

Rapport de l'itération 1 :

Le but de cette itération est que l'utilisateur puisse créer des tâches dans les deux colonnes/listes qui lui sont mises à disposition et que le squelette de l'application soit fait avec le nom de l'application etc...

Pour lancer notre application il faut lancer le fichier PrincipalFx.java

Fonctionnalités:

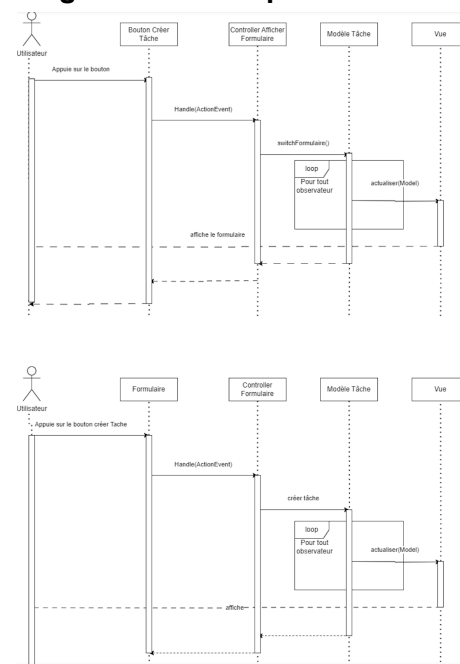
Base Graphique:

Notre application affiche une page principale avec deux colonnes et des tâches dedans, dans chaque colonne, on trouve un bouton ajouter tâche sur lequel l'utilisateur peut cliquer. Le nom de l'application est aussi affiché. **Fait par Timothée Benchergui**

Créer Tâche:

L'utilisateur pourra ajouter une tâche à la liste choisie en cliquant sur le bouton ajouter une tâche qui se trouve en bas de la liste/ colonne choisie. Quand l'utilisateur appuie sur le bouton alors un formulaire apparaît et il doit rentrer le nom et la description de la tâche. Ensuite il lui reste plus qu'à valider la création de la tâche en appuyant sur le bouton de validation et la tâche apparaît dans la liste/ colonne. **Fait par Hugo Rysak et Jawad Komodzinski**

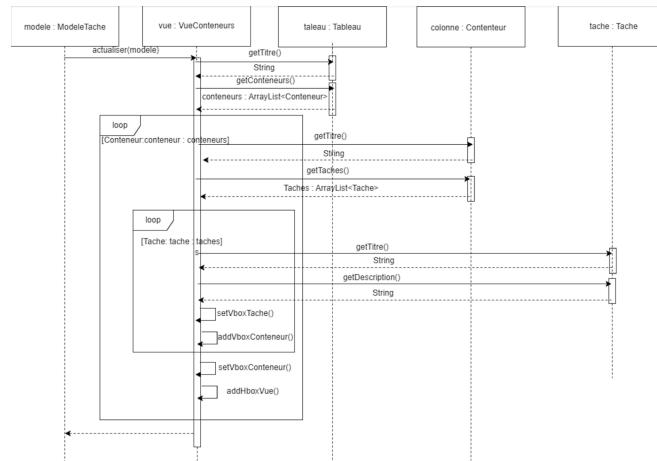
Diagramme de séquence liée à la fonctionnalité:



Afficher Tâche:

Les tâches s'affichent dans les bonnes listes/ colonnes choisies, on peut y voir son nom et sa description (elle sera visible plus tard dans la vue en détails). **Fait par Maxime Bechard**

Diagramme de séquence liée à la fonctionnalité:



Toutes les fonctionnalités ci-dessus fonctionnent et ont été testé.

Quelques informations sur le code:

Modèle :

Tâche:

Une tâche pouvant contenir plusieurs tâches enfants, nous avons décidé de faire un patron composite. Pour l'instant seul les tâches mère peuvent avoir des enfants. Nous avons donc la classe abstraite tâche qui possède un nom et une description ainsi qu'une méthode affichage. Une tâche mère possède en plus une liste de sous-tâches.

Conteneur:

Un conteneur contient plusieurs tâches et pour l'instant ne peut être affiché qu'en colonne. Le conteneur possède aussi un titre. On peut ajouter une tâche au conteneur ou supprimer une tâche.

Tableau:

Un tableau contient plusieurs conteneurs. Pour l'instant lors de l'itération 1, le tableau contient deux conteneurs de base et l'utilisateur ne peut pas en rajouter ni en supprimer

MVC:

Modèle:

Notre modèle possède une liste d'observateurs, un tableau, un entier représentant la colonne sélectionnée par l'utilisateur, il possède aussi un boolean formulaire utilisé pour savoir s' il faut montrer le formulaire ou non. Le Modèle

possède une méthode pour changer la colonne sélectionnée par l'utilisateur, une méthode pour créer une tâche et une méthode pour mettre le boolean formulaire à true si il est false et vice versa.

Controller:

ControllerAfficherFormulaire: Controller utilisé pour afficher le formulaire quand le bouton ajouter un tâche est cliqué

ControllerCréerTache: controller utilisé pour Créer la tache et retirer le formulaire de l'affichage quand le bouton créer tache ou annuler est cliqué. On peut noter qu'il possède deux attributs en plus du modèle qui sont deux textField pour que le controller puisse récupérer le nom et la description de la tâche donnée par l'utilisateur.

Vue:

VueContainers : vue qui gère l'affichage des tache dans les containers

VueFormulaire : vue qui gère l'affichage du formulaire