

# AN Desk

	In Progr.	Testing
	<div>us No ⑦</div>	<div>us No ⑤</div> <div>us No ⑥</div> <div>us ④</div>

Architecture Logicielle :

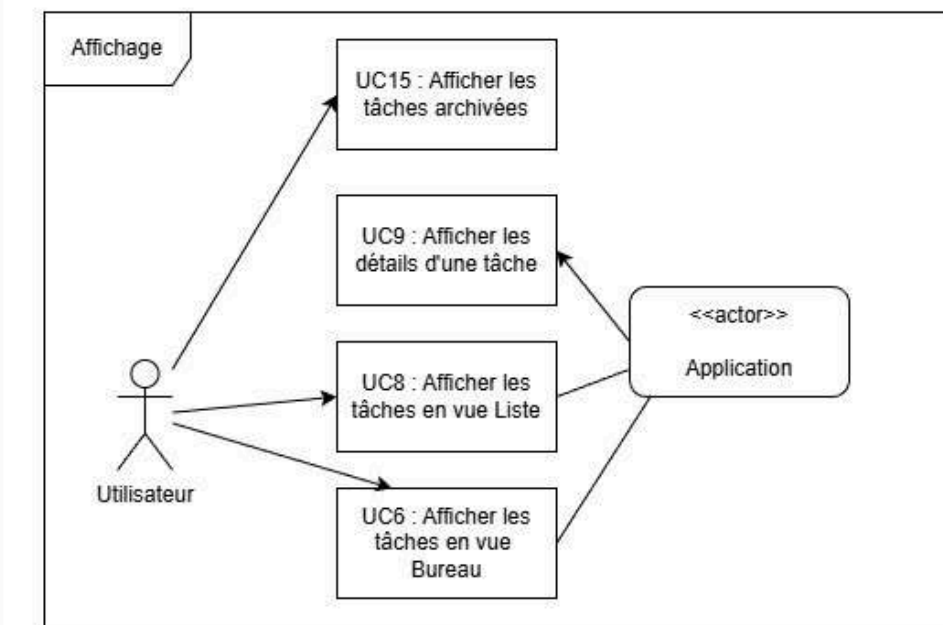
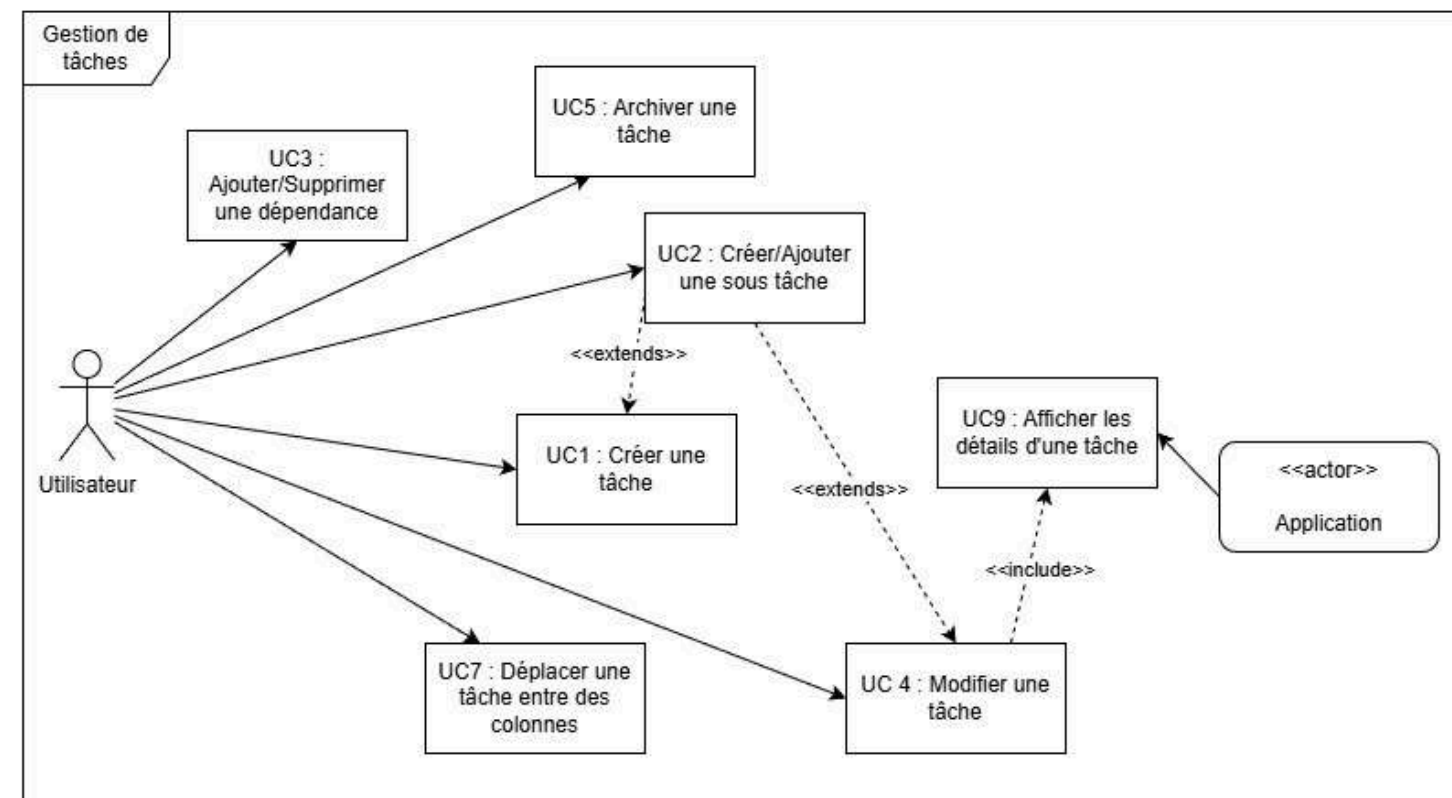
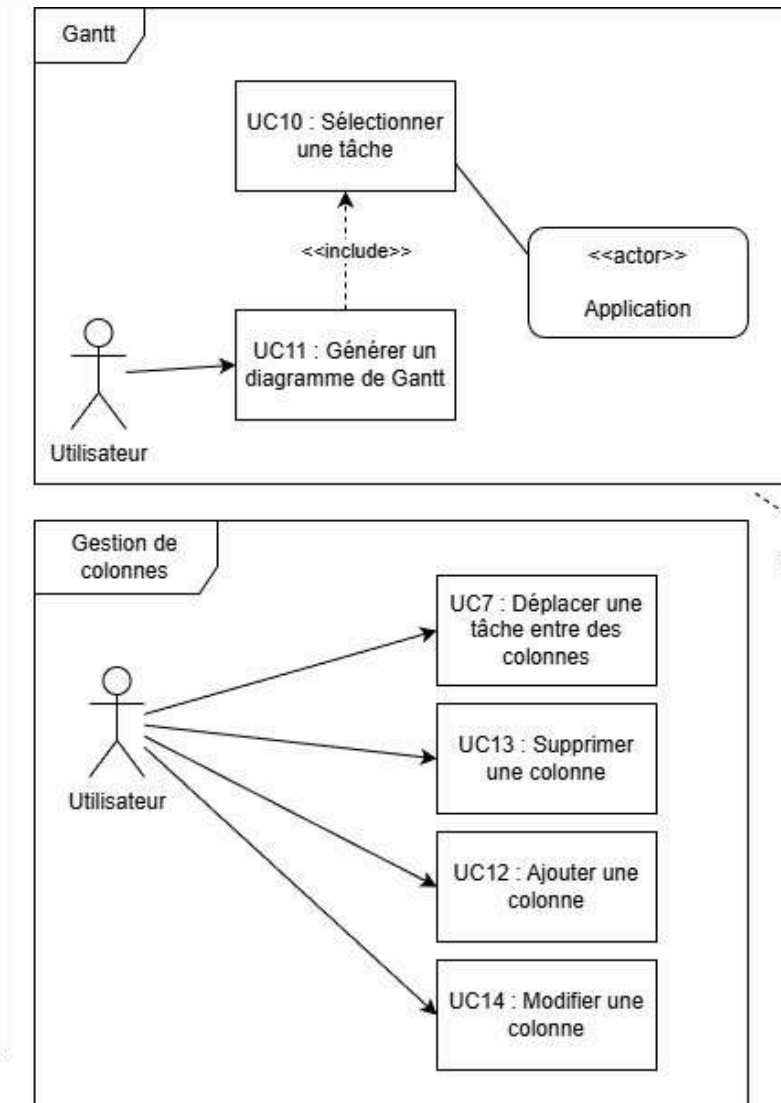
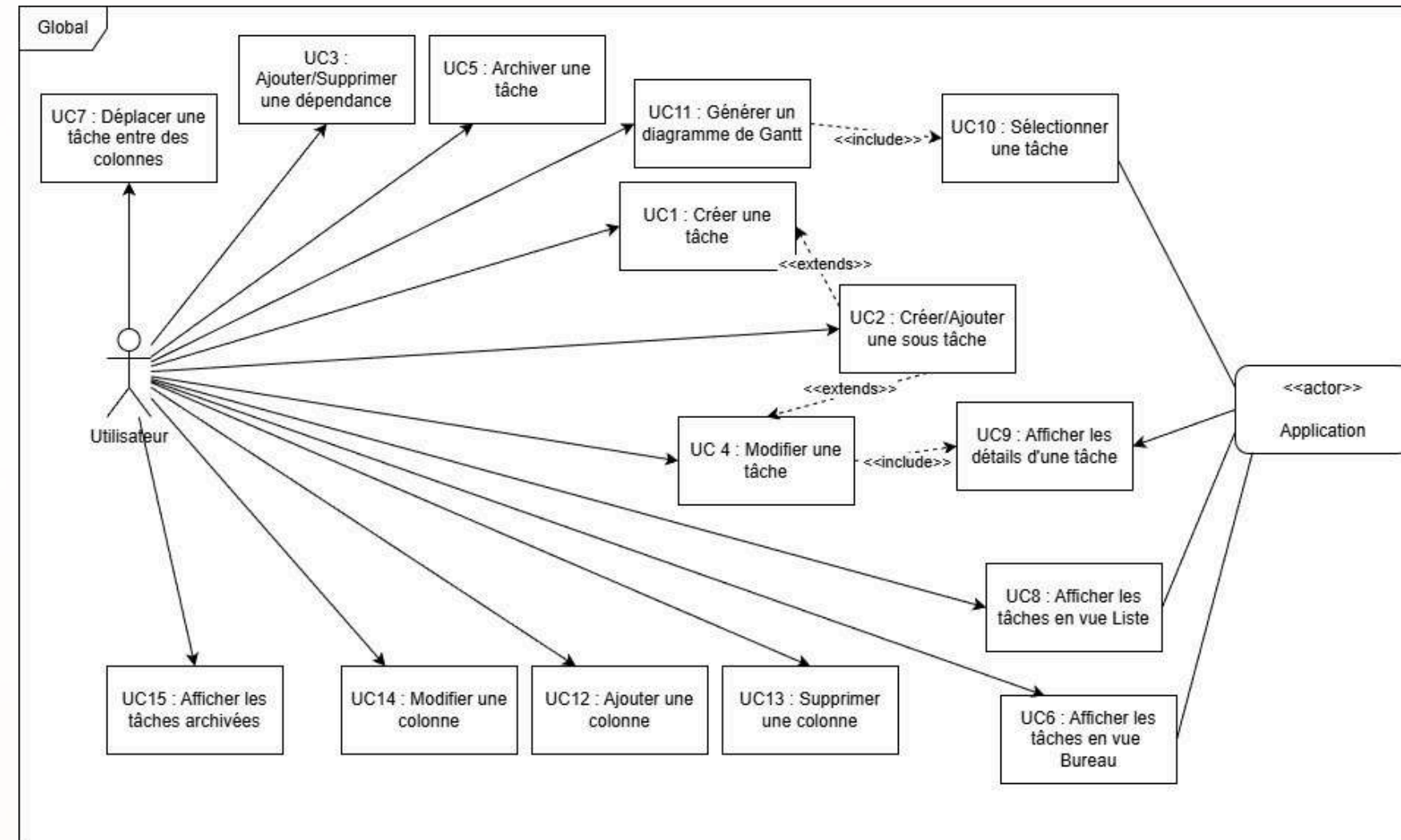
# Vue d'ensemble fonctionnelle

Le diagramme de cas d'utilisation définit le périmètre fonctionnel du système, centré sur l'utilisateur.

## Acteurs et Interactions

- Identification des acteurs principaux et secondaires.
- Définition des limites du système.
- Relations d'inclusion et d'extension entre les cas.

Cette modélisation assure l'alignement entre les besoins métier et les fonctionnalités techniques implémentées.



# Cycle de vie d'une tâche

## État Initial

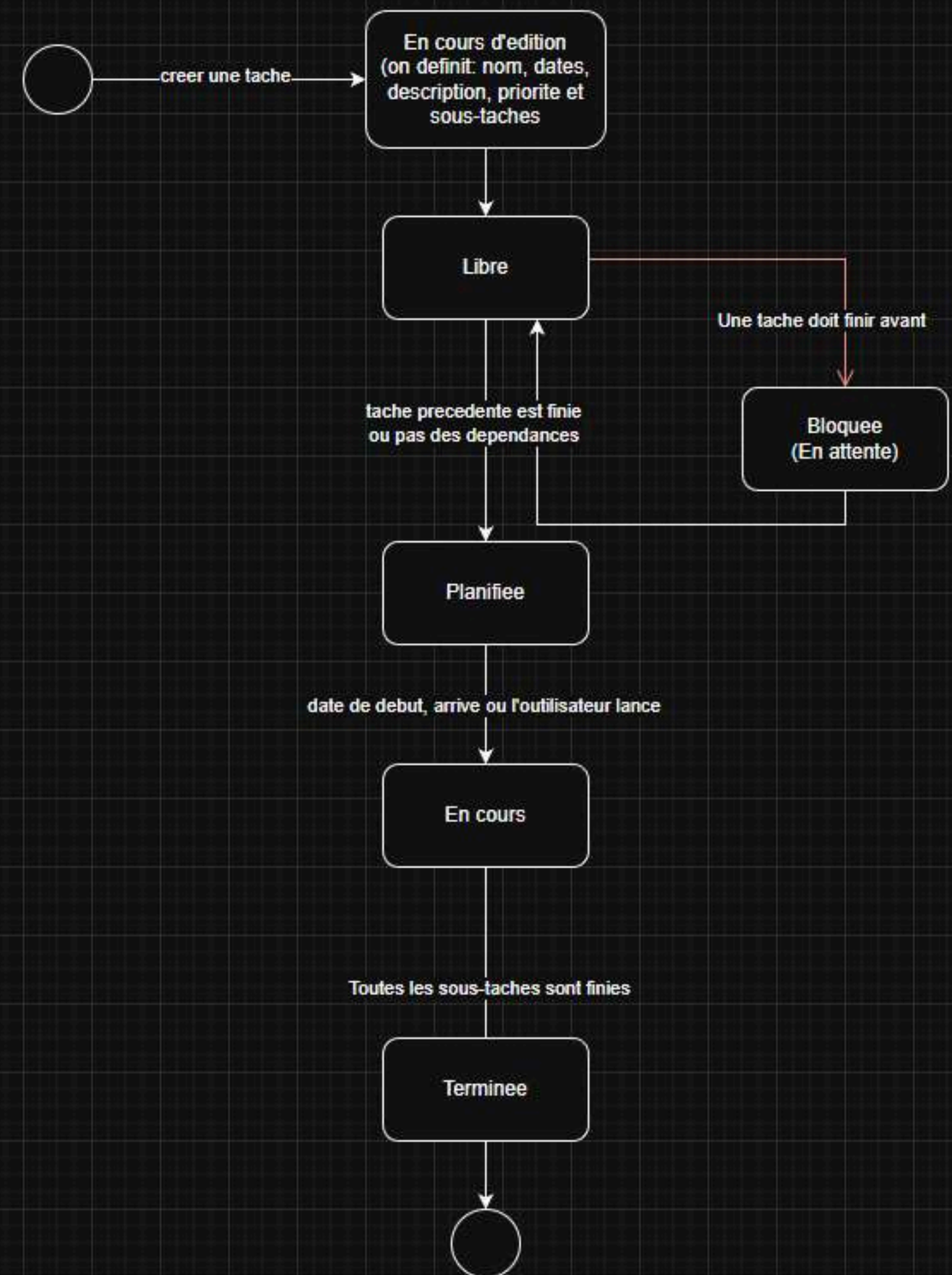
Création de l'instance de tâche. La tâche est en attente de traitement.

## Transitions

Passages logiques basés sur les événements déclencheurs (start, pause, complete).

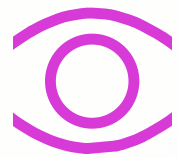
## État Final

Clôture ou archivage de la tâche une fois le processus terminé.



# Patrons de Conception

Implémentation de solutions éprouvées pour résoudre les problèmes récurrents de conception logicielle.



## Observer

Utilisé pour l'interface graphique. Permet de notifier les vues des changements d'état du modèle en temps réel.



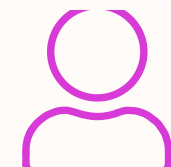
## Strategy

Interchangeabilité des algorithmes d'affichage. Permet de basculer dynamiquement entre les modes "Bureau" et "Liste".



## Composite

Gestion hiérarchique des tâches. Traite les Tâches simples et les Sous-Tâches de manière uniforme.



## Singleton

Garantit une instance unique pour la connexion à la base de données, optimisant les ressources.



# Architecture Globale et Persistance

## Modèle MVC

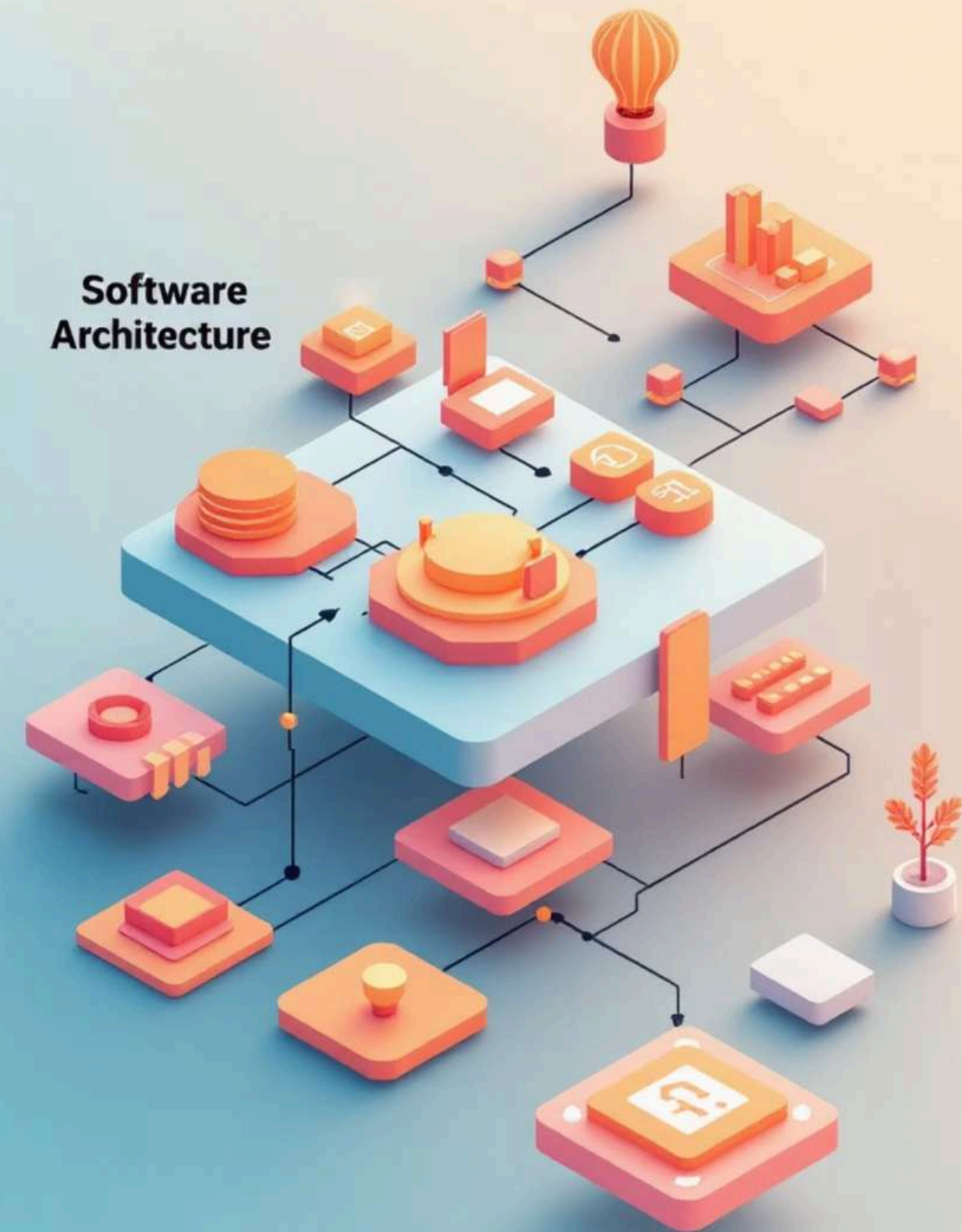
Séparation stricte des préoccupations :

- **Modèle** : Logique métier et données.
- **Vue** : Interface utilisateur et rendu des tâches.
- **Contrôleur** : Orchestration des interactions utilisateur.

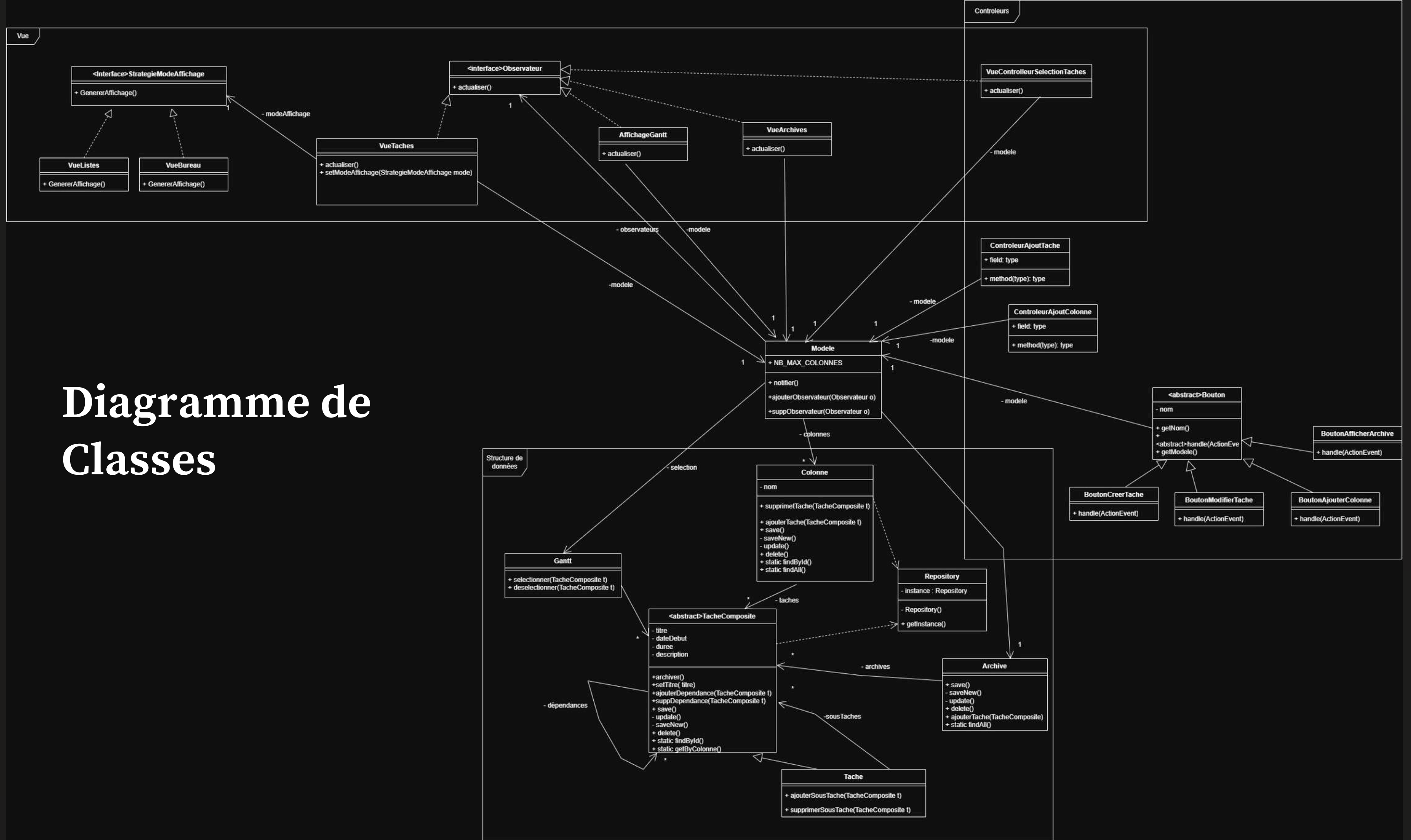
## Active Record

Pattern architectural pour la couche de données :

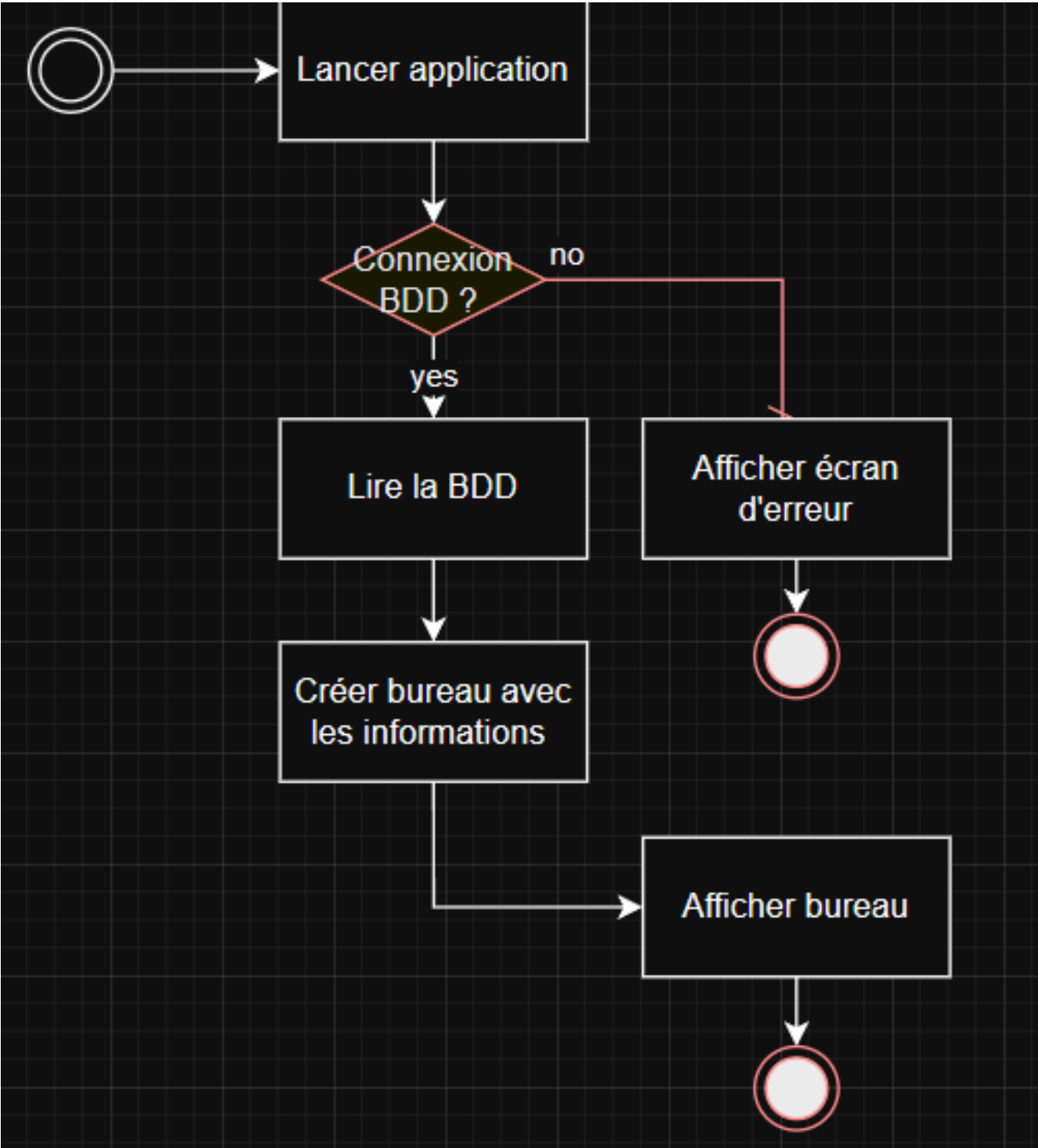
- Encapsulation de l'accès aux données dans les objets métier.
- Sauvegarde directe des Tâches et Colonnes.
- Simplification des opérations CRUD (Create, Read, Update, Delete).



# Diagramme de Classes

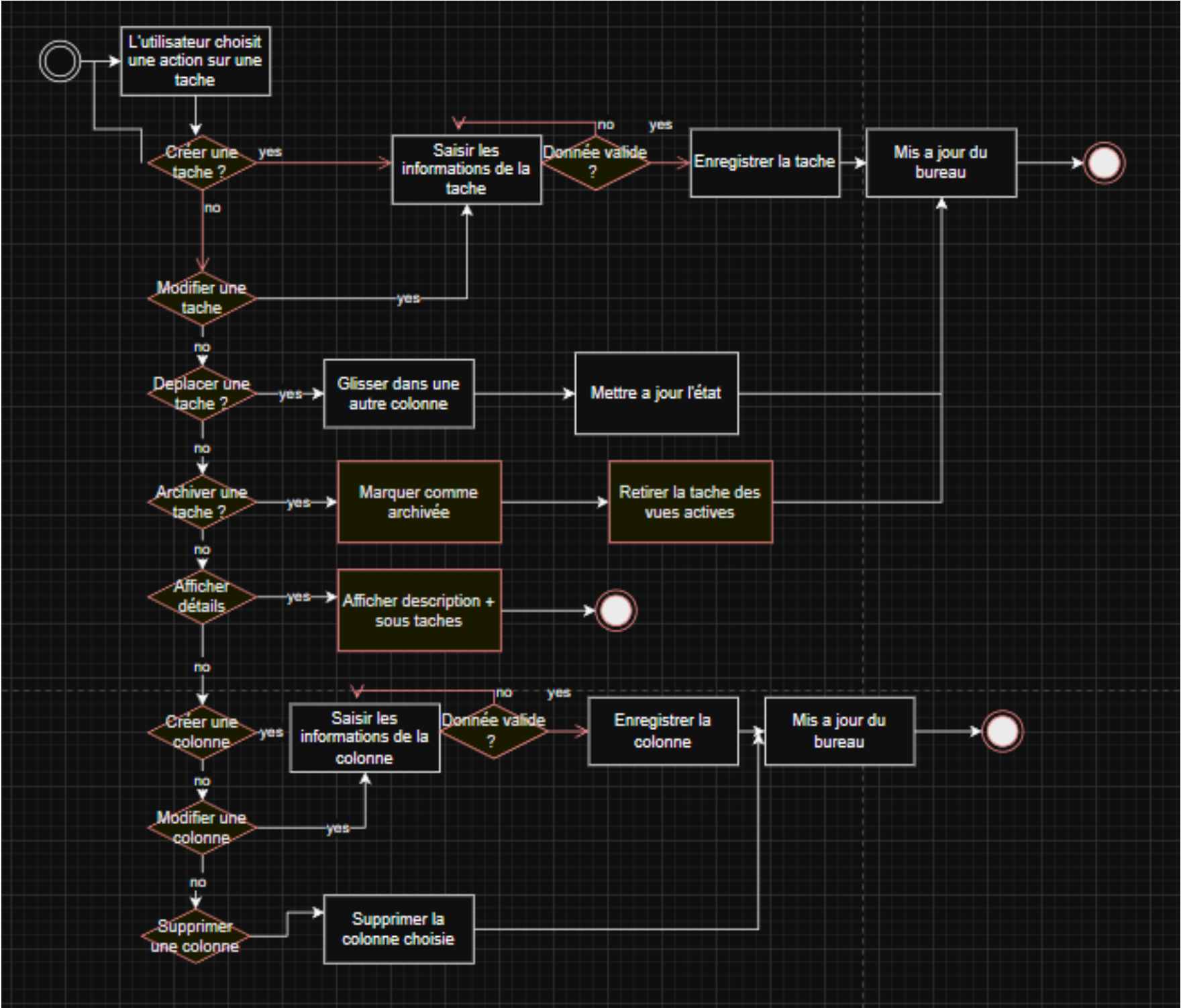


# Diagramme d'activité



## Démarrage

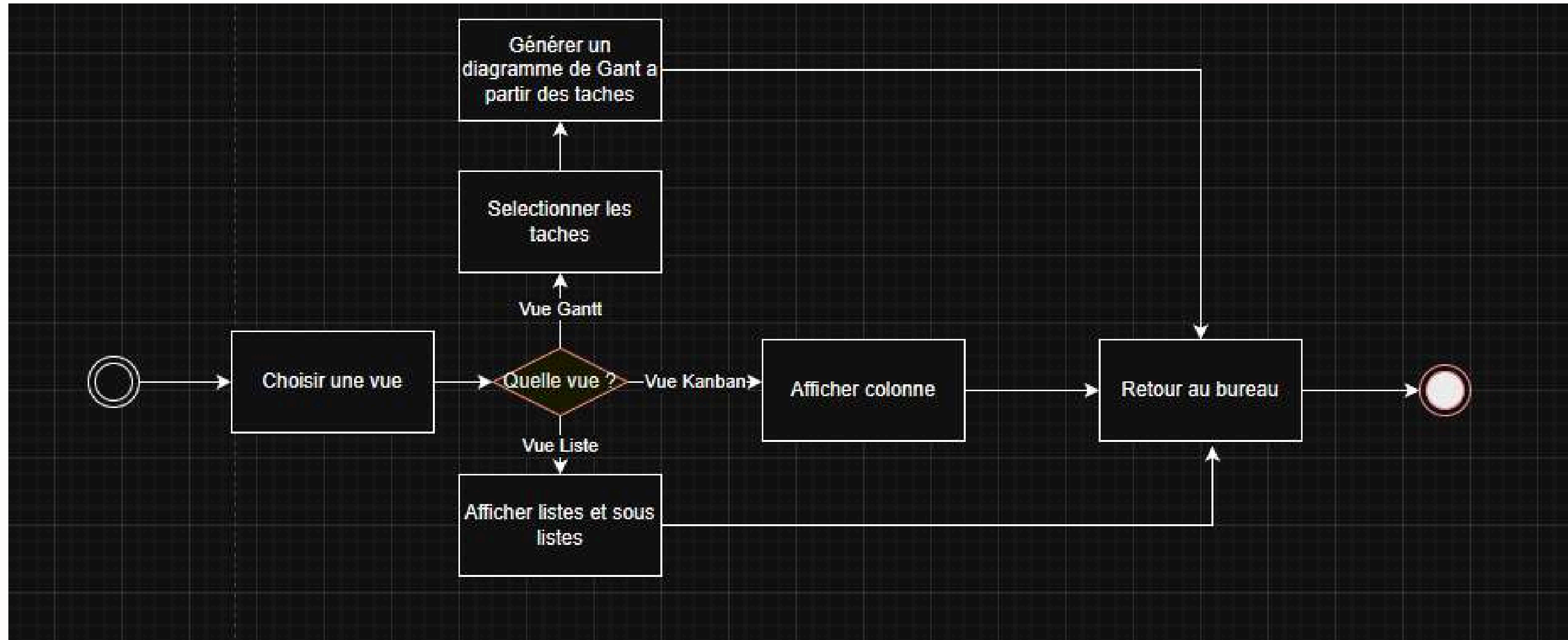
Initialisation des ressources système.



## Gestion

Logique de traitement des tâches.

# Diagramme d'activité



## Visualisation

Rendu graphique des données.



# Présentation de la maquette

