

# Application Web de Gestion de tâche collaborative

---

Projet tuteuré

GOURA NACIM, DELETRAZ THOMAS, DURILLON QUENTIN  
MICROSOFT

# Sommaire

## I. [Principe](#)

- A. [Objectif](#)
- B. [Les tâches](#)
- C. [Les actions](#)
- D. [Fonctionnalités complémentaires](#)
- E. [Utilisateurs](#)
- F. [La collaboration](#)
- G. [Schéma récapitulatif](#)

## II. [Technologies employées](#)

- A. [Présentation](#)
- B. [Comparaison](#)

## III. [Diagramme UML](#)

- A. [Diagramme de cas d' utilisation](#)
- B. [Diagramme de séquence](#)

## IV. [Annexe](#)

# **I - Principe**

## **A. Objectif**

Le but de l'application est de permettre à l'utilisateur de **planifier des tâches** (Administratives, professionnelles, personnelles ou autres) au cours d'une année.

Il aura aussi la possibilité de **réaliser ou d'automatiser des actions** qui lui permettront de suivre et de poursuivre le bon déroulement de ces tâches jusqu'à leur achèvement.

L'application est orientée principalement pour aider à la gestion de tâches administratives au sein d'un établissement scolaire et donc de faciliter les échanges entre élèves, professeurs et directeurs (de formation par exemple).

## **B. Les tâches**

Il est important de définir comment sera spécifiée une tâche. Une tâche possédera une **catégorie**, un **nom**, un **descriptif**, une **date** et une **durée**. Les catégories seront déterminées par l'utilisateur, elles permettront de créer des ensembles et ainsi de les filtrer à l'affichage. Il y aura un système d'inclusion de tâches afin de créer de sous-tâches et ceci à l'infini. Chaque tâche pourra être associée à un ou plusieurs tags, ce qui fournira une nouvelle façon de filtrer l'affichage.

Une tâche pourra être **individuelle** ou **collaborative**. Tout dépendra si les actions à réaliser pour les valider sont liées à l'utilisateur ou aux autres protagonistes.

Il sera facile **d'ajouter** des tâches ou de les **modifier**, c'est à dire de les **déplacer** dans le temps ou de faire **varier** leur durée. On pourra aussi les **supprimer** ou les **archiver**.

Un **système de visualisation pratique** servira à surveiller les tâches en cours et à venir, type de diagramme de Gantt.

## C. Les actions

Les actions, qui représentent toute l'originalité du projet, vont permettre de **subvenir** aux tâches. Elles seront accessibles à l'intérieur des tâches elles-mêmes. On pourra les organiser de **manière chronologique** et les **déclencher manuellement** ou **automatiquement**.

Elles sont recensées actuellement dans la liste suivante :

- un **envoi de mail** à une ou plusieurs personnes
- un **envoi automatique de mail** à une ou plusieurs personnes
- un **rappel** personnel (alerte)
- une **relance** automatique à une ou plusieurs personnes
- un **envoi de document** à une ou plusieurs personnes
- une **question**
- une **demande de document** à une ou plusieurs personnes
- un **rendez-vous**
- la **consultation d'un lien**
- gérer une **check-list**
- écrire une **note**
- un **événement** (réunion, visite...)

Toute nouvelle action supplémentaire définie par le client sera implémentée si le temps le permet.

## D. Fonctionnalités complémentaires

L'application prévoit de **créer, modifier** et **supprimer** des **listes de diffusion** afin de les associer à des actions de tâches.

L'utilisateur aura la possibilité de **sauvegarder un modèle annuel** de tâches afin de le transférer vers une autre année.

Il existera un système de création de **comptes** et un système de gestion de **groupe de travail**, ce que nous expliquerons par la suite. Chaque compte possédera son propre **planning** et ses propres **paramètres**.

## **E. Utilisateurs**

On distingue trois types de personnes autour de l'application :

- l'**utilisateur principal**, possesseur d'un compte  
(Un directeur par exemple)
- les **collaborateurs**, possesseurs eux aussi d'un compte  
(Des professeurs par exemple)
- les **intervenants extérieurs**, non possesseurs de compte  
(Des élèves par exemple)

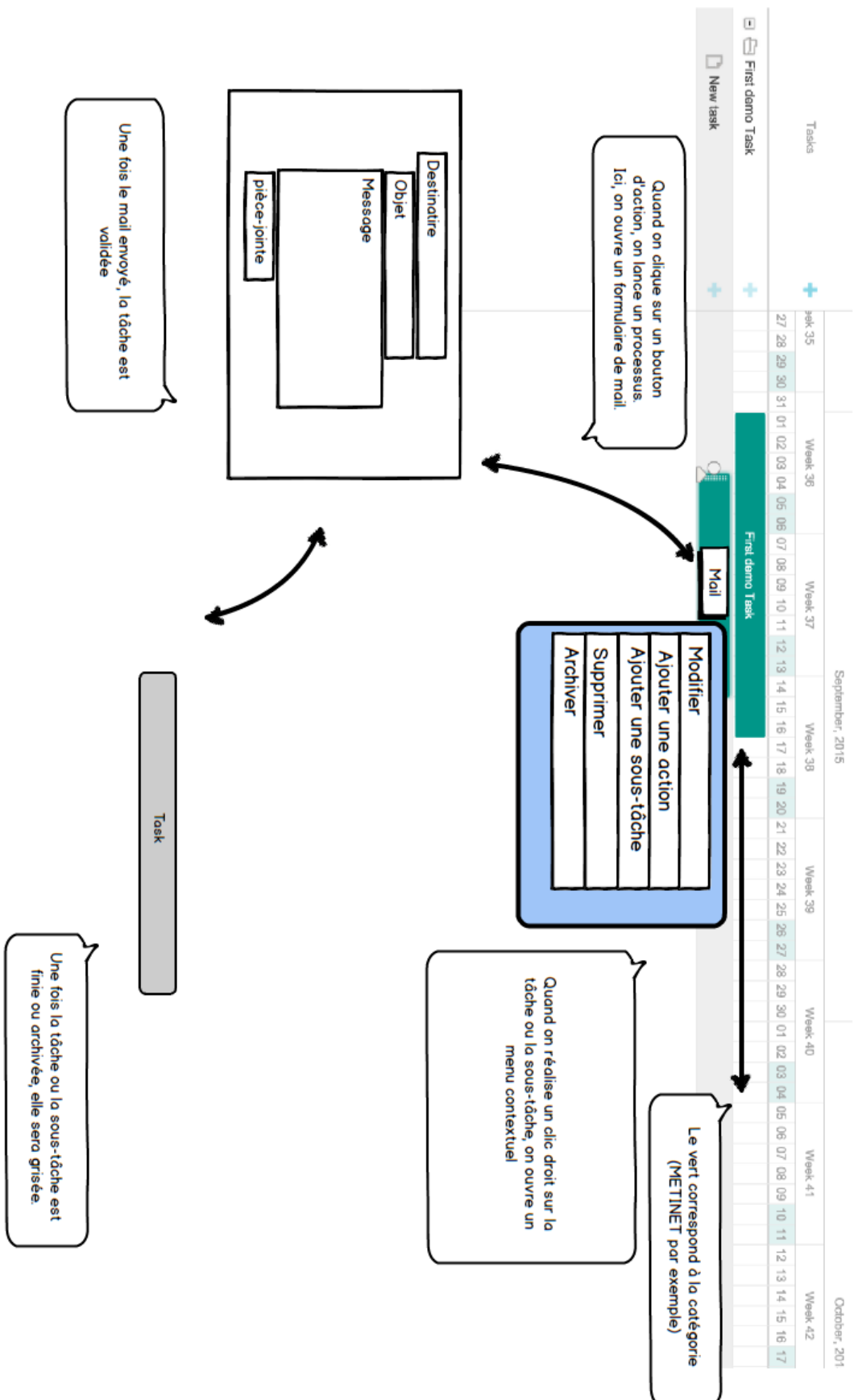
## **F. La collaboration**

La **notion collaborative**, deuxième maillon important du système, s'effectuera de deux manières : **interne** et **externe**.

Premièrement, l'utilisateur aura la possibilité d'**inviter un membre** de son groupe de travail (un collaborateur) à **participer à une tâche**, sollicitant ainsi son approbation. Une fois acceptée, la tâche sera ajoutée au planning du collaborateur. Ce dernier pourra ajouter une alerte à celle-ci, si besoin est.

Deuxièmement, certaines tâches ne seront terminées que quand les intervenants extérieurs auront **répondu à des actions** (telles que lors de la demande d'envoi de document). Ces "**retours**" **extérieurs** seront proposés et gérés par l'application grâce à des formulaires. Ils seront **surveillés** à partir du système et pourront donner lieu à des **actions supplémentaires** de l'utilisateur principal en les rajoutant à sa **tâche en cours** ou sous forme de **sous-tâche**.

### G. Schéma récapitulatif



## **II - Technologies employées :**

### **A. Présentation**

*Meteor* est un Framework open-source de développement Web réalisé en JavaScript et basé sur *Node.js*. Il permet d'utiliser le même langage niveau client et niveau serveur. Il offre aussi la possibilité de créer des applications multiplateformes : des web apps pour les navigateurs et des applications natives hybrides pour iOS et Android grâce à l'intégration de Cordova. Il gère l'actualisation des pages côté client donc agit comme une application desktop, rendant ainsi la navigation plus fluide. *Meteor* intègre son propre système de gestion des fonctionnalités serveur de NodeJS afin de profiter pleinement des ressources de celui-ci.

*Node.js* offre un environnement serveur différent des serveurs traditionnels par un système de mise à jour automatique de toutes ses pages. Cette technologie permet en effet d'actualiser une page de manière asynchrone, donc invisible pour l'utilisateur et d'afficher en temps réel les modifications extérieures à celui-ci.

*Meteor* intègre aussi *MongoDB*, un SGBD (système de gestion de base de données) orientée documents. Celui-ci permet de stocker des données semi-structurées, c'est-à-dire dont la structure n'est pas contrainte par un schéma mais dont le contenu est néanmoins formaté. L'unité de stockage est un document, généralement au format JSON ou XML.

Au niveau graphique, notre choix se porte sur l'utilisation du traditionnel couple *HTML5/CSS3*, complété du Framework Bootstrap. Les différents éléments graphiques et interactions dynamiques seront quant à elles réalisées grâce à des technologies telles que *D3.js*, une bibliothèque JavaScript qui permet d'afficher des données numériques. Nous réfléchissons encore à l'ajout de *Semantic UI*, un Framework Web ou encore *interact.js*, une bibliothèque JavaScript qui gèrent divers événements souris.

## **B. Comparaison avec les autres technologies**

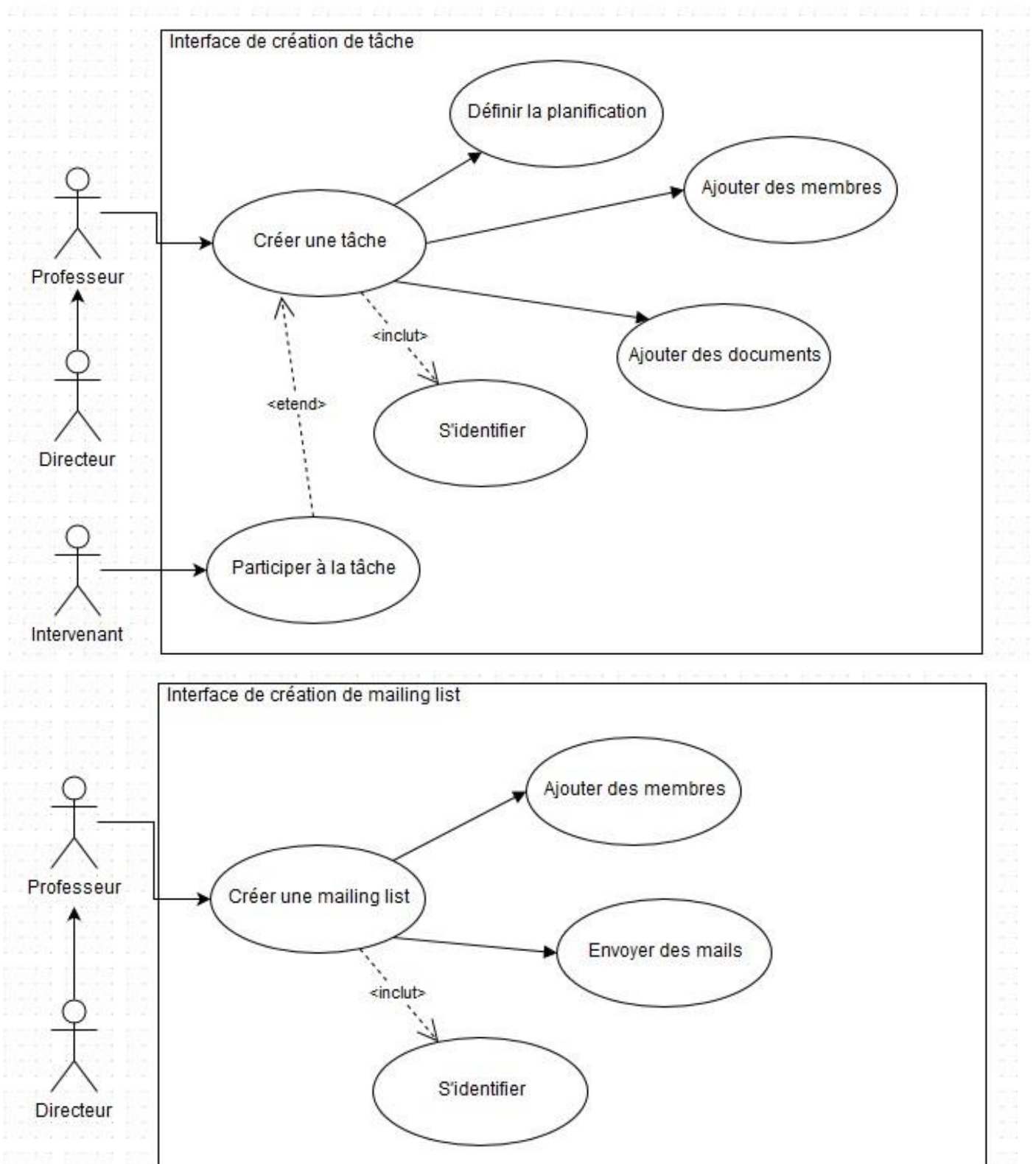
Nous avons comparé la technologie imposée avec les technologies existantes. La technologie se rapprochant le plus de notre projet est le couple *AngularJS* + *NodeJS*. *AngularJS* gère le côté client et en lui rajoutant *NodeJS*, il est aussi possible de réaliser les mêmes choses qu'avec *Meteor*. *Seulement*, la couche d'abstraction présente sur ce dernier simplifie grandement les choses afin de faire communiquer le serveur et le navigateur. Il aurait aussi été possible d'utiliser du *PHP* ou de l'*ASP.NET*, cependant cela aurait été trop contraignant à réaliser, et n'aurait pas été aussi fluide et réactif qu'en utilisant *Meteor*. Il semble donc que *Meteor* soit la meilleure solution pour le projet que nous devons réaliser.

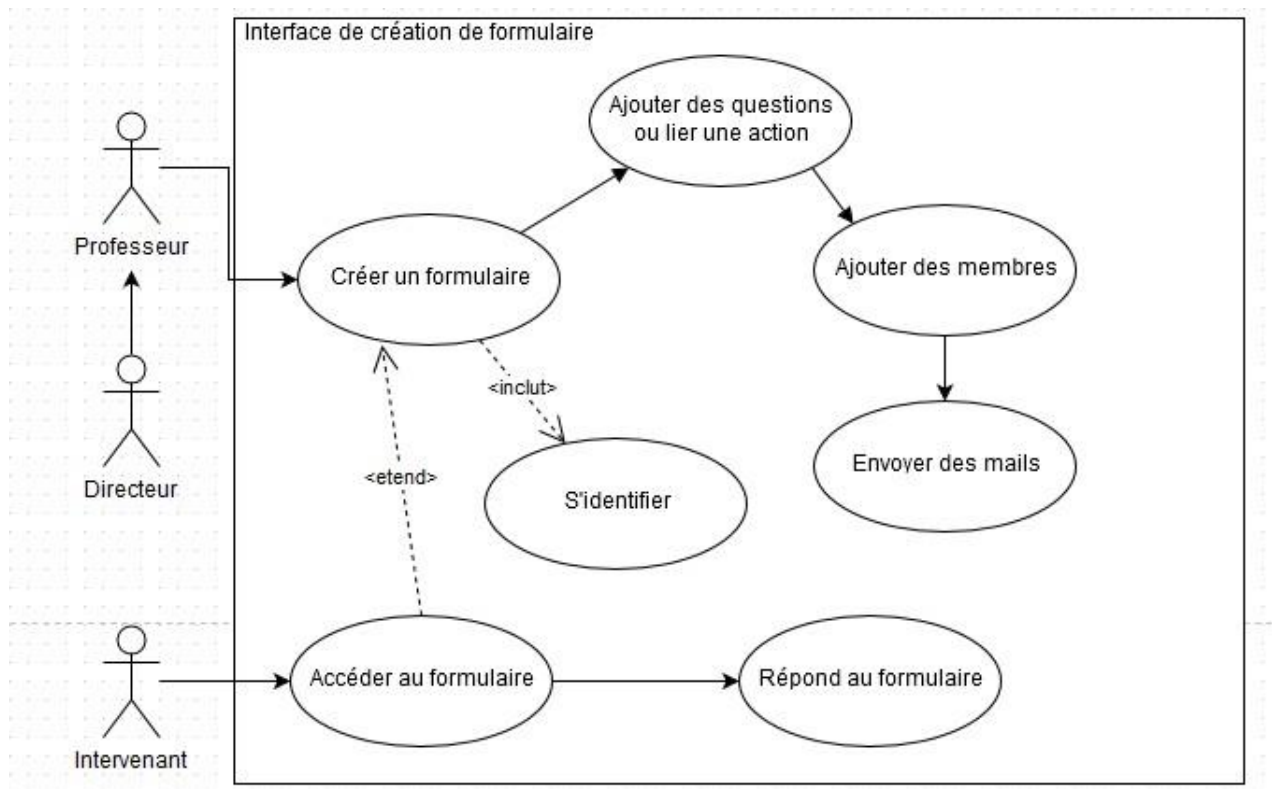
Nous pensons aussi qu'il s'agit d'une technologie pleine d'avenir dans le développement Web et qu'il serait bon de savoir la maîtriser pour de futurs projets.



## IV – Diagramme UML

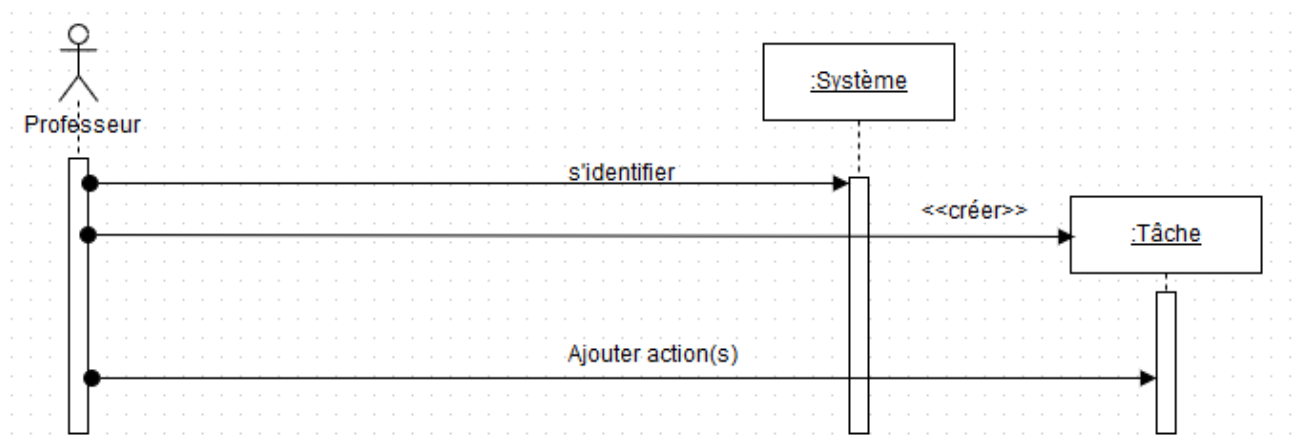
### A. Diagramme de cas d'utilisation



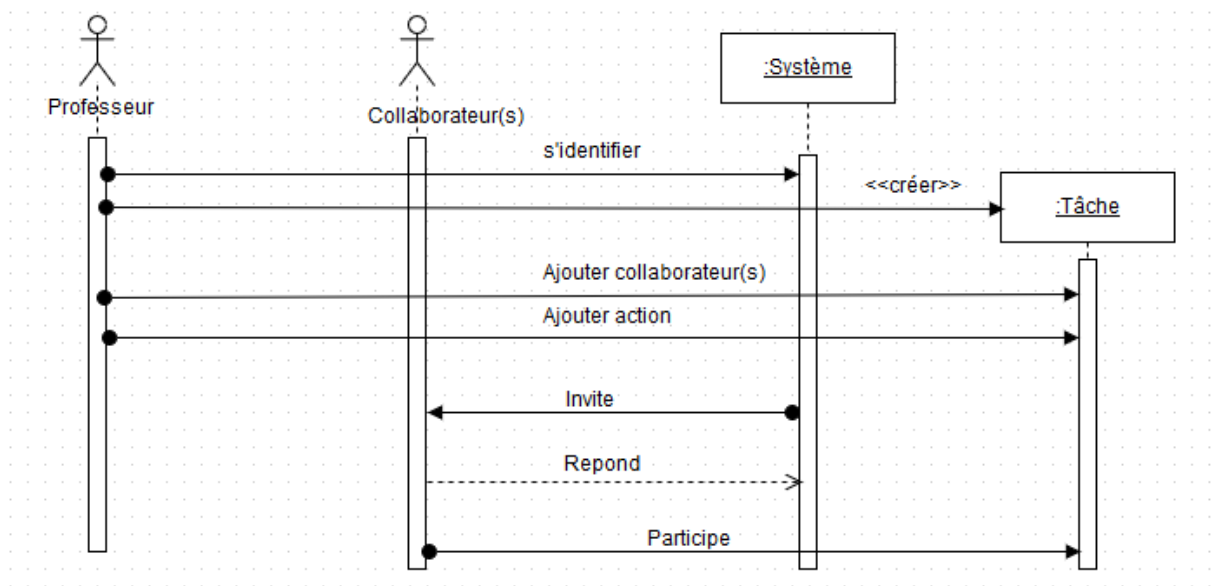


## B. Diagramme de séquence de création de tâche

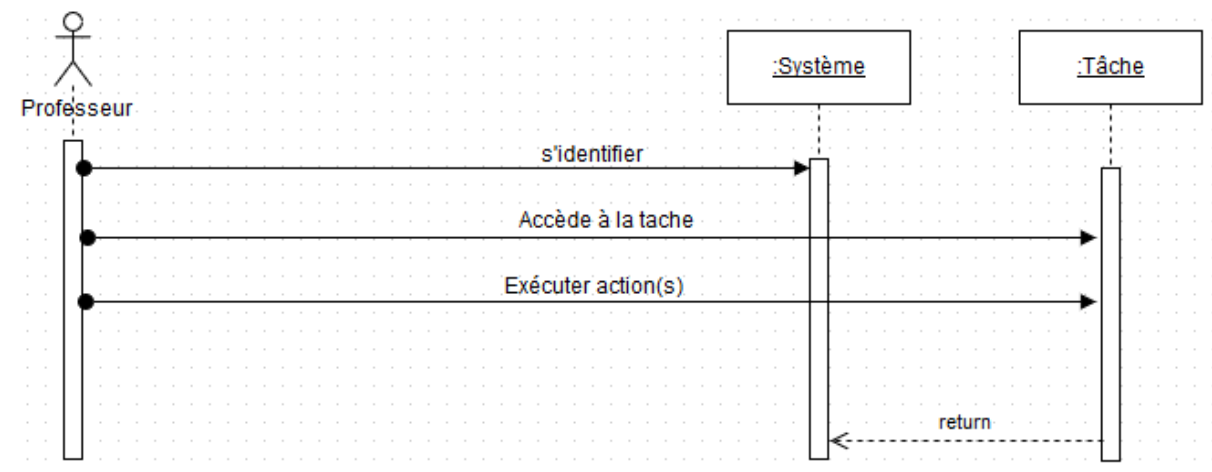
### 1- Diagramme de séquence de création de tâche individuelle sans intervenant



## 2- Diagramme de séquence de création de tâche collaborative



## 3- Diagramme de séquence de réalisation de tâche individuelle sans intervenant



## 4- Diagramme de séquence de réalisation de tâche individuelle avec intervenant(s)

