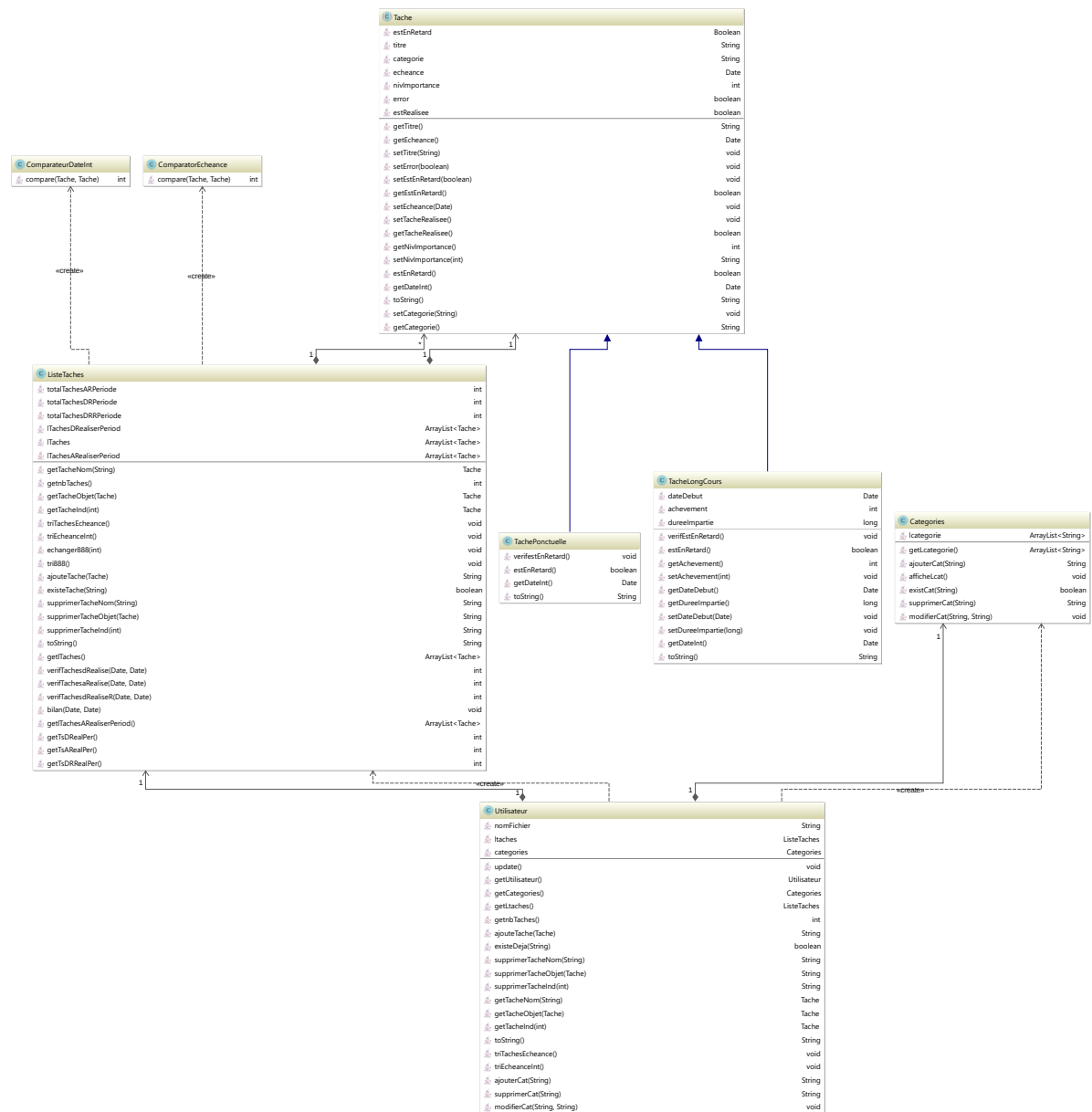


LES DIAGRAMMES

1. Diagrammes de classes

- La couche métier



TachePonctuelle et TacheLongCours héritent toutes les deux de Tâche (classe abstraite) dans laquelle on définit le comportement générique des tâches.

La classe Categorie représente les différentes catégories de tâches. Son seul attribut est une collection de chaînes de caractères qui représentent chacune une catégorie de tâches. Cette collection possède par défaut les catégories « Personnelle » et « Travail ». On y retrouve les différentes méthodes qui permettent d'ajouter, modifier et supprimer une catégorie.

ListeTache est une classe qui a en attribut une collection de tâches. Elle gère l'ajout, la suppression, la modification de tâches et le tri de ces dernières : les différentes méthodes de tris sont des méthodes de cette classe. Le tri par date d'échéances est réalisé grâce à un objet de la classe ComparatorEcheance implémentant l'interface Comparable et le tri par date d'échéances intermédiaires grâce à un objet de la classe CompareurDateInt qui implémente l'interface Comparator. Ils nous permettent d'imposer un ordre de tri à la collection.

La classe Utilisateur est celle qui s'occupe de sérialiser toutes les données. Elle représente l'utilisateur de l'application. Nous sommes partis du principe suivant : Un utilisateur possède une liste de tâches et définit leurs catégories.

- **Le modèle MVC**

Ce modèle permet une conception propre en séparant la couche métier de la partie graphique.

- a) Le modèle*

Il s'agit de la classe Utilisateur composée des collections de tâches et de catégories.

- b) La vue (classe TacheView)*

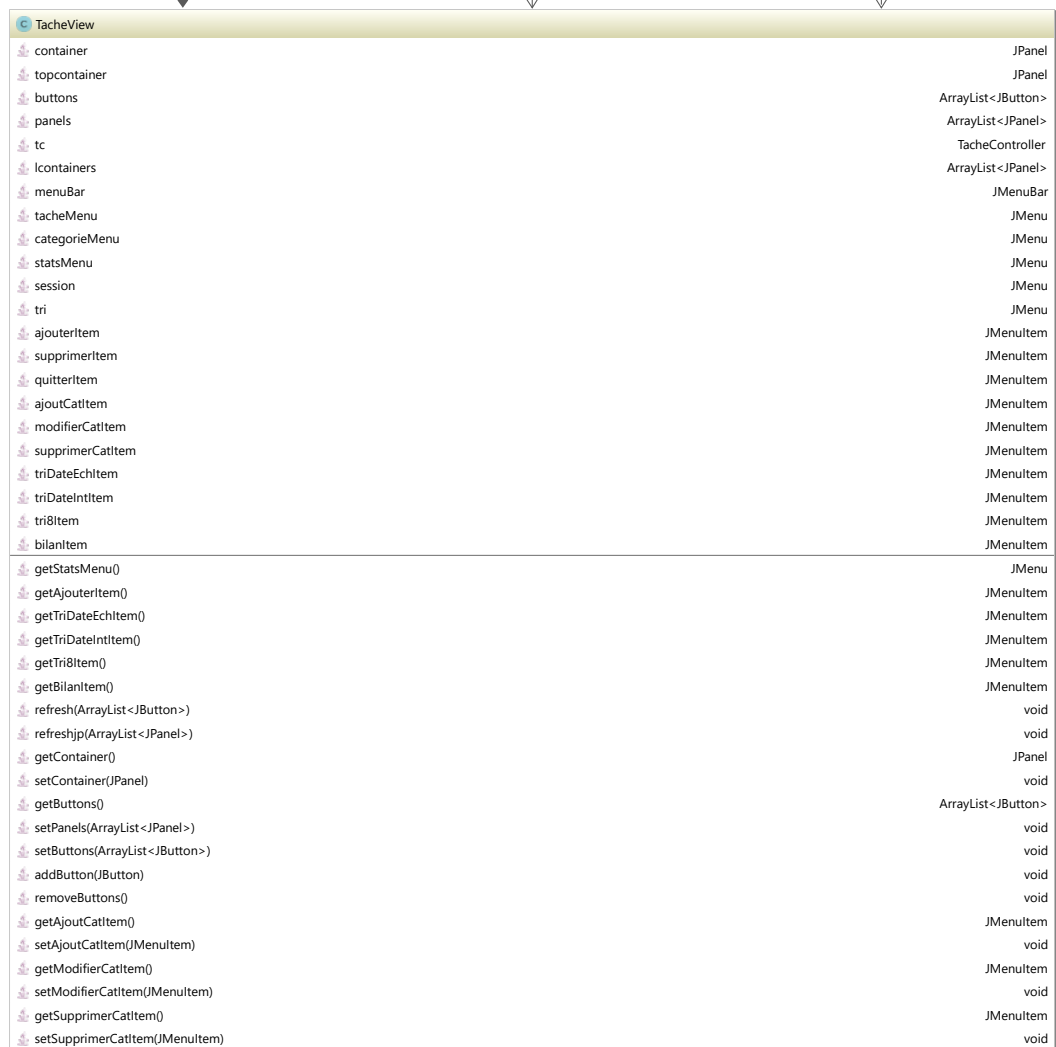
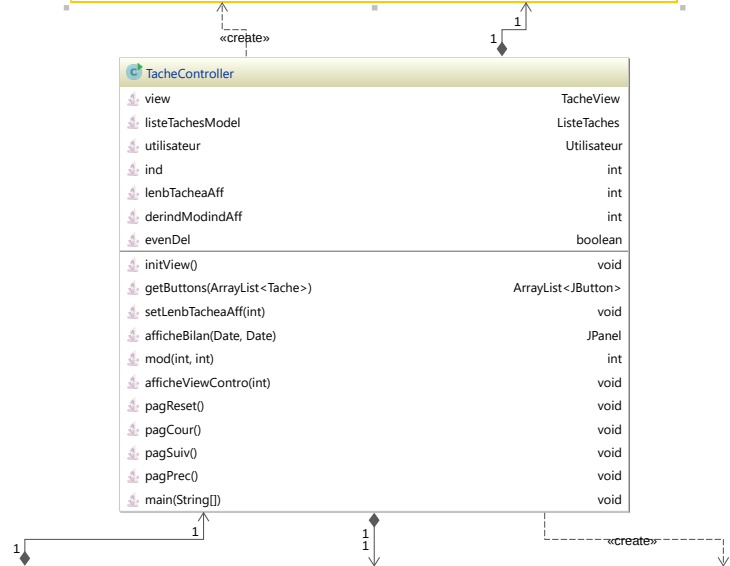
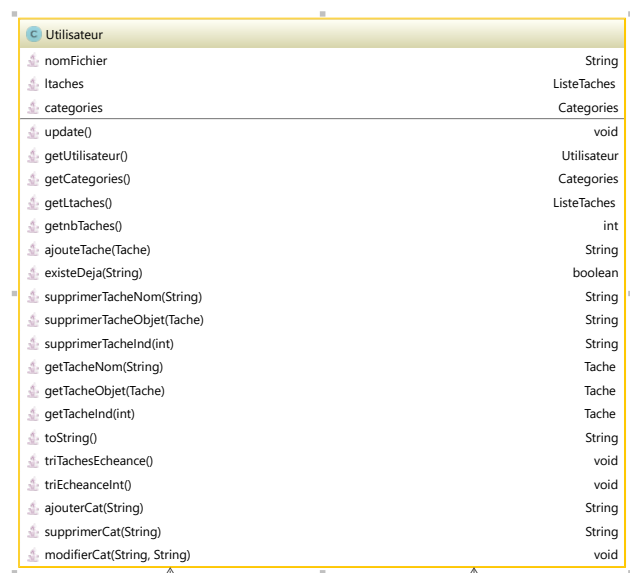
C'est l'interface graphique. Elle comprend tous les composants swing (JPanel, JButton, etc)

- c) Le contrôleur (classe TacheController)*

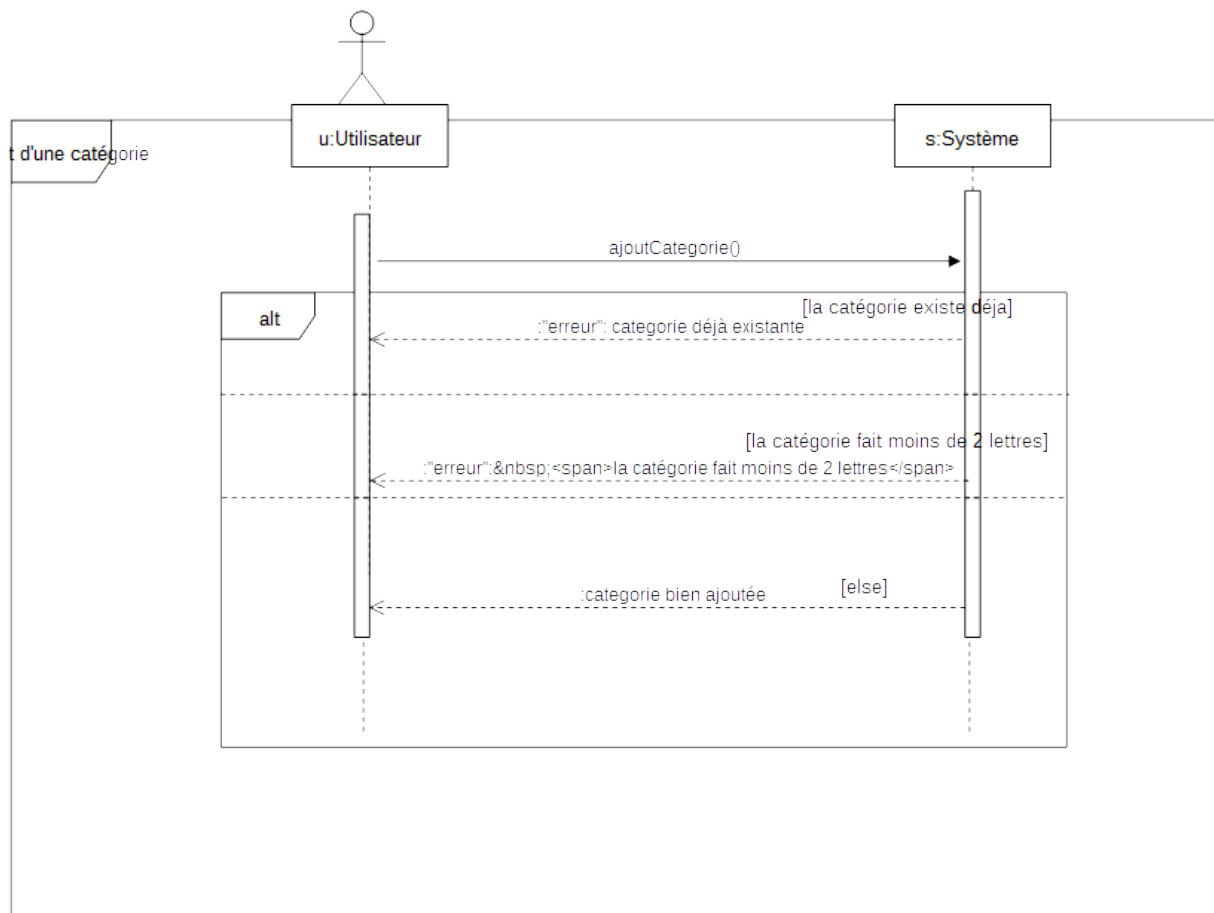
Il fait le pont entre la vue et le contrôleur. Il possède donc en attribut le modèle et la vue. Le modèle et la vue communiquent au moyen d'écouteurs d'événements qui sont placés sur les différents éléments de la vue par le biais d'accesseurs en lecture

- d) La classe PageSuivEcouteur*

C'est un écouteur d'événements qui permet de passer d'une page à l'autre (la page de l'application présente uniquement 12 tâches, il faut donc pouvoir voir les autres tâches



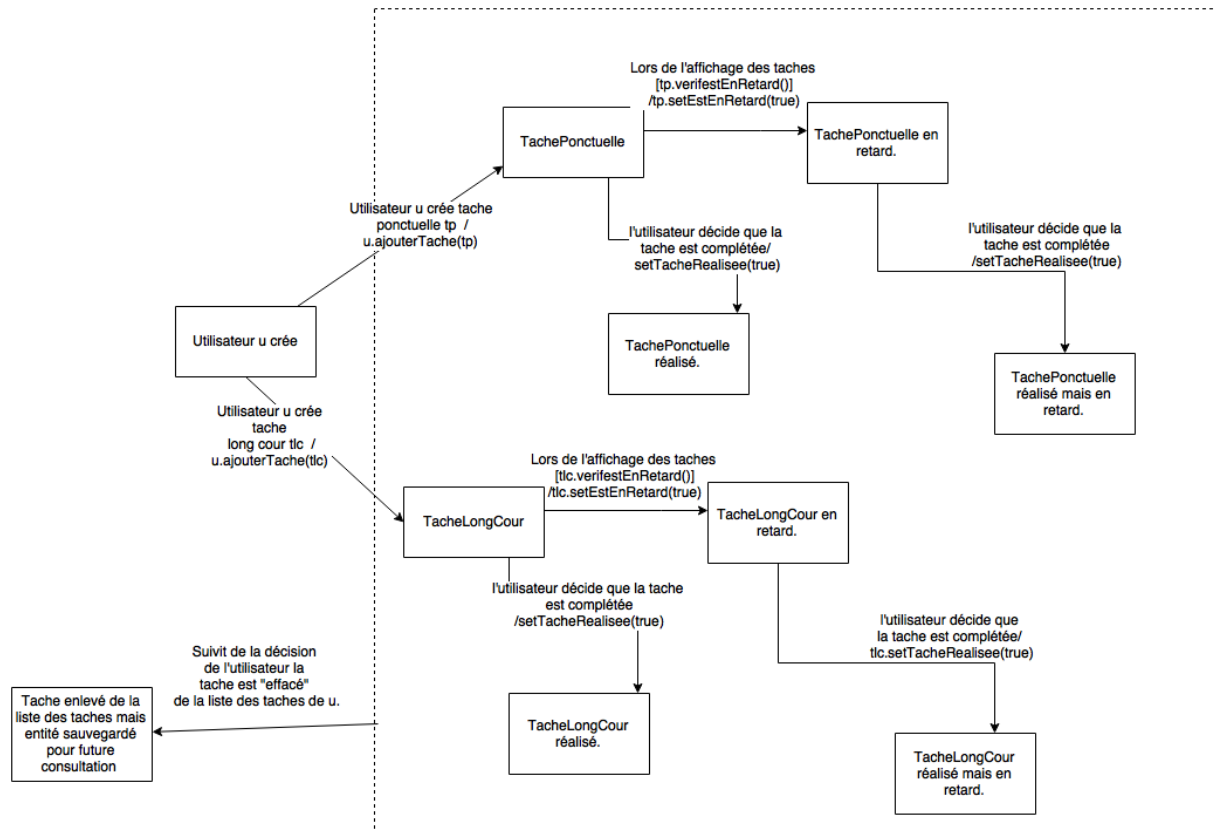
2. Diagramme de séquence : Ajout par l'utilisateur d'une catégorie.



L'utilisateur veut ajouter une nouvelle catégorie au système. Trois cas sont possibles :

- La catégorie existe déjà
- La catégorie fait moins de deux lettres
- La catégorie est bien.

3. Machines à états pour les tâches



Voici la machine à état de la classe Tache (sous-classes) incluse. On voit bien que la vie d'une tâche dépend principalement de l'utilisateur mais que l'attribut déterminant le retard d'une tâche est soigneusement attribué lors du lancement du logiciel. On s'aperçoit aussi que l'une des principales différences des sous-classes de Tache se retrouve lors du calcul de cet attribut retard.