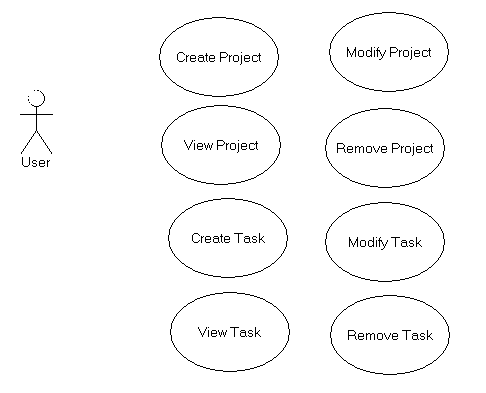
TimeSheet Project

Use Case



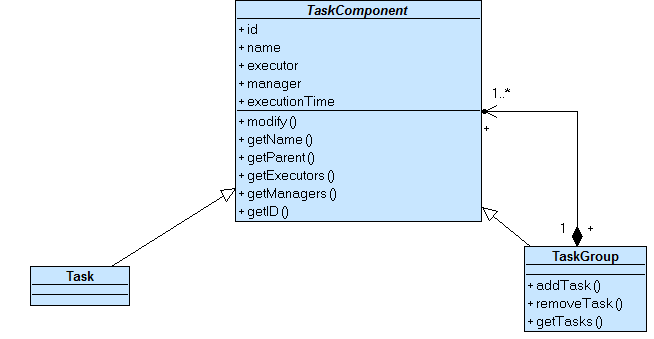
Task and Project definition

The task is the main element of the use case. We have to define what’s task?

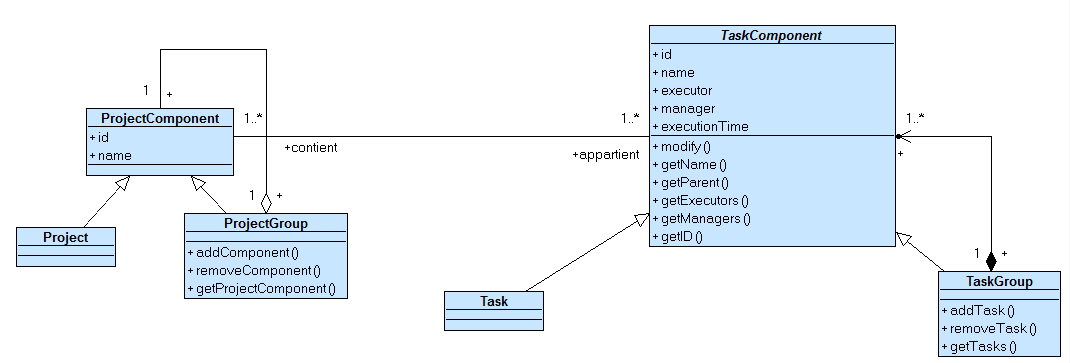
One task can be identified by his ID, his Name. The task can be executed by one or various executors. The task is tracked by one manager. The task is also defined by one execution time.

We can also imagine that task can be divided into sub-task for easier manager.

Following the last comment, we decide to use the Composite Pattern that allows realizing such hierarchy arborescent. Finally, we adopt the following modeling of the Task class.



The same modeling can be used for the Project Class.

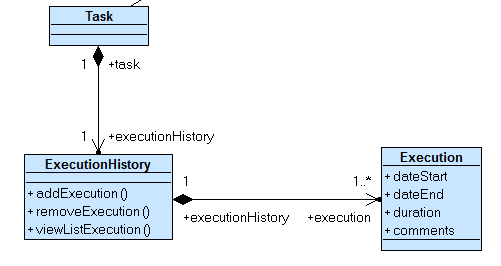


In this class diagram, tow new classes are emerging: Executor and Manager. But those attributes will be unheeded for the moment.

For the TimeSheet project, we now focalize to the execution time attribute. In fact, we want to:

* Calculate the time spent for the task by all executors
* View and Save all history of the time spent by executors only for task chosen

For the second functionalities, we need to add and attributes executionHistory.



Intéressons nous à la façon que nous voulons utiliser la function

1. L’utilisateur ouvre le logiciel
2. Il commence à configurer le logiciel (Nouveau use cas à ajouter) (cas de reconfiguration à envisager : l’utilisateur veut tout recommencer)
3. Une fois créer le projet et les sous projets, il peut commencer à créer un TaskComponent
   1. Choisir le projet (sous-projet) voulu
   2. Appuyer sur le bouton Add pour créer un nouveau TaskComponent
      1. Il faut choisir Task ou Group Task
   3. Une fois créer il faut qu’il apparaît dans la liste

En terme de rendu, comment on s’y attend

Vu normal du projet

Vu quand on demande de voir l’historique d’exécution de la tache

Si on sélectionne task(=> affichage de deux dernière exécution) et new (c'est-à-dire new execution = time sheet), alors

Première tentative : afficher la liste des exécutions de la Task