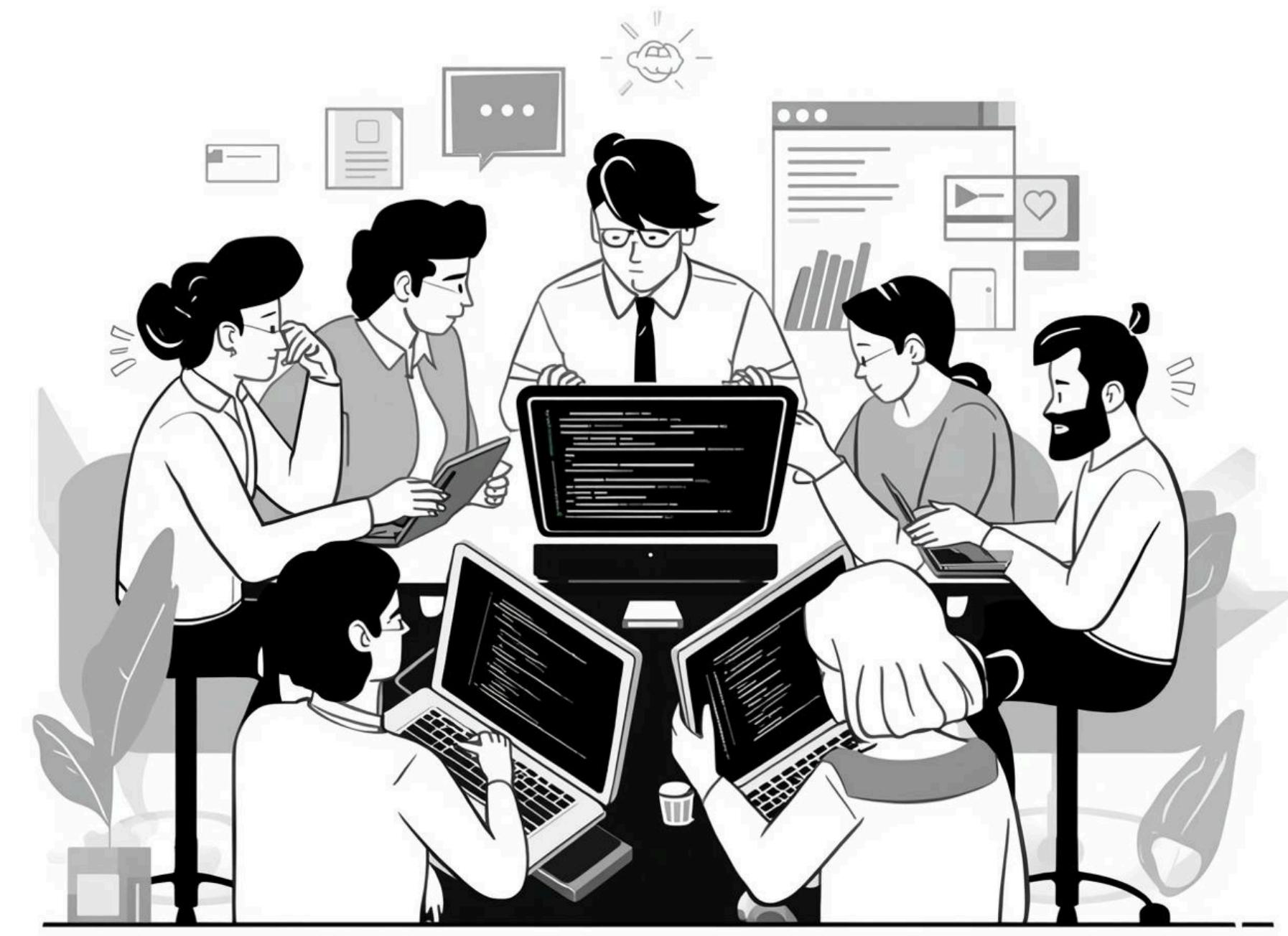


OLLERT

Rétrospection



Comparaison entre l'étude fait au préalable et le rendu final

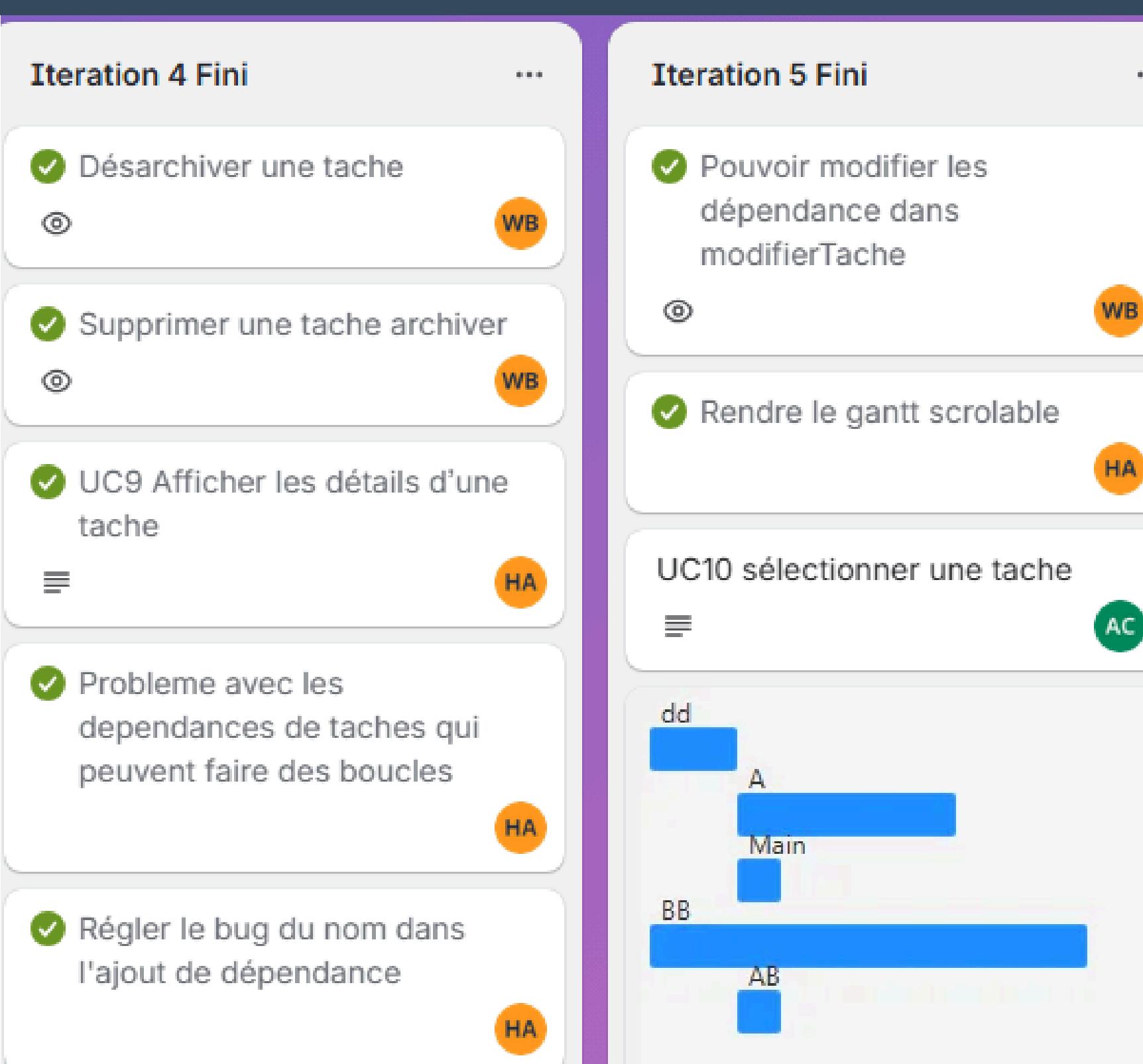
Dans la globalité - Planning d'itérations 4 et 5

Itération 4 : UC5, UC6, UC9, UC15

- Finaliser la vue « Bureau » de l'interface graphique
- Ajouter le système d'archivage des tâches
- Ajouter l'affichage des archives dans l'interface graphique (archiver une tâche et consulter les tâches archivées)
- Créer les tests pour l'interface graphique

Itération 5 : UC8, UC11

- Générer le diagramme de Gantt
- Tester la génération du diagramme de Gantt
- Ajouter la vue en liste
- Créer la vue d'affichage du diagramme de Gantt



Comparaison entre l'étude fait au préalable et le rendu final

Quelques différences..

PRÉVU

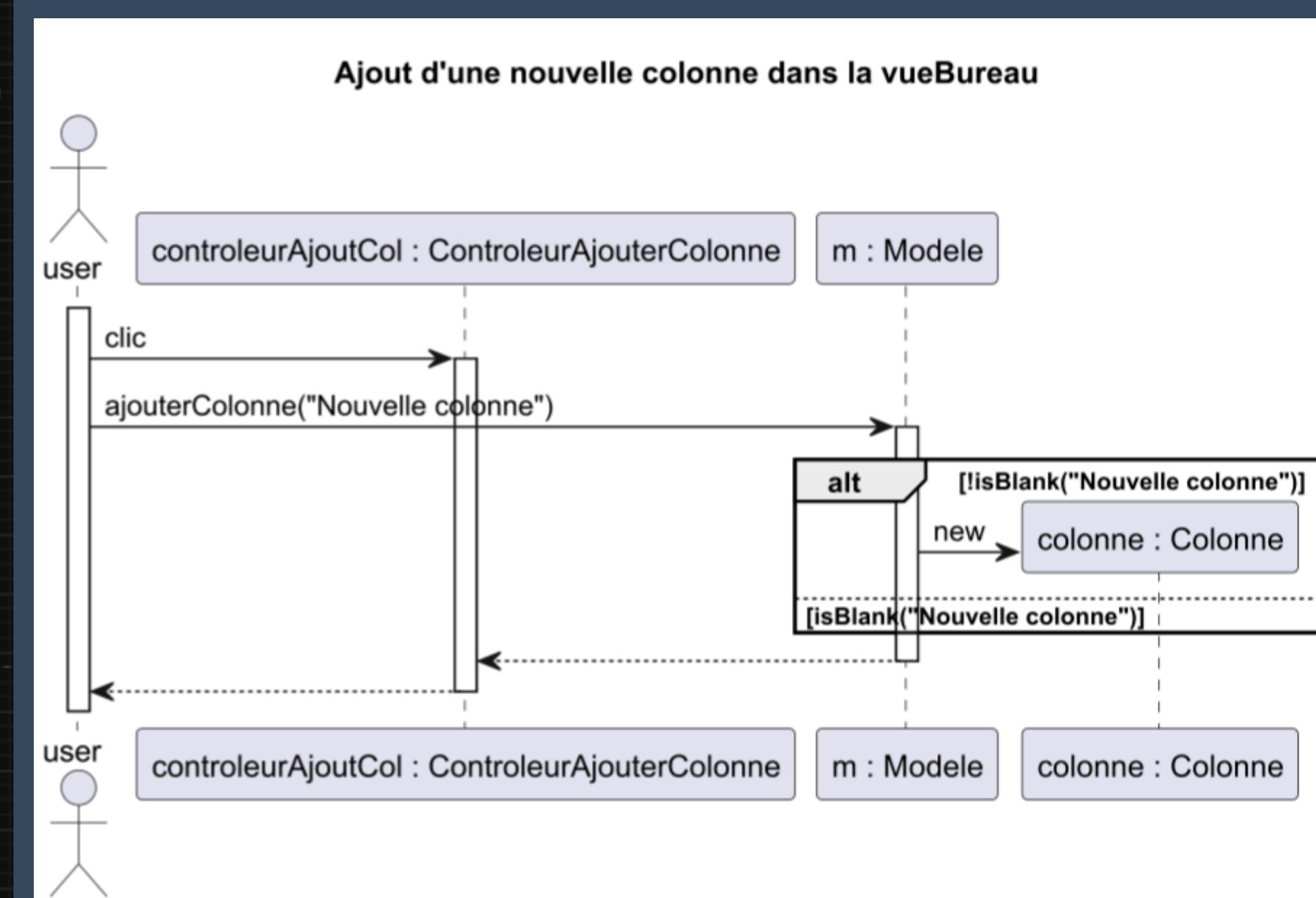
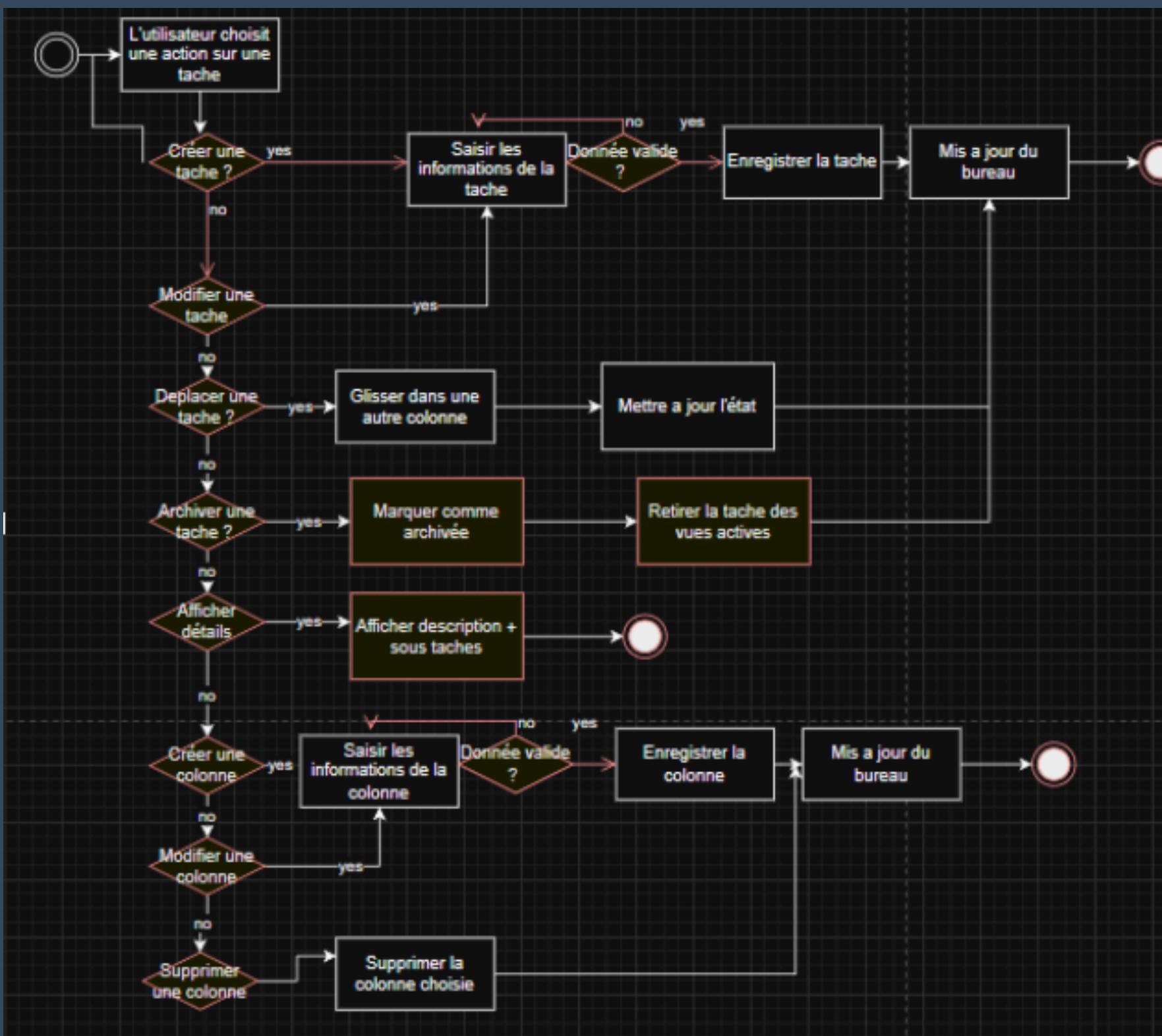
- Active Record - Base de donnée gérant la sauvegarde
- Un système d'états des tâches : Non fini, en pause, en cours, complétée

RÉALISÉ

- Repository Singleton - Sauvegarde dans un fichier texte
- Aucun système d'état n'a été fait

Comparaison entre l'étude fait au préalable et le rendu final

Quelques différences..



Iteration 1

Hugo

- Conception des classes de base Tache / TacheComposite
- UC2 : Création / ajout de sous-tâches.

Walid

- UC8 : Vue Liste : affichage des tâches sous forme de tableau.

Ilias

- Gestion des colonnes

Alexander

- Repository pour la sauvegarde des données dans un fichier
- Écriture de tests unitaires pour Repository

Iteration 2

Hugo

- UC3 : Définition d'une dépendance entre tâches.
- ControleurAjouterDependances
- Ajout des test junit pour les taches et dépendances
- ControleurModifierTache

Ilias

- Mise en place du modèle de données (classes principales + relations).

Walid

- UC4 : Modification d'une tâche (popup d'édition).
- Creation d'une tâche (popup d'édition).
- Vue Liste : changement de colonne d'une tâche directement dans le tableau.

Alexander

- Modification du Repository suite à la réunion de fin d'itération 1
- Début du développement de la vue bureau

Iteration 3

Hugo

- UC7 : Déplacement d'une tâche entre colonnes (drag & drop)
- ControleurAjouterTache
- ControleurAjouterSousTache

Walid

- UC5 : Archivage d'une tâche.
- UC15 : Affichage des tâches archivées (popup archive).

Ilias

- Gestion de l'Accueil (VueAccueil)

Alexander

- Finition de l'affichage des tâches et des colonnes dans la vue Bureau
- Création de nouvelle colonnes et Tâches par la vue bureau

Iteration 4

Hugo

- UC9 : Affichage des détails d'une tâche (popup détails)
- Problème identifié : dépendances pouvant former des boucles (cycles).
- Correctifs sur l'ajout de dépendances (bug de nom)

Walid

- Désarchivage d'une tâche (retour dans la première colonne).
- Suppression définitive d'une tâche archivée.

Ilias

- Ajout de la possibilité de Supprimer une Colonne qui archive les tâches à l'intérieur

Alexander

- Modification des nom des colonnes sur l'interface graphique

Iteration 5

Hugo

- Diagramme de Gantt scrollable (horizontal + vertical pour éviter le blocage écran)
- Gantt : tri / organisation visuelle des tâches liées aux mêmes dépendances
- UC11 : Génération du diagramme de Gantt
- Nettoyage des dépendances si une tâche supprimée/archivée était dépendance d'une autre (suppression des références)
- Affichage des sous-tâches dans le détail d'une tâche

Walid

- Correctif : bug désarchivage quand aucune colonne n'existe.
- Modification des dépendances dans "modifierTache".

Ilias

- Ajout du bouton “Archiver une tâche” pour chaque tâche

Alexander

- Modification de taches depuis la vue bureau
- Recherche de bugs
- Selection des taches pour le diagramme de Gantt
- Bouton pour sélectionner toutes les taches pour le diagramme de Gantt
- Ecriture de tests pour la class modele

Évolution de la conception

1. Départ : mise en place du cœur métier (Tache/TacheComposite) et des vues de base (création de tâche, liste).
2. Ajout progressif du Kanban : colonnes limitées + renommage + déplacement des tâches
3. Enrichissement des relations : dépendances et sous-tâches (avec contrôles dédiés).
4. Stabilisation métier : gestion des cas limites (cycles de dépendances, suppression/archivage qui doit nettoyer les références).
5. Ajout d'une visualisation avancée : Gantt basé sur calcul de dates “effectives” (dépendances) + sélection des tâches + ergonomie (scroll).

Patrons d'architecture et de conception utilisés

MVC (Modèle / Vues / Contrôleurs)

- Modèle porte l'état (colonnes, tâches, archive, sélection Gantt) et expose les méthodes métier (ajouter, modifier, archiver, dépendances...).
- Les vues (VueBureau, VueListes, VueGantt, popups) affichent et récupèrent les actions utilisateur.
- Les contrôleurs (EventHandler JavaFX) déclenchent les opérations du modèle.

Observer

Observateur + Modèle.notify() : dès qu'une action modifie les données, les vues observatrices se mettent à jour automatiquement (actualiser()).

Singleton

- Repository (instance unique) pour la sauvegarde/chargement.
- Certaines vues sont en instance unique (ex : VueBureau, VueListes) pour éviter de recréer l'UI et garder un comportement stable.

Composite

TacheComposite centralise attributs communs + listes de dépendances / sous-tâches, et Tache hérite pour représenter une tâche "concrète".

Strategy (stratégie d'affichage)

StrategieModeAffichage permet de changer dynamiquement le rendu (Bureau / Liste / Gantt) sans toucher au modèle.

Diagramme de classe

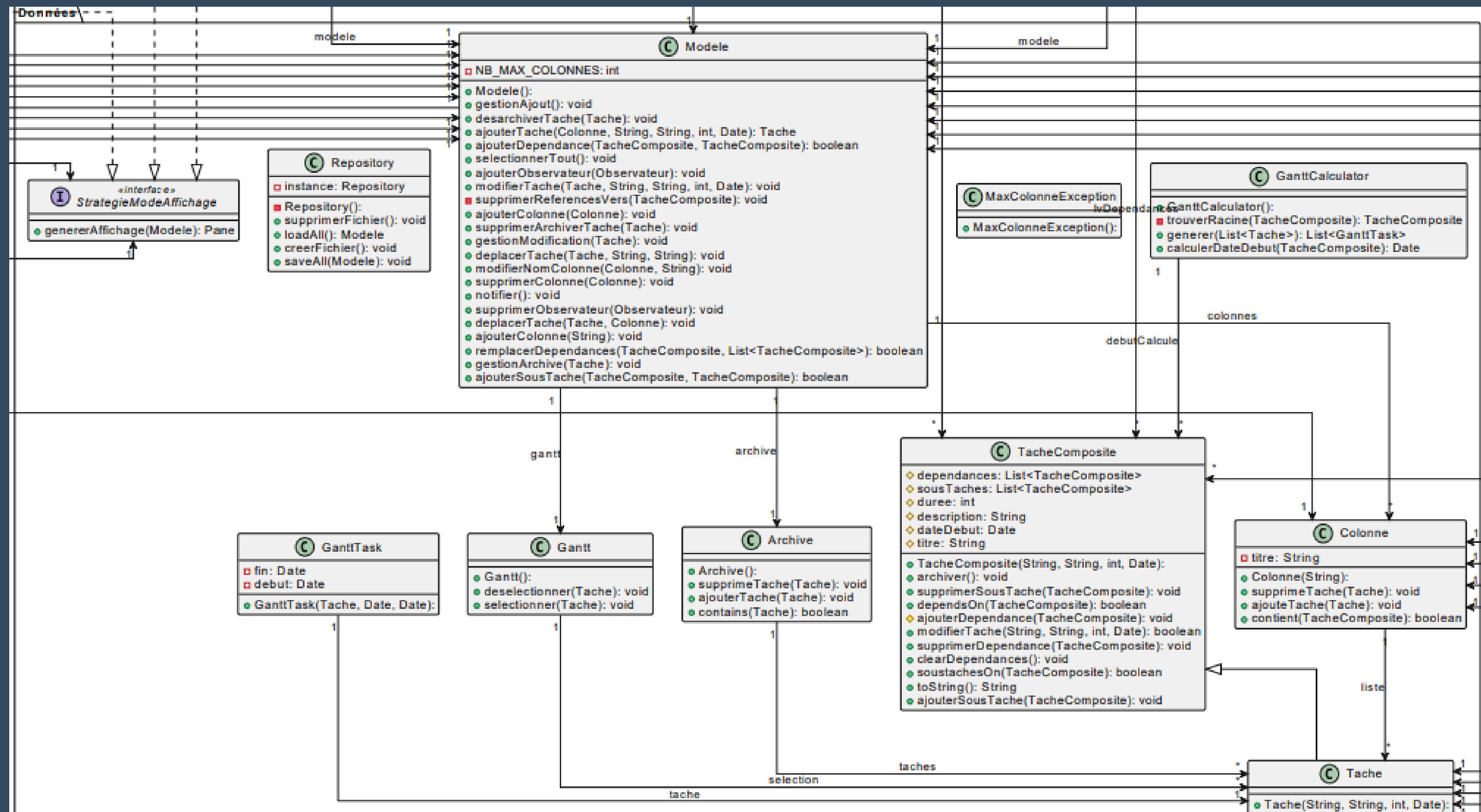


Diagramme de classe

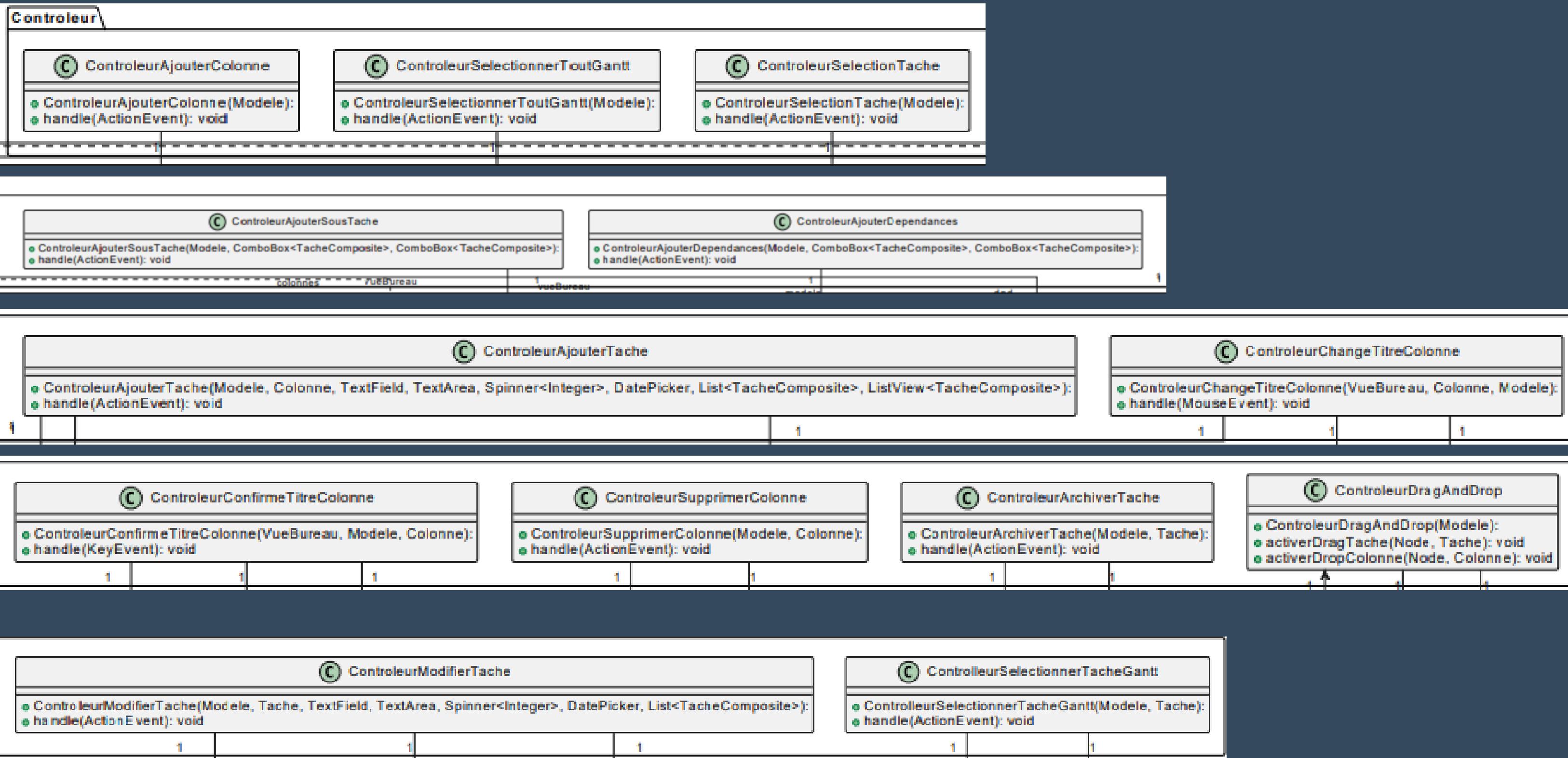


Diagramme de classe

