

# **Soutenance : Partie analyse**

SAE S3.01 : Développement d'une application JAVA



# Sommaire

## 1 Présentation du projet

- Objectif du projet
- En quoi le projet consiste-t-il
- Notre démarche d'analyse
- Planning

## 2 Fonctionnement de l'application

- Liste des uses cases
- Analyse de l'état d'une tâche
- Maquettage de l'application

## 3 Travail préliminaire à la programmation

- Diagramme de classe
- Description des patrons utilisés

# Présentation du projet

# L'objectif du projet

## Application

Faciliter l'organisation des tâches personnelles

## Dépendances

Gestion de dépendances entre tâches.

## Diagramme de Gantt

Permettre une visualisation claire de la progression

## Interface Utilisateur

Visualisation intuitive de l'avancement grâce à une interface simple et ergonomique.

## Reflexion personnelle

Possibilité d'ajouter des fonctionnalités imaginatives non demandées.

# En quoi consiste le projet

## Utilité

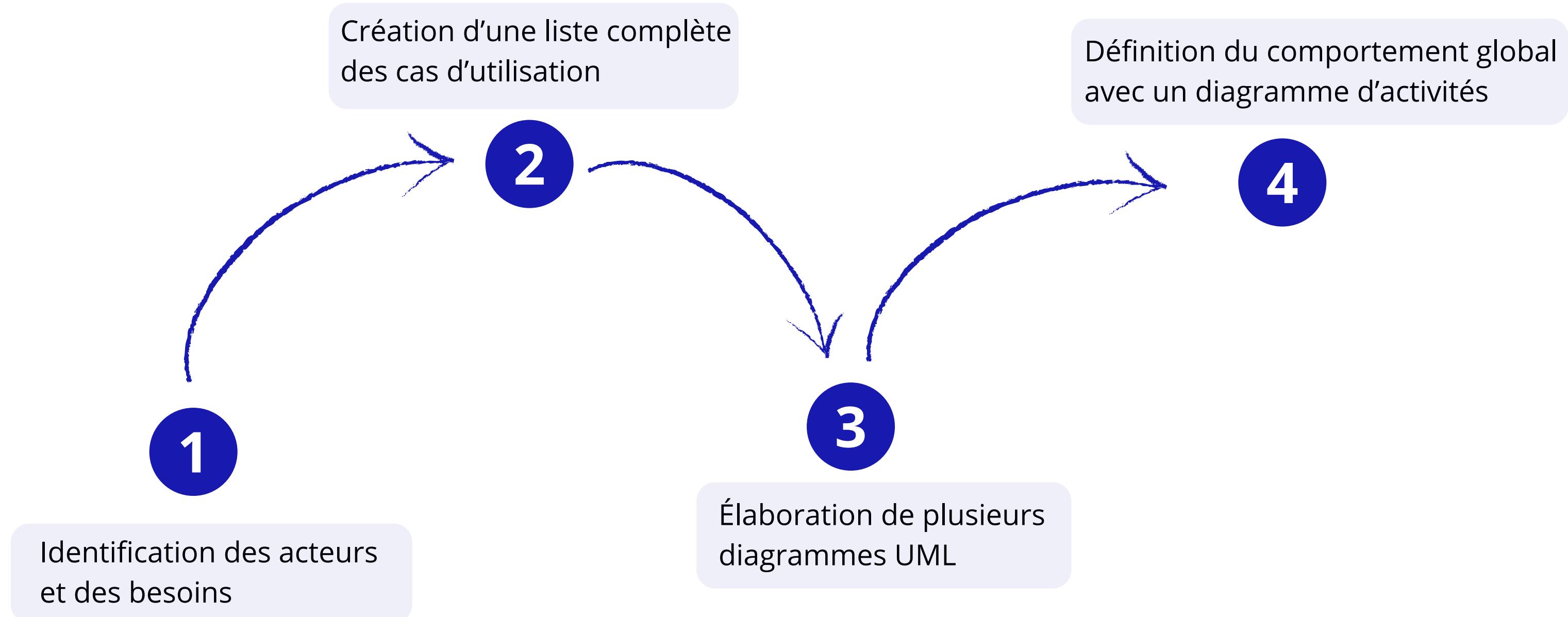
- Création, modification des tâches
- Archiver les tâches terminées
- Gérer les sous-tâches et les dépendances logiques
- Générer un diagramme de Gantt

## Interface basée sur JavaFX

Sur la base du patron d'architecture Model View Controller

- Organisation par colonne
- Glisser déposer les tâches
- Vue en liste hiérarchique
- Historique des modifications
- Affichage des details d'une tâche

# Notre démarche d'analyse



# Planning des itérations

Numéro de l'itération	Objectifs	Risques
Itération 1	<ul style="list-style-type: none"> <li>Création d'une base de données</li> <li>Mettre en place le modèle du patron MVC</li> <li>Implémenter la création de tâche (données minimales : titre, description, date, ...)</li> <li>Faire les premiers tests</li> </ul>	<ul style="list-style-type: none"> <li>S'assurer d'un bon squelette de bd</li> <li>S'assurer du bon modèle du patron MVC</li> <li>S'assurer des bons attributs</li> <li>S'assurer de la bonne synchronisation entre les membres du groupe</li> </ul>
Itération 2	<ul style="list-style-type: none"> <li>Implémenter la gestion des dépendances</li> <li>Mettre à jour les diagrammes</li> <li>Mettre en place la validation des formulaires (création/modification)</li> <li>Faire les tests unitaires sur la logique de dépendance</li> </ul>	<ul style="list-style-type: none"> <li>Détection de cycle mal gérée</li> <li>Modèle de donnée insuffisant pour représenter les dépendances</li> </ul>
Itération 3	<ul style="list-style-type: none"> <li>Implémenter la gestion des archivages de tâche</li> <li>Implémenter l'affichage filtrée</li> <li>Développer la vue "liste" avec développement des sous tâches</li> <li>Amélioration de l'ergonomie (séparation claire liste/détails)</li> <li>Tests sur les filtres et l'affichage dynamique</li> </ul>	<ul style="list-style-type: none"> <li>Risque de confusion entre archivé et terminé</li> <li>Complexité de l'affichage</li> <li>Risques de dépendances défectueuse</li> </ul>

Numéro de l'itération	Objectifs	Risques
Itération 4	<ul style="list-style-type: none"> <li>Implémenter la vue "Bureau" avec colonnes (a faire / en cours / terminé)</li> <li>Gérer le drag &amp; drop des tâches entre colonnes</li> <li>Mettre à jour en direct l'ordre des tâches dans une colonne</li> <li>Tests sur les interactions graphiques</li> <li>Ajout de la gestion de l'historique</li> </ul>	<ul style="list-style-type: none"> <li>Gestion compliquée entre modèle et interface graphique</li> <li>Mauvaise synchronisation entre le drag et le drop</li> <li>Bug graphique du drag &amp; drop</li> </ul>
Itération 5	<ul style="list-style-type: none"> <li>Sélection des tâches pour le diagramme de Gantt</li> <li>Génération du diagramme (positionnement, durées, dépendances visuelles)</li> <li>Ajout des controllers pour la finalisation du patron MVC</li> <li>Finalisation des scénarios d'utilisation complets</li> <li>Tests d'intégration</li> </ul>	<ul style="list-style-type: none"> <li>Le rendu graphique du Gantt peut être difficile à stabiliser</li> <li>Problèmes d'échelle temporelle ou d'alignement des barres</li> <li>Performances si beaucoup de tâches sont chargées</li> </ul>
Itération 6	<ul style="list-style-type: none"> <li>Intégration finale et validation globale</li> <li>Correction des bugs restants + nettoyage du code</li> <li>Rédaction du dossier final</li> <li>Préparation de la soutenance</li> <li>Ajout possible de fonctionnalités bonus (si tout est stable)</li> </ul>	<ul style="list-style-type: none"> <li>Manque de temps si les itérations précédentes ont pris du retard</li> <li>Risque d'oublier des tests ou des cas limites</li> <li>Difficulté à stabiliser les dernières fonctionnalités complexes (drag &amp; drop, Gantt)</li> </ul>

# **Fonctionnement de l'application**

# Lister les uses cases

- **Gestion des tâches :**
  - Créer une tâche : permet à l'utilisateur d'ajouter une nouvelle tâche en renseignant ses informations principales, qui seront ensuite enregistrées par le système
  - Ajouter une sous-tâche : offre la possibilité d'associer une sous-tâche à une tâche existante afin de structurer le travail de manière hiérarchique
  - Ajouter une dépendance entre 2 tâches : permet de définir un lien de dépendance entre deux tâches, le système vérifiant automatiquement l'absence de cycle
  - Modifier une tâche : autorise l'utilisateur à mettre à jour les informations d'une tâche existante, avec validation du formulaire par le système
  - Supprimer une tâche : permet à l'utilisateur de retirer définitivement une tâche, le système mettant à jour les données en conséquence
  - Archiver une tâche : offre la possibilité de masquer une tâche sans la supprimer, en la plaçant dans l'archive tout en conservant ses données
- **Organisation visuelle :**
  - Afficher les tâches dans la vue "Bureau" : le système affiche l'ensemble des tâches sous forme de colonnes visuelles afin de permettre à l'utilisateur de les organiser facilement
  - Déplacer une tâche dans une colonne (drag and drop) : l'utilisateur peut réorganiser ses tâches en les déplaçant par glisser-déposer, et le système met à jour leur position
  - Modifier l'ordre des tâches dans une colonne : permet de réorganiser les tâches d'une colonne en ajustant leur ordre d'affichage selon les besoins de l'utilisateur
  - Afficher les détails d'une tâche depuis la vue "Bureau" : l'utilisateur peut sélectionner une tâche pour consulter ses informations détaillées, que le système charge et affiche
- **Vue liste hiérarchique**
  - Afficher les tâches dans la vue "Liste" : le système affiche l'ensemble des tâches sous forme de liste hiérarchique afin de permettre une visualisation structurée
  - Déployer ou replier les sous-tâches : l'utilisateur peut ouvrir ou fermer l'affichage des sous-tâches, et le système met à jour la liste en conséquence
  - Naviguer dans les sous-tâches : permet à l'utilisateur de parcourir les différents niveaux de la hiérarchie des tâches et sous-tâches
  - Afficher les détails d'une tâche depuis la vue "Liste" : l'utilisateur peut sélectionner une tâche dans la liste pour afficher ses informations détaillées chargées par le système



## Gestion des tâches :

- Créer une tâche : permet à l'utilisateur d'ajouter une nouvelle tâche en renseignant ses informations principales
- Ajouter une sous-tâche : offre la possibilité d'associer une sous-tâche à une tâche existante

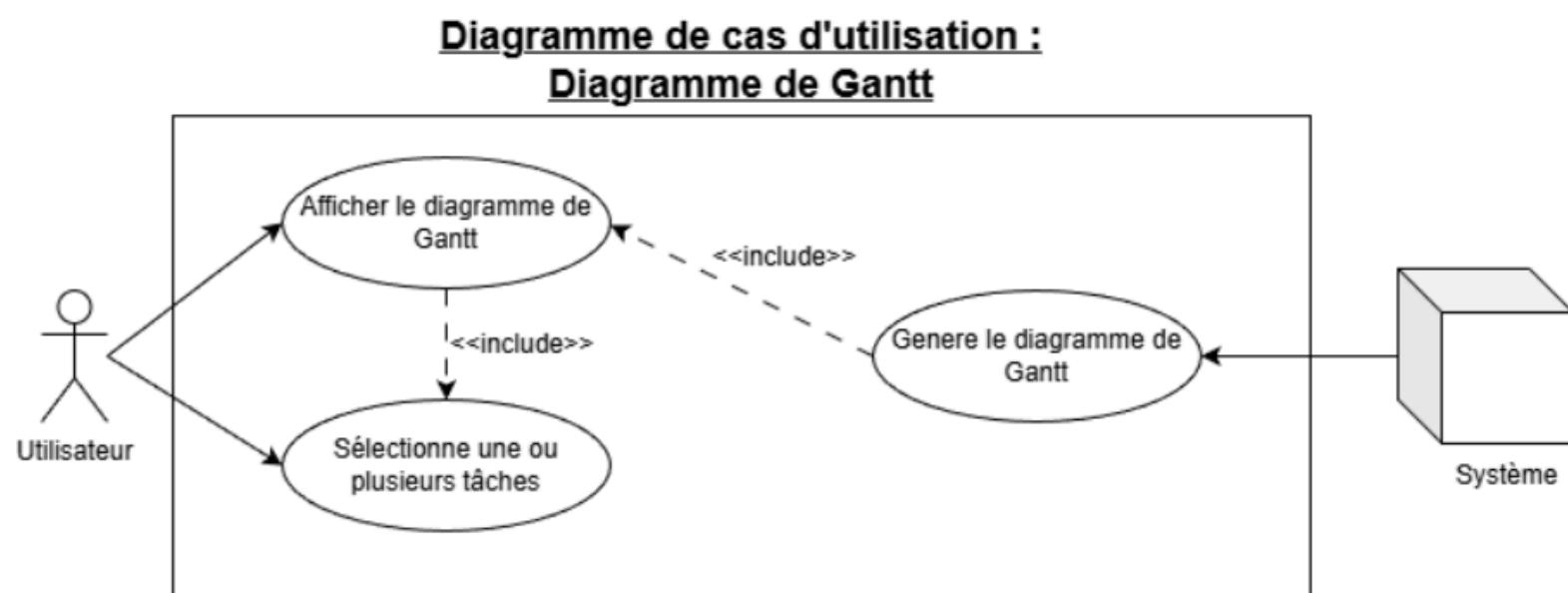


## Vue liste hiérarchique

- Naviguer dans les sous-tâches : permet à l'utilisateur de parcourir les différents niveaux de la hiérarchie des tâches
- Déployer ou replier les sous-tâches : l'utilisateur peut ouvrir ou fermer l'affichage des sous-tâches

# Quelques diagrammes

## Cas d'utilisation : Générer Gantt



### Précondition

- Le système affiche la liste des tâches du projet, l'utilisateur a la possibilité de sélectionner des tâches
- Aucune tâche sélectionnée ne doit être archivée

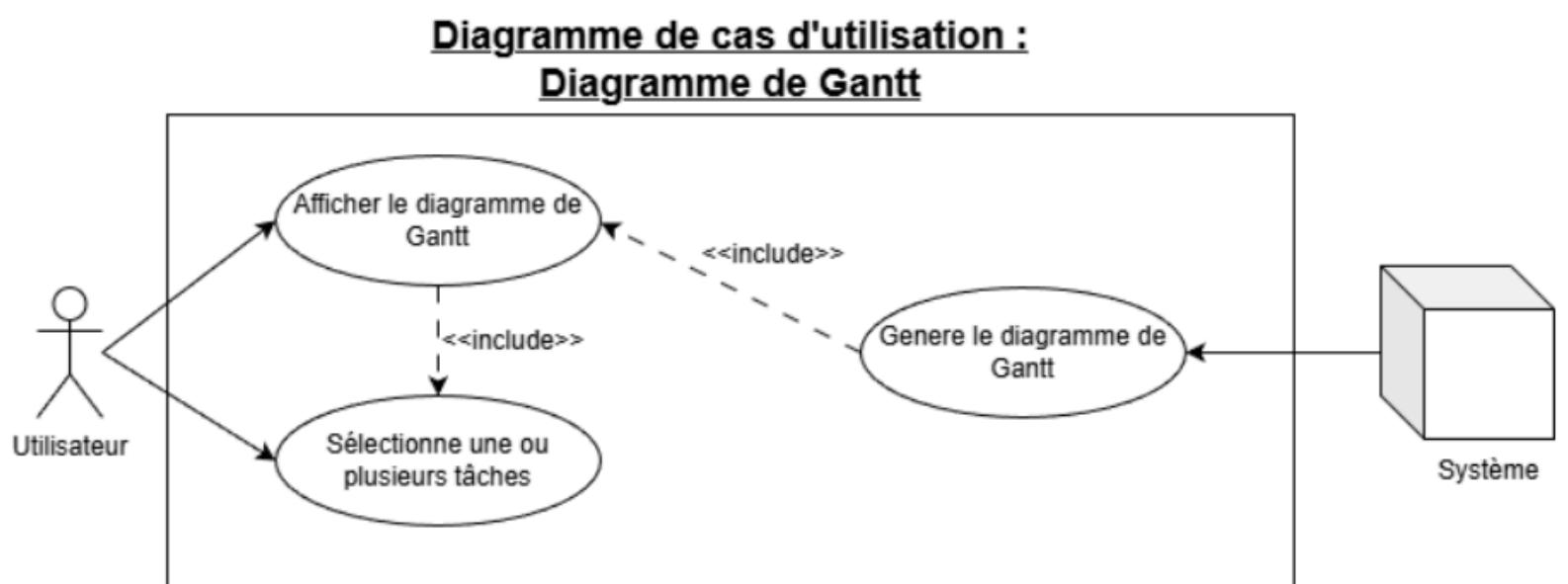
### Postcondition

- Le diagramme de Gantt est généré à partir des tâches sélectionnées et affiché à l'utilisateur

### Déroulement normal

1. L'utilisateur clique sur le bouton générer Gantt
2. Le système affiche la liste des tâches disponibles pour la génération du Gantt
3. L'utilisateur sélectionne une ou plusieurs tâches à inclure dans le diagramme
4. L'utilisateur confirme la génération du diagramme
5. Le système récupère les informations des tâches sélectionnées (dates, durée, dépendances, ...)
6. Le système génère automatiquement le diagramme de Gantt en utilisant ces données
7. Le système affiche à l'utilisateur le diagramme de Gantt généré, sous forme visuelle

## Variantes



### (A) Aucune tâche sélectionnée

À l'étape 3 ou 4, si aucune tâche n'a été choisie :

- Le système affiche un message d'erreur indiquant qu'au moins une tâche doit être sélectionnée
- Le scénario reprend à l'étape 3

### (B) Tâche invalide pour le Gantt

Si une tâche sélectionnée ne possède pas suffisamment d'informations (ex : dates manquantes) :

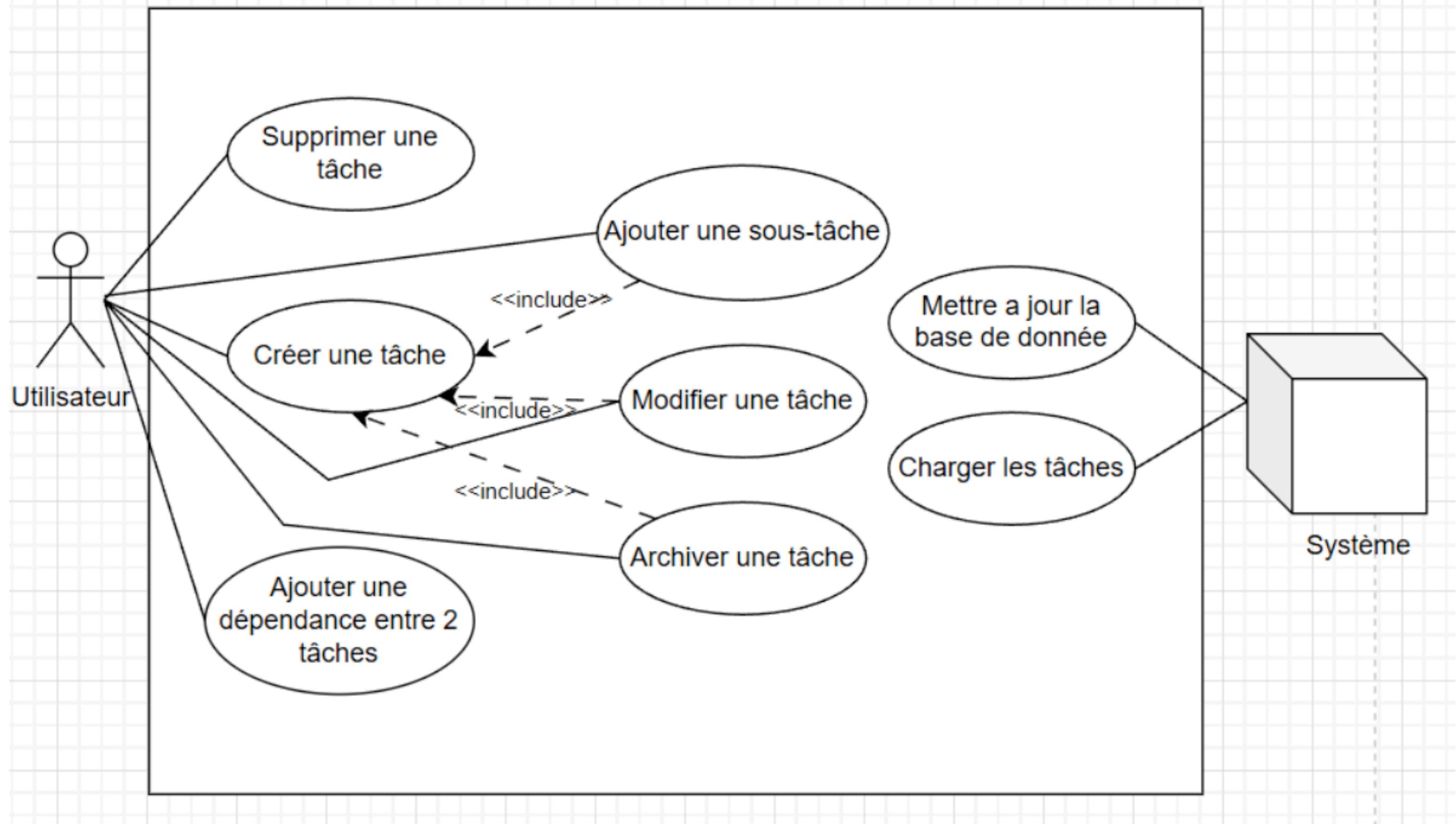
- Le système informe l'utilisateur du problème
- La génération est annulée tant que la tâche n'est pas complétée ou dé-sélectionnée

### (C) Erreur lors de la génération

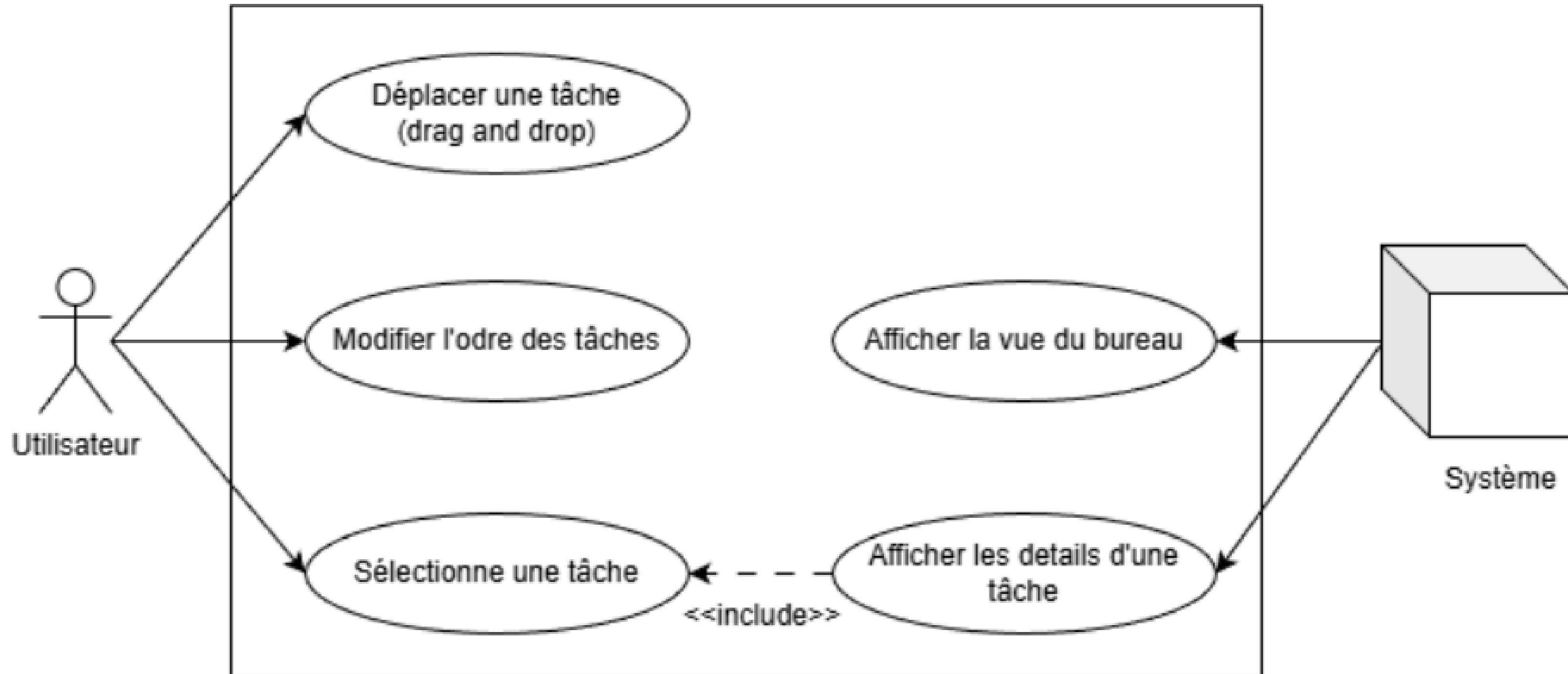
Si le système rencontre une impossibilité technique (données incohérentes, dépendances impossibles) :

- Le système signale l'erreur
- Le diagramme n'est pas généré et l'utilisateur doit corriger les données

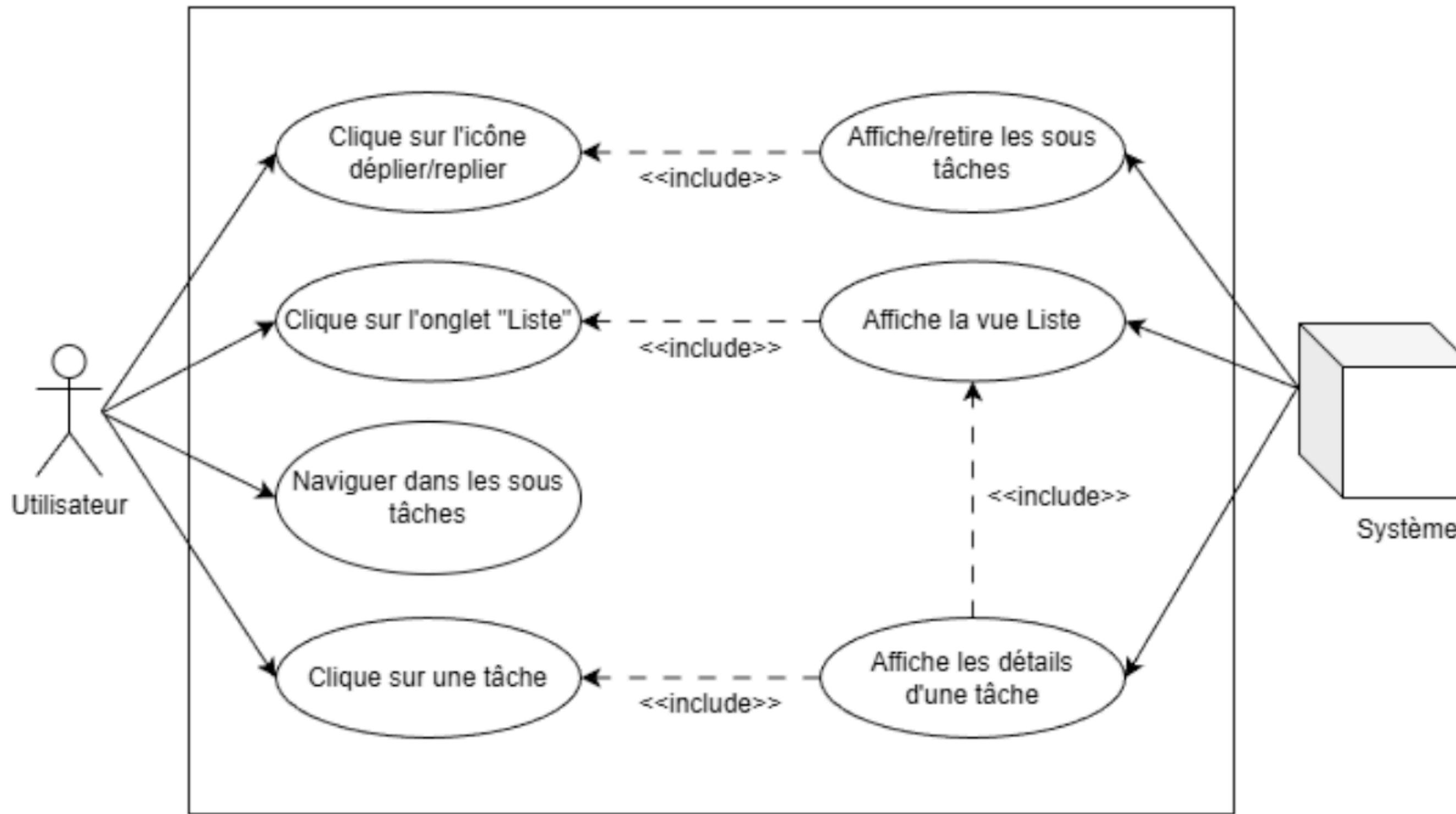
## Gestion des tâches



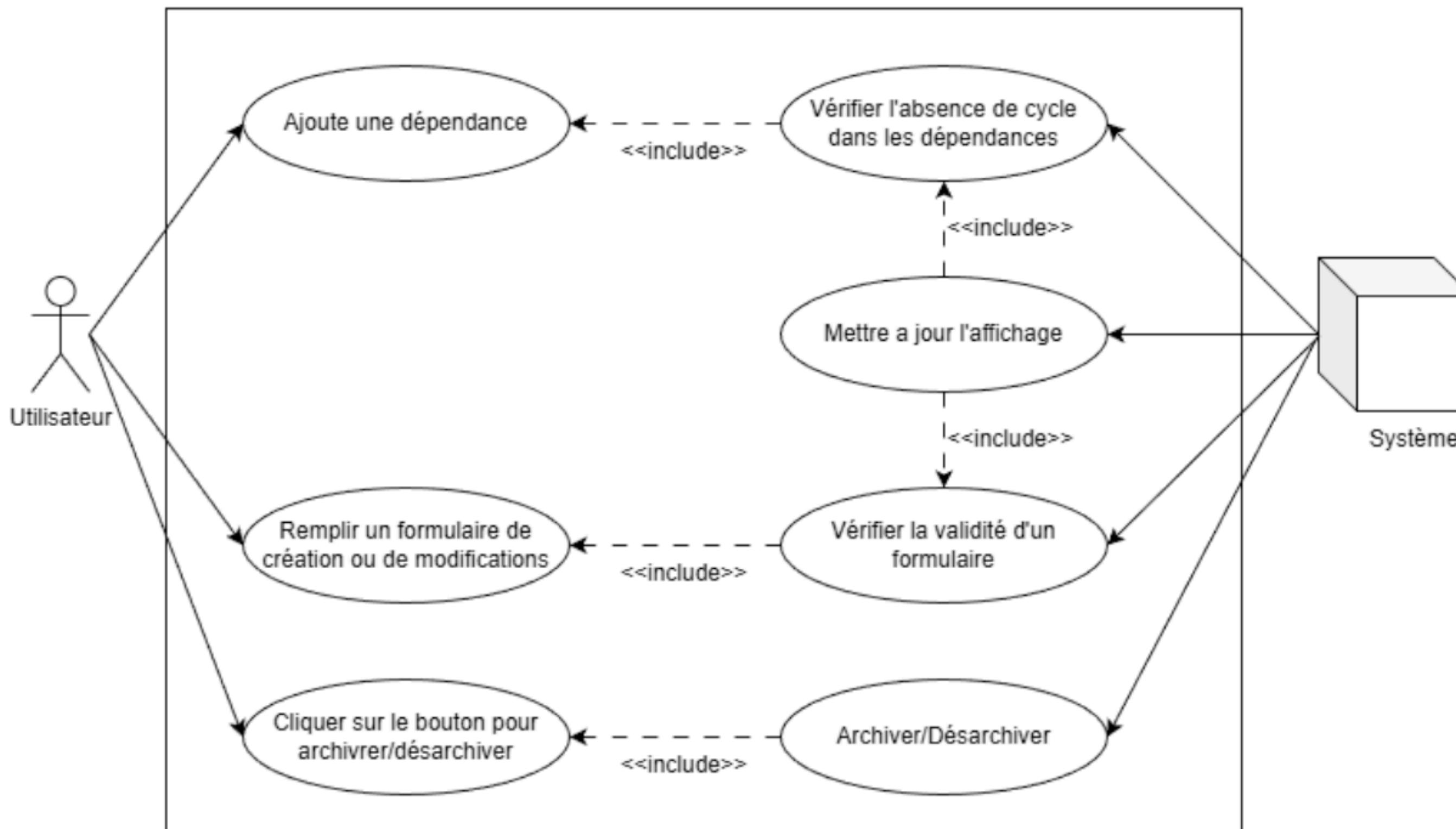
## Diagramme du cas d'utilisation : Organisation visuelle



## Diagramme de cas d'utilisation : Vue liste hiérarchique

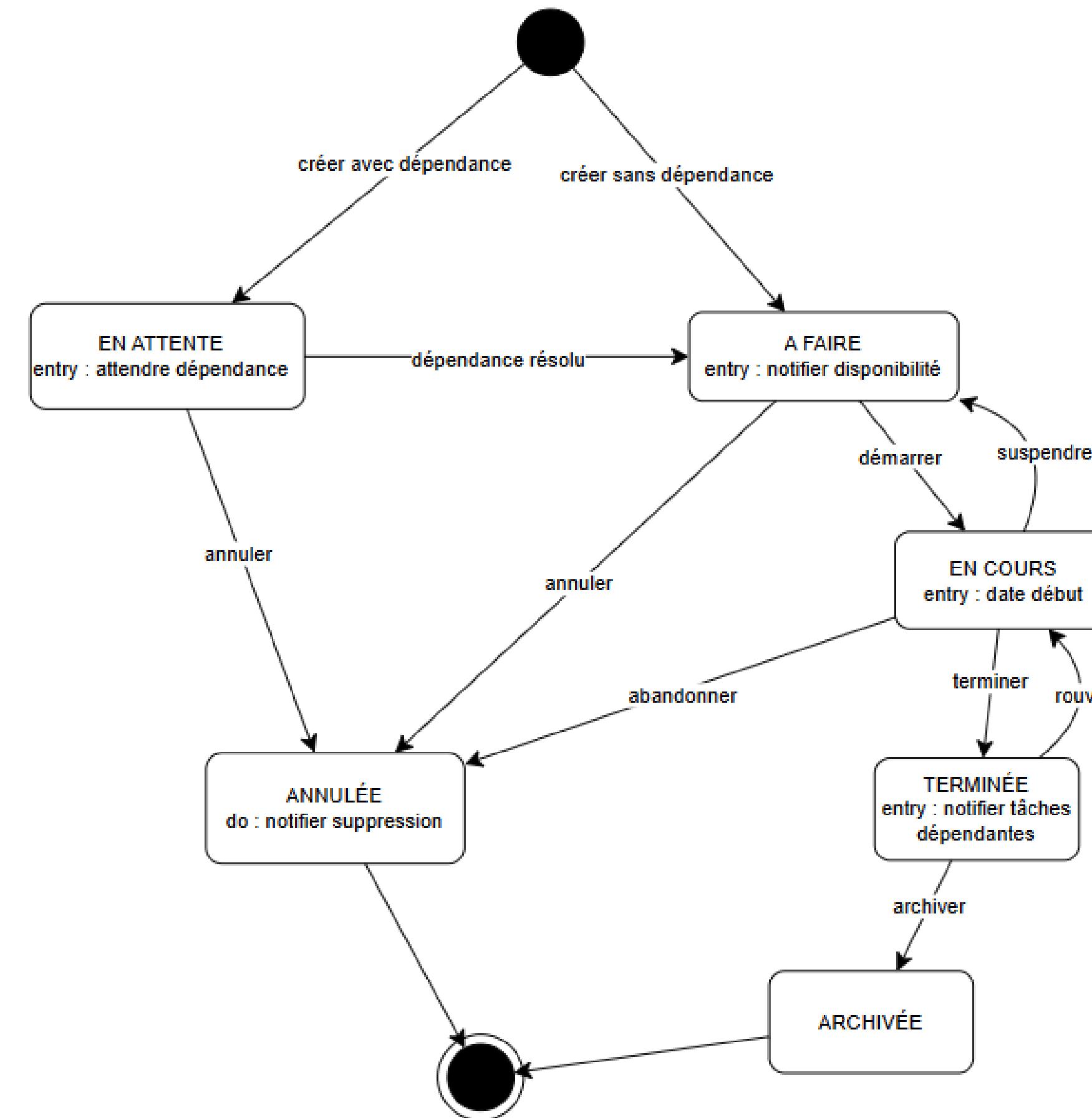


## Diagramme de cas d'utilisation : Fonctionnalités système

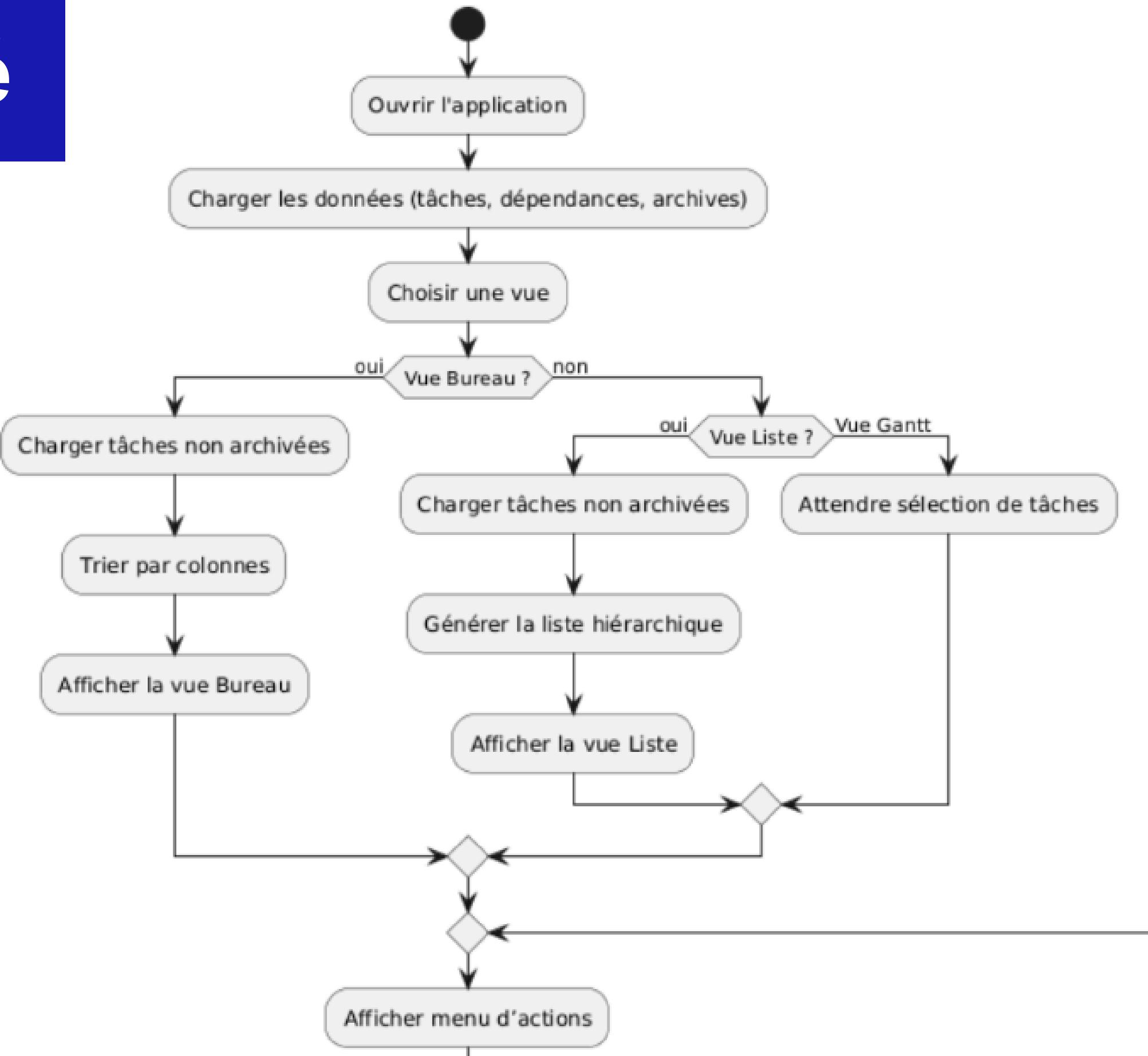


# Diagramme d'état

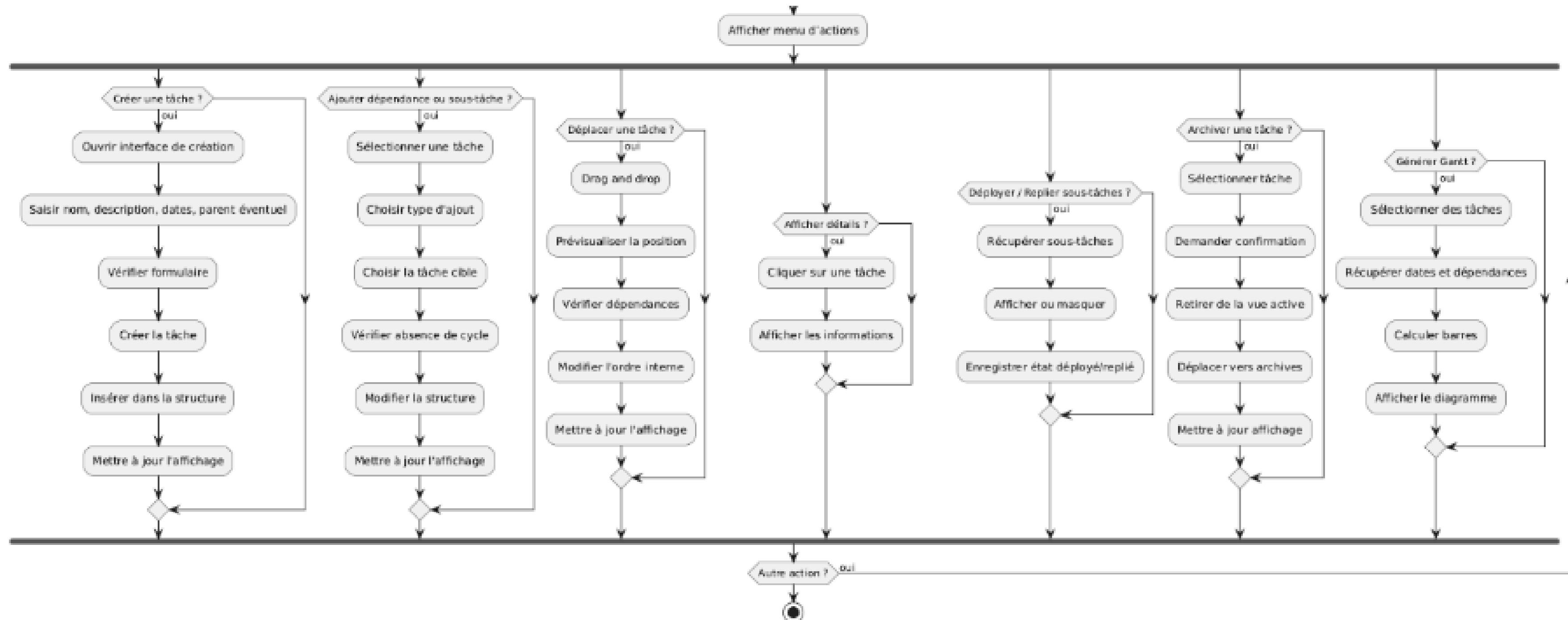
Diagramme d'état de l'objet global de l'application



# Diagramme d'activité



# Diagramme d'activité BIS



# Maquettage de l'application

## Tableau de bord

Nouvelle tâche    Supprimer tâche    Diagramme de Gantt    Tâches archivée

**Tâche 1** Archiver BACKLOG Mardi 23 Novembre

Description de la tâche nommée Tâche 1 avec le contenu du travail attendu

Dépendances :

aucune

Sous-Tâches V

- **Sous-Tâche 1**  
Description de la sous-tâche 1 avec le travail attendu
- **Sous-Tâche 2**  
Description de la sous-tâche 2 avec le travail attendu

**Tâche 3** Désarchiver TERMINÉE Lundi 29 Novembre

Description de la tâche nommée Tâche 3 avec le contenu du travail attendu

Dépendances :

Tâche 1  
Tâche 2

Sous-Tâches >

**Tâche 4** Archiver TODO Lundi 29 Novembre

Description de la tâche nommée Tâche 4 avec le contenu du travail attendu

Dépendances :

Nom de colonne

Tâche 1  
Tâche 2  
Tâche 3  
Tâche 4  
Tâche 5  
Tâche 6  
Tâche 7

Nom de colonne

Tâche 8  
Tâche 9  
Tâche 10  
Tâche 11  
Tâche 12

Nom de colonne

Tâche 13  
Tâche 14  
Tâche 15

Nom de colonne

Tâche 16  
Tâche 17  
Tâche 18  
Tâche 19

Historique :

- Déplacement de la tâche X vers la colonne [nom de colonne](#)
- Ajout d'une nouvelle tâche : NOM : [nom de la tache](#) / Description : [description de la tâche](#)

# Maquettage de l'application

## Supprimer une tâche

Supprimer une tâche

Selection de la tâche

Réaliser l'emploi du temps

Supprimer les tâches dépendantes

Dépendances

1. Composition des classes

# Maquettage de l'application

## Générer le diagramme de Gantt

Générer un diagramme de Gantt

Sélectionner toutes les tâches

Selection de la/les tâche(s)

<input type="checkbox"/> Tache 1	<input type="checkbox"/> Tache 2	<input type="checkbox"/> Tache 3
<input type="checkbox"/> Tache 4	<input type="checkbox"/> Tache 5	<input type="checkbox"/> Tache 6
<input type="checkbox"/> Tache 7	<input type="checkbox"/> Tache 8	<input type="checkbox"/> Tache 9
<input type="checkbox"/> Tache 10	<input type="checkbox"/> Tache 11	<input type="checkbox"/> Tache 12
<input type="checkbox"/> Tache 13	<input type="checkbox"/> Tache 14	<input type="checkbox"/> Tache 15

**Générer le Gantt**

# Maquettage de l'application

## Description du travail :

### L'objectif dans la réalisation des maquettes :

- visualisation de l'affichage
- décomposition des vues
- apprêhender un graphe de scène organisé et optimisé

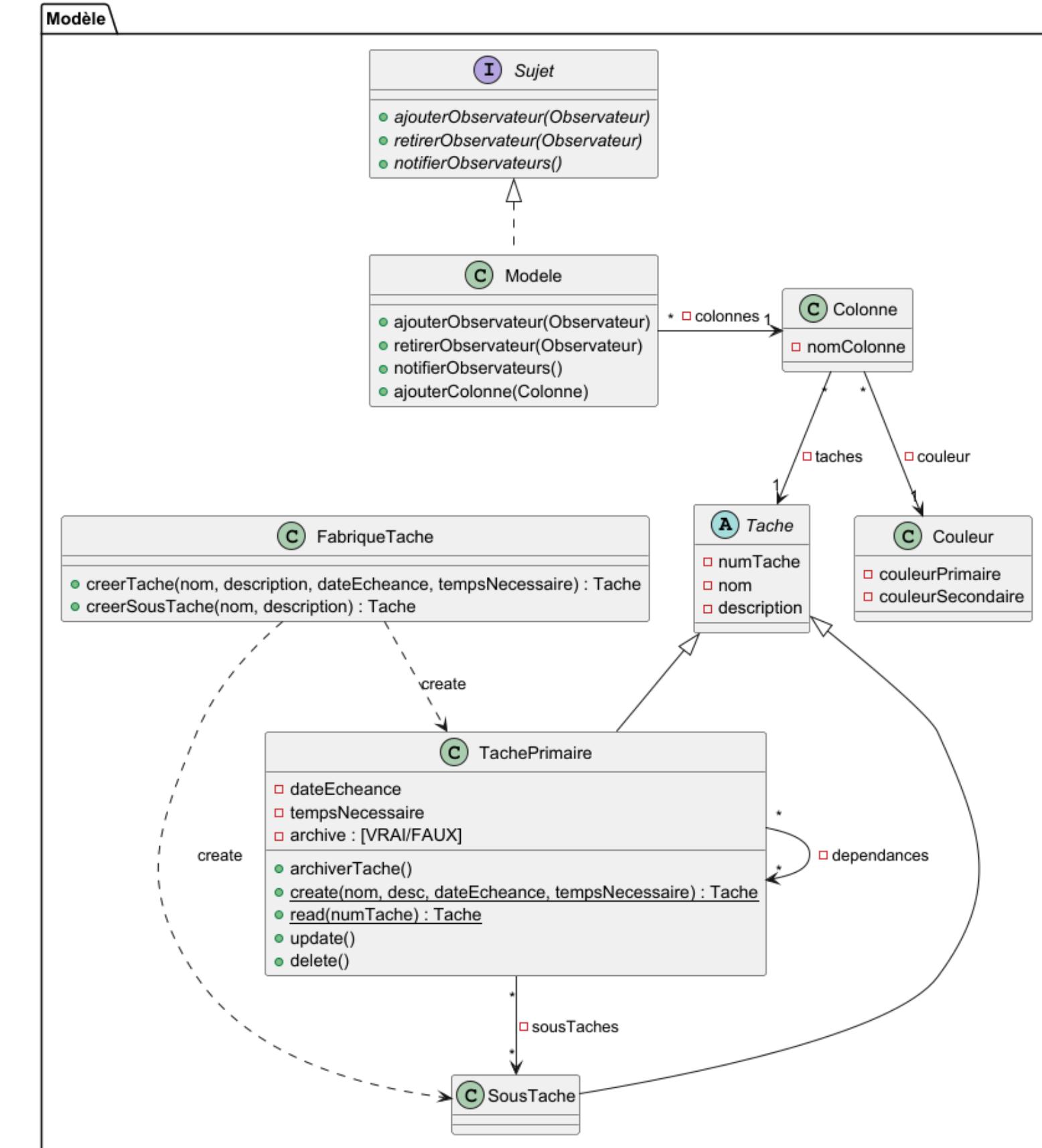
# Préparation à la programmation

## Patrons d'architecture :

- **MVC (Modèle/Vue/Contrôleur)**
- Active Record

## Patrons de conception :

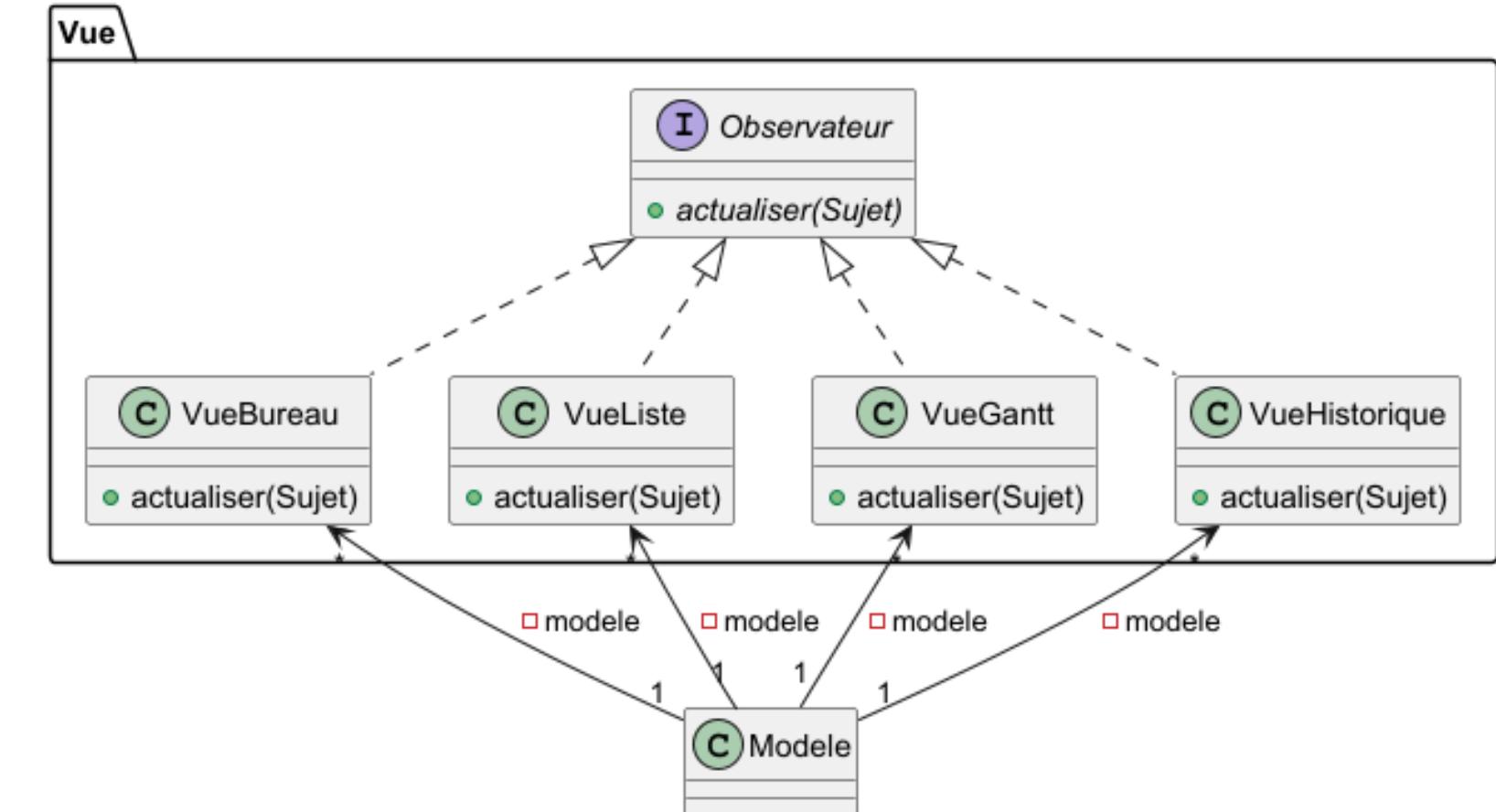
- Observateur
- **Composite**
- **Fabrique**
- Singleton



# Préparation à la programmation

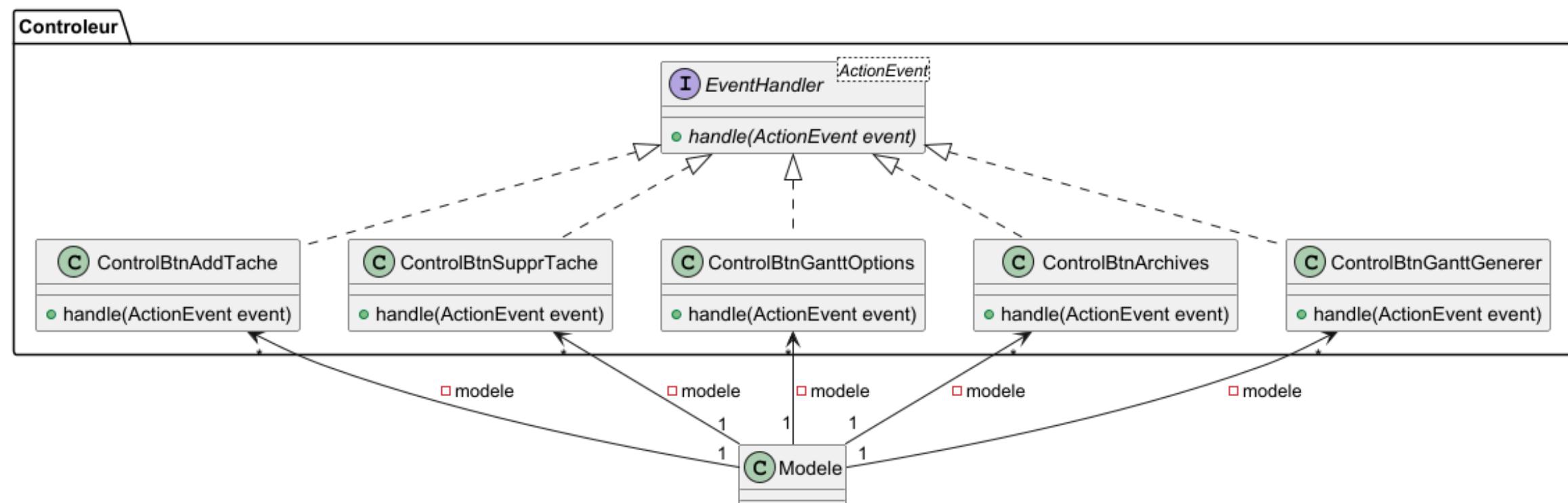
## Patrons d'architecture :

- **MVC (Modèle/Vue/Contrôleur)**
- Active Record



## Patrons de conception :

- **Observateur**
- Composite
- Fabrique
- Singleton



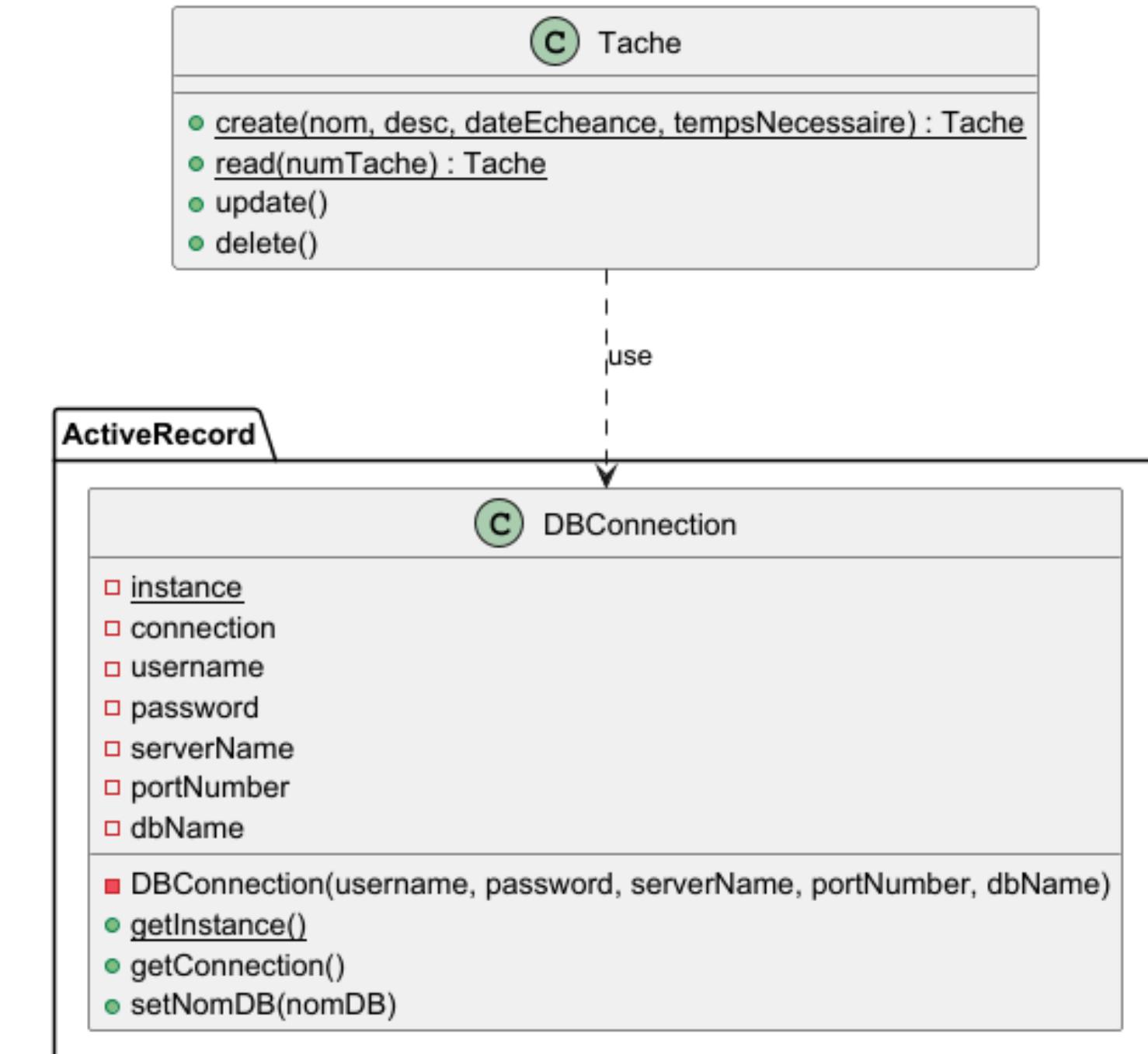
# Préparation à la programmation

## Patrons d'architecture :

- MVC (Modèle/Vue/Contrôleur)
- **Active Record**

## Patrons de conception :

- Observateur
- Composite
- Fabrique
- **Singleton**



# En conclusion

**Qu'est ce que nous a apporté cette partie**

- Identifier les besoins
- Réduire les ambiguïté
- Proposer une conception selon les fonctionnalité repérées
- Visualisation de l'interface