

# Rapport d'Iteration – Projet Application de Gestion de Tâches

Projet : Application de Gestion de Tâches

Itération : 1

Groupe : *Marchal Enzo, Nacime Laghezali, Cabot Matthieu, Vincent Julien*

Date : 12/12/2025

---

## 1. Objectif de l'itération

L'objectif principal de cette itération était d'établir la base fonctionnelle de l'application en posant une architecture claire et évolutive permettant de gérer efficacement des projets, des colonnes ainsi que des tâches hiérarchisées. L'accent a été mis sur la mise en place des mécanismes essentiels au fonctionnement du système : création, suppression, visualisation et déplacement d'éléments.

Plus précisément, cette itération visait à :

- Structurer la hiérarchie **tâche / sous-tâche** afin de permettre une organisation détaillée du travail ;
- Développer les fonctionnalités fondamentales du noyau applicatif, notamment :
  - L'ajout et la suppression **d'un projet**,
  - L'ajout et la suppression **d'une colonne**,
  - L'ajout et la suppression **d'une tâche**,
  - Le **déplacement** d'une tâche d'une colonne à une autre,
  - La **visualisation** des projets, colonnes et tâches,
  - L'ajout de **dépendances** entre tâches,
  - Le **changement d'état** d'une tâche (ex. : A FAIRE → EN COURS → TERMINÉE).

L'ensemble de ces fonctionnalités a été testé via **une application textuelle**, permettant d'obtenir rapidement un retour utilisateur simple et efficace pour valider les règles de gestion.

## 2. Fonctionnalités et cas d'utilisation

## **2.1 Liste des fonctionnalités développées**

Dans cette itération, plusieurs fonctionnalités essentielles ont été implémentées afin de constituer la base du fonctionnement de l'application. Chaque fonctionnalité ci-dessous est décrite selon quatre axes : son objectif, son déroulement dans l'application, l'utilisateur impliqué et le membre du groupe responsable de son développement.

### **Ajout d'un projet**

Objectif : permettre la création d'un nouveau projet vide dans l'application.

Déroulement : l'utilisateur saisit un nom ; le système crée l'objet projet et l'intègre à la liste des projets.

Utilisateur : utilisateur principal de l'application.

Développement : *Nacime Laghezali*

### **Suppression d'un projet**

Objectif : retirer un projet existant et toutes ses données associées.

Déroulement : l'utilisateur sélectionne un projet ; l'application le supprime de la structure interne.

Utilisateur : utilisateur principal.

Développement : *Nacime Laghezali*

### **Ajout d'une colonne**

Objectif : permettre au projet de disposer de colonnes (ex : A FAIRE, EN COURS...).

Déroulement : l'utilisateur choisit un projet et ajoute une colonne ; l'objet colonne est créé et inséré dans le projet.

Utilisateur : utilisateur principal.

Développement : *Matthieu Cabot*

### **Suppression d'une colonne**

Objectif : retirer une colonne et les tâches qu'elle contient.

Déroulement : l'utilisateur sélectionne la colonne ; le système la supprime du projet actif.

Utilisateur : utilisateur principal.

Développement : *Matthieu Cabot*

### **Ajout d'une tâche / sous-tâche**

Objectif : permettre de construire une hiérarchie de tâches et sous-tâches.

Déroulement : l'utilisateur choisit une colonne, saisit les informations de la tâche ; l'objet tâche est créé et rattaché à l'élément parent (colonne ou tâche).

Utilisateur : utilisateur principal.

Développement : *Julien Vincent*

### **Suppression d'une tâche**

Objectif : supprimer une tâche ou sous-tâche de la hiérarchie.

Déroulement : l'utilisateur sélectionne une tâche dans une colonne ; elle est supprimée de la structure.

Utilisateur : utilisateur principal.

Développement : *Julien Vincent*

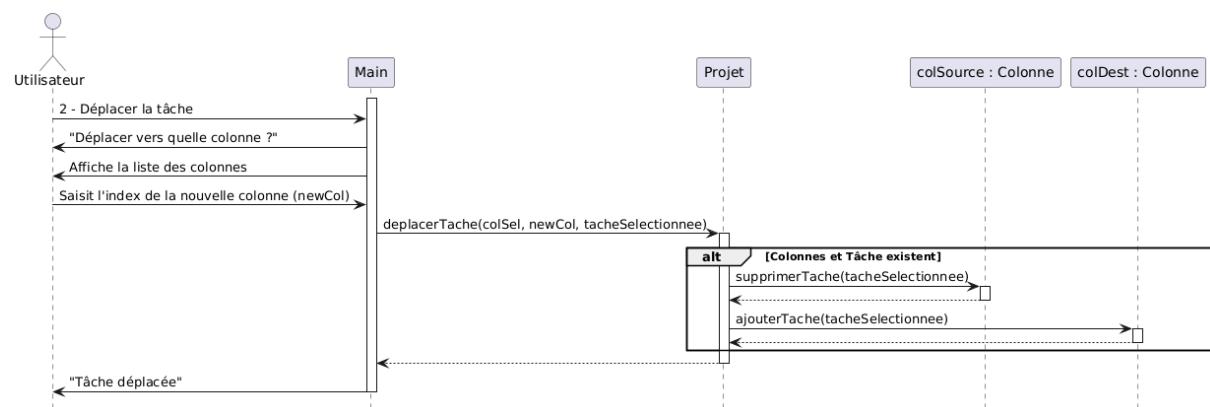
### Déplacement d'une tâche

Objectif : changer la colonne d'appartenance d'une tâche (ex : A FAIRE → EN COURS).

Déroulement : l'utilisateur sélectionne une tâche, choisit la colonne cible ; le système met à jour la structure.

Utilisateur : utilisateur principal.

Développement : *Julien Vincent*



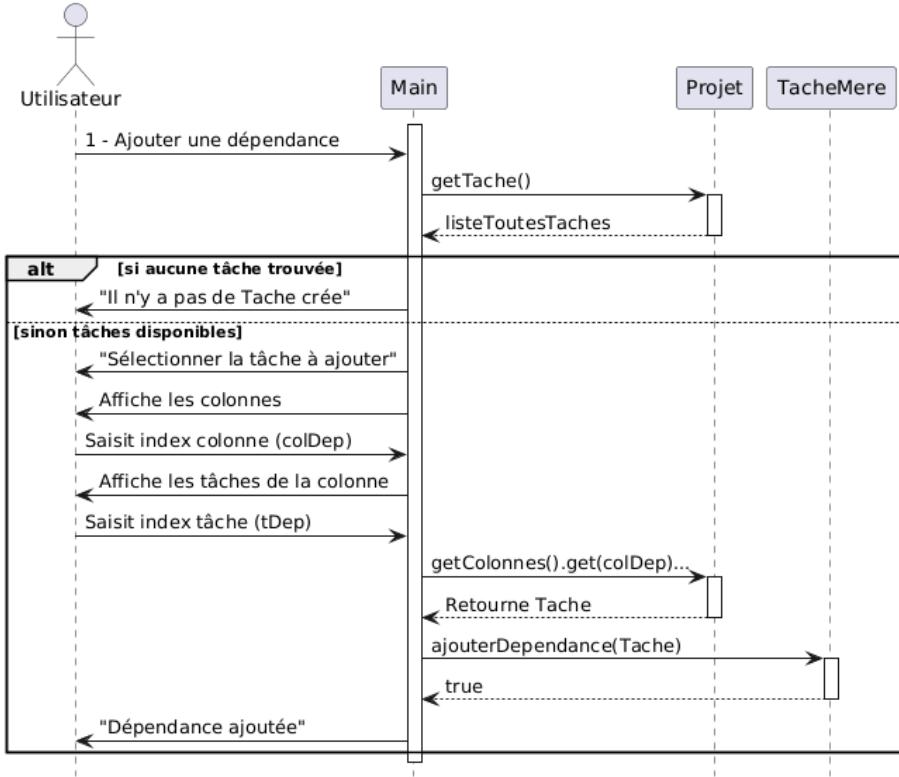
### Ajout d'une dépendance

Objectif : exprimer un lien logique entre deux tâches.

Déroulement : l'utilisateur sélectionne deux tâches ; une relation de dépendance est créée.

Utilisateur : utilisateur principal.

Développement : Enzo Marchal



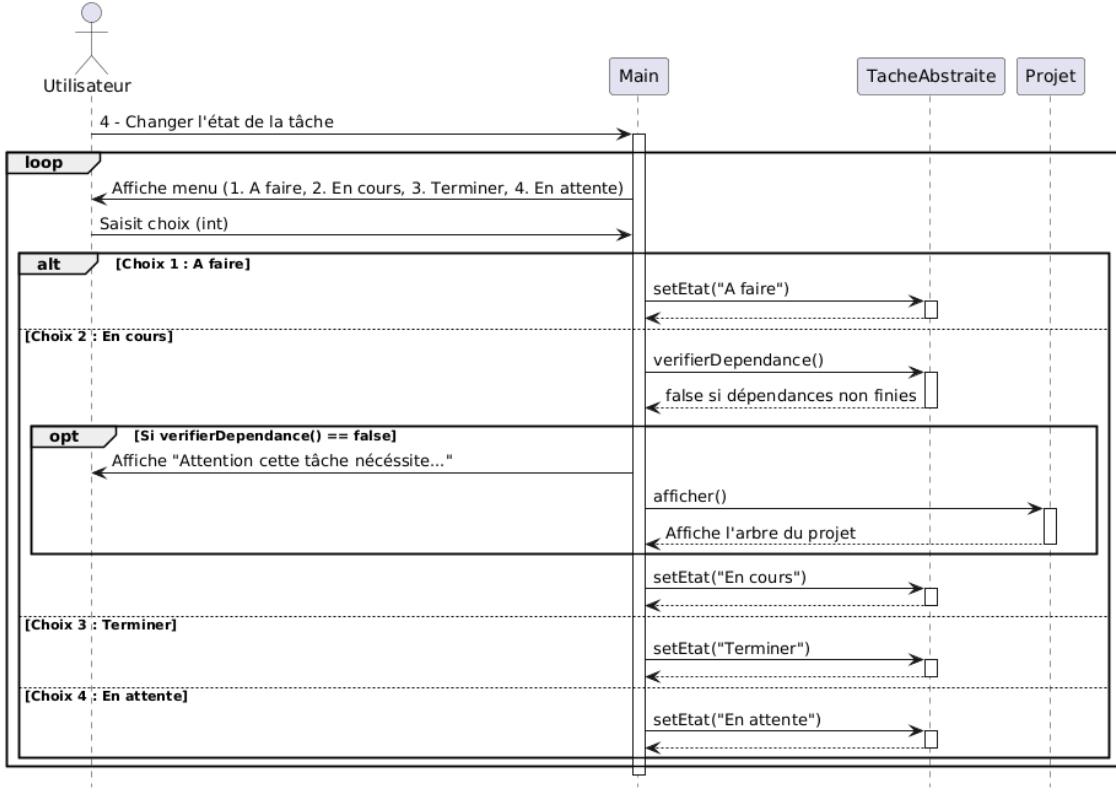
## Changement de l'état d'une tâche

Objectif : faire évoluer une tâche dans son cycle de vie (A FAIRE, EN COURS, TERMINÉE).

Déroulement : l'utilisateur sélectionne une tâche ; une mise à jour de son état interne est effectuée.

Utilisateur : utilisateur principal.

Développement : Enzo Marchal



## Visualisation textuelle

Objectif : permettre à l'utilisateur de consulter la structure complète d'un projet en mode texte.

Déroulement : l'application affiche les projets, colonnes, tâches et sous-tâches de manière hiérarchique.

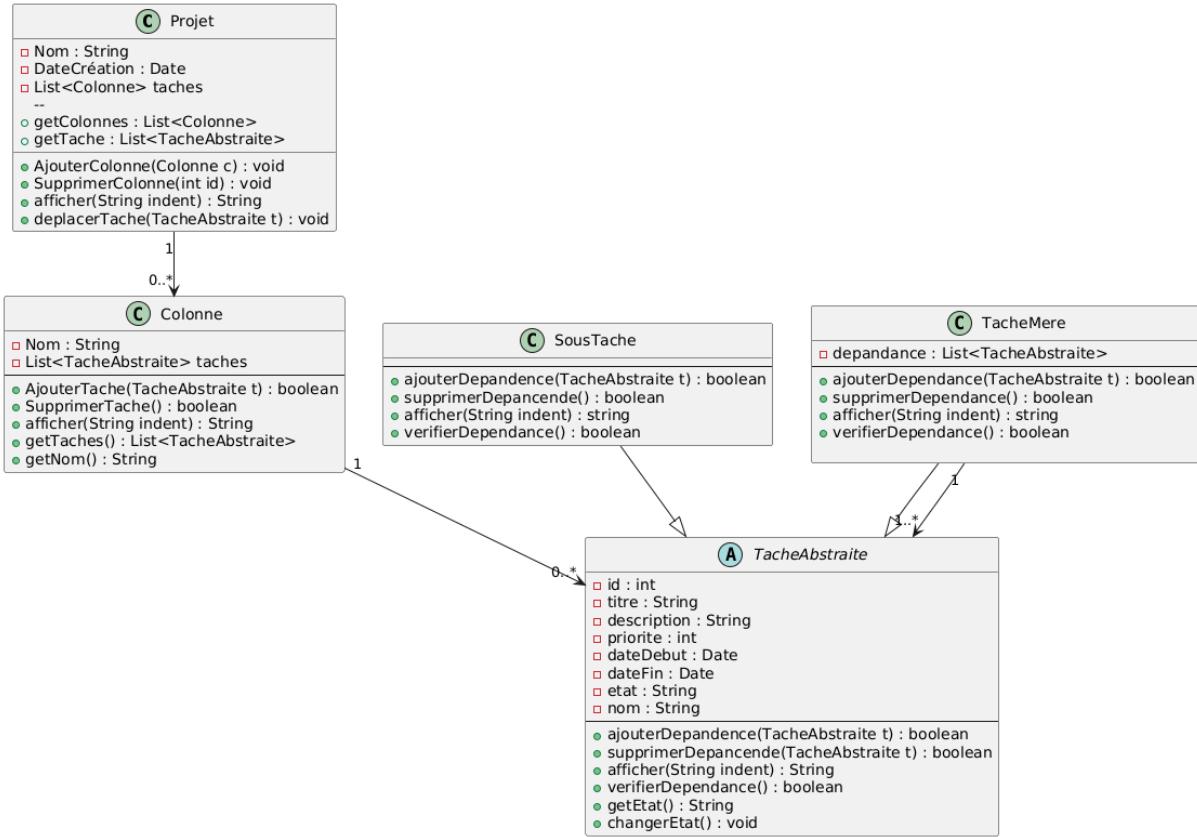
Utilisateur : utilisateur principal.

Développement : Enzo Marchal

## 3. Diagramme de classes et choix de conception

Un diagramme de classes global représentant :

- La structure des projets, colonnes et tâches,
- La hiérarchie tâche/sous-tâche,
- Les dépendances entre les classes,
- Les relations d'agrégation, composition ou association.



### 3.1 Choix de conception

L'architecture retenue pour cette itération repose sur une structure claire et hiérarchisée permettant d'assurer l'évolutivité et la cohérence de l'application. Le modèle suit une organisation centralisée : Projet → Colonnes → Tâches → Sous-tâches. Cette approche facilite la navigation, la manipulation des données et leur structuration logique.

#### Logique d'architecture

Le projet constitue l'élément racine regroupant l'ensemble des colonnes. Chaque colonne contient une liste de tâches, lesquelles peuvent elles-mêmes contenir des sous-tâches, créant une hiérarchie naturelle et intuitive. Ce découpage rend possible :

- La modularité des fonctionnalités,
- Une gestion isolée des éléments (projets, colonnes, tâches),
- Une extension simple pour les futures itérations (ex : affichage graphique, gestion avancée des dépendances).

#### Patron utilisé

Un patron de conception a été utilisé pour cette première itération :

- **Patron Composite** : il s'applique naturellement à la hiérarchie *tâche / sous-tâche*, permettant de manipuler l'ensemble comme une structure arborescente uniforme.

### Conformité avec l'étude préalable

Les choix effectués dans cette itération sont conformes à l'étude préalable :

- La structure hiérarchique projets-colonnes-tâches a été respectée,
- Les fonctionnalités essentielles prévues initialement ont été implémentées,
- Les bases architecturales posées permettent une évolution cohérente vers les itérations suivantes.

## 4. Bilan de l'itération

Cette première itération nous a permis de poser les bases pour le développement de l'application de gestion de tâches. L'ensemble de l'architecture centrale projets, colonnes, tâches et sous-tâches a été défini et implanté conformément à l'étude préalable, permettant d'obtenir un modèle cohérent, extensible et prêt à pour les évolutions futures. Les fonctionnalités essentielles ont été réalisées avec succès : création, suppression et visualisation des éléments, gestion des dépendances, déplacement des tâches et changement d'état. Nous avons utilisé le patron Composite pour structurer la hiérarchie de tâches. Nous avons aussi effectué des tests via l'interface textuelle et les tests JUnit qui ont confirmé le bon fonctionnement de l'application gestion de tâche. Cette itération constitue ainsi une base fiable sur lequel les prochaines fonctionnalités pourront être implémenter dans les prochaines itérations.

## 5 Bugs

Nous n'avons pas rencontré de réel bug lors de cette itération.