

Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

SAE 3.01 Développement d'applications

Rapport final Du logiciel d'organisation de tâches personnelles



Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

SOMMAIRE

I. Introduction.....	3
II. Réalisation du projet.....	4
• La liste des fonctionnalités réalisées.....	4
• La répartition du travail.....	5
➤ Nino.....	5
➤ Maëlle.....	5
➤ Célie.....	6
➤ Yehor.....	6
• Le diagramme de classe final.....	7
Voici le diagramme de classe final de notre application :.....	7
III. Architecture et conception.....	8
• Les éléments modifiés par rapport à l'étude préalable.....	8
• Les patrons de conception et d'architecture mis en place.....	10
IV. Interface graphique.....	11
• Les graphes de scène de l'interface.....	11
➤ La Vue Globale.....	11
➤ La Vue Bureau.....	12
➤ La Vue Liste.....	13
➤ La Vue Gantt.....	13
➤ La Vue Archive.....	14
V. Conclusion.....	15

Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

I. Introduction

Dans la suite de notre étude préalable, notre équipe a pu développer une application dédiée à l'organisation des tâches personnelles. Nous avons voulu que ce projet offre une solution intuitive et pratique, inspirée de Trello mais adaptée aux fonctionnalités demandés.

Ce compte rendu détaille notre parcours, met en avant les fonctionnalités implémentées, la répartition des tâches, le diagramme de classe final, l'architecture de l'application, ainsi que les aspects visuels de l'interface utilisateur.

Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

II. Réalisation du projet

• La liste des fonctionnalités réalisées

Lors de l'implémentation de notre application, nous avons réussi à réaliser les fonctionnalités prévues lors de l'étude préalable mais aussi à en ajouter un certain nombre pour faire évoluer l'application de base.

Voici la liste des fonctionnalités réalisés dont celles qui n'étaient pas prévues lors de l'étude préalable en vert :

- La gestion d'une tâche (supprimer, ajouter, modifier, archiver)
- La gestion d'une colonne (modifier, supprimer)
- La gestion d'un tableau (modifier, supprimer)
- La gestion des dépendances d'une tâche
- Générer le diagramme de Gantt en fonction d'une sélection de tâches
- L'affichage du tableau sous forme de liste
- L'affichage du tableau sous forme de bureau
- L'affichage de la liste des tâches archivées
- La gestion de la liste des tâches archivées
- Le Drag and Drop des colonnes dans la vue Bureau
- Le Drag and Drop des tâches dans la vue Bureau et la vue Liste
- Gérer un système de plusieurs tableau

Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

- **La répartition du travail**

Durant l'implémentation de cette SAE, nous avons en effet travaillé ensemble sur beaucoup de fonctionnalités de bases majoritairement afin d'être tous en accord sur les bases de notre application, mais nous avons aussi dû en répartir pour pouvoir avancer plus efficacement.

- **Nino**

- Gestion de l'esthétique de l'interface de l'application
- Vue Bureau et ses contrôleurs
- La gestion de la sélection de tâches pour générer le diagramme de Gantt
- La gestion des pop up d'ajout et de modification des tâches et des colonnes

- **Élément original dont tu es fière:**

“L'ajout dont je suis le plus fier est celui des pop-ups d'ajout et de modification des tâches, car ces fonctionnalités sont particulièrement efficaces pour ajouter ou mettre à jour des tâches complexes (c'est-à-dire des tâches avec des sous-tâches et des sous-tâches de sous-tâches, etc.). Ces ajouts utilisent de nombreuses récursivités, ce qui permet de gérer divers cas d'utilisation qui pourraient être particuliers. Ces fonctionnalités permettent de générer une arborescence complexe des tâches. Le problème qui se posait était de trouver comment récupérer les informations des champs remplis pour créer l'arborescence des tâches. La solution qui s'est montrée efficace était de créer un formulaire graphique qui respecte l'arborescence que l'utilisateur pouvait construire et de récupérer graphiquement les informations pour créer les tâches.”

- **Maëlle**

- Tests des classes “principales” du projet (Tâche, Colonne, Tableau)
- Test pour la gestion de l'archivage
- Vue Archive et ses contrôleurs
- Vérifications des dates des sous tâches pour l'ajout d'une tâche et le diagramme de Gantt

- **Élément original dont tu es fière:**

“L'un des éléments de cette itération dont j'ai été personnellement fière a été la mise en place des méthodes de vérifications pour les durées et les dates de chaque tâche et sous-tâches. Ce point du projet était de plus important pour pouvoir avoir un diagramme de Gantt cohérent. Dans notre application, nous voulions que les sous-tâches d'une tâche

Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

mère soit incluses dans ses dates, mais qu'elles ne se chevauchent pas entre elles."

➤ **Célie**

- Vue Liste et ses contrôleurs
- Vue Système (changement de tableau courant)
- Gestion de l'unicité des tâches et des colonnes
- La gestion de l'ajout des antécédents d'une tâche

○ **Élément original dont tu es fière:**

"Je suis particulièrement fière de l'ajout de la vue Système. Travailler sur cette partie m'a permis de comprendre l'importance de l'ordre d'actualisation des vues. J'ai trouvé le fait d'utiliser la classe Vue pour contrôler l'affichage des vue du modèle, intéressant. Comme la Vue était un observateur de Système il était intelligent qu' elle contrôle les vues du modèle contrôlé par Système."

➤ **Yehor**

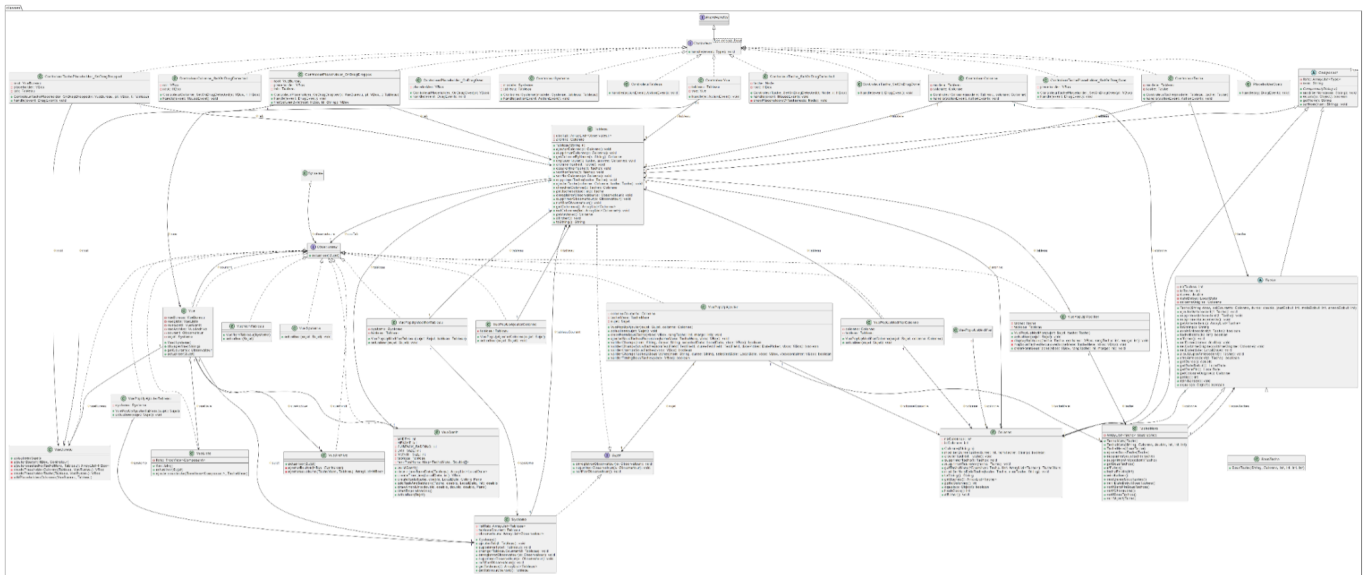
- Drag and drop des colonnes et des tâches dans la vue Bureau
- Vue Gantt et ses contrôleurs
- Drag and drop des tâches dans la vue Bureau
- Drag and drop des tâches dans la vue Liste

○ **Élément original dont tu es fier:**

"En tant qu'étudiant, je suis vraiment fier de l'implémentation de la fonctionnalité de drag and drop dans notre projet de gestion des tâches. Ce travail m'a permis de développer mes compétences en programmation et en conception d'interface utilisateur. Réussir à rendre cette fonctionnalité à la fois fluide et intuitive, tout en assurant la cohérence des données en arrière-plan, était un défi passionnant. Cette réalisation a renforcé ma compréhension de l'importance d'une interaction utilisateur efficace dans les applications modernes."

- **Le diagramme de classe final**

Voici le diagramme de classe final de notre application :



(une version .SVG est disponible dans le dossier)

Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

III. Architecture et conception

- Les éléments modifiés par rapport à l'étude préalable

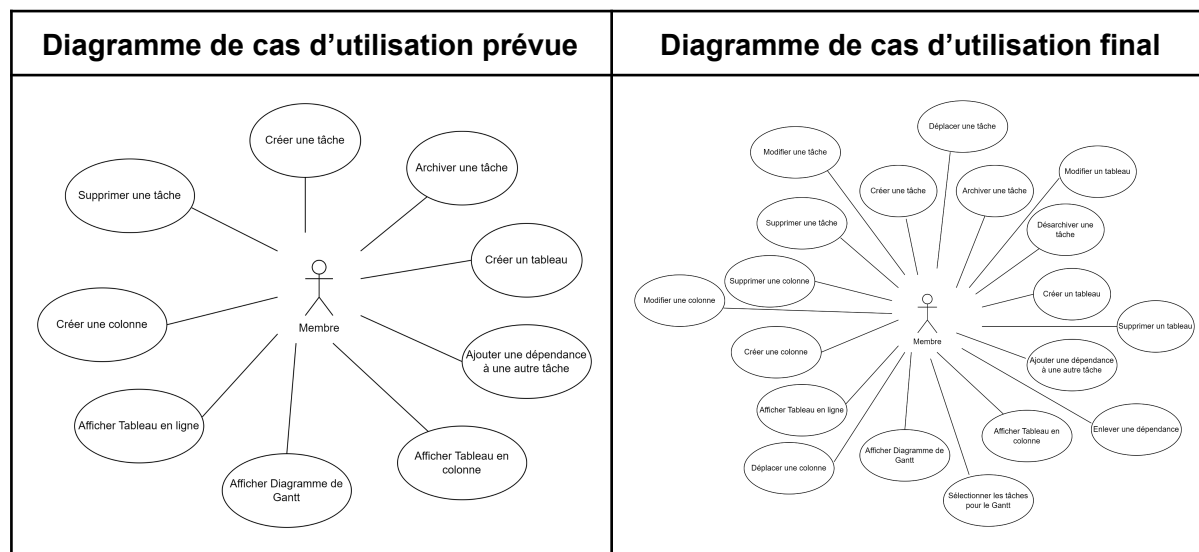
L'étude préalable de notre application n'a pas spécifiquement subi de modifications, mais plutôt des évolutions.

Tout d'abord, notre liste de fonctionnalités à évoluer. Lors de l'étude préalable, nous avons prévu les fonctionnalités de base de *Trello*, une vision simpliste de notre application. C'est pour cela que nous avons pu ajouter à notre planning d'itérations des fonctionnalités supplémentaires.

Fonctionnalités prévues	Fonctionnalités réalisées
<ul style="list-style-type: none"> • Gérer une tâche (supprimer, ajouter, modifier) • Gérer une colonne • Gérer un tableau • Gérer les dépendances d'une tâche • Générer le diagramme de Gantt • Afficher le tableau en liste • Afficher le tableau en bureau • Gérer la liste des tâches archivées 	<ul style="list-style-type: none"> • Gérer une tâche (supprimer, ajouter, modifier, archiver) • Gérer une colonne (modifier, supprimer) • Gérer un tableau (modifier, supprimer) • Gérer les dépendances d'une tâche • Générer le diagramme de Gantt en fonction d'une sélection de tâches • Afficher le tableau en liste • Afficher le tableau en bureau • Gérer la liste des tâches archivées • Déplacer les colonnes de place • Déplacer les tâches de place • Gérer un système de plusieurs tableau

Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

Notre diagramme de cas d'utilisation de l'étude préalable a donc lui aussi évolué.



Pour ce qui est du diagramme de classe, nous avons conservé les mêmes patrons que prévu et la même base du diagramme. La seule différence se trouve aussi sur l'ajout de fonctionnalités et donc l'ajout de vues et contrôleurs supplémentaires. (cf *diagramme de classe final dans le dossier*)

Enfin, nos maquettes ont également évolué. Les bases des maquettes sont restées les mêmes mais nous avons fait évoluer l'application en lui donnant plus d'aspects esthétiques, avec l'ajout de couleurs, de police différentes et de pop up. Ceci a rendu l'application plus ergonomique et améliore ainsi l'expérience utilisateur. (cf *présentation de la soutenance dans le dossier*)

• Les patrons de conception et d'architecture mis en place

Les patrons de conception et d'architecture utilisés étaient les mêmes que nous avions prévu lors de l'étude préalable et nous n'avons pas eu besoin de nouveaux.

- Le patron de conception **Observateur**:

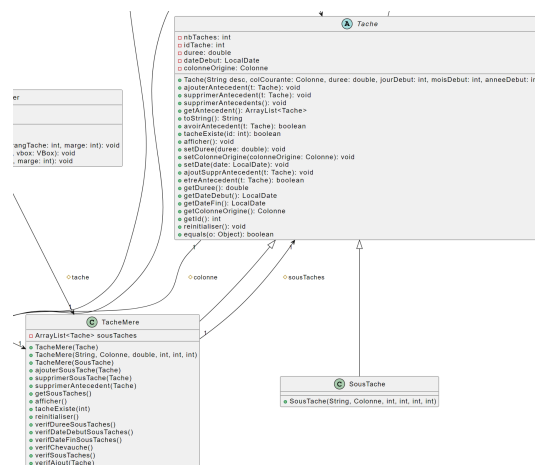
Ce patron nous a été utile pour la gestion et l'actualisation de nos différentes et nombreuses vues sans "casser" le projet. Grâce à ce patron, nous avons pu implémenter 13 vues en comptant les vues des pop up , des différents affichages et de la fonctionnalité du Drag and Drop.



(Voir dans le dossier ZIP pour accéder au format SVG)

- Le patron de conception **Composite**:

Ce patron nous a permis de gérer au mieux les tâches et les sous-tâches. pour prendre en compte les tâches et les sous-tâches. Grâce au patron Composite, les tâches mère peuvent avoir des sous-tâches de sous-tâches jusqu'à l'infini.



- Le patron d'architecture **MVC**:

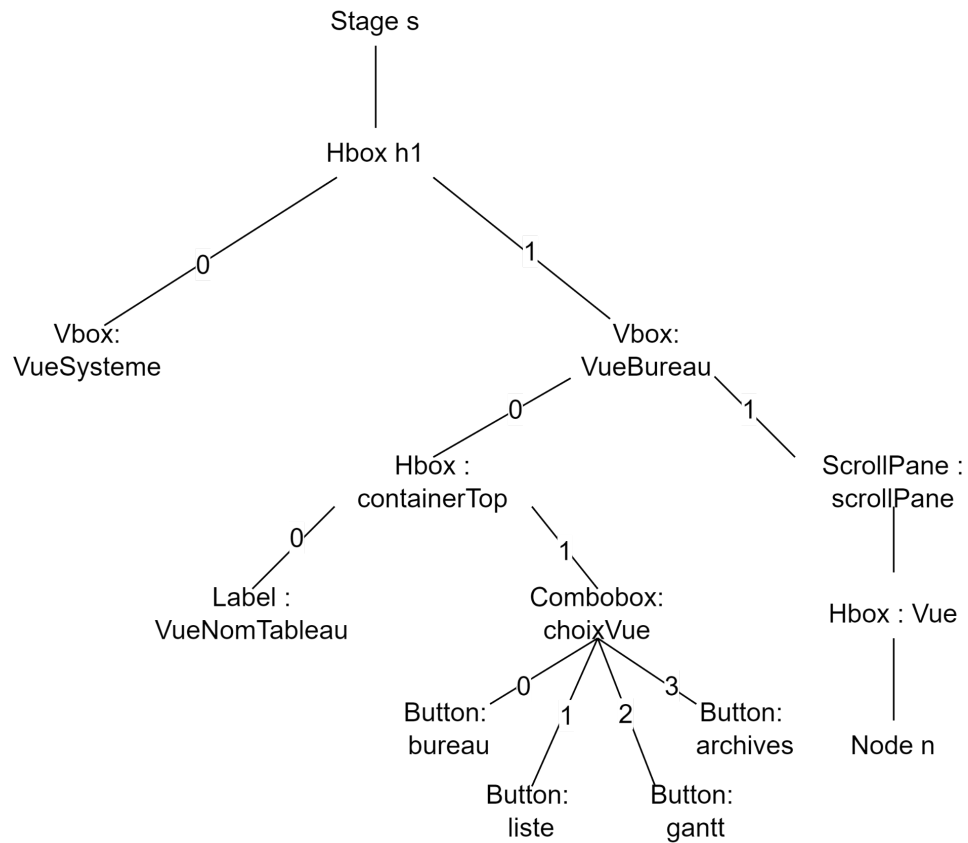
Ce patron nous a permis de structurer notre projet et de mettre en place les différents contrôleurs et vues. (Voir diagramme de classe format SVG dans le dossier)

IV. Interface graphique

- Les graphes de scène de l'interface

L'interface de notre application se divise en 5 vues dont la vue globale. Voici donc les différents graphes de scène qui illustreront ce qu'il y a "derrière" ces différentes vues.

➤ La Vue Globale

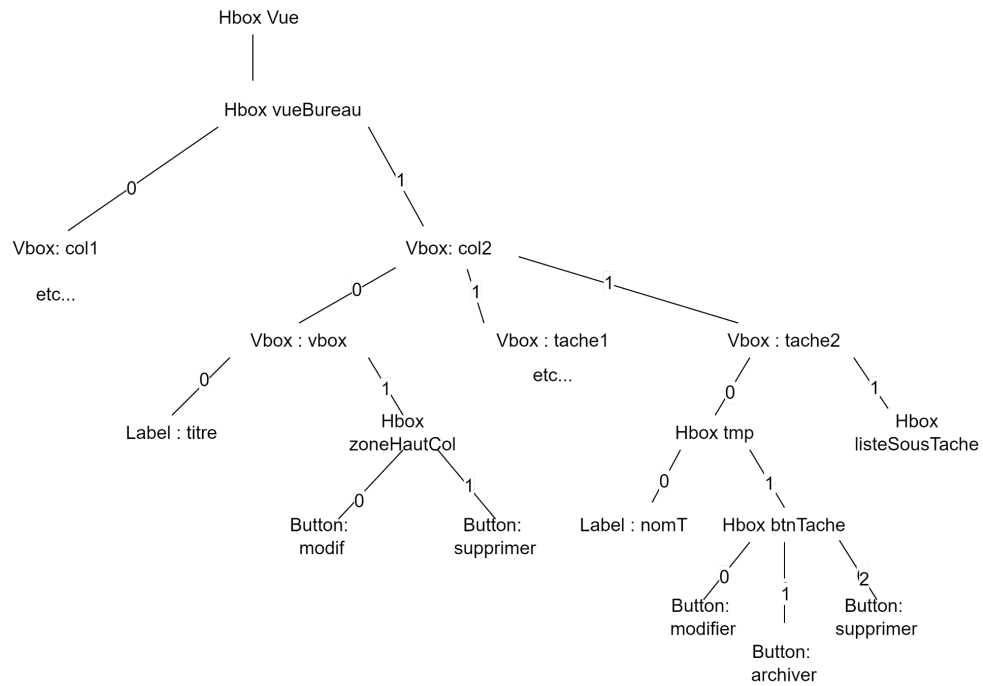


Node dépend de la vue qui est en cours (Bureau, Liste, Gantt ou Archives).

➤ La Vue Bureau

Dans ce diagramme, nous supposons que dans le bureau il y a :

- colonne1
- colonne2
 - tâche 1
 - tâche 2

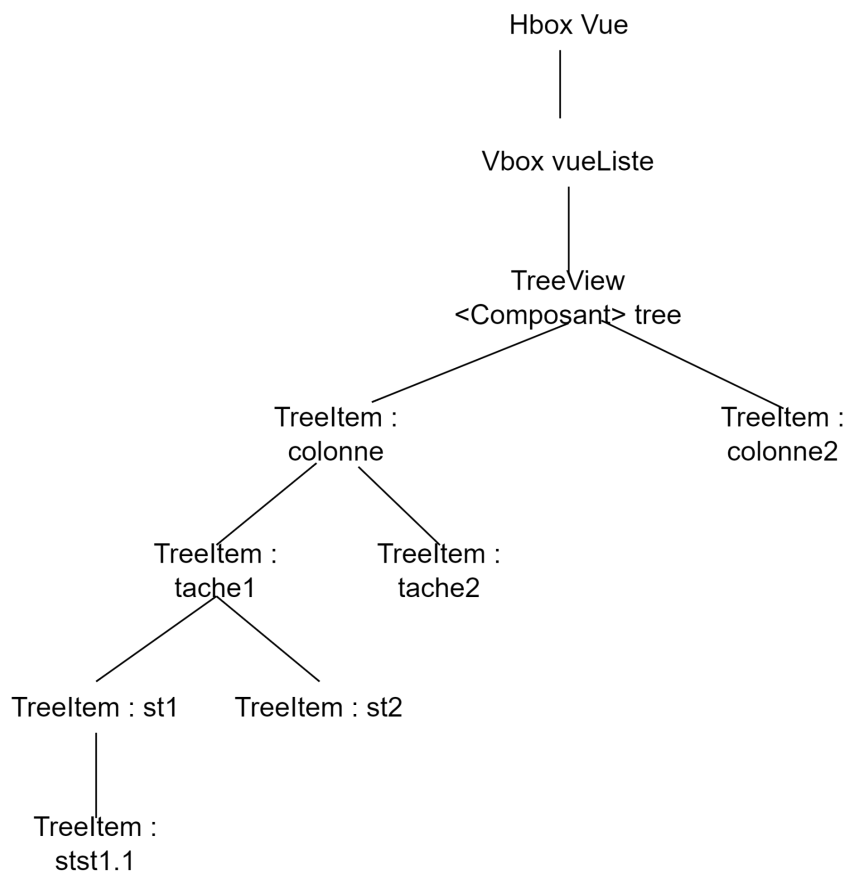


Par soucis de place, les placeholder ne sont pas représentés dans ce graphe.

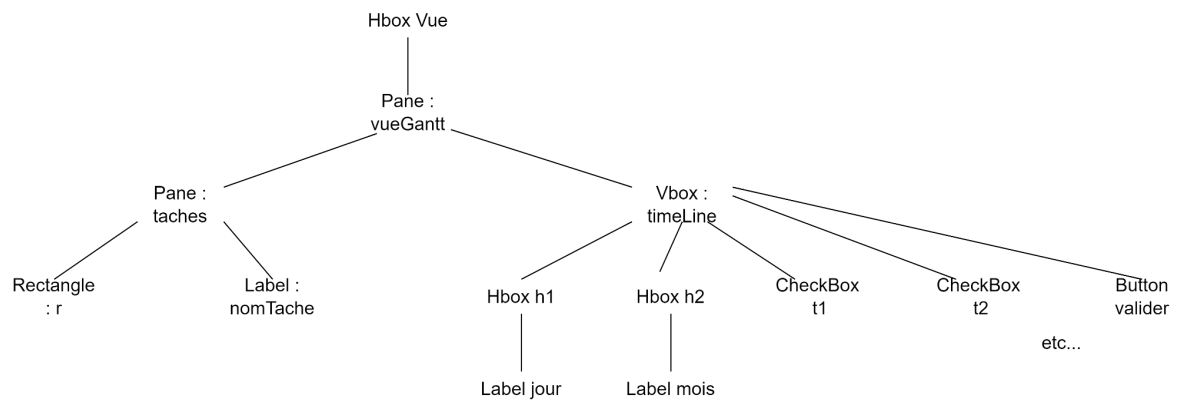
➤ **La Vue Liste**

Dans ce diagramme, nous supposons que dans la liste il y a :

- colonne1
 - tâche 1
 - sous tâche 1.1
 - sous tâche 2
 - tâche 2
- colonne 2

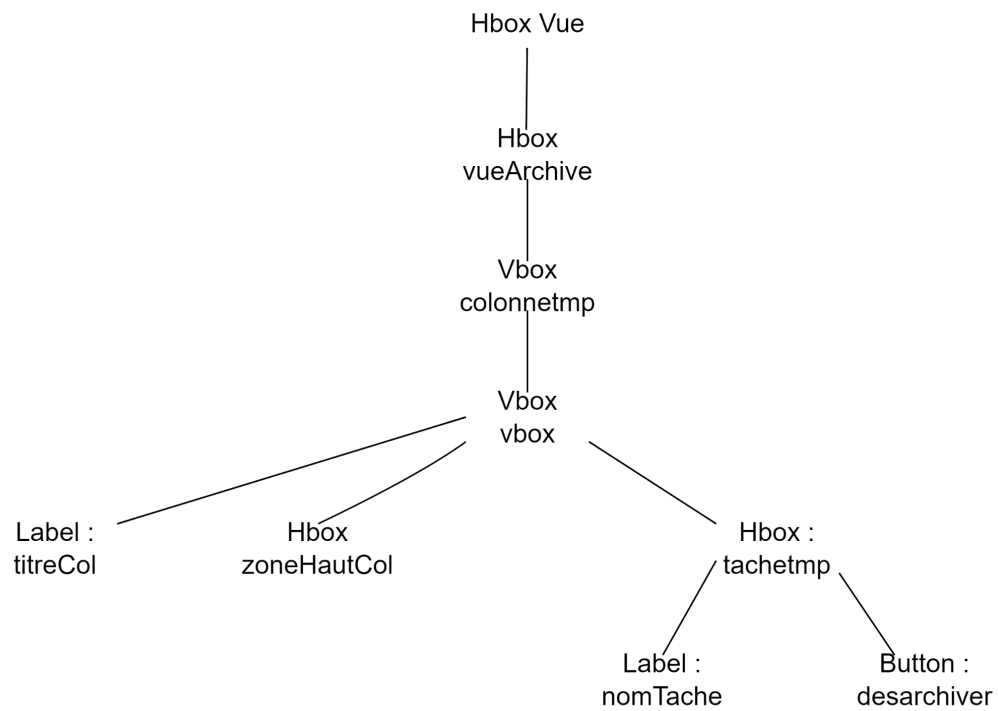


➤ **La Vue Gantt**



Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

➤ **La Vue Archive**



Nino Arcelin S3D	Maëlle Bitsindou S3D	Célie Ponroy S3D	Prykhodko Yehor S3D
------------------	----------------------	------------------	---------------------

V. Conclusion

Au terme de cette SAÉ, nous sommes fiers de présenter une application qui va au-delà de nos attentes initiales et fonctionnelle. La répartition du travail a permis à chacun de contribuer de manière significative, renforçant la qualité globale du projet.

Le diagramme de classe reflète la solidité de notre architecture et les interfaces graphiques, illustrées à travers les graphes de scène, assurent une expérience utilisateur fluide

En conclusion, notre application de gestion de tâches personnelles est le résultat de nos efforts. Ce projet nous a permis de voir que nous pouvions concevoir une application fonctionnelle et ergonomique.