SAE 3.01 – DEVELOPPEMENT

-

LOGICIEL D'ORGANISATION DE TACHES PERSONNELLES

Etude préalable

Alann MONNIER, Fabien TOUBHANS, Farid MARI

17/12/2023

Table des matières

1. INTRODUCTION

1.1 Contexte et objectifs du projet

Ce rapport détaille le processus de conception et de développement entrepris lors du projet de fin de module, visant à créer un logiciel d'organisation de tâches personnelles. Dans le cadre de la SAE 3.01, nous avons abordé la complexité de la conception d'un système agile et adaptable, capable de répondre aux exigences diversifiées de gestion des tâches dans un environnement personnel.

Le document s'articule autour de la réalisation d'une application qui non seulement s'aligne avec les outils de gestion de projet modernes tels que Trello, mais qui aussi innove en termes de fonctionnalités et d'ergonomie. L'accent a été mis sur l'identification des besoins utilisateur spécifiques et l'intégration de ces derniers dans un design fonctionnel et assez esthétique.

À travers une méthodologie itérative et incrémentale, sous forme d'itérations, ce rapport expose notre démarche analytique, les décisions de conception et les stratégies de développement adoptées. Nous avons mis en œuvre une architecture MVC en JavaFx, tout en respectant les principes de conception appris, afin de garantir un produit final, qui saurait répondre aux attentes.

En conclusion, ce rapport sert non seulement de compte rendu de notre travail, mais également de référentiel pour les futures itérations du développement de l'application.

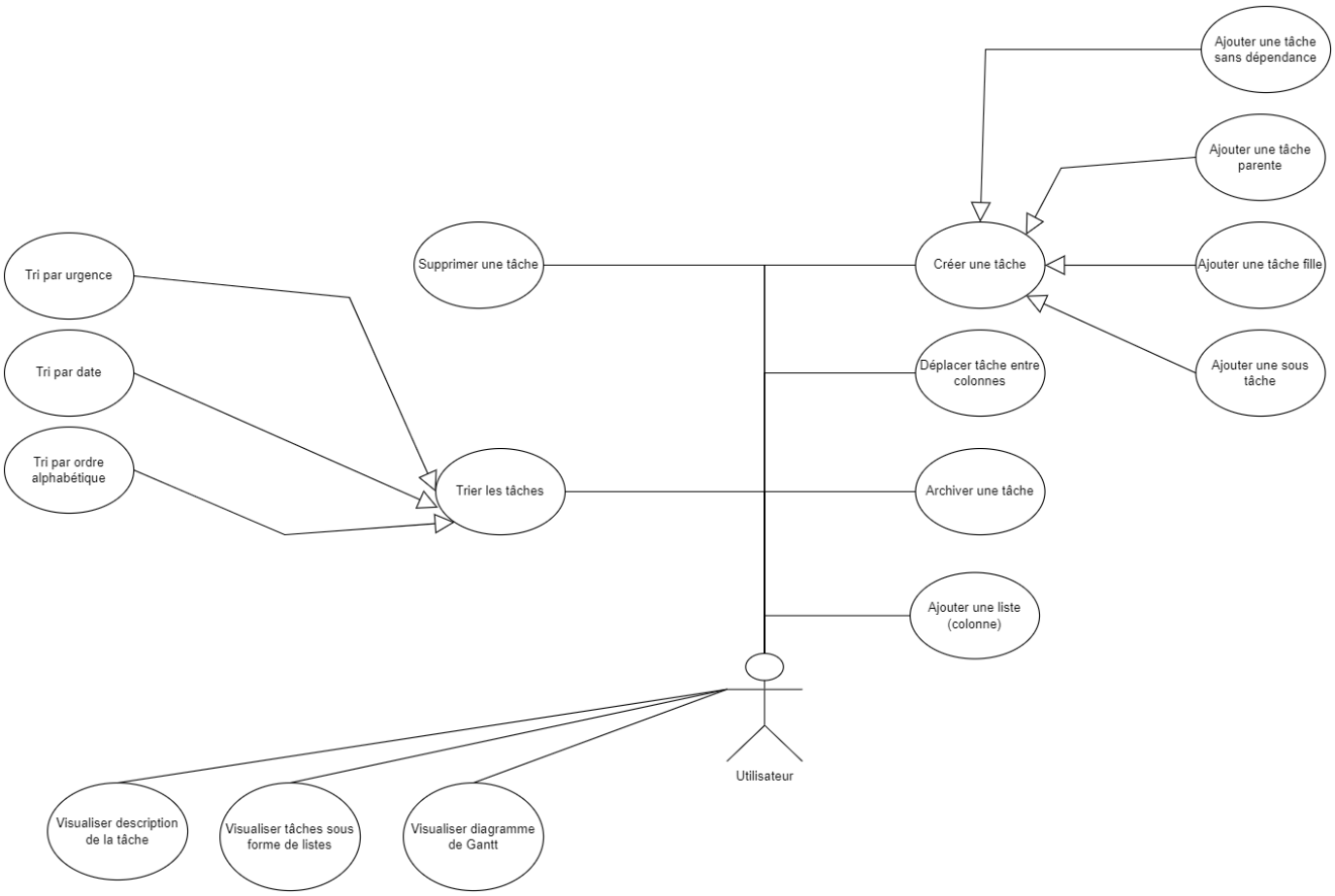
2. LISTE DES FONCTIONNALITES

Création de tâches avec dépendances :	Cette fonctionnalité permet aux utilisateurs de créer des tâches qui sont interdépendantes, offrant la possibilité de définir des prérequis pour le démarrage de certaines actions. Par exemple, une tâche secondaire ne pourra être initiée qu'après l'achèvement de sa tâche parente, assurant une séquence logique et une progression méthodique des activités.
Créer de nouvelles colonnes (3 de bases)	Les utilisateurs pourront créer de nouvelles colonnes en plus des trois colonnes de base fournies. Ces colonnes peuvent être personnalisées pour refléter différents stades ou aspects du workflow des tâches, telles que 'À faire', 'En cours', et 'Terminée', offrant ainsi une flexibilité dans l'organisation des tâches.
Supprimer une tâche	Une option pour supprimer les tâches inutiles ou terminées permet de maintenir l'espace de travail clair et organisé. Les tâches peuvent être supprimées individuellement ou en groupe, en fonction des besoins de l'utilisateur.
Archiver une tâche	Les tâches complétées peuvent être archivées pour conserver un historique des travaux accomplis sans encombrer l'espace de travail actif. L'archivage contribue à une meilleure visibilité des tâches en cours et aide à suivre les progrès tout en conservant les données pour des références futures.
Dépendances entre tâches	Il est possible d'établir des liens de dépendance entre les tâches, permettant à une tâche de débuter uniquement après la finalisation d'une autre. Cette fonction est cruciale pour les projets où l'ordre d'exécution est essentiel et contribue à une gestion de projet plus précise.
Visualisation en liste et sous-listes	Les utilisateurs peuvent visualiser les tâches sous forme de listes imbriquées, ce qui est particulièrement utile pour les projets complexes avec de multiples sous-tâches. Cette structure hiérarchique aide à comprendre la relation entre les tâches et les différentes étapes nécessaires à leur achèvement.

Génération de diagrammes de Gantt	Cette fonctionnalité avancée permet aux utilisateurs de sélectionner des tâches spécifiques et de générer un diagramme de Gantt, facilitant une visualisation graphique de la chronologie et de la durée de chaque tâche. Les utilisateurs peuvent ainsi anticiper les interactions entre les tâches, gérer les délais et optimiser la planification globale du projet.
Archiver des tâches	L'archivage automatique est une fonctionnalité qui déplace les tâches complétées vers une archive, nettoyant ainsi l'espace de travail tout en conservant un enregistrement des tâches terminées. Cela permet de réduire le désordre visuel et de se concentrer sur les tâches actives, tout en gardant un accès facile aux tâches antérieures pour référence ou audit.
Bonus Supp / Archiv / Depl soit avec clic-droit et un menu s'ouvre ou directement avec des boutons	Pour une gestion de tâches plus intuitive, les utilisateurs peuvent effectuer des actions communes telles que supprimer, archiver ou déplacer des tâches en utilisant soit un menu contextuel accessible par clic droit, soit des boutons dédiés. Cette flexibilité permet une interaction rapide et personnalisée selon les préférences de l'utilisateur, rendant la navigation dans l'application plus fluide et moins encombrée.

3. MODELISATION UML

3.1 Diagramme de cas d'utilisation



- **Utilisateur** : Représente la personne qui interagit avec le système de gestion de tâches.

- **Créer une tâche** : Cette fonctionnalité centrale permet à l'utilisateur de créer de nouvelles tâches. Il y a trois extensions à cette fonctionnalité :

    - **Ajouter une tâche sans dépendance** : Créer une tâche qui n'est pas liée à d'autres tâches.

    - **Ajouter une tâche parente** : Créer une tâche qui aura des sous-tâches (tâches enfants) associées.

    - **Ajouter une tâche fille** : Créer une sous-tâche qui est dépendante d'une tâche parente.

- **Déplacer tâche entre colonnes** : L'utilisateur peut changer le statut d'une tâcheen la glissant d'une colonne à une autre, par exemple, de "À faire" à "En cours".

- **Archiver une tâche** : Une fois qu'une tâche est terminée, l'utilisateur peut l'archiver pour la retirer de la vue active tout en conservant son historique.

- **Ajouter une liste (colonne)** : Permet à l'utilisateur de créer des colonnes supplémentaires pour organiser les tâches.

- **Trier les tâches** : Cette fonctionnalité donne à l'utilisateur la possibilité de trier les tâches selon différents critères :

    - **Tri par urgence** : Classe les tâches en fonction de leur niveau de priorité ou de leur urgence.

    - **Tri par date** : Organise les tâches par date d'échéance ou de création.

    - **Tri par ordre alphabétique** : Classe les tâches par ordre alphabétique.

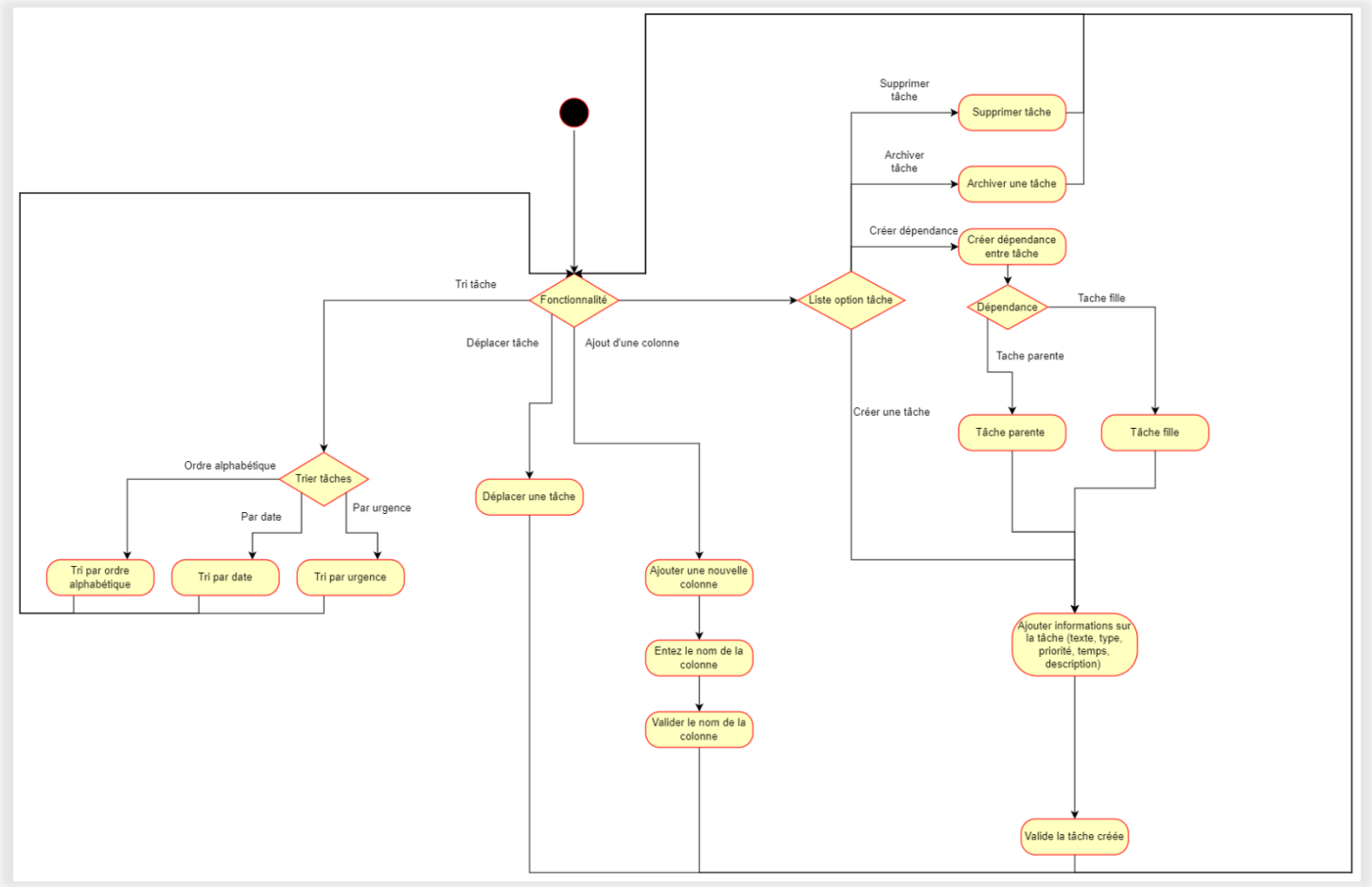
- **Supprimer une tâche** : Offre la possibilité d'effacer définitivement une tâche de la liste.

- **Visualiser description de la tâche** : Permet à l'utilisateur de voir les détails d'une tâche spécifique.

- **Visualiser tâches sous forme de listes** : Affiche toutes les tâches dans une liste ou une sous-liste.

- **Visualiser diagramme de Gantt** : L'utilisateur peut générer et voir un diagramme de Gantt pour suivre la chronologie et la progression des tâches choisies.

3.2 Diagramme d'activités



- **Fonctionnalité** : Point de départ pour l'utilisateur, indiquant qu'il s'apprête à interagir avec le système pour exécuter différentes fonctions.

- **Liste option tâche** : Cette branche présente les options disponibles lorsqu'une tâche est sélectionnée, telles que :
  - **Supprimer tâche** : Permet à l'utilisateur de supprimer une tâche.
  - **Archiver une tâche** : Donne la possibilité d'archiver la tâche pour la retirer de la vue active.
  - **Créer dépendance entre tâche** : Permet de créer une relation de dépendance entre les tâches, avec des sous-branches pour Tâche parente et Tâche fille, indiquant qu'une tâche peut être dépendante d'une autre.

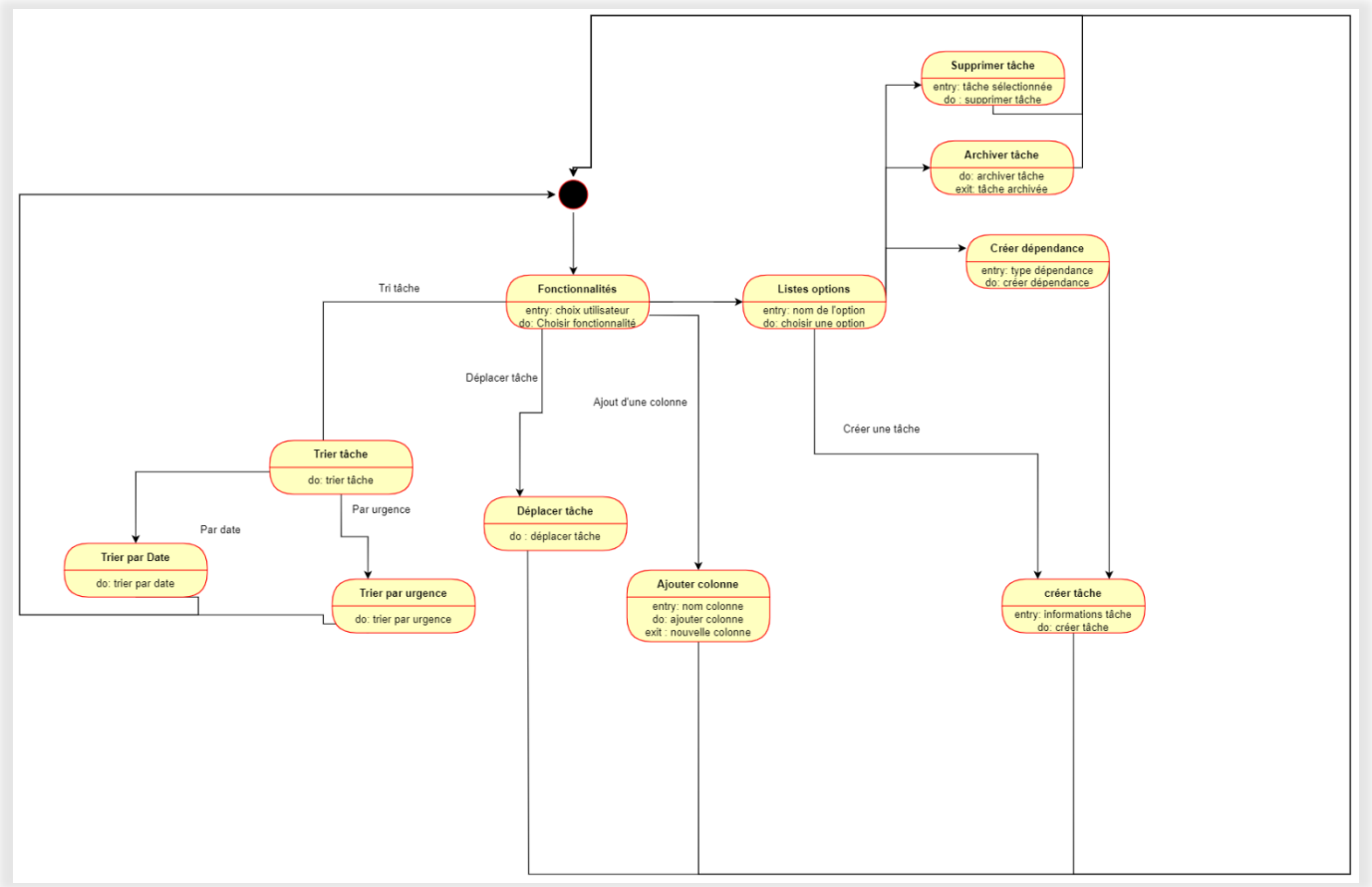
- **Créer une tâche** : Cette activité comprend plusieurs étapes pour la création d'une tâche, y compris l'ajout de texte, le type, la priorité et le temps accordé à la tâche, finissant par la description et la validation de la tâche créée. C'est l'option principale, qui donne la possibilité à l'utilisateur de créer une tâche.

- **Ajout d'une colonne** : Cette branche du diagramme montre les étapes pour ajouter une nouvelle colonne au système, qui comprend l'entrée et la validation du nom de la colonne.

- **Déplacer tâche** : Représente l'action de déplacer une tâche d'une colonne à une autre.

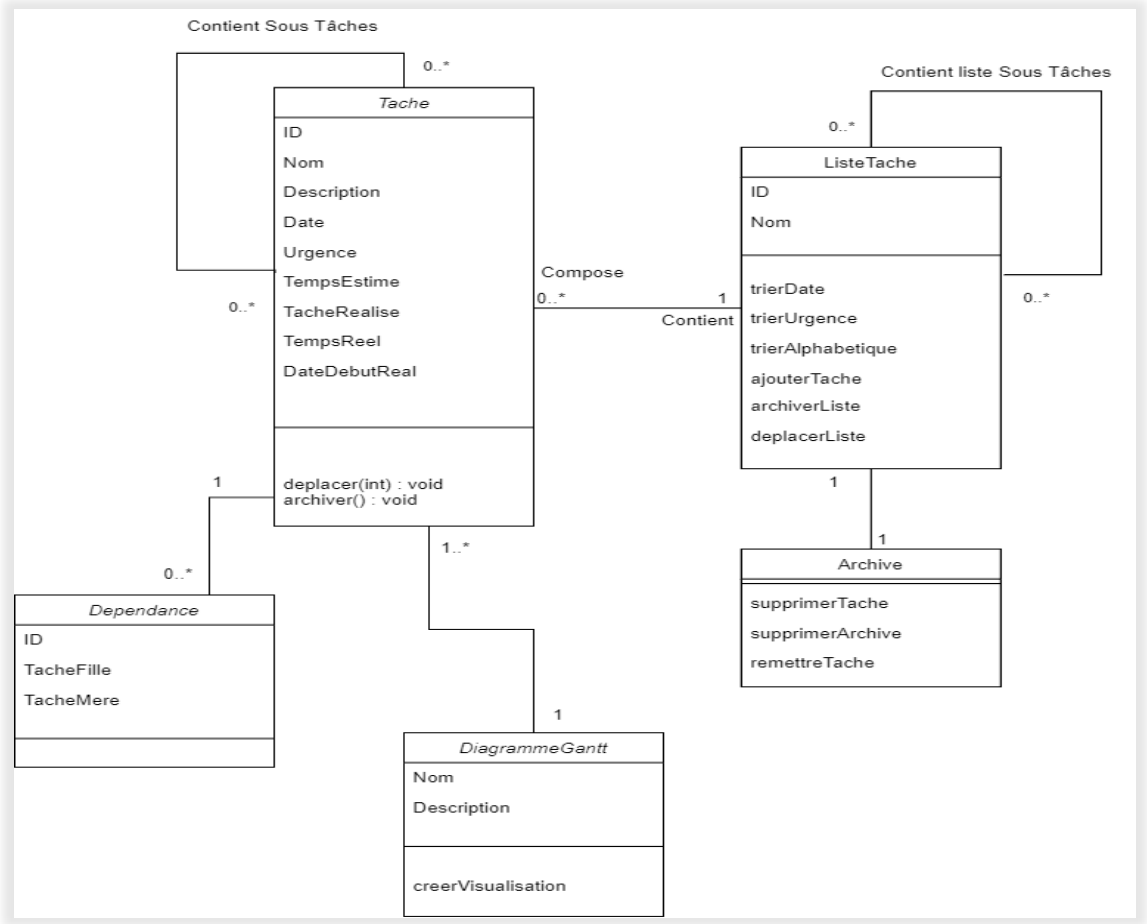
- **Trier tâche** : Offre des options pour trier les tâches selon l'ordre alphabétique, la date, ou l'urgence, permettant à l'utilisateur de personnaliser l'affichage des tâches selon ses préférences.

3.3 Diagramme d'état



3.4 Diagramme de classe

3.4.1 Analyse



La classe centrale est **Tâche**, qui a des attributs de base tels que ID, nom, description, date, urgence, temps estimé et temps réel. Elle contient aussi des méthodes pour déplacer et archiver la tâche. La **Tâche** peut avoir zéro ou plusieurs sous taches.

La classe **Dépendance** est une relation entre les tâches où une tâche peut dépendre de zéro ou plusieurs autres tâches, et inversement. L'association avec une multiplicité de 0..* des deux côtés, indiquant qu'une tâche peut avoir plusieurs dépendances et peut être une dépendance pour plusieurs tâches.

La **ListeTache** gère un ensemble de tâches et offre des fonctionnalités pour trier les tâches par date, urgence, et ordre alphabétique, ainsi que pour ajouter, archiver, et déplacer des tâches au sein de la liste.

L'Archive est une classe connectée à la **ListeTache** et gère les tâches archivées avec des méthodes pour supprimer des tâches ou les retirer de l'archive.

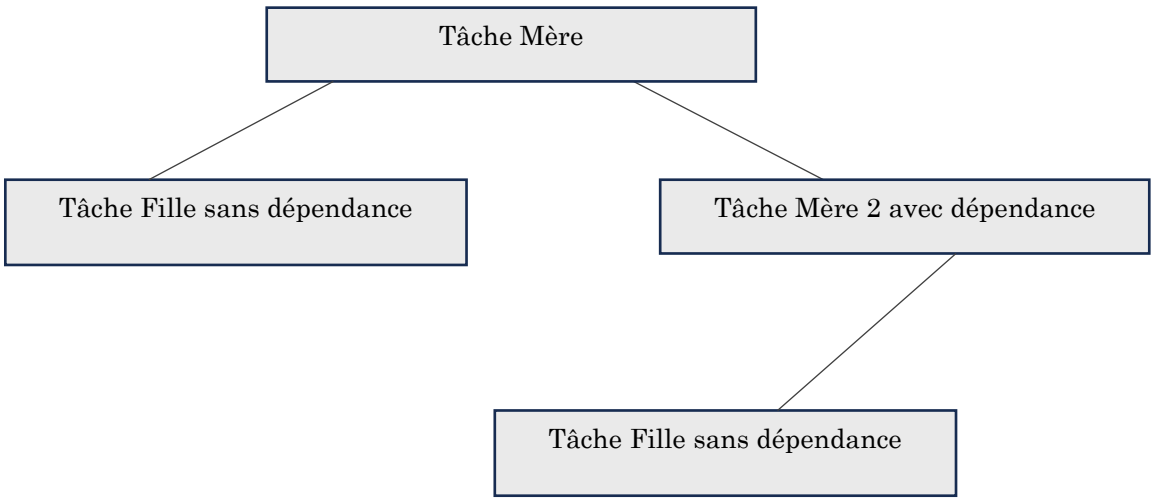
Enfin, nous avons la classe **DiagrammeGantt** qui va servir à créer une visualisation des tâches en termes de planification et de suivi de projet.

3.4.2 Conception

Diagramme imposant, voir dossier → Diagramme_de_classe.

4. PATRONS DE CONCEPTION

Le patron de conception **Composite** sera employé pour modéliser où les tâches sont le système de sous taches interdépendantes et structurées hiérarchiquement, ainsi que pour le diagramme de Gantt. Cela sera conceptualisé comme un arbre où les nœuds feuilles représentent des tâches pouvant elles-mêmes avoir des dépendances. Le patron Composite est donc particulièrement adapté pour ce modèle puisqu'il facilite la représentation et la manipulation de structures arborescentes complexes.



Le patron de conception **Stratégie** est mis en œuvre pour gérer les diverses méthodes d'affichage des tâches, que ce soit en colonnes, en lignes, ou sous la forme d'un diagramme de Gantt. Ce patron nous permettra de modifier dynamiquement l'approche d'affichage selon l'interaction de l'utilisateur avec l'interface, garantissant ainsi une flexibilité optimale. De la même manière, le patron **Stratégie** sera utilisé pour varier les critères de tri des tâches, permettant un classement selon l'urgence, la date, ou tout autre paramètre pertinent.

L'exclusivité de la classe Archive est assurée par l'implémentation du patron de conception **Singleton**. Ce choix assure qu'une seule instance de la classe Archive peut être créée, consolidant ainsi la gestion centralisée des tâches archivées au sein de l'application.

Le patron de conception **Fabrique** est appliqué pour la création de tâches, à travers trois fabriques distinctes. La première fabrique crée des tâches autonomes sans

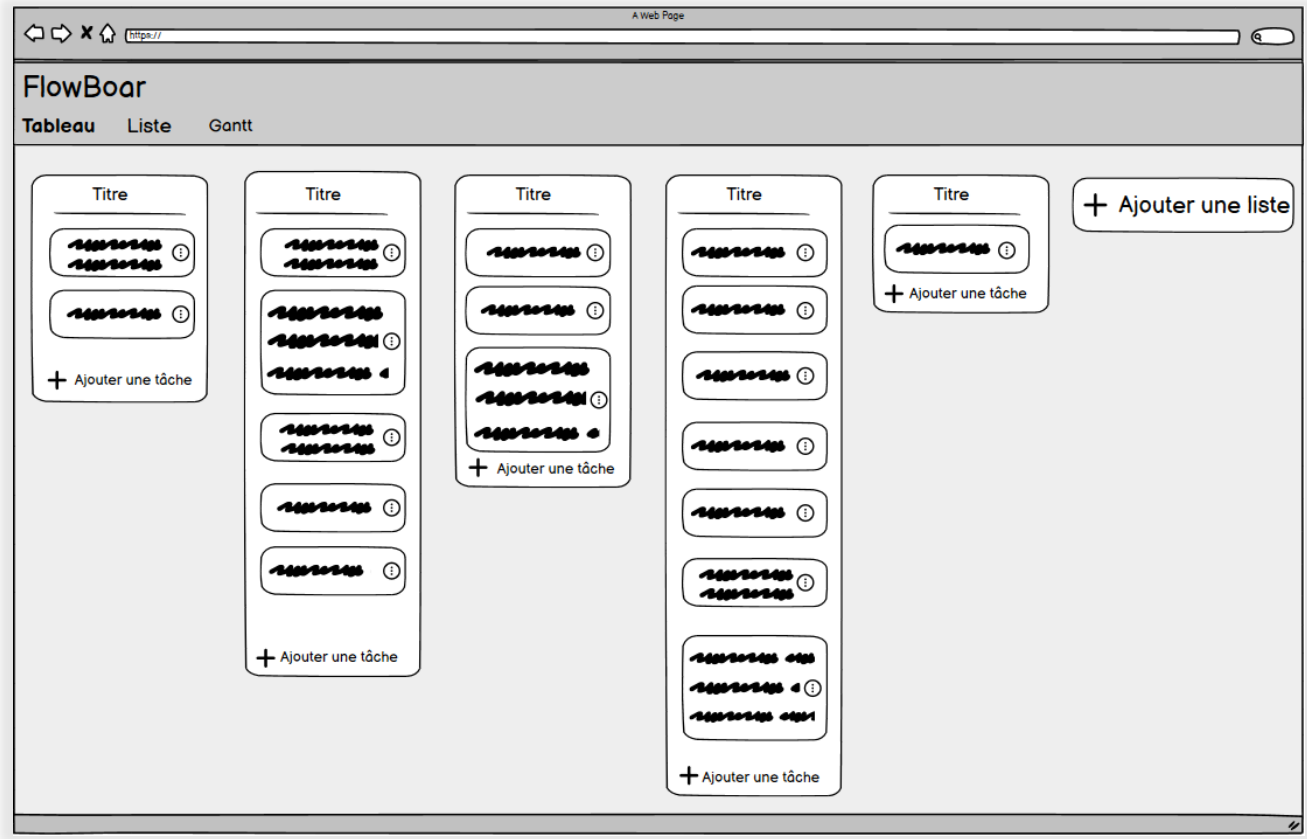
dépendances. La seconde est spécialisée dans la création de tâches ayant des relations de dépendance de type "fille", tandis que la troisième fabrique des tâches avec des relations de dépendance de type "mère". Cette approche modulaire offre une flexibilité dans la génération de tâches et leur organisation relationnelle.

Patron **Observateur** pour la partie graphique : parfait pour la gestion des mises à jour de l'interface graphique en réponse aux changements d'état des tâches. Lorsqu'une tâche est modifiée, ajoutée ou supprimée, la partie graphique de l'application, qui observe ces changements, peut automatiquement se mettre à jour pour refléter les nouvelles informations.

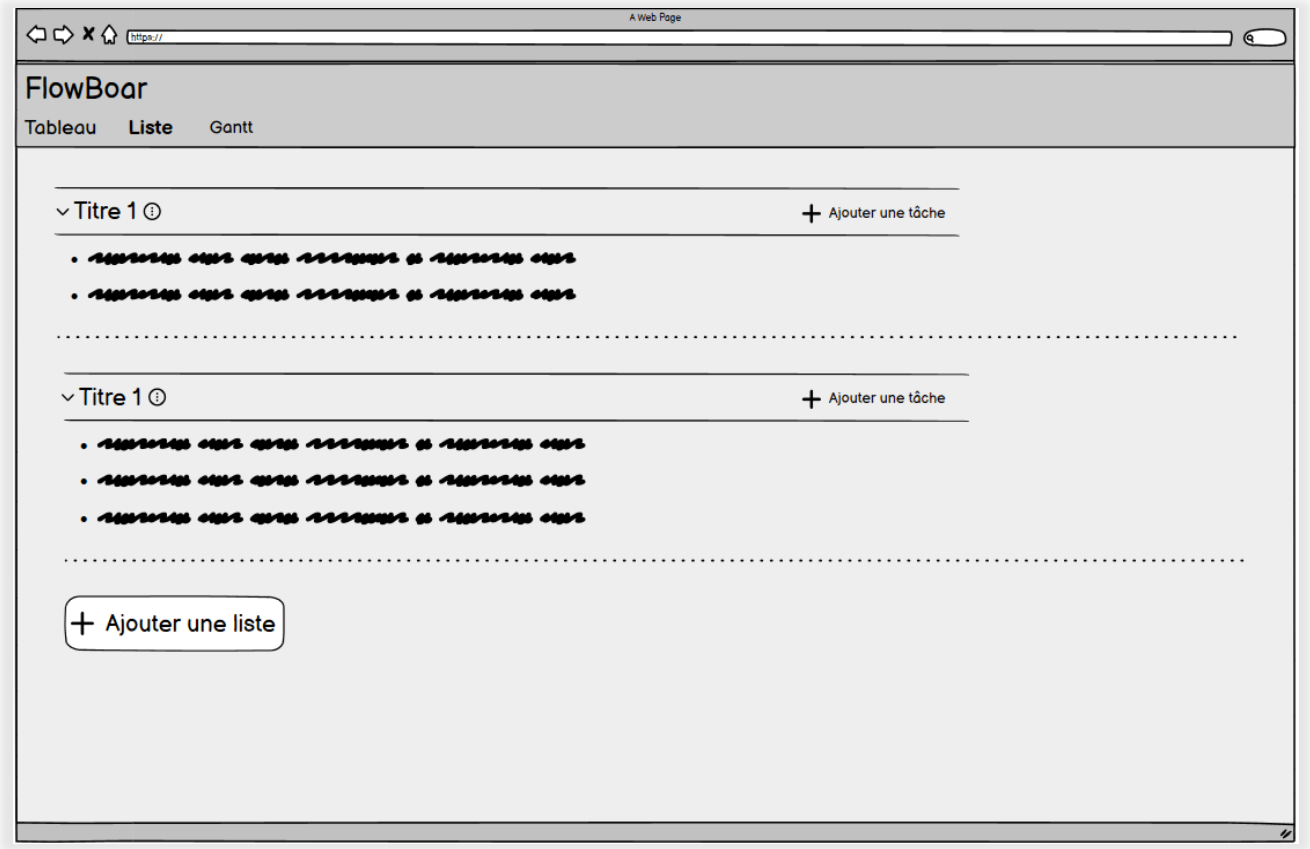
Base de données et **Active Record** : Ce patron sera utilisé pour mapper les structures de données orientées objet avec la base de données relationnelle. Chaque tâche ou ensemble de tâches est représenté par un enregistrement dans la base de données, ce qui simplifie la création, la lecture, la mise à jour et la suppression des données de tâches que l'utilisateur sera amené à faire.

5. MAQUETTE

*Maquette Balsamiq de la vision en mode **Tableau***



*Maquette Balsamiq de la vision en mode **Liste***



6. PLANNING

1. Créer des tâches avec dépendance, sous tâches + test :
   - Création de la classe Tache et de ses méthodes
   - Mise en place du patron Composite sur Tache pour SousTacheFinale et SousTache afin de créer les dépendances
   - Test des méthodes et des dépendances

2. Visualiser l'ensembles des tâches sous forme d'un bureau. Création nouvelle colonne + test.
   - Visualiser l'ensemble des tâches sous forme de listes et de sous listes si on a demandé le déploiement des sous tâches + tests :
   - Création de la classe ColonneLigne et de ses méthodes
   - Mise en place du patron Iterator sans coder les méthodes de Tri pour le moment
   - Création de la classe Archive et du patron Singleton sur celle-ci.
   - Intégration de ColonneLigne en JavaFX (Développement de l'interface en parallèle)

- Tests correspondants
- Eventuellement début de MVC / Observateur
- Implémentation de la visualisation des sous tâches
- Implémentation éventuelle de certaines fonctionnalités sur Tache et ColonneLigne

3. Glisser tâche d'une colonne à l'autre:
   - Mise en place des différentes vues et contrôleurs, finalisation éventuelle de l'interface JavaFX
   - Mise en place du patron Observateur et premières implémentations de fonctionnalités comme le drag & drop d'un ColonneLigne à une autre
   - Tests correspondants

4. Description qui peut être visualisée après sélection avec la souris + archiver des tâches. Gestion vue de l'archivage. Gestion des différents tris + test + début diagramme de Gantt:

   - Implémentation de la visualisation des tâches en détail
   - Implémentation des trois différentes méthodes de tris après les avoir codées dans Iterator
   - Implémentation de ces tris sur l'interface
   - Ajout des méthodes d'archivages et implémentation sur interface avec vue Archive
   - Tests correspondants

5. Génération le diagramme de Gantt. + test + ajout de fonctionnalités bonus (supprimer, déplacer tâche dans un sous menu):
   - Implémentation de la génération du diagramme de Gantt
   - Gestion de VueGantt et de l'affichage du diagramme sur l'interface JavaFX
   - Tests correspondants

6. Bilan du projet. Préparation compte rendu, diaporama, tests finaux, implémentations finales.