

SAE 3.01 Etude préalable

*BECHARD Maxime
BENCHERGUI Timothée
KOMODZINSKI Jawad
RYSACK Hugo*

Introduction

Le cœur du projet de la SAE 3.01 est de développer une application de gestion de tâches personnelles. Pour se faire nous avons réalisé une étude préalable comprenant une partie analyse et conception.

Dans la phase d'analyse, nous nous concentrons sur la définition claire des fonctionnalités attendues de l'application et les cas d'utilisation qui guideront son interaction avec l'utilisateur. Ce qui inclut la liste des fonctionnalités, les diagrammes de cas d'utilisations ainsi que des descriptions textuelles, des DSS ou encore des diagrammes d'activités et d'états.

Quant à la phase de conception, nous déterminerons la conception générale de l'application. Ce qui inclut un diagramme de classe de la première itération, une maquette balsamiq de l'application et un planning des 6 itérations prévu et les objectifs de chacune.

Ce rapport servira de document de référence pour guider le développement de l'application au cours des semaines à venir.

Commençons la partie analyse avec la liste des fonctionnalités. Cette liste servira de ligne directrice pour l'ensemble de la phase d'analyse.

Liste des fonctionnalités

Création de tâches :

- Permet à l'utilisateur de créer de nouvelles tâches.
- Possibilité de définir un nom, une description et des détails pour chaque tâche.
- Prise en charge de la création de tâches dépendantes d'autres tâches.

Affichage des détails de la tâche :

- Sélection d'une tâche pour afficher ses détails.
- Affichage des sous-tâches, des dépendances et d'autres informations pertinentes.

Archivage des tâches :

- Permet à l'utilisateur de marquer une tâche comme réalisée.
- Archivage des tâches réalisées pour ne pas encombrer la vue principale.

Visualisation du bureau :

- Affichage graphique des tâches sous forme de bureau avec des colonnes représentant différents états (à faire, en cours, terminé).
- Possibilité de déplacer les tâches entre les colonnes pour refléter leur progression.

Déplacement des tâches entre les colonnes :

- Permet à l'utilisateur de déplacer une tâche d'une colonne à une autre sur le bureau.
- Mise à jour en temps réel de l'état de la tâche.

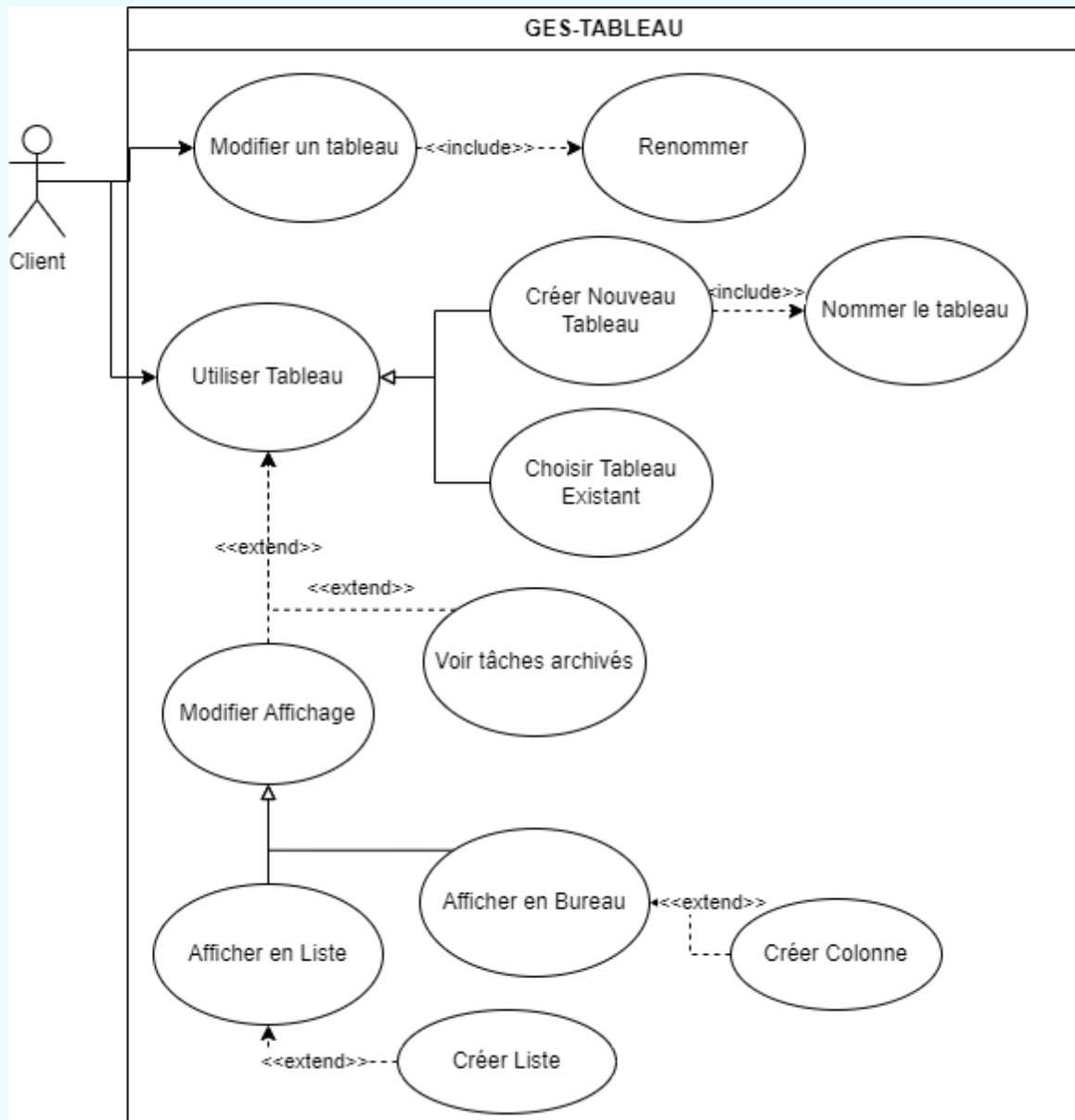
Visualisation de la liste des tâches :

- Affichage en liste hiérarchique des tâches avec la possibilité de déployer les sous-tâches.
- Vue détaillée de toutes les tâches et de leurs relations.

Génération du diagramme de Gantt :

- Génération d'un diagramme de Gantt pour visualiser la chronologie des tâches sélectionnées.
- Représentation des dépendances entre les tâches.

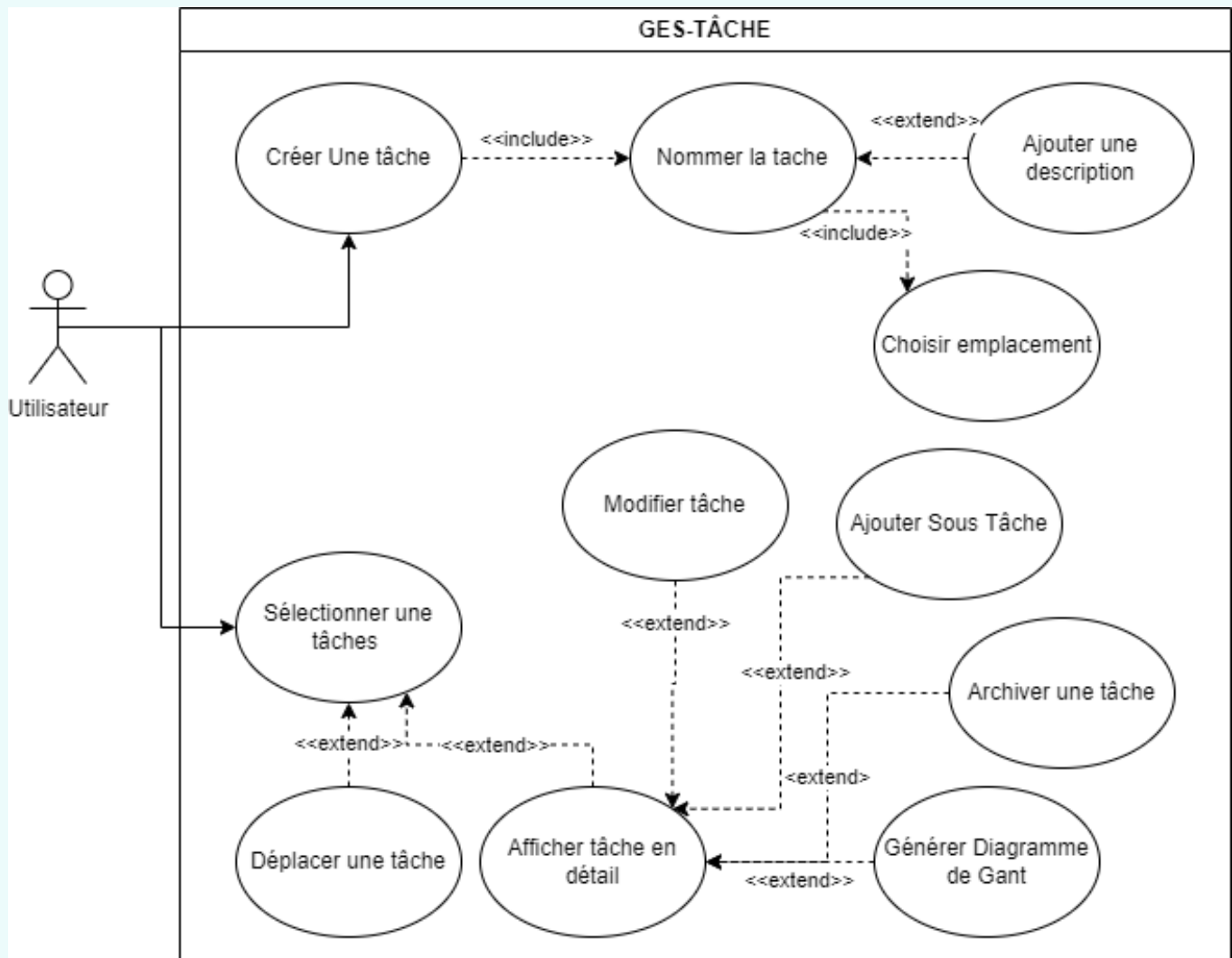
Diagramme de cas d'utilisation de la gestion du tableau



Explication

Un tableau agit comme un conteneur principal, fournissant une disposition en bureau ou en listes pour catégoriser les différentes tâches à accomplir. Pour utiliser un tableau, l'utilisateur doit soit le créer ou en choisir un existant. L'utilisateur peut modifier le tableau en le renommant.

Diagramme de cas d'utilisation de la gestion des tâches



Explication

Lors de la création d'une tâche, l'utilisateur doit lui attribuer un nom, choisir son emplacement dans le tableau, et ajouter une description si voulu. L'utilisateur peut interagir avec une tâche de différentes manières. Lorsqu'une tâche est sélectionnée, deux principales options s'offrent à l'utilisateur : la déplacer ou afficher les détails complets de la tâche. L'affichage détaillé offre un accès direct à des fonctionnalités avancées. Il permet de modifier les informations de la tâche, de l'archiver, d'ajouter des sous-tâches, ou encore de générer un diagramme de Gantt.

Description textuelle (archivage d'une tâche)

Précondition :

L'utilisateur doit être connecté pour pouvoir accéder au tableau.

Postcondition :

La tâche ne se trouve plus dans aucune colonne ou liste, et se trouve dans la section des tâches archivés.

Déroulement normal :

(1) L'utilisateur sélectionne la tâche qu'il souhaite. (2) L'utilisateur clique sur le bouton « Archiver ». (3) Le système affiche un message d'alerte pour demander à l'utilisateur de confirmer son choix. (4) L'utilisateur appuie sur le bouton « Oui » pour confirmer. (5) Le système archive la tâche.

Variantes :

(A) L'utilisateur n'a pas les droits d'archiver une tâche. (1) L'application ne lui affiche pas le bouton pour archiver.

(B) L'utilisateur appuie sur le bouton de « non » et n'archive pas la tâche.

Contraintes non fonctionnelles :

(A) Performance : l'application doit réagir dans un délai inférieur à 4 secondes quel que soit l'action de l'utilisateur.

(B) Sécurité des données : le Système doit garantir la sécurité des données de l'utilisateur.

Explication

Nous avons fait une description textuelle pour l'archivage d'une tâche afin d'assurer la clarté et la cohérence de cette fonctionnalité centrale au sein de notre application.

Description textuelle (création d'une tâche)

Précondition :

L'utilisateur est connecté à l'application et doit travailler sur un tableau.

Postcondition :

Si la tâche a pu être créée celle-ci apparaît sur le tableau dans la colonne ou la liste où l'utilisateur a voulu la créer.

Déroulement normal :

(1) L'utilisateur clique sur le bouton « Créer Tâche ». (2) Le système demande à l'utilisateur de nommer la tâche, d'éventuellement lui donner une description et l'endroit où il veut la créer. (3) L'utilisateur tape le nom de la tâche, facultativement une description et choisit l'endroit où il veut la créer. (4) Le système vérifie que les informations entrées par l'utilisateur sont valides. (5) la tâche est créée et s'affiche.

Variantes :

(A) Les données entrées par l'utilisateur sont jugées invalides par le système (2) Si les données sont jugées invalides le système affiche un message d'erreur et revient au menu principal
(B) L'utilisateur appuie sur le bouton retour pour quitter la création de tâche.

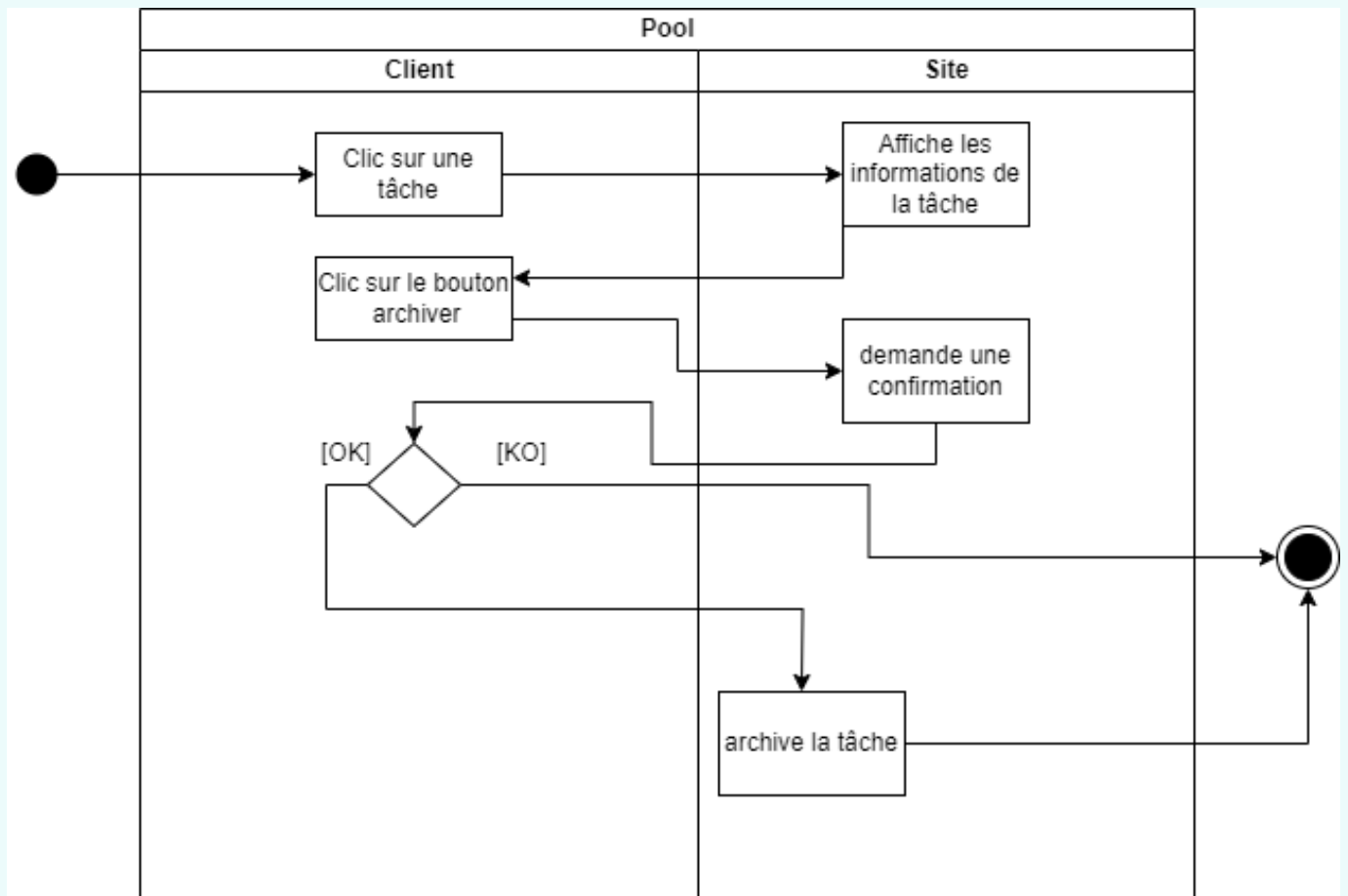
Contraintes non fonctionnelles :

(A) Le système doit réagir dans un délai inférieur à 4 secondes
(B) Sécurité des données : le Système doit garantir la sécurité des données de l'utilisateur.

Explication

Nous avons fait une description textuelle pour la création de tâches afin d'assurer la clarté et la cohérence de cette fonctionnalité centrale au sein de notre application.

Diagramme d'activité : archiver une tâche



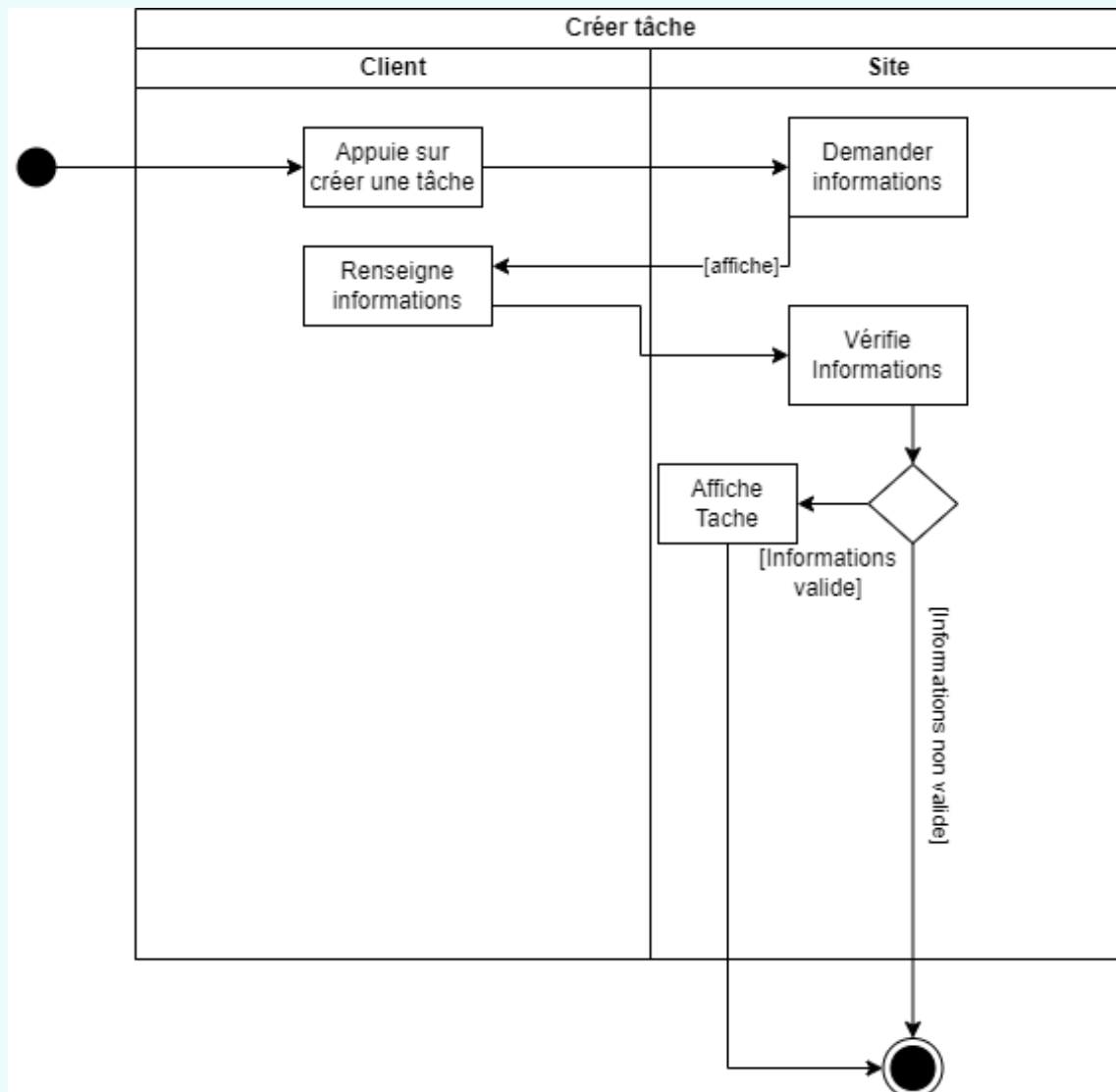
Explication

Notre diagramme d'activité pour l'archivage d'une tâche illustre de manière détaillée le cheminement du processus.

Déroulement :

1. **Déclenchement par le Client :** Le processus débute dans la colonne « client » lorsque l'utilisateur clique sur une tâche qu'il souhaite archiver.
2. **Affichage des Informations dans la Colonne Site :** Ensuite, le diagramme indique une transition vers la colonne « site » où les informations détaillées de la tâche sont affichées.
3. **Demande de Confirmation :** De retour dans la colonne « client », l'utilisateur clique sur le bouton d'archivage. Une transition est ensuite faite vers la colonne « site » où une demande de confirmation est présentée à l'utilisateur.
4. **Traitement de la Confirmation :** Une condition est introduite dans la colonne « client », qui détermine la suite en fonction de la réponse de l'utilisateur. Si la réponse est « OK » (pour « Oui »), le processus passe à la colonne « site » où la tâche est effectivement archivée, et le processus se termine.
5. **Fin du Processus :** Si la réponse est « KO » (pour « Non »), le processus se termine immédiatement sans archiver la tâche, respectant ainsi le choix de l'utilisateur.

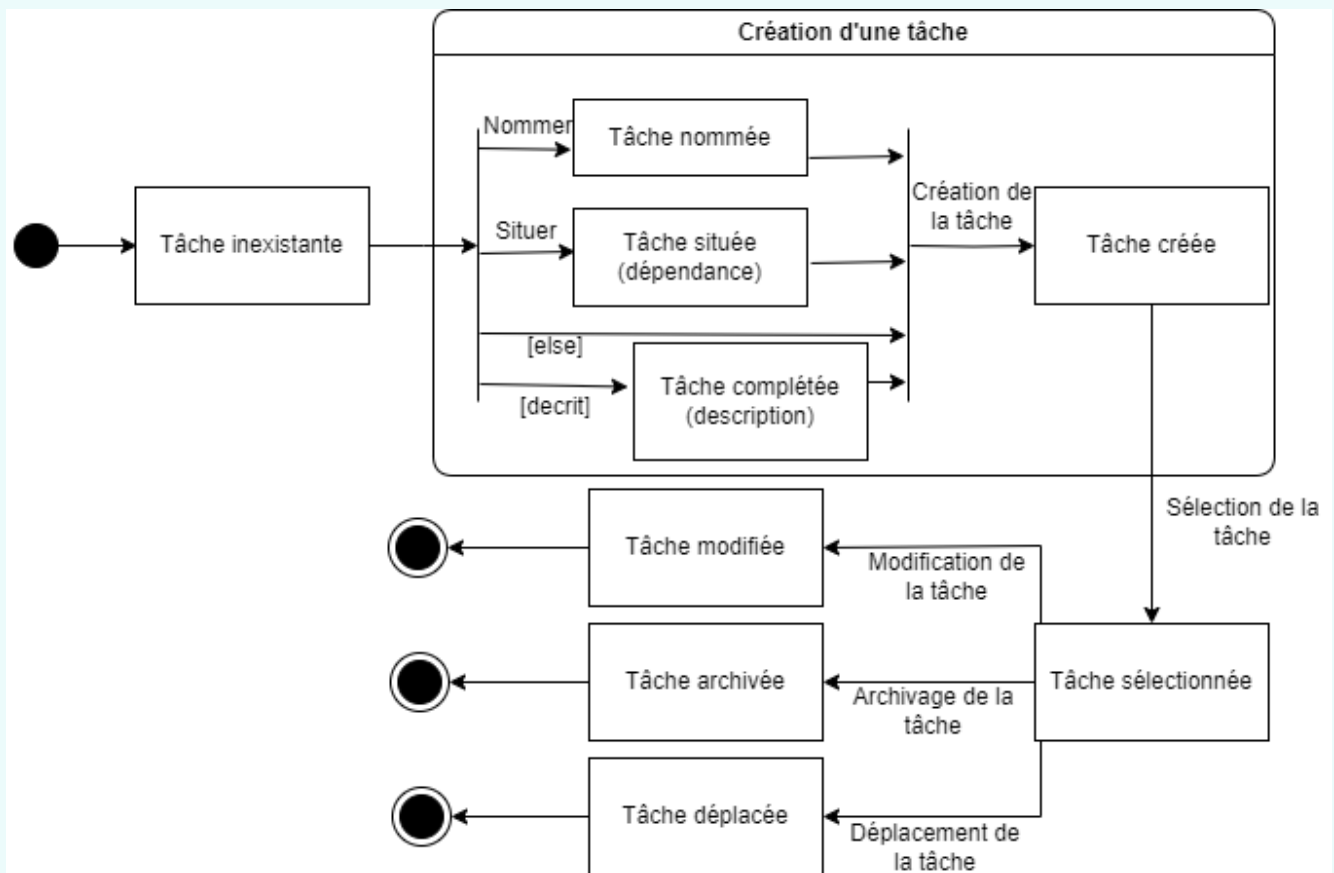
Diagramme d'activité : créer une tâche



Explication

Quand l'utilisateur créer un tâche il doit appuyer sur le bouton « créer une tâche ». Après cela le système va lui demander de renseigner les informations pour la tâche et si les informations sont valide alors le système affiche la tâche et c'est la fin du processus. Si les informations ne sont pas valide alors la tâche n'est pas créée et c'est la fin du processus.

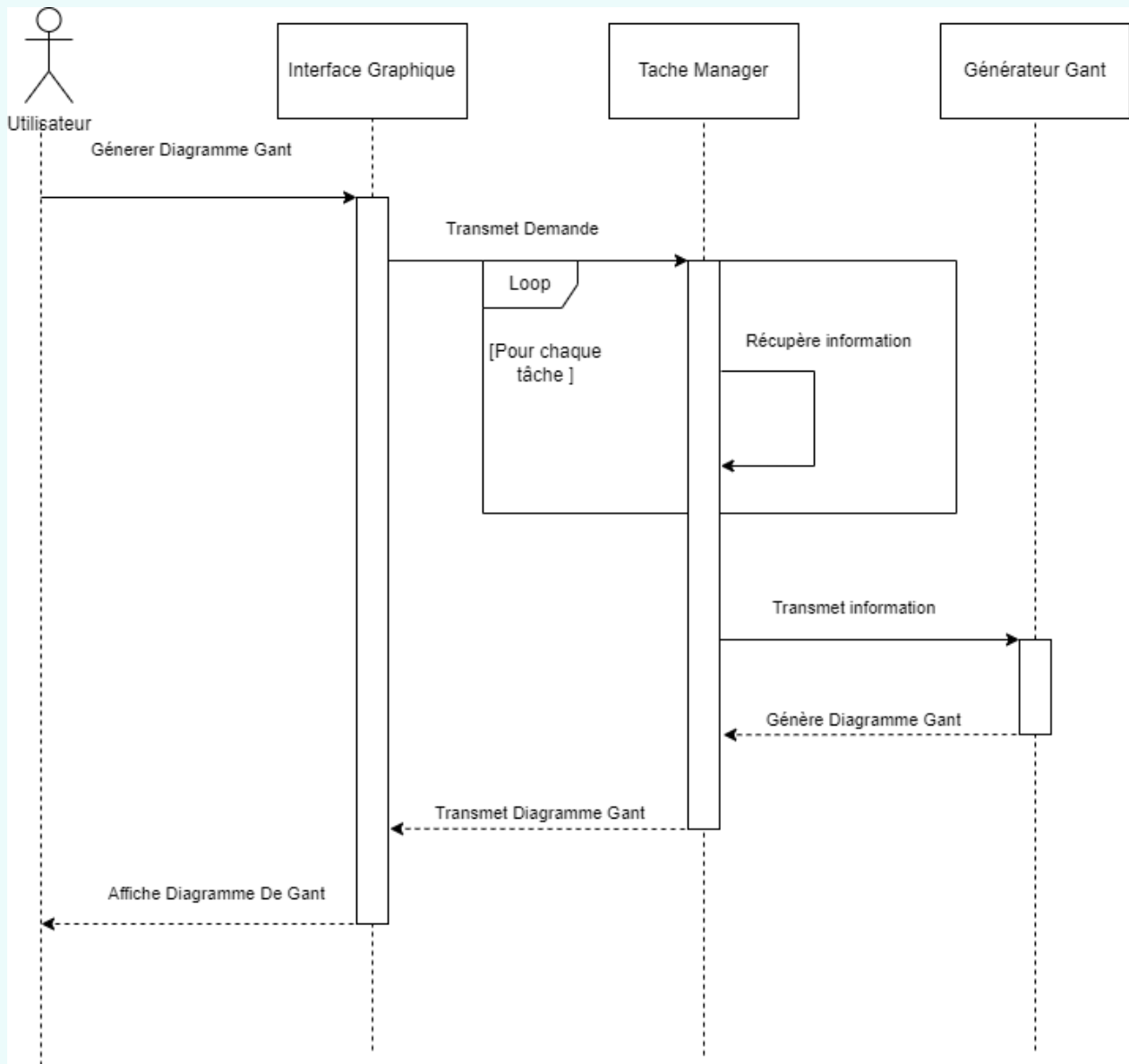
Diagramme d'état : gérer une tâche



Explication

Pour créer une tâche il faut dans un premier temps qu'elle soit inexistante. Puis ensuite en parallèle il faut nommer la tâche, la située (dans qu'elle colonne) et faire une description (optionnel). Une fois la tâche créée on peut la sélectionner pour la modifier, l'archivée ou la déplacer.

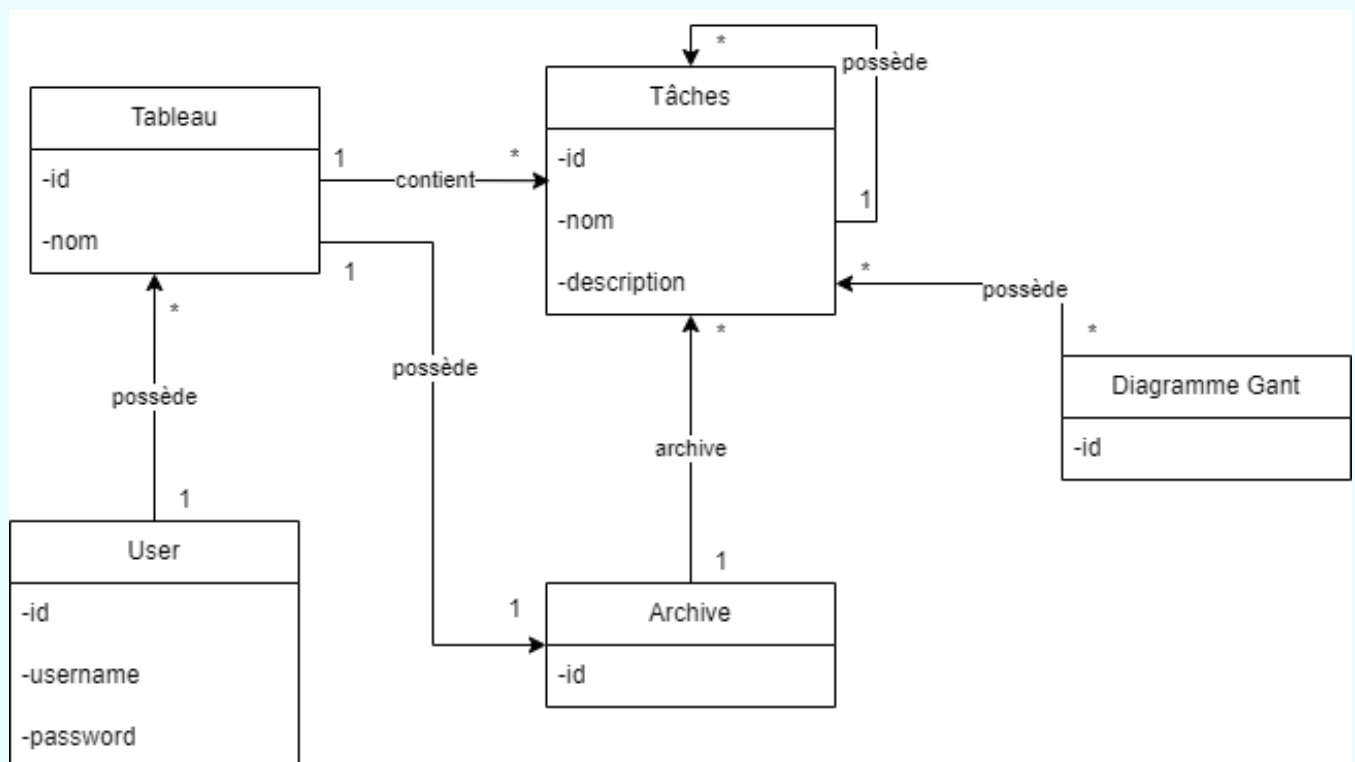
Diagramme de séquence système : générer un diagramme de Gantt



Explication

Pour générer le diagramme de gantt, l'utilisateur appuie sur le bouton qui génère la diagramme, ensuite l'interface graphique va transmettre la demande au manager des tâche qui lui va répéter pour chaque tâches la récupération des informations des tâches, puis ensuite il transmet les informations au générateur de gantt qui lui va générer le diagramme au tache manager qui lui va le transmettre à l'interface graphique qui lui va l'afficher à l'utilisateur.

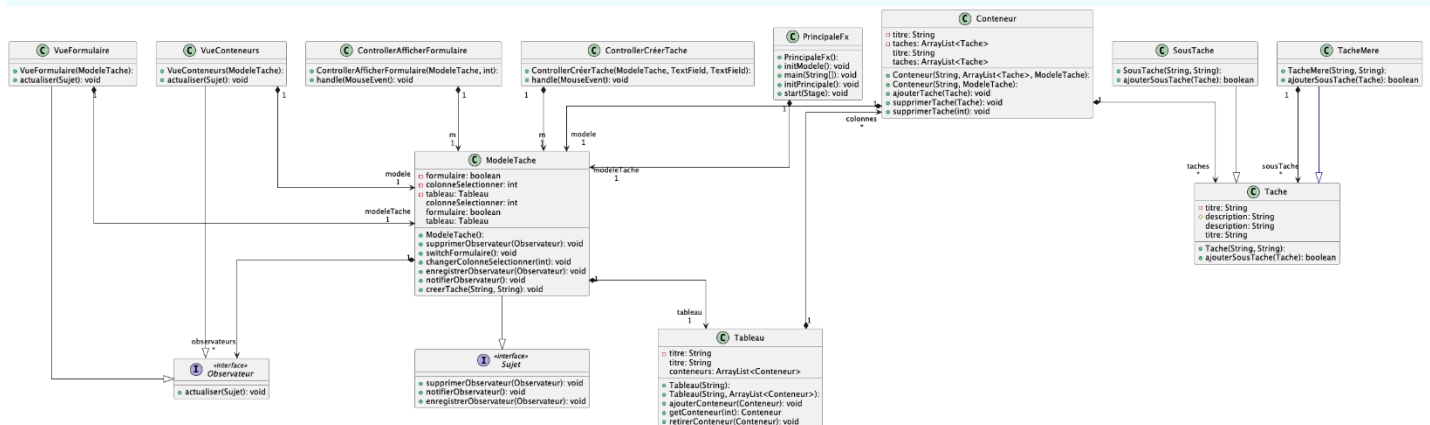
Diagramme de classe de l'analyse



Explication

Un Utilisateur possède un ou plusieurs tableaux qui eux même contient des tâches et des archives. Les archives possède des tâches archivées. Les tâches possède aussi d'autre tâches pour faire des sous-tâches et pour finir un diagramme de Gantt possède les tâches qui lui sont utile pour générer son diagramme.

Diagramme de classe de la première itération



Explication

Pour la première itération, nous utilisons le patron d'architecture MVC, le patron de conception Observateur et Composite.

L'objectif est d'avoir un affichage de base avec des colonnes précréé et de pouvoir ajouter des tâches.

1. **Patron Composite :**

- **Objectif :** Gérer des structures d'objets de manière hiérarchique. Utilisé pour représenter des tâches, où une tâche peut être soit une tâche simple, soit une tâche composée de plusieurs sous-tâches.
- **Exemple dans le code :** La classe abstraite "Tâche" et ses sous-classes représentent la structure hiérarchique des tâches.

2. **Patron MVC (Modèle-Vue-Contrôleur) :**

- **Objectif :** Séparer les préoccupations liées aux données, à l'affichage et à la logique de contrôle. Utilisé pour organiser la structure de votre application en trois composants distincts.
- **Exemple dans le code :**
 - **Modèle :** Représente les données et la logique métier.
 - **Vue :** Gère l'affichage des données et interagit avec l'utilisateur.
 - **Contrôleur :** Gère les actions de l'utilisateur et met à jour le modèle et la vue en conséquence.

3. **Patron Observateur (Observer) :**

- **Objectif :** Établir une dépendance d'un-à-plusieurs entre objets, de sorte que lorsque l'état d'un objet change, tous ses dépendants sont notifiés et mis à jour automatiquement. Utilisé pour permettre aux vues d'être informées des changements dans le modèle sans connaître les détails de mise à jour.
- **Exemple dans le code :** Le modèle possède une liste d'observateurs, et ceux-ci sont notifiés lorsque des changements, tels que la création d'une nouvelle tâche, surviennent.

Planning des Itérations

Itération 1 :

- Créer tâche dans colonne
- Afficher Tâche dans colonne
- Squelette MVC

Objectifs :

Pour cette itération, l'objectif est de créer une base fonctionnelle qui servira de socle pour les prochaines itérations.

A la fin de cette itération, on pourra ajouter des tâches à des colonnes préconstruites.

Itération 2 :

- Créer sous tâche
- Afficher Détails tâche (sous tâche, description)
- Modifier tâche

Objectifs :

Ici, on se concentre sur les fonctionnalités liées aux tâches car c'est l'élément central de l'application.

A la fin de cette itération, on pourra créer des tâches dépendantes, voir leurs détails et les modifier.

Itération 3 :

- Affichage liste
- Affichage bureau
- Patron Stratégie

Objectifs :

Lors de cette itération, nous nous concentrons sur l'affichage des tâches. Soit sous forme de bureau (colonnes) ou de listes. Nous utiliserons le patron stratégie pour implémenter ces fonctionnalités.

A la fin de cette itération, on pourra changer l'affichage.

Itération 4 :

- Créer tableau
- Changer Tableau
- Permettre d'ajouter des colonnes

Objectifs :

Pour cette itération, nous ajouterons la possibilité d'avoir de nouveau tableau et le fait d'ajouter de nouvelles colonnes ou listes. Il n'y aura plus de limite.

A la fin de cette itération, nous pourrons changer de tableau, et ajouter autant de colonnes ou listes souhaité.

Itération 5 :

- Archiver
- Archiver tâche
- Enlever tâche archive

Objectifs :

Ici, on se concentre sur l'archivage. Consulter l'archive, ajouter des tâches à l'archivage et les enlever.

Itération 6 :

- Déplacer tâche
- Générer Diagramme Gantt

Objectifs :

Pour cette dernière itération, nous ajouterons le DragAndDrop pour déplacer les tâches entre les tableaux/listes et nous pourrons générer un diagramme de Gantt.

Maquette Balsamiq

New Tableau

Choix Tableau

Visualiser sous forme de bureau

Col1

Tâche One

Tâche Two

Tâche Three

Tâche Four

Col2

A Web Page

SAE301/Tableau/

Voir tâches archivées

Créer Colonne

Créer Tâche

Tâche

Création Tâches

Nom de la tâche :

NewTâche

Description :

Je suis une nouvelle Tâche

Créer Tâche

Retour

New Tableau

Choix Tableau

Visualiser sous forme de bureau

Visualiser sous forme de liste

Col1

Tâche One

Tâche Two

Tâche Three

Tâche Four

NewTâche

Col2

A Web Page
SAE301/Tableau/

Voir tâches archivées

Créer Colonne

Créer Tâche

NewTâche

Description :

Je suis une nouvelle Tâche

Sous Tâche :

Retour

Générer Gantt

Archiver

Modifier

Ajouter Sous Tâche

Ceci conclut cette étude préalable. Nous sommes impatients de concrétiser ces concepts dans la phase suivante de la SAE.