

Rapport Final - SAE 3.01 : Développement d'une Application

ARDHUIN Louis / GRAINE Mathis / ATTENOT Pierre

Projet : Gestionnaire de Tâches type trello

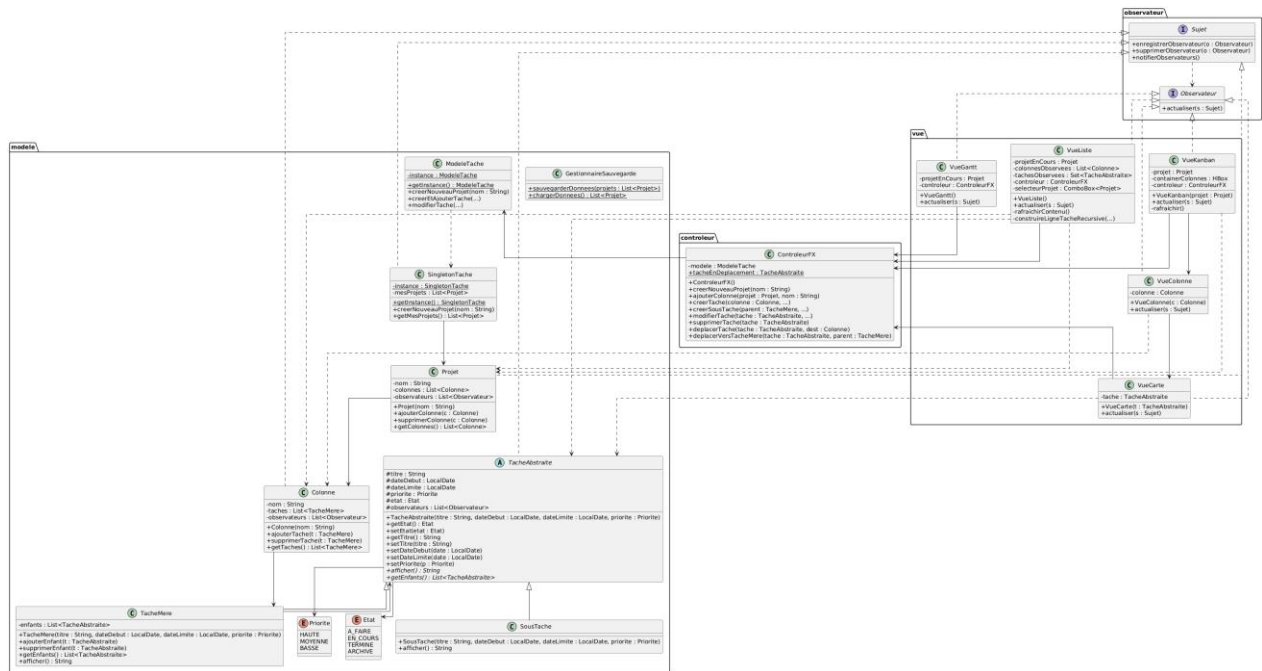
1. Liste des fonctionnalités réalisées

L'application permet la gestion de projets et de tâches à travers différentes vues.

Fonctionnalités :

- Gestion des Projets et Tâches :
 - Création de projets.
 - Création, modification et suppression de tâches et de colonnes.
 - Ajout de sous-tâches.
 - Gestion des priorités (Basse, Moyenne, Haute) avec chacune une couleur associée.
 - Gestion des dates (date de début et date de fin).
- Navigation et Vues :
 - Vue Kanban : Affichage par colonnes avec Drag & Drop pour déplacer les tâches.
 - Vue Liste : Affichage trier par jour et dépliable. Possibilités de créer et changer de projet.
 - Vue Gantt : Visualisation temporelle des tâches sous forme de barres colorées suivant leurs priorités.
- Limites :
 - Une sous-tâche ne peut pas commencer avant son parent ni finir après lui. Si l'utilisateur tente de le faire, les dates sont automatiquement modifiées à la date d'aujourd'hui.

2. Diagramme de classe final



3. Répartition du travail entre étudiants

- Mathis : Architecture, Modèle et Vue Kanban J'ai mis en place le patron MVC et le Contrôleur, j'ai conçu la classe Tache et l'énumération Priorité en intégrant le patron Composite pour permettre la gestion des tâches et sous-tâches. J'ai également implémenté la première version du Singleton qui a été modifié par la suite et sur la partie graphique j'ai travaillé avec Louis sur la vue Kanban ainsi que sur la vue liste et sur la vue Gantt.
- Louis : J'ai commencé principalement par la mise en place de diagrammes de classes et de séquences ainsi que la mise en place du Trello. Puis après avoir réglé un gros souci de JavaFX qui ne nous permettait pas de pouvoir lancer l'appli depuis la version du dépôt git on a pu commencer avec Mathis la première version de la vue Kanban. Pendant les vacances j'ai pu restructurer le code afin de repartir sur des bases plus saines, avec notamment la vue Projet, la vue Colonne et la vue Carte, j'ai ensuite permis l'affichage des sous-tâches et implémenté le drag & drop sur cette vue Kanban. Par la suite, nos efforts se sont portés sur les 2 dernières vues qu'on a fait en collaboration avec Mathis sur les 2 dernières itérations
- Pierre : Je me suis principalement occupé de la sérialisation des données. J'ai choisi le format .bin plutôt que le JSON, car il est natif à Java et ne nécessite aucune

bibliothèque externe. Lors de cette implémentation, j'ai dû trouver une solution pour vider et recréer les listes observables, car les objets JavaFX ne sont pas sérialisables, afin d'assurer un chargement et une sauvegarde fonctionnels. En parallèle, j'ai contribué aux premières itérations du projet par un travail de veille et de réflexion sur la conception, notamment pour la vue liste. Même si j'ai été moins présent sur le développement que Mathis et Louis, je les ai aidés régulièrement via le groupe Discord, en fournissant du code et des recherches ciblées selon leurs besoins. J'ai également participé à la conception globale en réalisant quelques diagrammes de classes et de séquences, principalement au début du projet.

4. Présentation d'un élément original dont vous êtes fiers

Nous avons choisi de présenter le système qui empêche de faire des erreurs de dates.

Dans une gestion de projet, il n'est pas logique qu'une sous-tâche finisse après la tâche principale dont elle dépend. Plutôt que d'afficher des messages d'erreur bloquants à chaque fois que l'utilisateur se trompe, l'application corrige les dates toute seule et instantanément.

Par exemple, si on essaie de placer une sous-tâche trop loin dans le calendrier (que ce soit en remplissant le formulaire ou en la déplaçant à la souris), elle est automatiquement ajustée pour rentrer dans les délais de la tâche principale.

5. Éléments modifiés par rapport à l'étude préalable

Au cours du développement, certaines architectures ont été restructurer :

1. De base, nous étions partis sur une structure de tâches mères imbriquées, un projet étant vu comme une tâche mère contenant des tâches mère. Nous avons changé pour une structure basée sur des listes de tâches mères dans des Colonnes.

2. Nous avons déplacé toute la logique de vérification des dates dans le Contrôleur. Cela permet à la règle de s'appliquer à toutes les vues.
3. Nous avons remplacé l'idée de fenêtres multiples par une seule fenêtre comme conteneur principal, permettant de changer de vue.

6. Patrons de conception et d'architecture

Le patron MVC, le patron Singleton et le patron Composite.

7. Graphe de scène et Organisation graphique

L'interface graphique est structurée autour d'un conteneur racine, la classe `VueMenu`, qui hérite du composant JavaFX `BorderPane`.

Organisation hiérarchique du graphe de scène :

- En haut on a l'accueil et la barre de navigation, structurée dans une `HBox` contenant les boutons de contrôle.
- Le centre est la zone dynamique qui affiche le nœud principal (les objets `VueKanban`, `VueListe` ou `VueGantt`).

Les trois vues sont instanciées une seule fois au démarrage de l'application pour conserver leur état. Lorsqu'on clique sur un bouton du menu, la méthode standard `setCenter` du `BorderPane` est appelée. Elle permet de remplacer instantanément le composant graphique central par la vue demandée, sans avoir à recharger toute la fenêtre.

8. Mode d'emploi pour lancer l'application

Prérequis :

- JDK : Version 23.

Installation et Lancement :

1. Cloner le depots git <https://github.com/Mirtille/SAE-3.01-D-veloppement-d-une-application>
2. marquer le dossier src/main/java comme "Sources Root"
3. Dépendances : le fichier pom.xml doit contenir les dépendances javafx-controls et javafx-fxml.
4. Pour lancer l'application il suffit de se rendre dans le mainfx dans le package app et de le lancer.
5. Fonctionnalités :
 - 1.1. Au lancement, un projet "Projet" est créé et l'appli démarre sur la vue kanban.
 - 1.2. Utilisez le menu en haut pour parcourir les différentes vues (kanban, liste, gantt).
 - 1.3. En dessous vous pouvez choisir votre projet, en crée un autre ou crée une nouvelle colonne.
 - 1.4. Dans la vue kanban il faut cliquer sur ajouter une tâche présent en bas de chaque colonne pour ajouter une tâche à la dite colonne, pour modifier il suffit de cliquer sur le petit stylo présent à droite des cartes de tâche et pour les supprimer il faut cliquer sur la croix rouge, vous pouvez aussi supprimer n'importe quelle colonne en cliquant sur la croix en haut à droite de celle-ci.
 - 1.5. Pour la vue liste vous avez un formulaire en bas qui permet de créer une tâche en renseignant son titre sa date de début et de fin et sa priorité, vous pouvez déplier et plier la liste de tâche en cliquant sur la date en haut de chaque liste, pour modifier ou supprimer c'est comme dans la vue kanban avec le crayon et la croix et vous pouvez ajouter une sous tâche en cliquant sur le plus à côté du crayon.

- 1.6. Pour la vue gantt c'est comme la vue liste pour ajouter une tache, vous pouvez obtenir les infos des tache en cliquant sur le point au milieu de chaque trait et depuis le pop up d'information vous pouvez modifier la tache ou ajouter une sous tache a cette tache