

Membres du groupe :

- Baptiste Besoin
- Thomas Charpentier
- Éloi Duchêne
- Alexis Zott

Étude préalable-Partie Analyse

Trello

Liste des fonctionnalités	3
Cas d'utilisation	4
Diagrammes de cas d'utilisation	5
Diagramme d'activité et d'état	7
Diagramme d'activité	7
Diagramme d'état de l'objet central de l'application	7
Diagramme de classe et patrons	8
Les patrons de conception et d'architecture prévues	8
Modèle relationnel	8
Domaine	8
Diagrammes de classe	9
Domaine	9
Conception	9
Conception (pour la première itération)	11
Maquette	12
Accueil :	12
Itérations	15
Premier sprint	15
Deuxième sprint	15
Troisième sprint	16
Quatrième sprint	16
Cinquième sprint	16
Sixième sprint	16

Liste des fonctionnalités

- Créer des listes
- Créer des tâches
- Créer des projets
- Mettre des titres aux projets
- Archiver des tâches
- Confirmer l'archivage
- Visualiser les tâches (sous forme de liste et de sous-liste ou de bureau)
- Dépendances entre tâches
- Générer un diagramme de Gantt des tâches sélectionnées
- Déplacer une tâche d'une liste à une autre
- Déplacer une liste
- Dupliquer une liste
- Dupliquer un tableau
- Dupliquer une tâche
- Se s'inscrire
- Se connecter
- Se déconnecter
- Affecter une / des personnes à des tâches
- Mettre des tâches en urgentes;
- Changer la couleur de fond des listes
- Ajouter quelqu'un dans un tableau
- Supprimer quelqu'un dans un tableau
- Système d'admin de tableau (lui seul a les droits d'ajouter et d'enlever des personnes)

L'application doit implémenter les fonctionnalités suivantes :

- créer des tâches qui peuvent dépendre d'autres tâches, par exemple une tâche incluse dans une autre tâche ou encore une tâche ne pouvant démarrer que si une autre est terminée
- archiver des tâches une fois celles-ci réalisées
- visualiser l'ensemble des tâches sous forme d'un bureau avec des colonnes regroupant des tâches, une tâche pouvant être glissée déposée d'une colonne à une autre. La description d'une tâche (et de ses sous-tâches) pourra être visualisée après sélection avec la souris
- visualiser l'ensemble des tâches sous forme de listes et de sous-listes si le déploiement des sous-tâches est demandé
- générer le diagramme de Gantt d'une sélection de tâches afin de visualiser la chronologie de la réalisation de celles-ci.

Cas d'utilisation

Ajouter une liste

Ajouter une tâche

Créer un tableau

Déplacer une tâche

Déplacer une liste

Archiver tâche

Archiver liste

Archiver tableau

Modifier nom liste

Modifier texte tâche

Ajouter nom tableau

Visualiser les tâches :

- Liste
- Bureau

Ajouter des dépendances entre les tâches

Générer un diagramme de Gantt

Dupliquer une liste

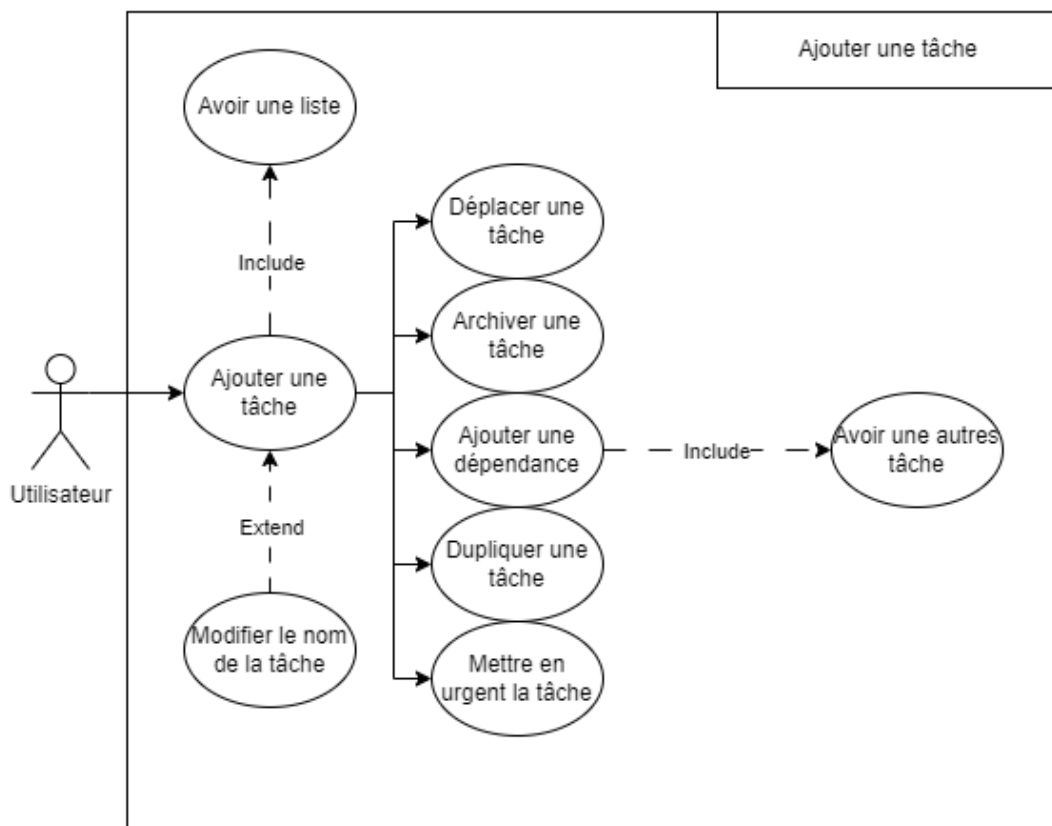
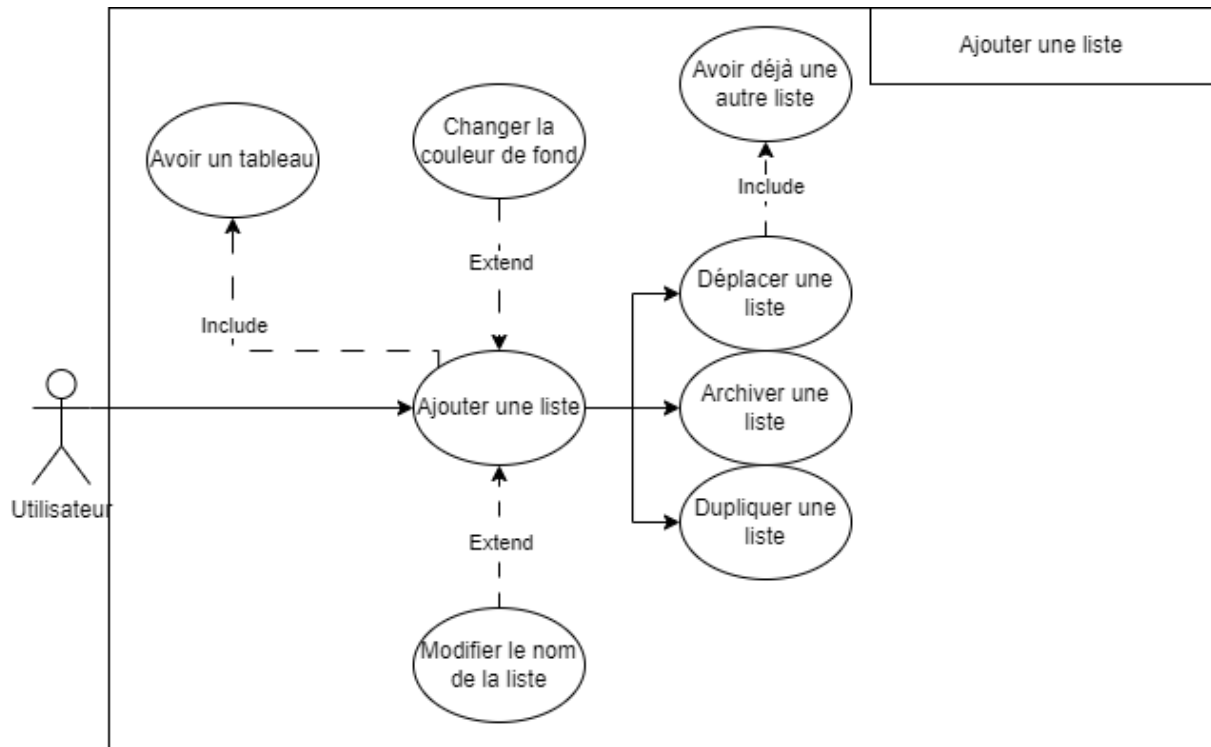
Dupliquer une tâche

Dupliquer un tableau

Modifier couleur liste

Pouvoir mettre en urgent une tâche

Diagrammes de cas d'utilisation



La description textuelle prend en compte le fait de générer un diagramme de Gantt :

Générer un diagramme de Gantt à partir d'une liste de tâches

Préconditions : Avoir un tableau, au moins une liste et une tâche.

Postconditions : L'utilisateur peut visualiser le diagramme de Gantt de son projet.

Déroulement normal :

L'utilisateur crée un tableau

Il ajoute au moins une liste

Il ajoute au moins une tâche à cette liste

Il clique sur générer un diagramme de Gantt

Variantes :

L'utilisateur ajoute des dépendances entre les tâches.

Contraintes non fonctionnelles :

Performance : le système doit générer le diagramme dans un délai inférieur à 5 secondes.

Sécurité : l'application doit être résistante à tout type d'attaques.

Le diagramme de Séquence Système représente l'archivage de toutes les tâches d'une liste

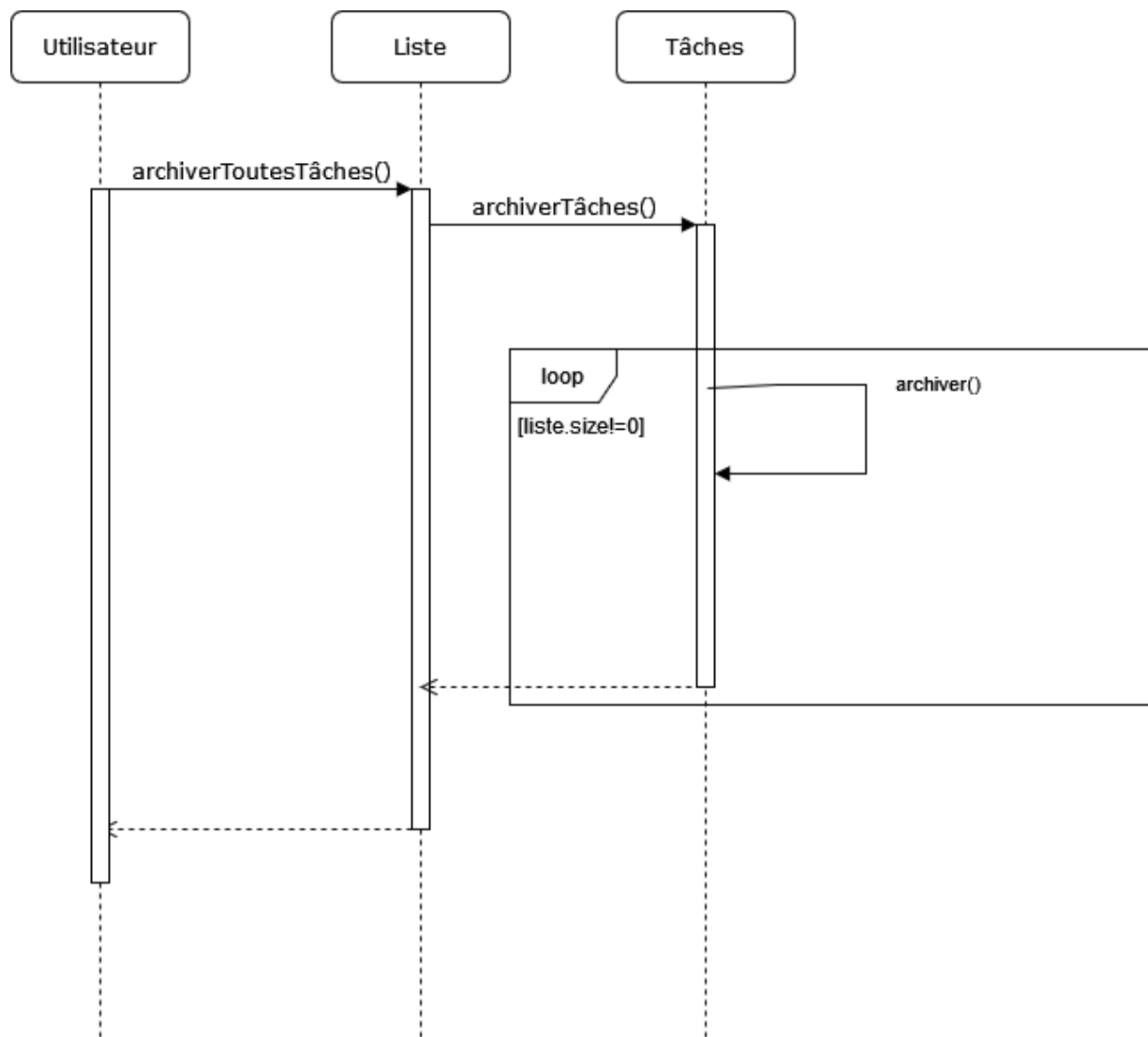


Diagramme d'activité et d'état

Diagramme d'activité

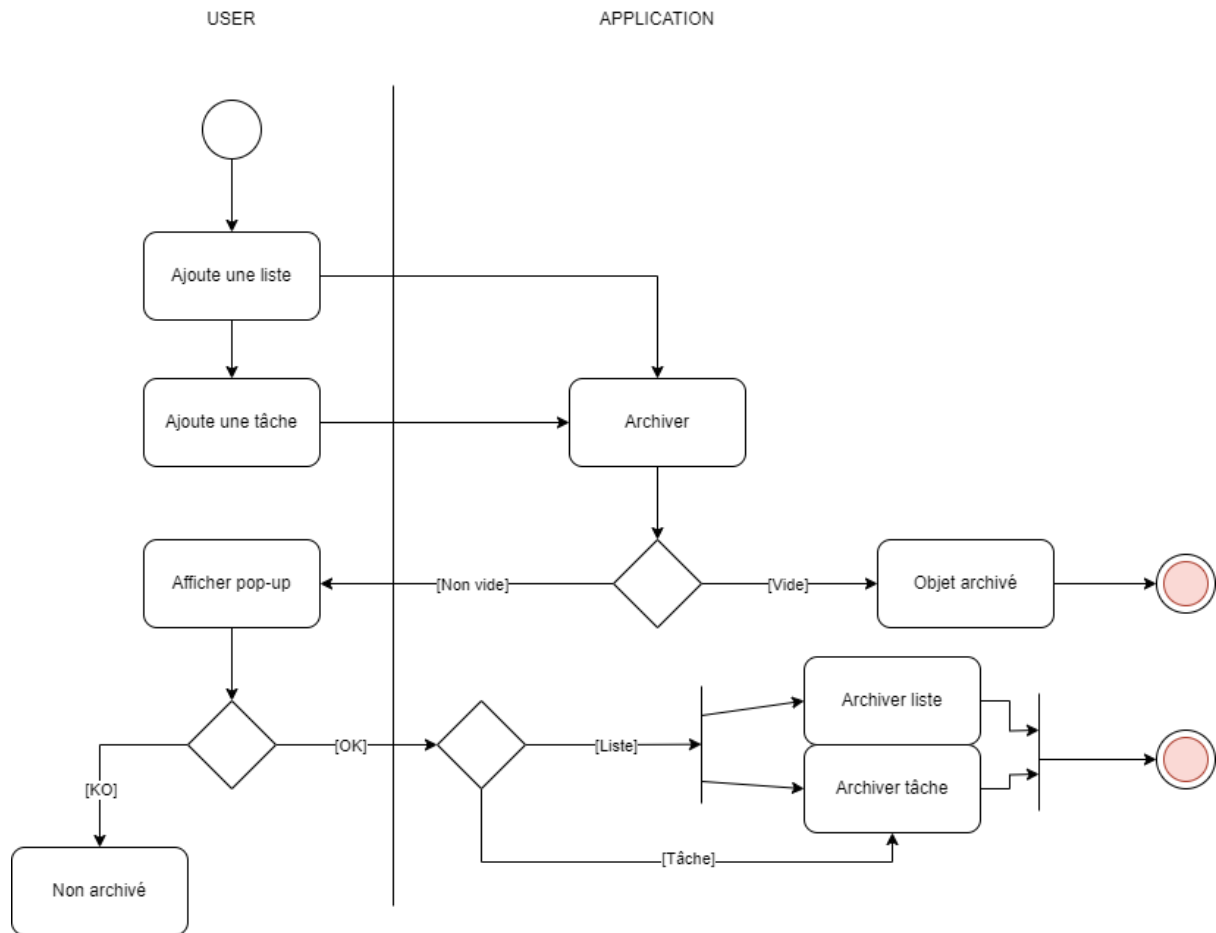


Diagramme d'état de l'objet central de l'application

Pour le diagramme d'état, nous avons choisi de prendre comme objet central les données (le modèle). C'est vraiment le cœur de notre application, c'est ici qu'il se passera le plus de changement (ajout tableau / liste / tâche, déplacer liste / tableau, modification d'une liste / tâche ...). Voici le diagramme d'état de notre application :

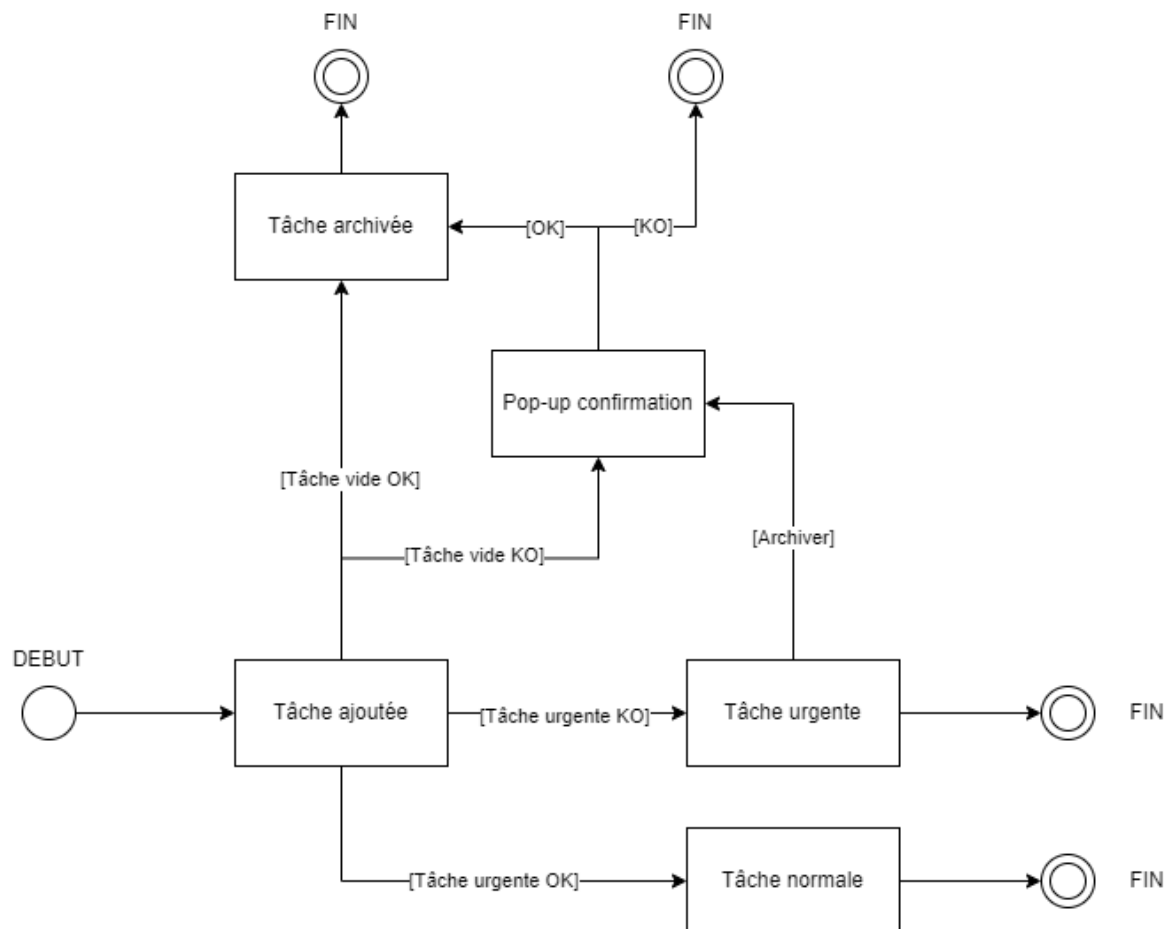


Diagramme de classe et patrons

Les patrons de conception et d'architecture prévus

Tout d'abord, nous comptons utiliser le patron d'architecture MVC et ainsi le patron de conception Observateur/Observé. Ces deux patrons nous permettront d'afficher et de gérer les mouvements des tâches et des listes. Nous allons aussi utiliser le patron d'architecture Active Record, ce qui nous permettra de gérer la base de données. Pour pouvoir utiliser ce dernier patron, il faut aussi utiliser le patron de conception Singleton qui sera utile pour créer une seule instance de la connexion à la base de données.

Nous n'excluons pas la possibilité d'utiliser le patron Décorateur afin d'implémenter des fonctionnalités supplémentaires sur les listes où les tâches.

Modèle relationnel

Domaine

Utilisateur(**idUtilisateur**, nom, prenom, mail, mdp)

Tableau(**nomTab**)

Liste(**idListe**, #idTableau)

Tache(**idTache**, nomTache, contenu)

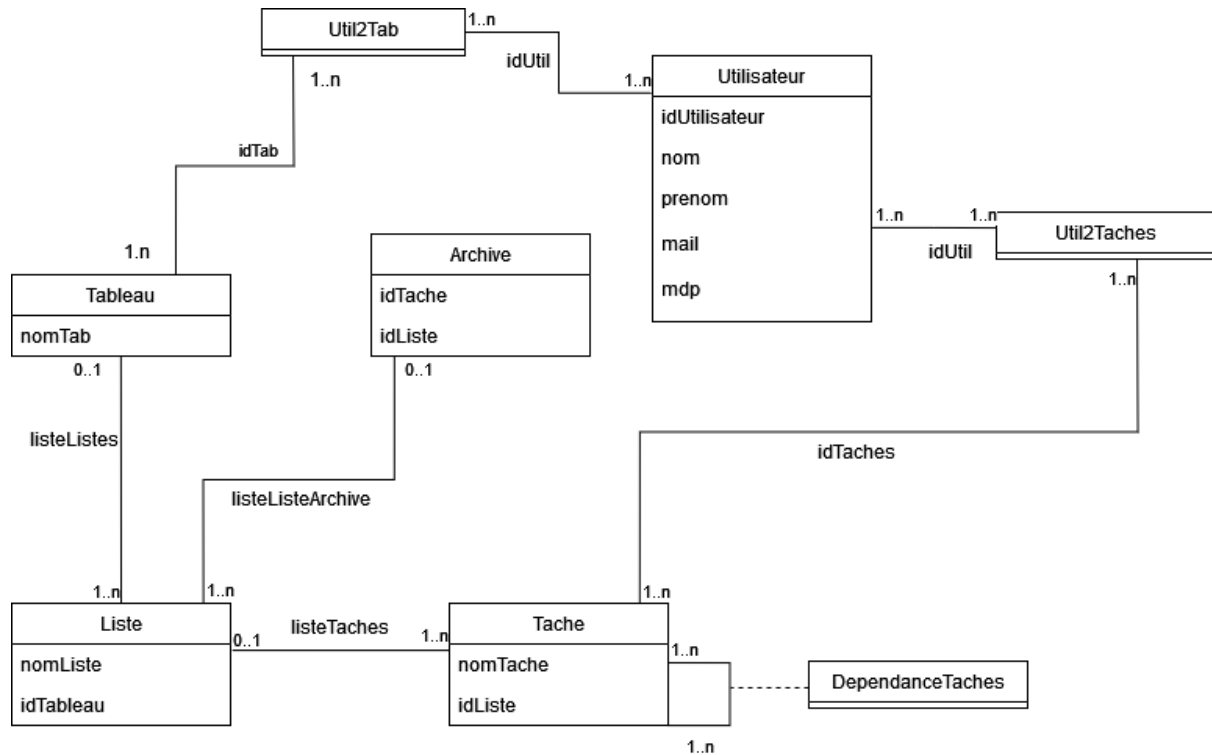
Tableau(**#idTache**, **#idListe**)

DependanceTache(**#tacheMere**, **#tacheFille**)

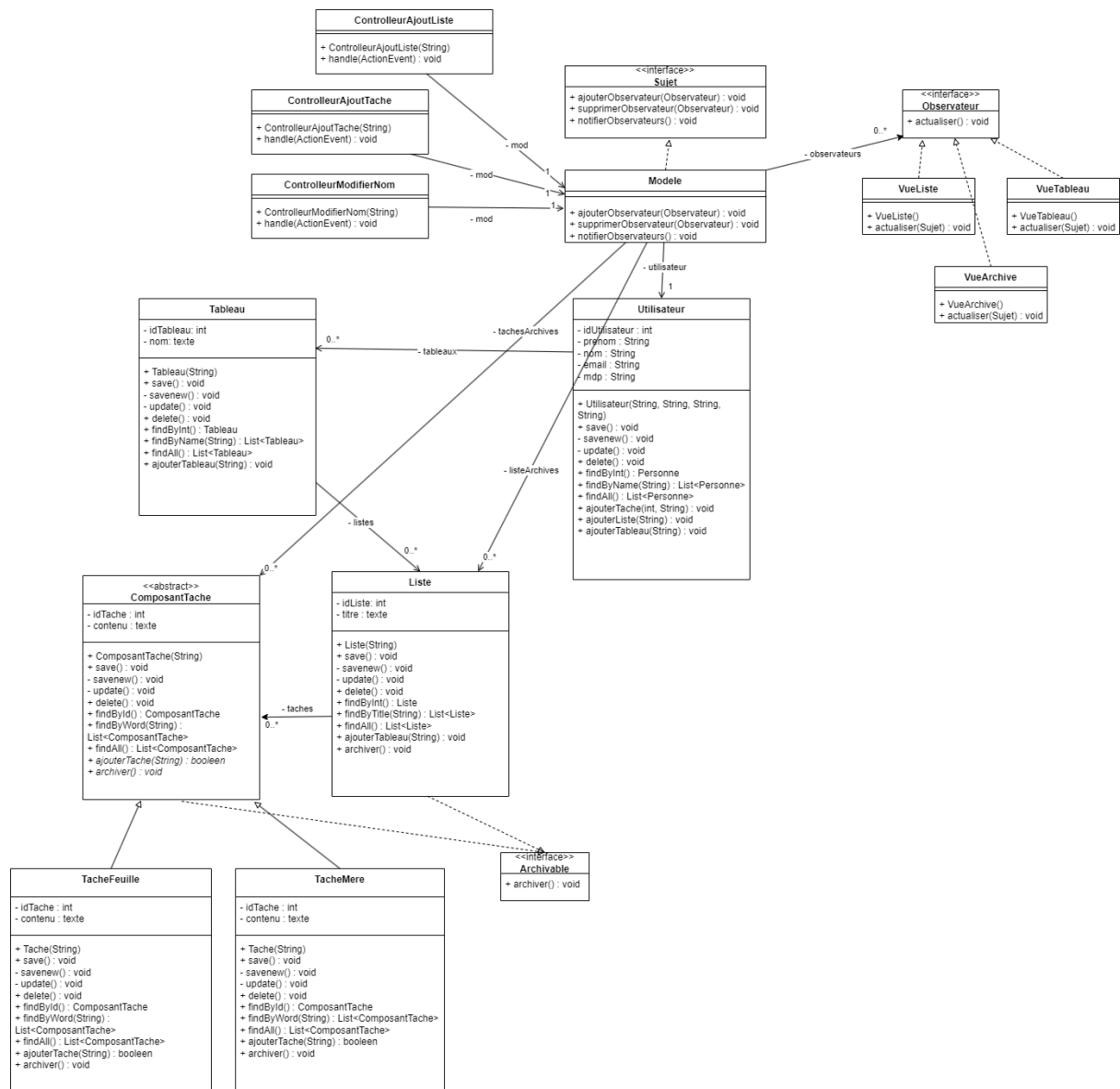
Archive(**#idListe**, **#idTache**)

Diagrammes de classe

Domaine

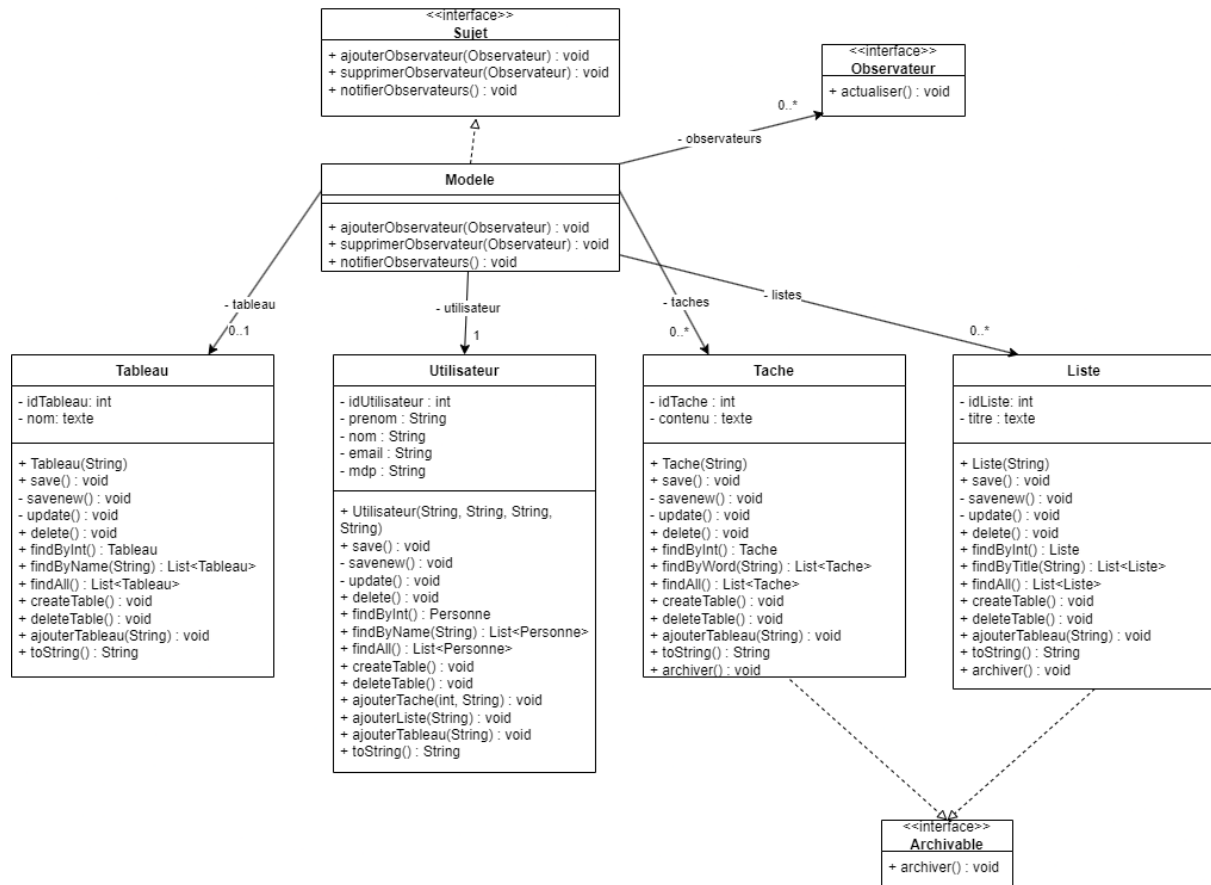


Conception



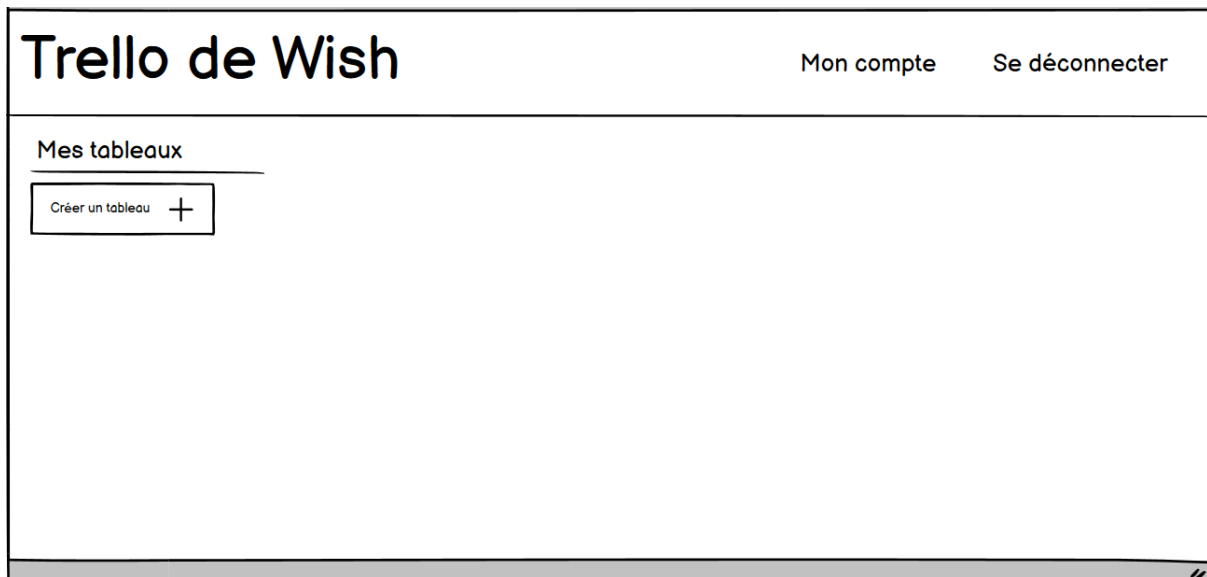
Conception (pour la première itération)

Utilisation de l'architecture logicielle MVC, et des patrons Observateur/Observé, Active Record. Nous envisageons d'utiliser d'autres patrons dans les prochaines itérations selon nos besoins, nous aurons sûrement besoin du patron Composite pour créer une arborescence de tâches afin qu'on puisse manipuler des tâches dépendantes ou non. La classe Utilisateur étant une

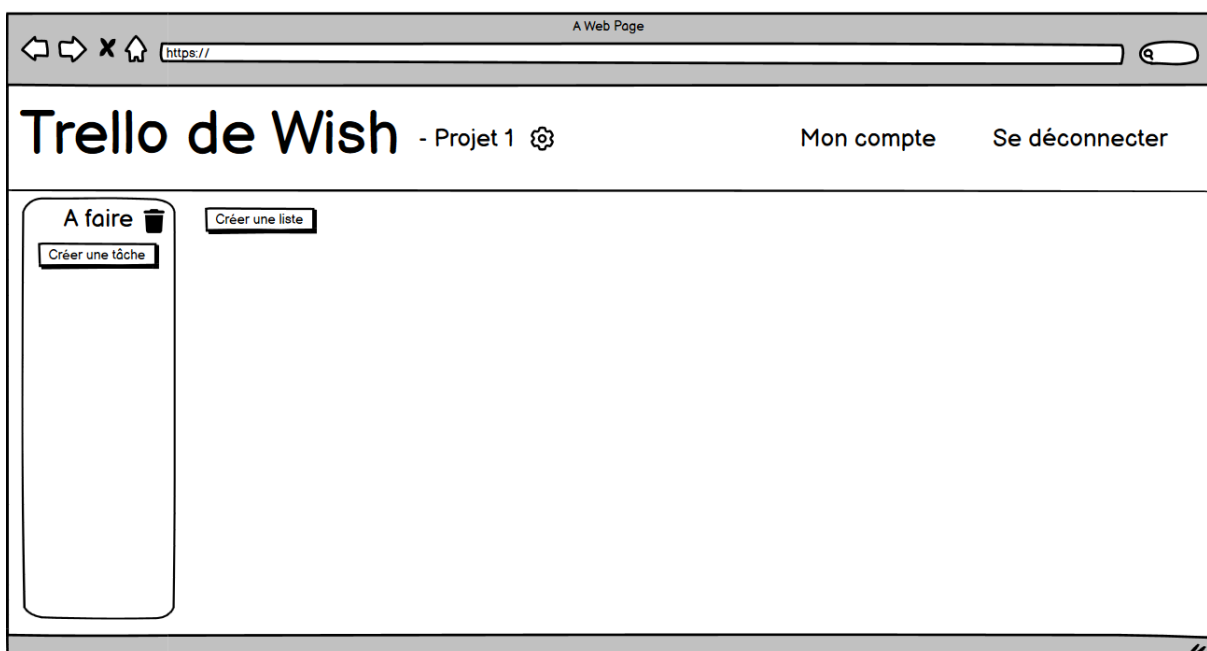


Maquette

Accueil :



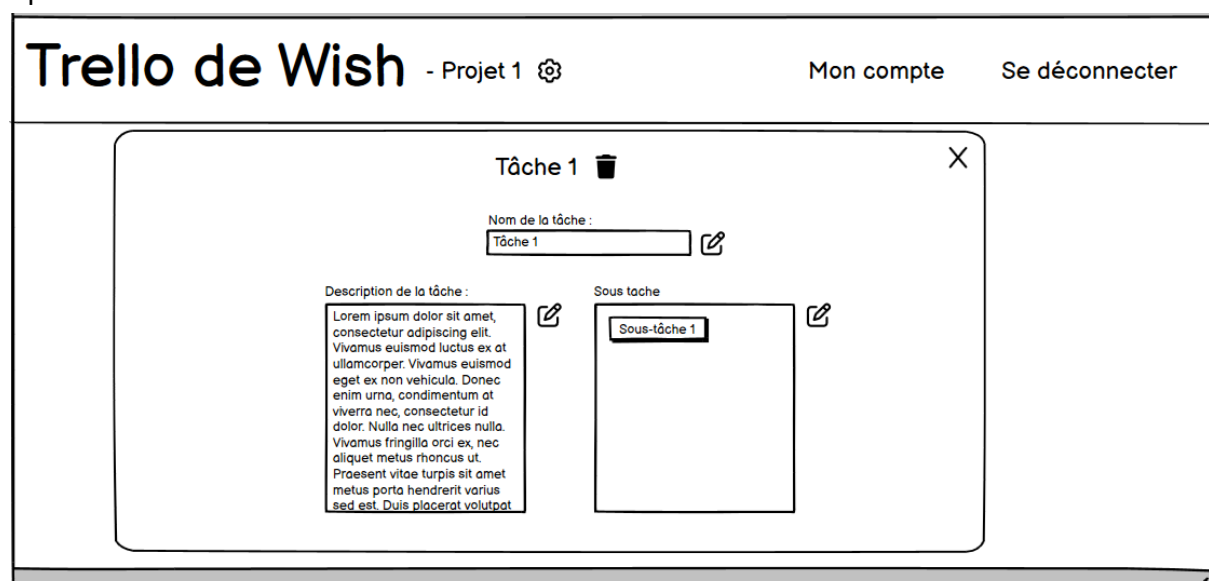
Trello après une liste créée :



Trello déployé, avec des tâches et sous tâche créées :



Options d'une tâche :



Options du tableau :

A Web Page

https://

Trello de Wish

- Projet 1

Mon compteSe déconnecter

Nom du tableau :

Projet 1

Membres : +

johnDoe54

thomasChorp57

eloiDuch

AlexisZotttttt

Couleur de fond :

Générer Diagramme de Gantt

Itérations

Le planning des 6 itérations prévues et les objectifs de chacune (en termes de cas d'utilisations) avec identification des risques.

Premier sprint

Objectifs :

- Créer la structure graphique (utilisation de classes JavaFX)

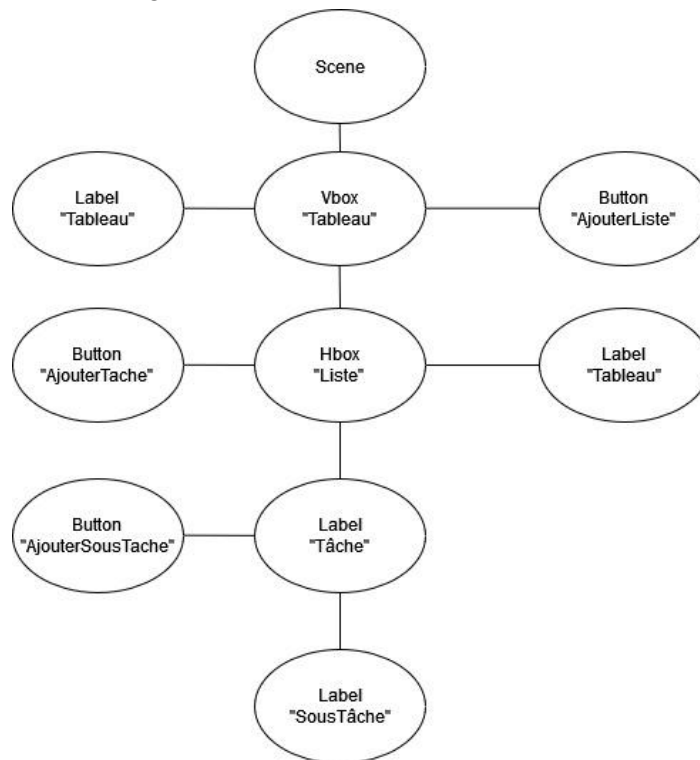


Diagramme de scène que nous allons certainement utiliser

- Création des méthodes de création de listes et de cartes.
 - Condition de validation : Les listes possèdent un nom qui peut être modifié et sont vides à la création, mais peuvent contenir des tâches. Les tâches peuvent posséder un nom qui peut être modifié et peuvent avoir une sous tâches (même comportement qu'une tâche).

Identifications des risques :

- Soucis de gestion de l'interface graphique.

Deuxième sprint

Objectifs :

- Création des tableaux
 - Condition de validation : un nouveau tableau vide peut être sélectionné
- Pouvoir archiver les tâches et les listes

- Condition de validation : la tâche ou la liste est bien déplacée dans la classe archivée et n'est plus affichée dans le tableau correspondant.

Identifications des risques : Vérifier que les listes sont affichées dans le bon tableau.

Troisième sprint

Objectifs :

- Pouvoir sauver les données et les lire
- Pouvoir mettre une tâche en urgente (elle devient prioritaire sur les autres et un moyen visuel est défini pour la remarquer)

Identifications des risques : Risques de problème à la lecture des données enregistrées.
Vérification de la mise en urgence de la tâche et du moyen de visualisation

Quatrième sprint

Objectifs :

- Méthodes de générations d'un diagramme de Gantt, donc il faut pouvoir ajouter une date de début et de fin à une tâche
- Duplication des listes et des tâches

Identifications des risques : Faire attention au placement des cartes et prendre en compte le cas des tâches sans dates. Vérifier que la tâche dupliquée ne modifie pas les deux tâches et de même pour les listes

Cinquième sprint

Objectifs : Pouvoir masquer des listes et masquer des tâches

Identifications des risques : Vérifier que seulement la liste ou la tâche sélectionnée n'est pas affichée, mais pas supprimé ou archivée.

Sixième sprint

Objectifs :

- Ergonomie et accessibilité
- Design et charte graphique

Identifications des risques : Les tâches ou les listes ne pourraient pas s'afficher correctement