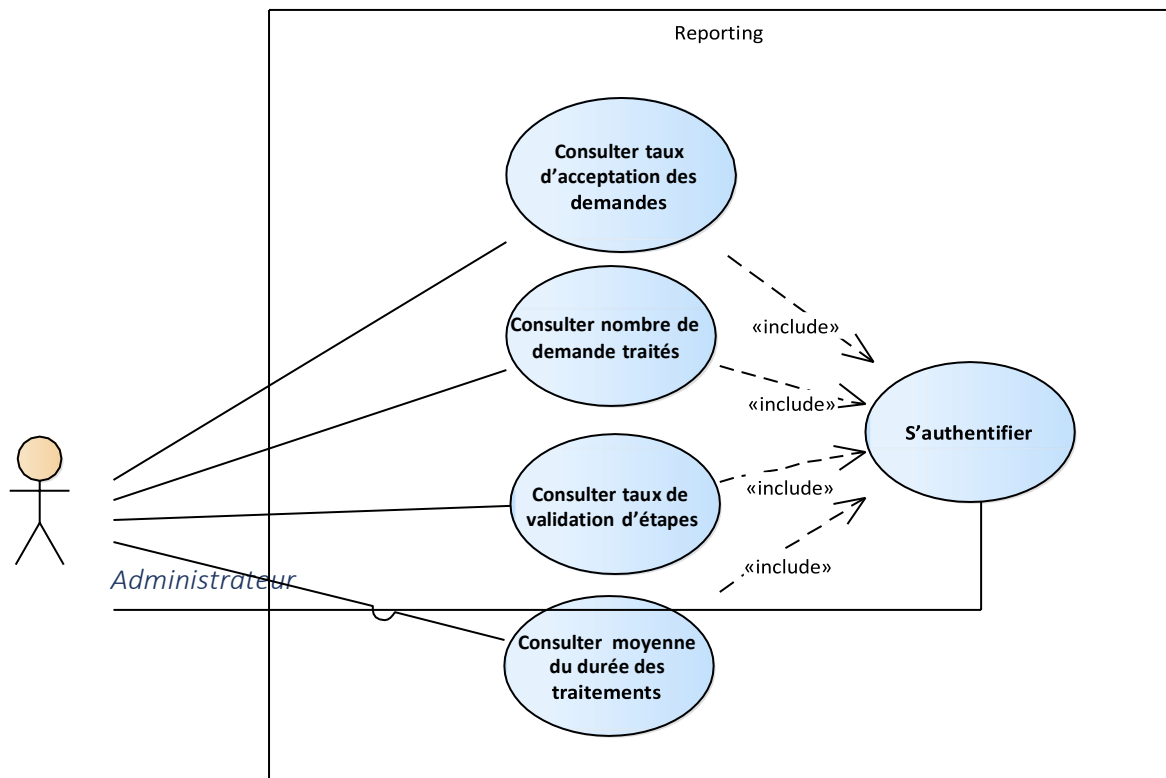


Figure 7: Diagramme de cas d'utilisation détaillé de Reporting

### 2.2. Diagrammes de séquence :

*En nous référant aux descriptions textuelles dans la livraison précédente, nous présentons les diagrammes de séquences adéquats. Sur cette base nous exposons les cas d'utilisation vus précédemment.*

- **Diagramme de séquence de cas d'utilisation « S'authentifier » :**



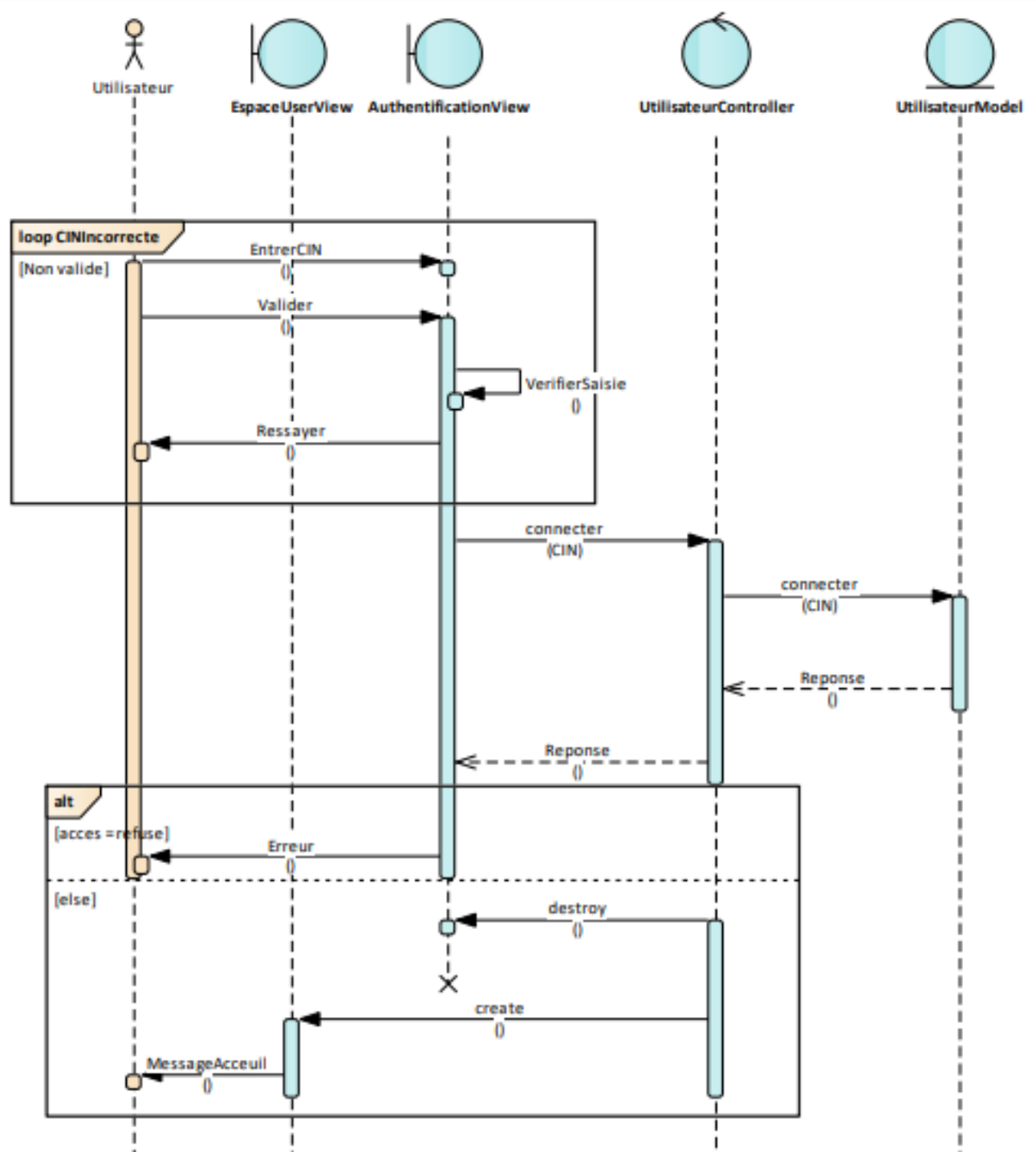


Figure 8: Diagramme de séquence s'authentifier

## Diagramme de séquence de cas d'utilisation « Déposer Demande » :

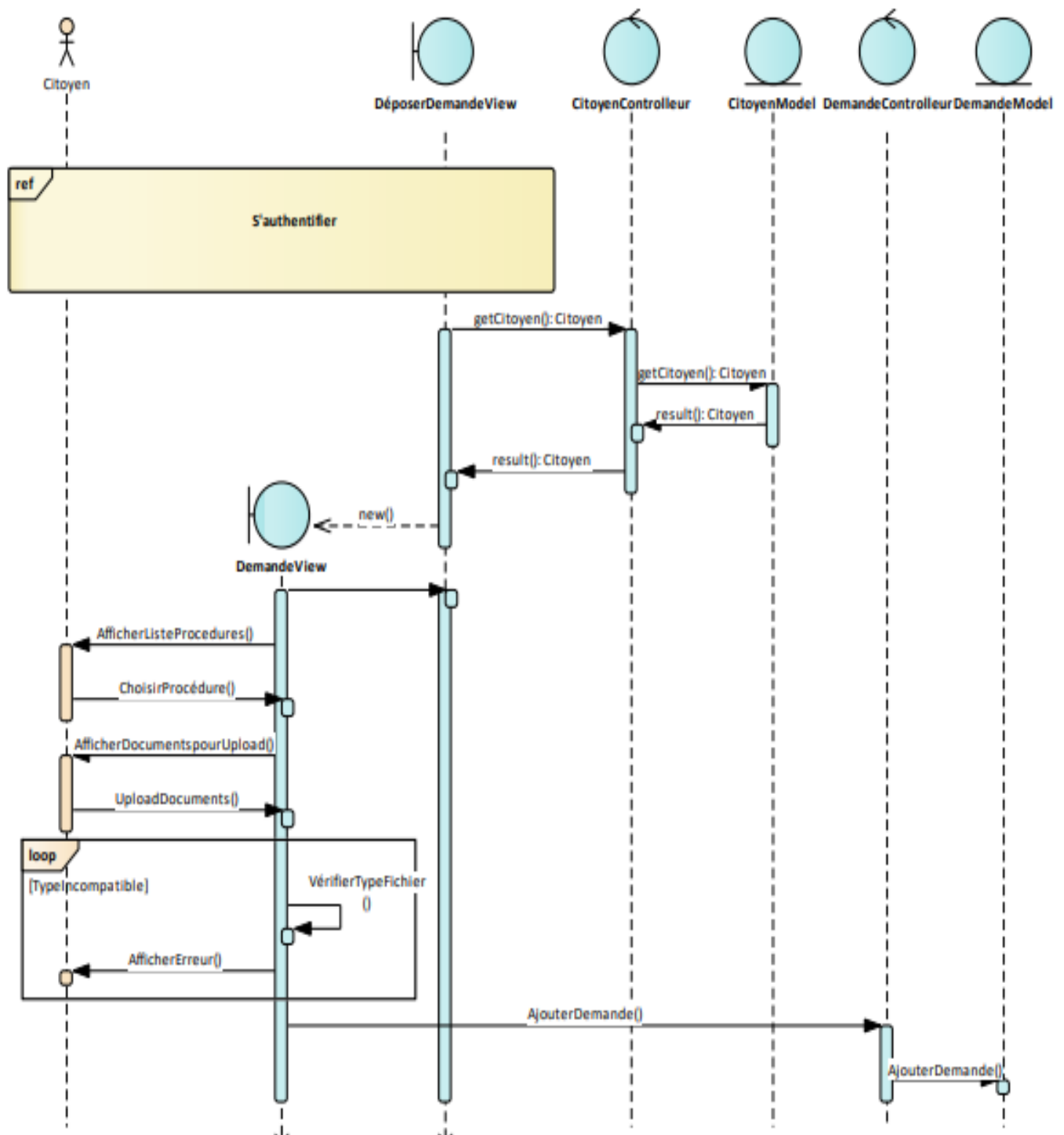


Figure 9: Diagramme de séquence Déposer Demande

➤ **Diagramme de séquence de cas d'utilisation « Ajouter Procédure » :**

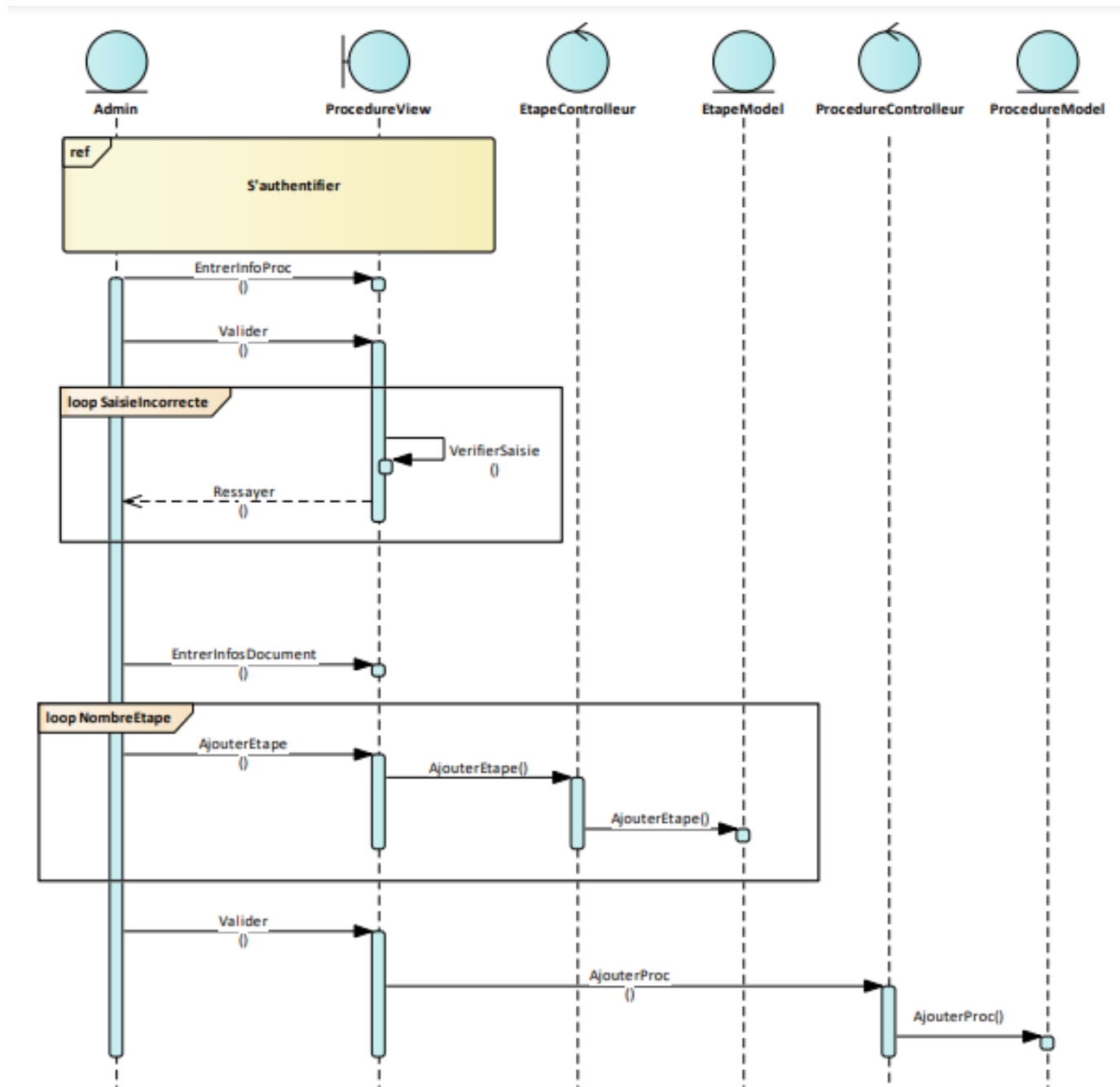


Figure 10: Diagramme de séquence Ajouter Procédure

➤ **Diagramme de séquence de cas d'utilisation « Suivre Demande » :**

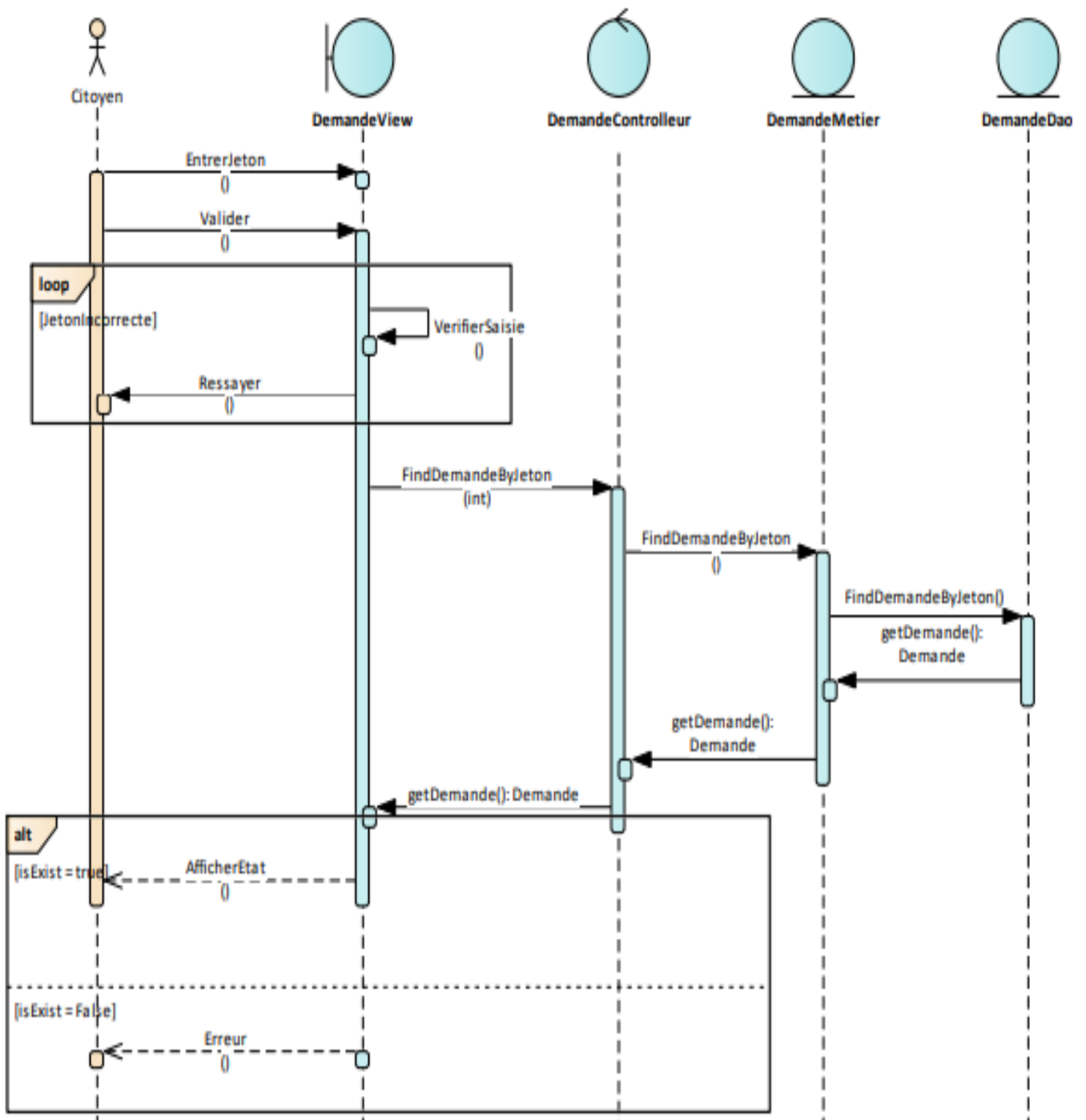


Figure 11: Diagramme de séquence Suivre Demande

## **2.1. Diagramme de classe détaillé :**

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

