



TRIBOLLO

SAE 3.01 KEISER BEKRI CHERCHARI

Présentation générale

Tribollo est une application de gestion de projet conçue pour simplifier l'organisation et le suivi des tâches. voici notre compte rendu du projet SAE 3.01, incluant ses fonctionnalités, la répartition du travail, les innovations, les choix techniques, les modifications apportées et les instructions d'utilisation.

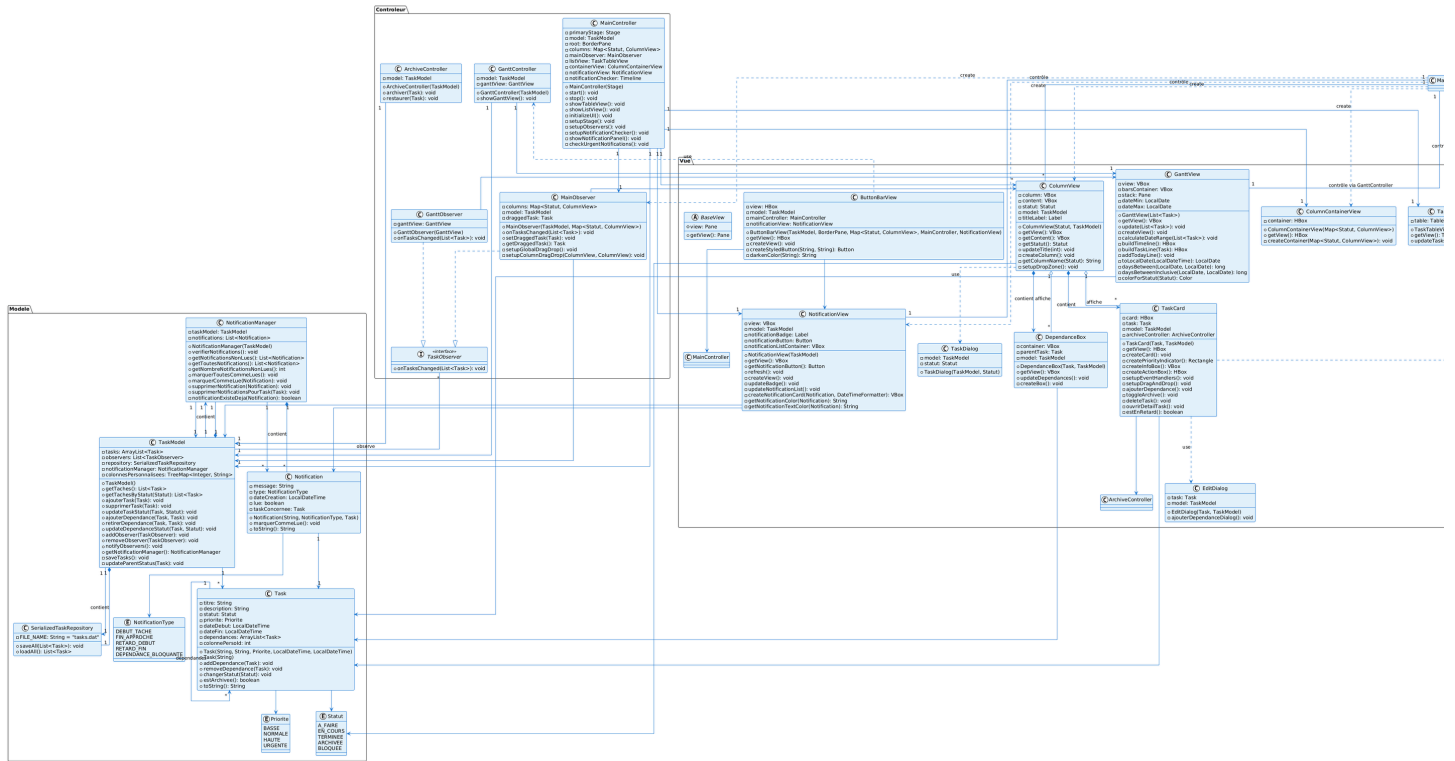
Fonctionnalités principales

- **Gestion des tâches** : Création, modification, déplacement et suppression de tâches.
- **Suivi de l'avancement** : Visualisation de l'état d'avancement des tâches (à faire, en cours, terminé).
- **Diagramme de Gantt** : Représentation graphique de la planification du projet.
- **Notifications** : Alertes en temps réel pour les échéances et les mises à jour.
- **Archivage des tâches** : Conservation des tâches terminées par exemple.
- **Interface utilisateur intuitive** : Facilité d'utilisation pour tous les membres de l'équipe.

Répartition du travail

Itération 1	Itération 2	Itération 3	Itération 4	Itération 5	Itération 6
<p>Une mauvaise vision du travail à faire :</p> <ul style="list-style-type: none"> ● Alexandre faisait les diagrammes ● Kenan faisait les programmes ● Samy faisait les tests unitaires <p>Nous avons réalisé :</p> <ul style="list-style-type: none"> ● La création des tâches, <ul style="list-style-type: none"> ● La modification des tâches, ● Archivage des tâches 	<p>Chacun a travaillé entièrement sur une fonctionnalité en faisant tout ce qui était en rapport avec la fonctionnalité à savoir diagramme, programmation, tests unitaires.</p> <p>Nous avons donc fait :</p> <ul style="list-style-type: none"> ● Serialisation, ● L'ajout d'une sous-tâche, ● L'implémentation des sous-tâches ● Sauvegarde 	<p>Nous avons continué sur le même fonctionnement que l'itération 2. Cependant, avant de se mettre entièrement dans l'itération 3, nous avons corrigé les problèmes relevés par le professeur.</p> <p>Nous avons fait :</p> <ul style="list-style-type: none"> ● Affichage du détail d'une tâche ● Gestion de la durée d'une tâche ● Début des dépendances ● Vue sous forme d'une liste 	<p>Pour cette itération, nous avons décidé de retirer le système de durée pour privilégier le fait de choisir des dates de début et de fin. De plus nous avons aussi rajouter des fonctionnalités.</p> <p>Il y a donc les nouveautés :</p> <ul style="list-style-type: none"> ● gestion de la date ● Ajout et suppression de colonnes personnalisées ● création du diagramme de Gantt 	<p>Ici, nous avons été dans la continuité de l'itération 4 en finalisant ce qui n'avait pas été terminée. En effet le projet était à cette étape déjà quasiment terminé c'est pourquoi il n'y a pas beaucoup d'ajout sur cette itération.</p> <p>Pas vraiment d'ajout important et début de l'amélioration de l'interface graphique</p>	<p>Pour la fin de ce projet, nous nous sommes majoritairement concentré sur la restructuration du code afin de le rendre plus propre. De plus le projet étant différent de ce que nous avions prévu dans l'étude préalable nous avons refait le diagramme de classe.</p> <p>Le seul vrai ajout est une amélioration graphique de l'interface afin de rendre l'utilisation plus agréable.</p>

Diagramme de classe final



Innovations et éléments originaux

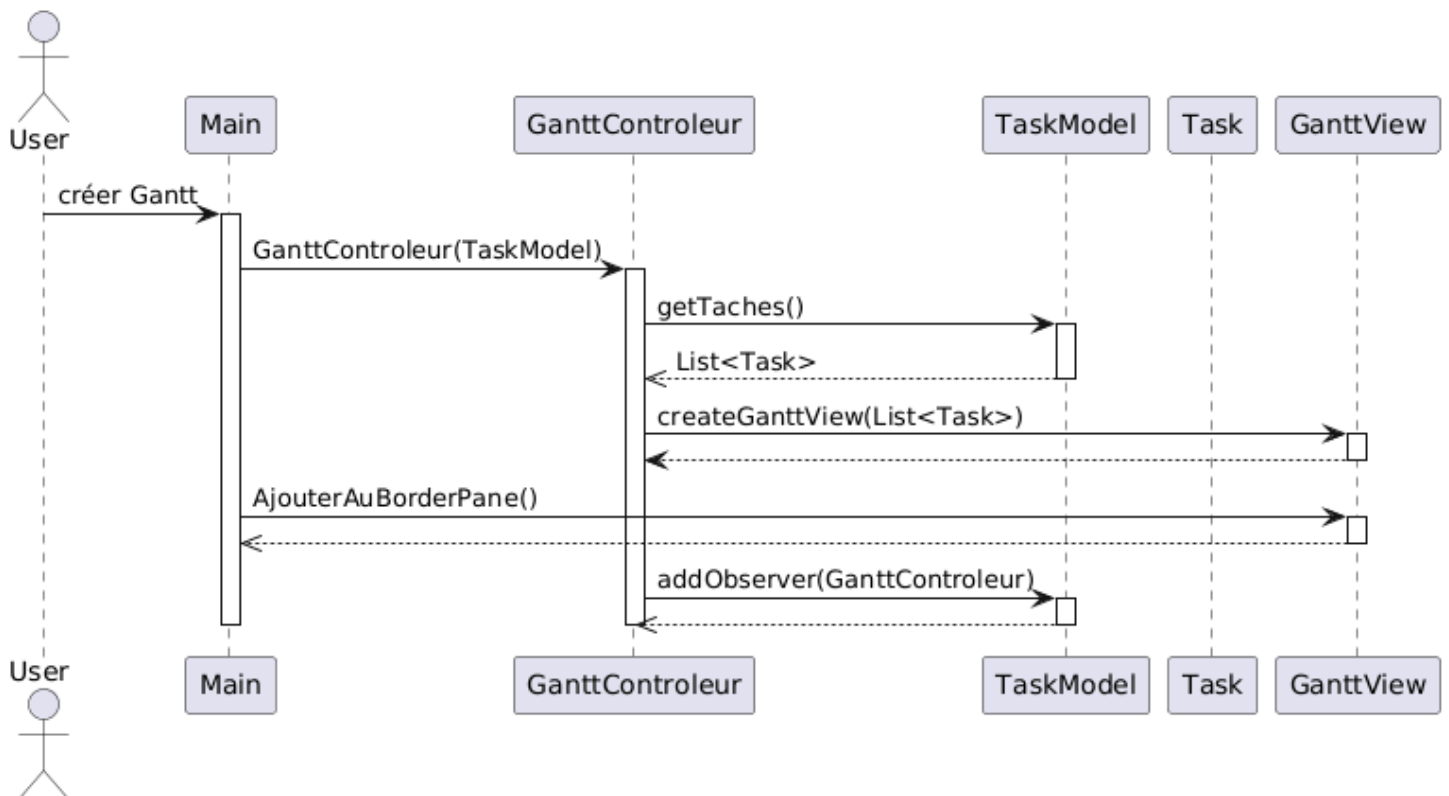
Tribolbo se distingue par:

- **Intégration du diagramme de Gantt en temps réel** : Permet de visualiser instantanément l'impact des modifications sur la planification.
- **Système de notifications** : Les utilisateurs peuvent être alerter.

Kenan :

- **Élément Original Dont Je Suis Fier : Le Diagramme de Gantt**
- Nous avons ajouté un **diagramme de Gantt** à notre application, ce qui n'existe pas dans Trello original. Ça permet de voir les tâches dans le temps, pas seulement par statut.
- **Fonctionnalités clés** :
 - **Timeline automatique** : Les dates s'ajustent selon les tâches
 - **Ligne "Aujourd'hui" rouge** : Pour voir où on en est
 - **Couleurs par statut** : Mêmes codes couleurs que le tableau
 - **Mise à jour en temps réel** : Quand une tâche change, le Gantt se met à jour
- **Ce que ça apporte** :
 - **Visualisation temporelle** : Voir les chevauchements et retards
 - **Planification** : Mieux organiser les tâches dans le temps
 - **Simplicité** : Un clic pour passer du tableau au Gantt
- **Pourquoi j'en suis fier** :
 - Parce que c'est une fonctionnalité qui dépasse la simple copie de Trello. C'est utile, et ça permet de gérer une réelle planification, permettant à l'utilisateur une bonne gestion.

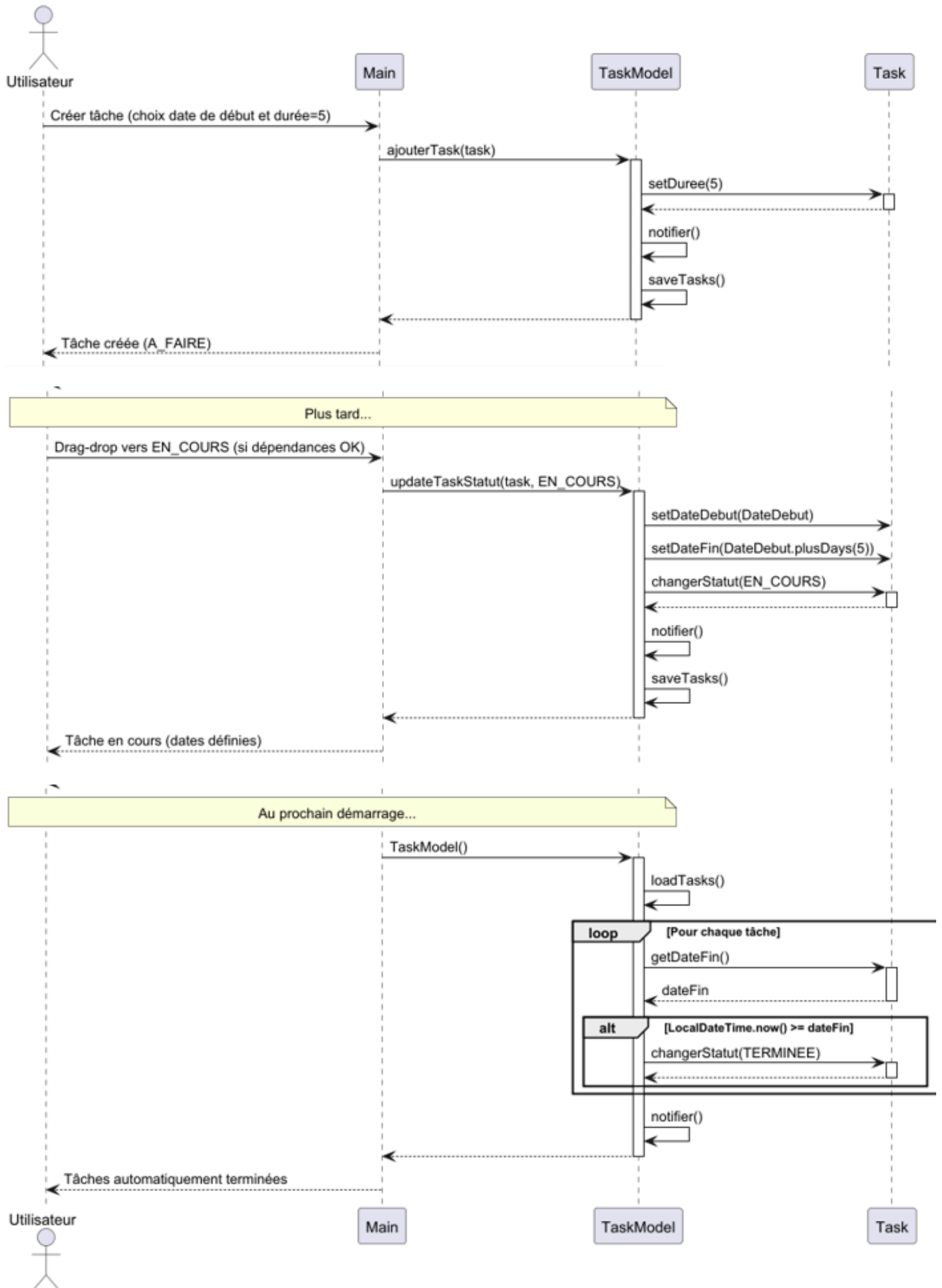
- Séquence :



Samy :

- **Élément dont je suis fier : Gestion des dates**
- Nous avons utilisé une gestion des dates relativement simple permettant à l'utilisateur de rester au courant de l'avancement de son projet grâce à des comparaisons avec la date du jour
- **Fonctionnalités clés :**
 - **Sélection des dates :** l'utilisateur peut pleinement planifier son projet avec la sélection des dates de début et de fin
 - **Mise en valeur des tâches en retard :** Les tâches dont la date de fin est passée apparaissent en rouge afin de montrer à l'utilisateur qu'il est temps de faire cette tâche.
- **Ce que ça apporte :**
 - **Visualisation :** Les tâches peuvent apparaître d'une certaine façon selon les dates permettant à l'utilisateur de savoir ce qu'il doit faire pour ne pas prendre de retard
 - **Planification :** Les dates sont gérées par l'utilisateur donc il peut planifier son projet comme bon lui semble
- **Pourquoi j'en suis fier :**
 - Bien que ça ne soit pas la fonctionnalité la plus compliquée, j'avais du mal avec l'utilisation des dates en Java. Avec cette fonctionnalité, je comprends mieux comment ça fonctionne.
- **Séquence**

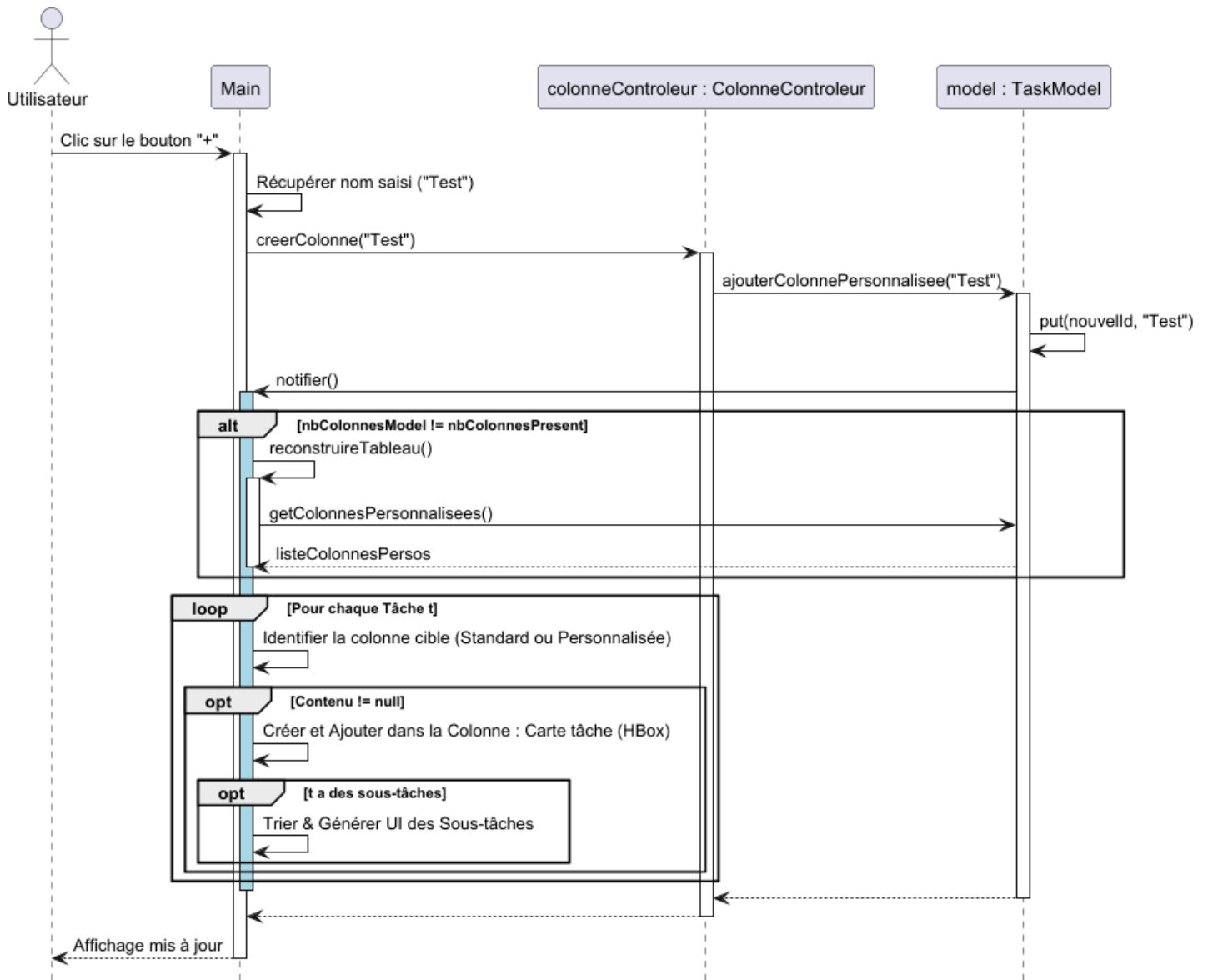
Gestion de la Durée



Alexandre :

- **Élément dont je suis fier : Colonnes Personnalisées**
- Nous avons étendu le tableau de notre application au-delà des statuts figés en permettant à l'utilisateur de créer ses propres espaces de rangement.
- **Fonctionnalités clés :**
 - **Création dynamique :** Via un bouton « + » intégré directement dans la barre d'outils, permettant d'ajouter de nouvelles colonnes nommées par l'utilisateur.
 - **Indépendance du statut :** Contrairement aux colonnes standards qui sont liées à l'énumération Statut (À faire, En cours...), les colonnes personnalisées fonctionnent via un système d'ID unique.
- **Ce que ça apporte :**
 - **Liberté d'organisation :** L'utilisateur n'est pas forcé de suivre le cycle de vie strict d'une tâche. Il peut créer des colonnes contextuelles (ex: "Idées", "En attente retour") qui ne sont pas des "états" d'avancement mais des lieux de classement.
 - **Flexibilité :** Le code gère deux logiques d'affichage en parallèle sans conflit : la logique métier (Statut) et la logique visuelle (ID de colonne).
 - **Continuité :** L'expérience utilisateur est fluide, il n'y a pas de différence de comportement entre une colonne native et une colonne ajoutée par l'utilisateur.
- **Pourquoi j'en suis fier :** C'est la réflexion architecturale en amont. J'ai réalisé que les colonnes personnalisées ne pouvaient pas être gérées comme les colonnes standards (liées au Statut). J'ai donc implémenté une TreeMap pour les gérer par ID. À la suite de cela, le défi s'est porté sur la couche d'affichage, où j'ai réussi à intégrer une logique sans casser les fonctionnalités existantes (des sous-tâches, persistance) et en gardant le code des colonnes standards propre.
- **Séquence**

Ajout simple d'une colonne



Choix techniques

Architecture

- Modèle Vue Contrôleur (MVC)
- JavaFX pour l'interface graphique

Patrons de conception

- Observateur (pour les notifications)
- Fabrique (pour la création des tâches)
- Singleton (pour la gestion de la configuration)

Modifications par rapport à l'étude préalable

Par rapport à l'étude préalable, les modifications suivantes ont été apportées :

- **Ajout du système de notifications** : Pour améliorer le suivi des échéances.
- **Amélioration de l'interface utilisateur** : Pour une meilleure expérience utilisateur.
- **Abandon de l'export en png et de la création de colonne personnalisé.**

Instructions de lancement

1. Ouvrez IntelliJ IDEA.
2. Importez le projet Tribollo.
3. Exécutez la classe **MAIN** dans **demo** pour lancer l'application.

Mode d'emploi

Ajout d'une tâche:

Cliquez sur le bouton "Ajouter une tâche", renseignez les informations nécessaires (nom, description, échéance) et validez.

Modification d'une tâche:

Double cliquer sur la tâche, apportez les modifications et validez.

Déplacement d'une tâche:

Faites glisser la tâche vers la colonne correspondant à son état d'avancement (à faire, en cours, terminé).

Archivage d'une tâche:

Cliquez sur le bouton "A" de la tâche à archiver.

Suppression d'une tâche:

Cliquez sur le bouton "x" de la tâche à supprimer.

Accès au diagramme de Gantt:

Cliquez sur l'onglet "Gantt" pour visualiser la planification du projet.

Accès aux notifications:

Cliquez sur la cloche en haut à droite.

Résumé

Tribollo est une application de gestion de projet plutôt intuitive, conçue pour optimiser le travail d'équipe et le suivi des tâches. Grâce à ses fonctionnalités, son interface utilisateur conviviale et ses choix, elle offre une solution efficace pour la gestion de projets.