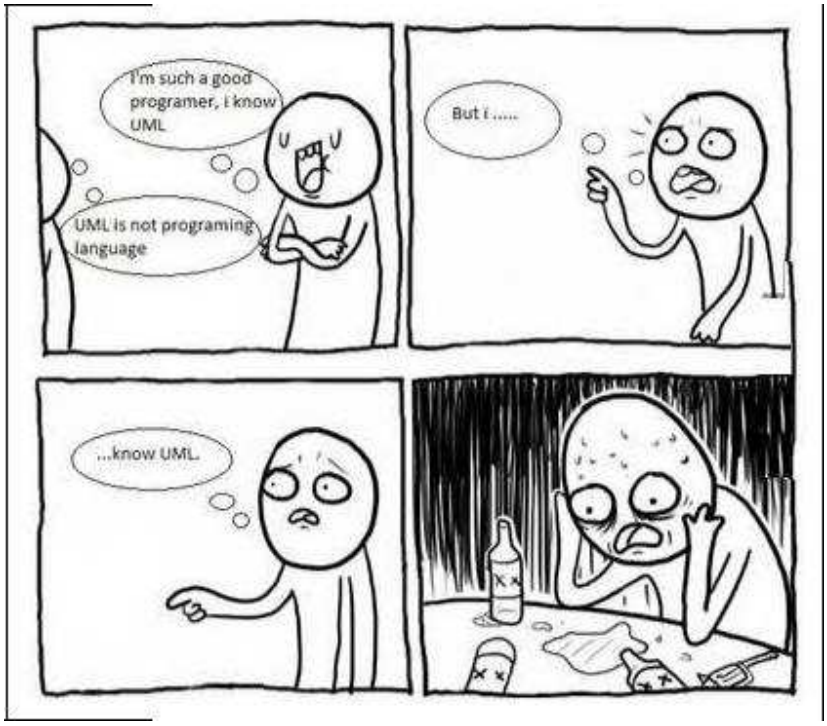
Modélisation Objet

Logiciel de gestion de projet



ANDUJAR Stanislas et CALLE BERT Maxime

Introduction

Durant ce projet nous devions modéliser un logiciel de gestion de projet décrit de la façon suivante :

Un projet est un ensemble de tâches, les tâches doivent être réaliser et dirigées par différents acteurs, des chefs de projet, des responsables et des contrôleur de tâches ainsi que des intervenant.

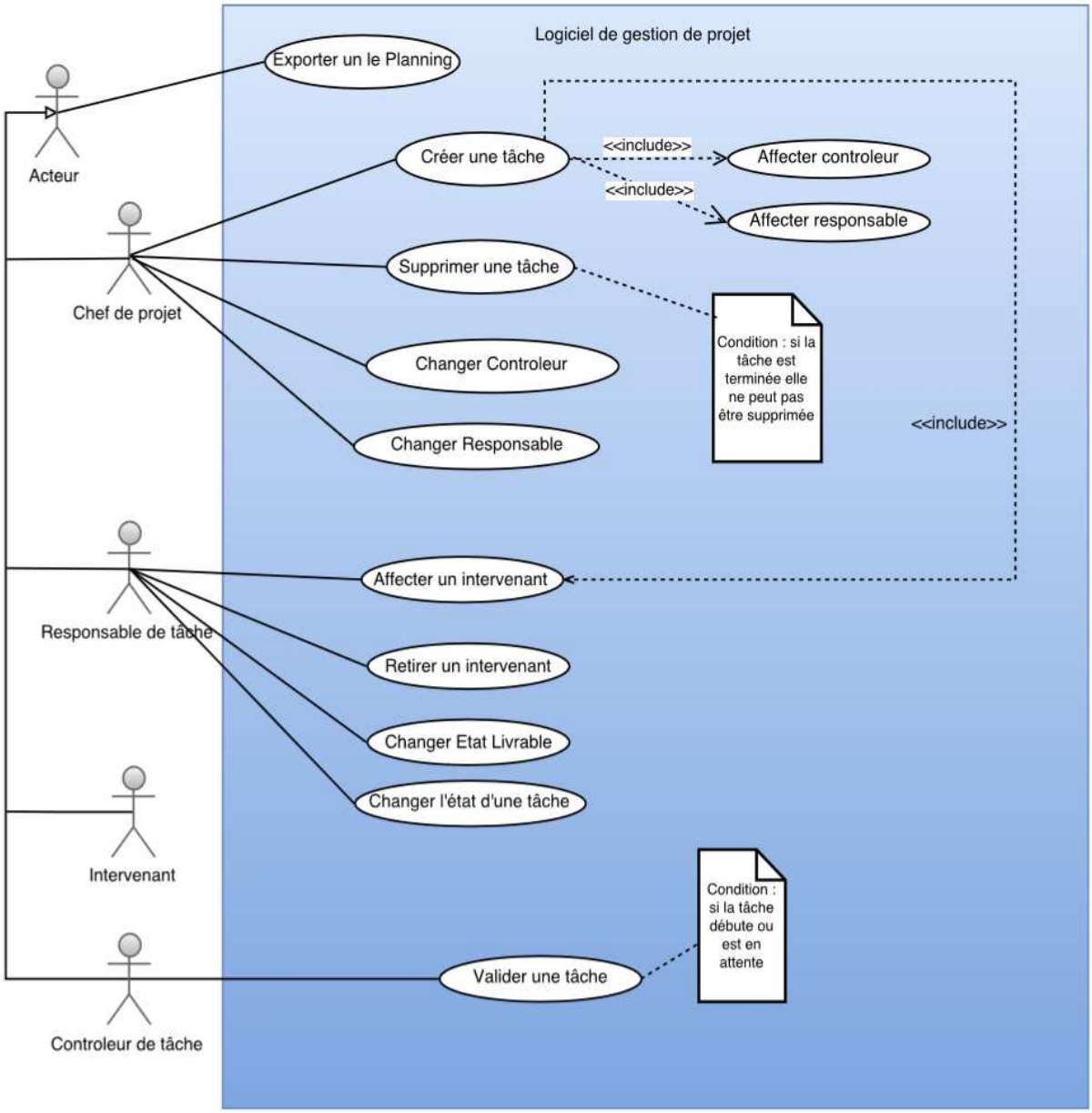
Plusieurs contraintes doivent être respectées, des contraintes de type financière et temporelles entre autre. Il existe une multitude de liens entre les différents acteurs et chaque acteurs possèdes des droits sur les autres et sur les tâches.

Afin de mener au mieux la modélisation, nous allons passer par la réalisation de différents modèles sous forme de diagramme.

SOMMAIRE :

Modèle de cas d'utilisation :

Diagramme des cas d'utilisation :



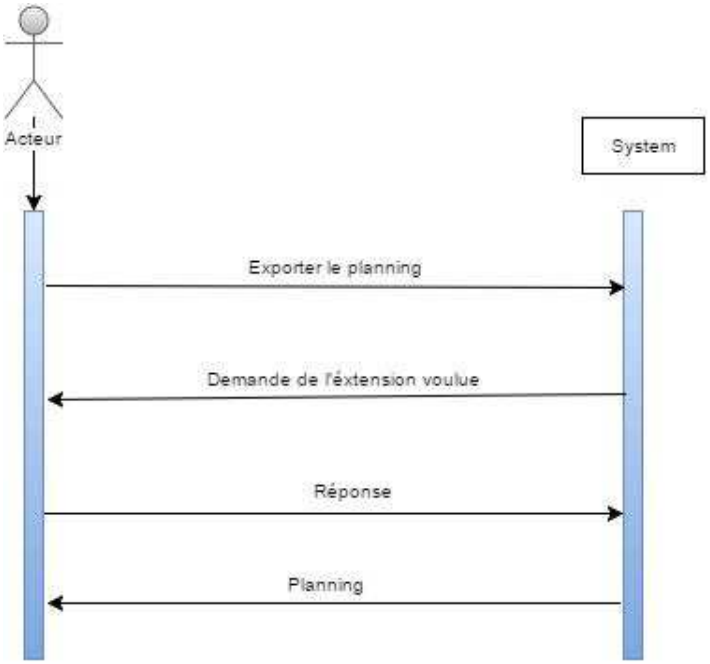
Ce diagramme met en évidence les différentes actions attribuées aux différents acteurs ainsi que les liens entre les cas d'utilisation.

Tout le monde est Acteurs.

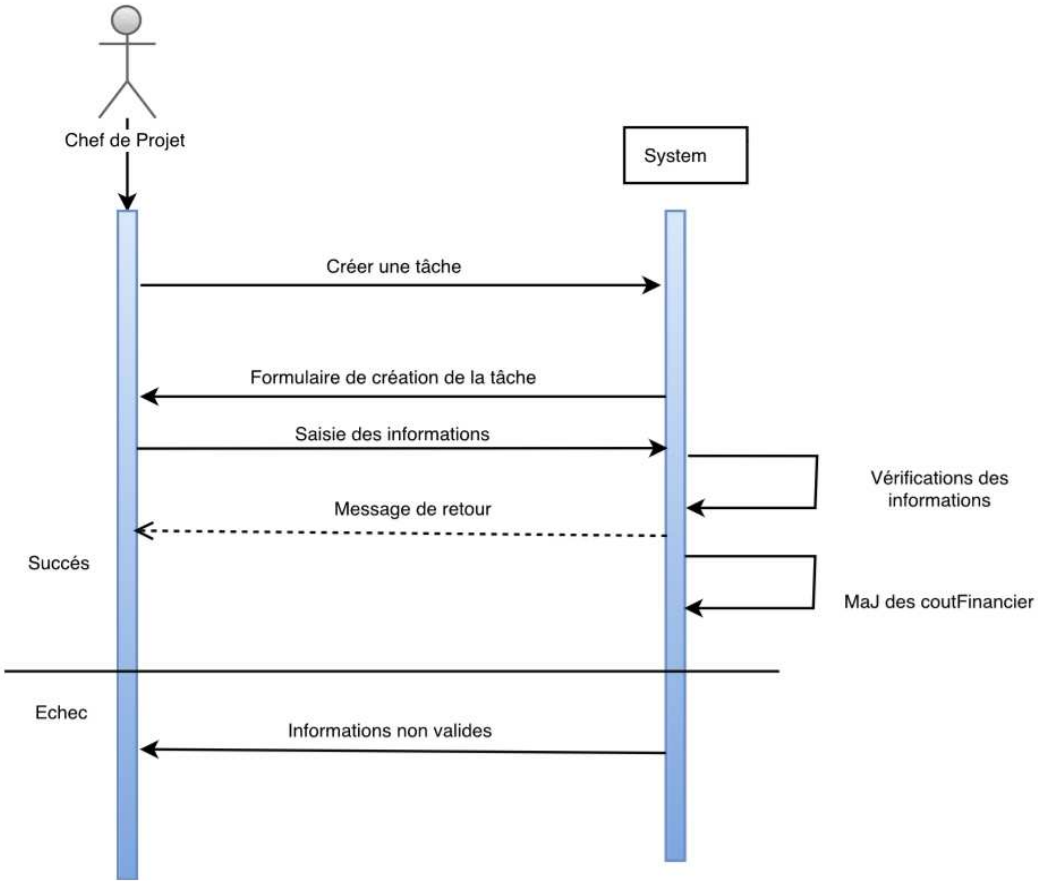
Modèle d'analyse :

Diagrammes interactions :

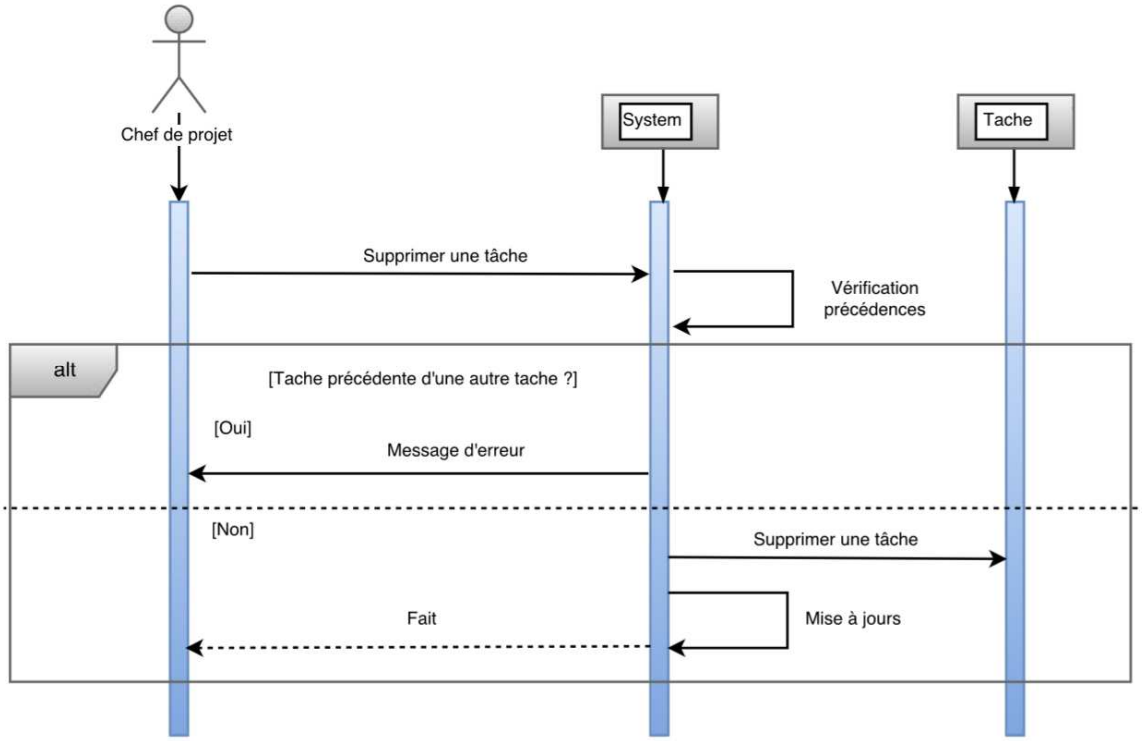
Acteur : Export Planning



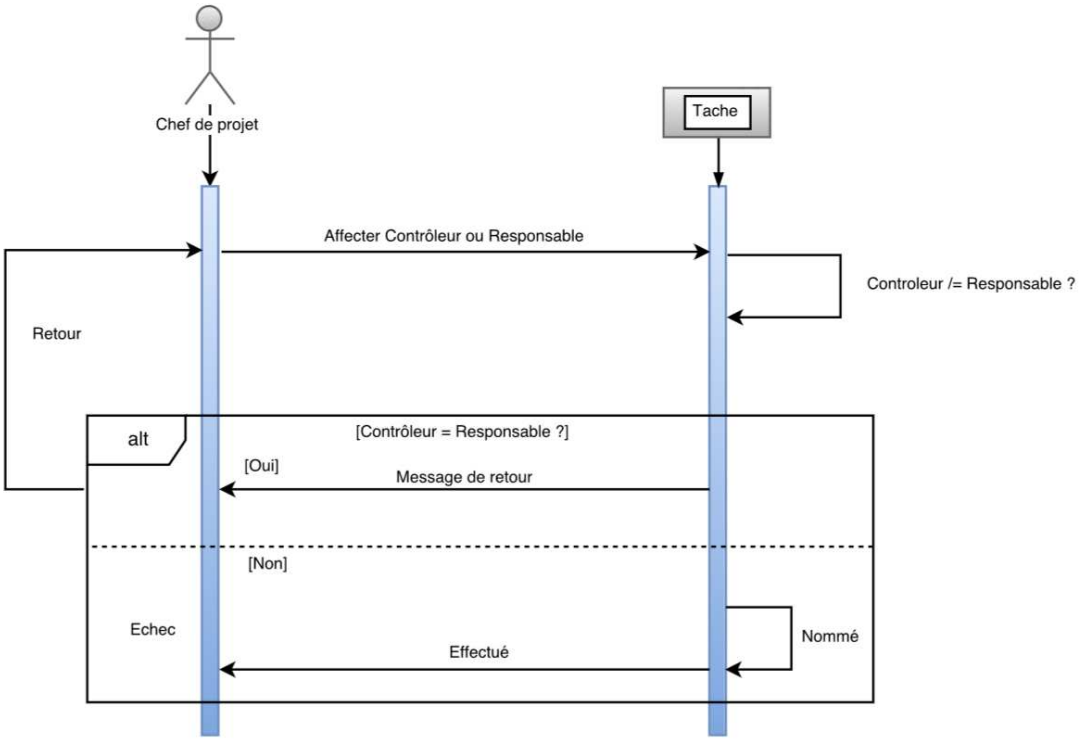
Chef de projet : Créer tâche



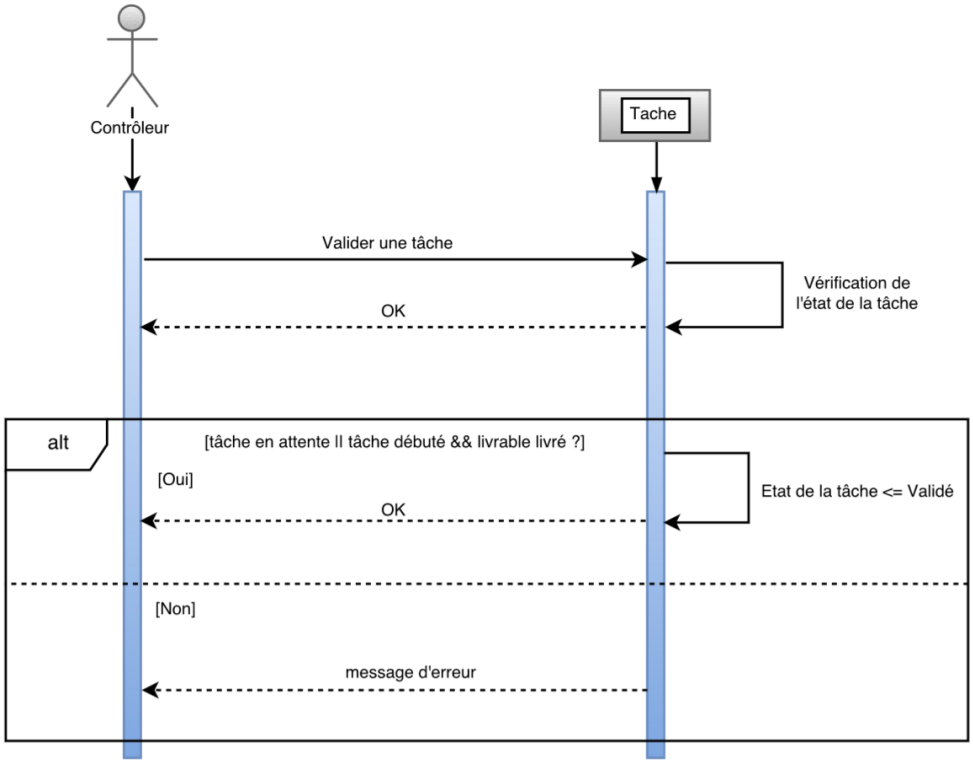
Chef de projet : Supprimer une tâche



Chef de projet : Affecter Contrôleur ou Responsable



Contrôleur de tâche - Valider une tâche



Responsable de tâche - Gérer un intervenant

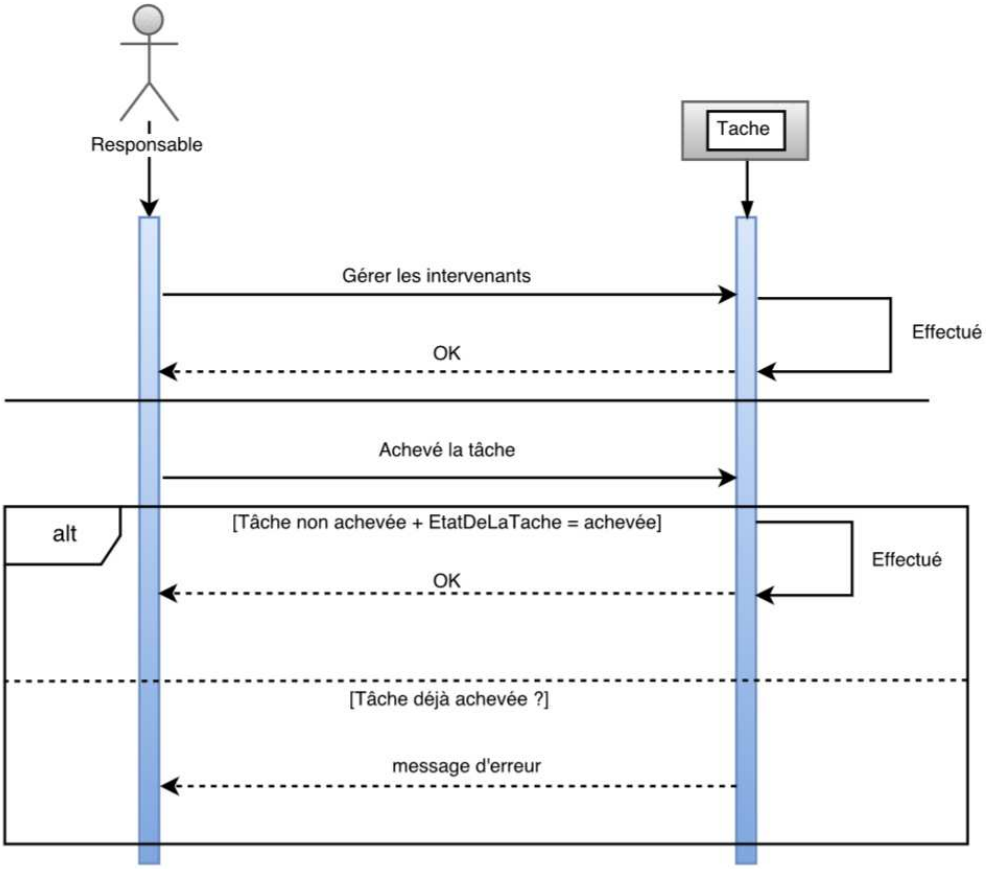
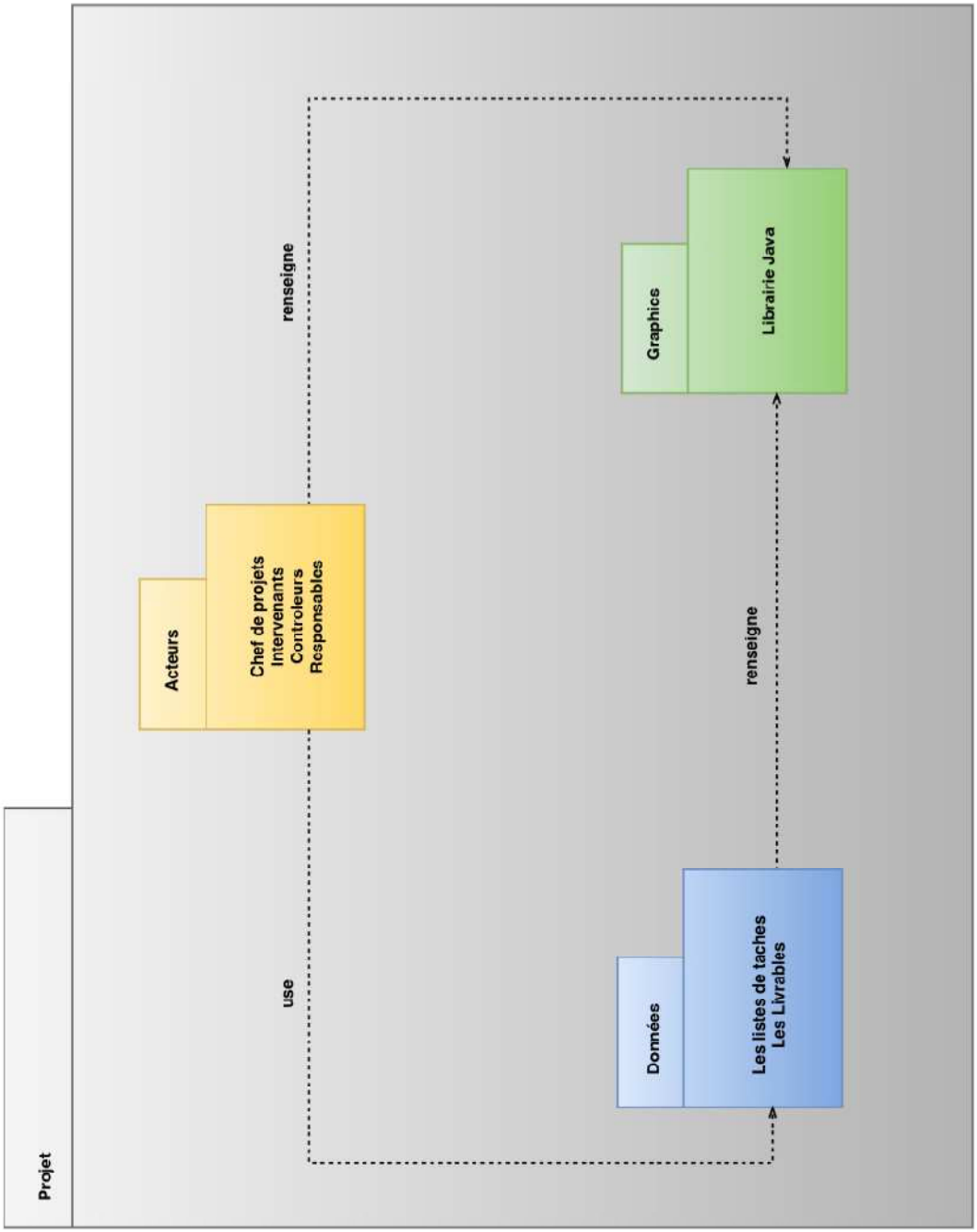


Diagramme de Package :



Ce Diagramme de package est un diagramme UML qui fournit une représentation graphique de l'organisation de notre application, il nous aide à identifier les liens de généralisation et de dépendance entre les packages. (cependant, nous n'avons pas d'interface).

Diagramme des classes :

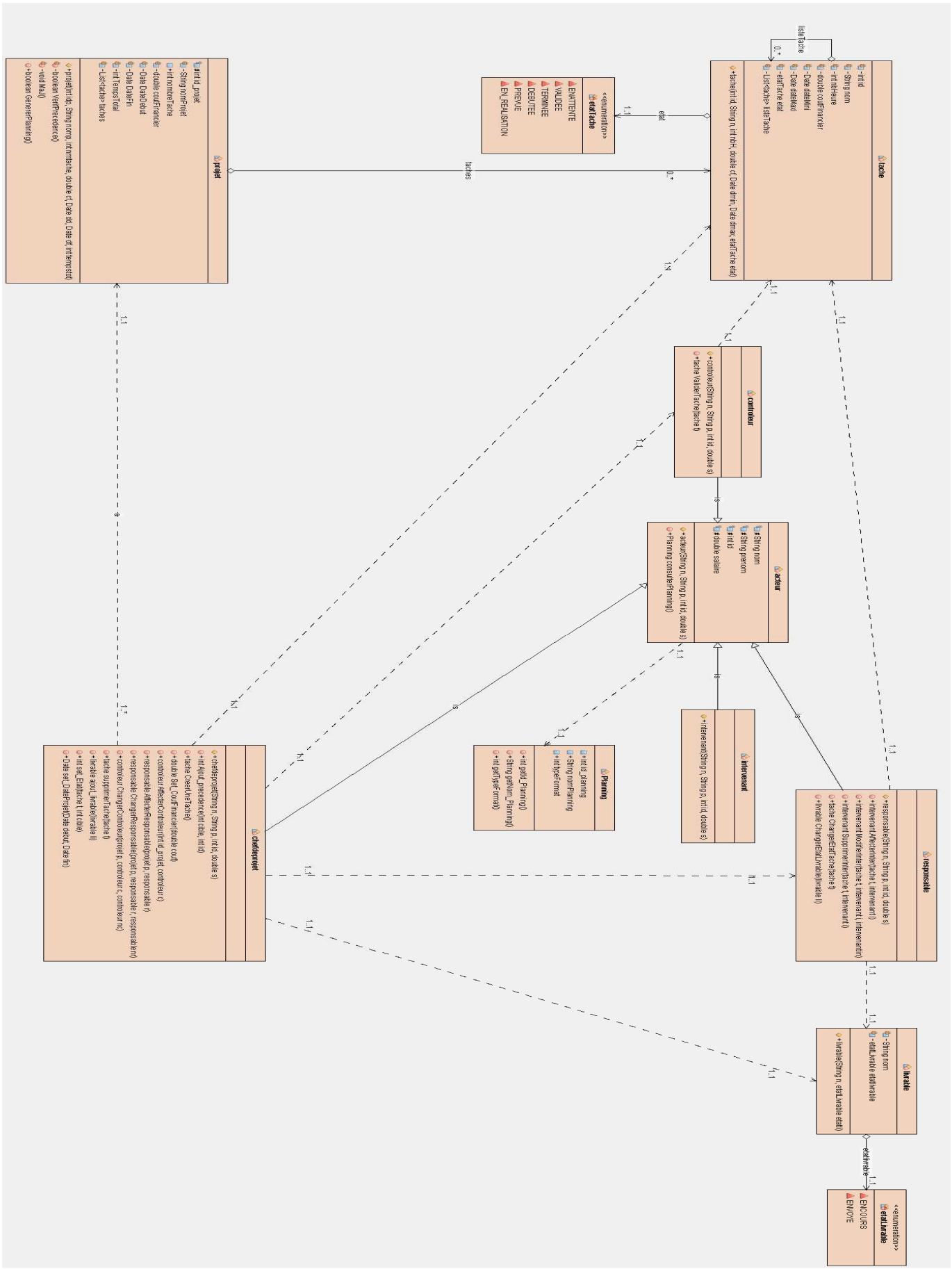
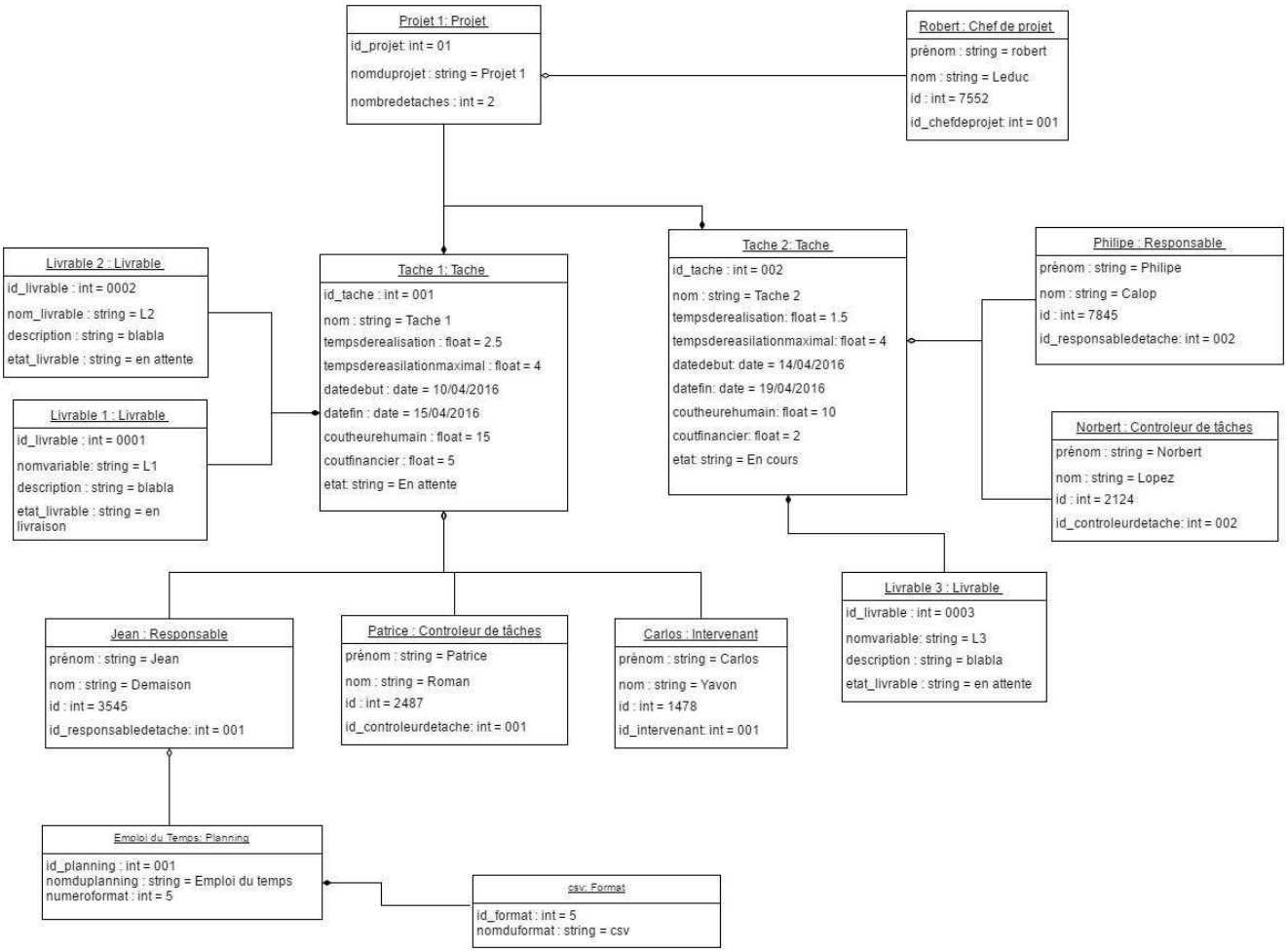


Diagramme d'objets :



(plus visible en PDF)

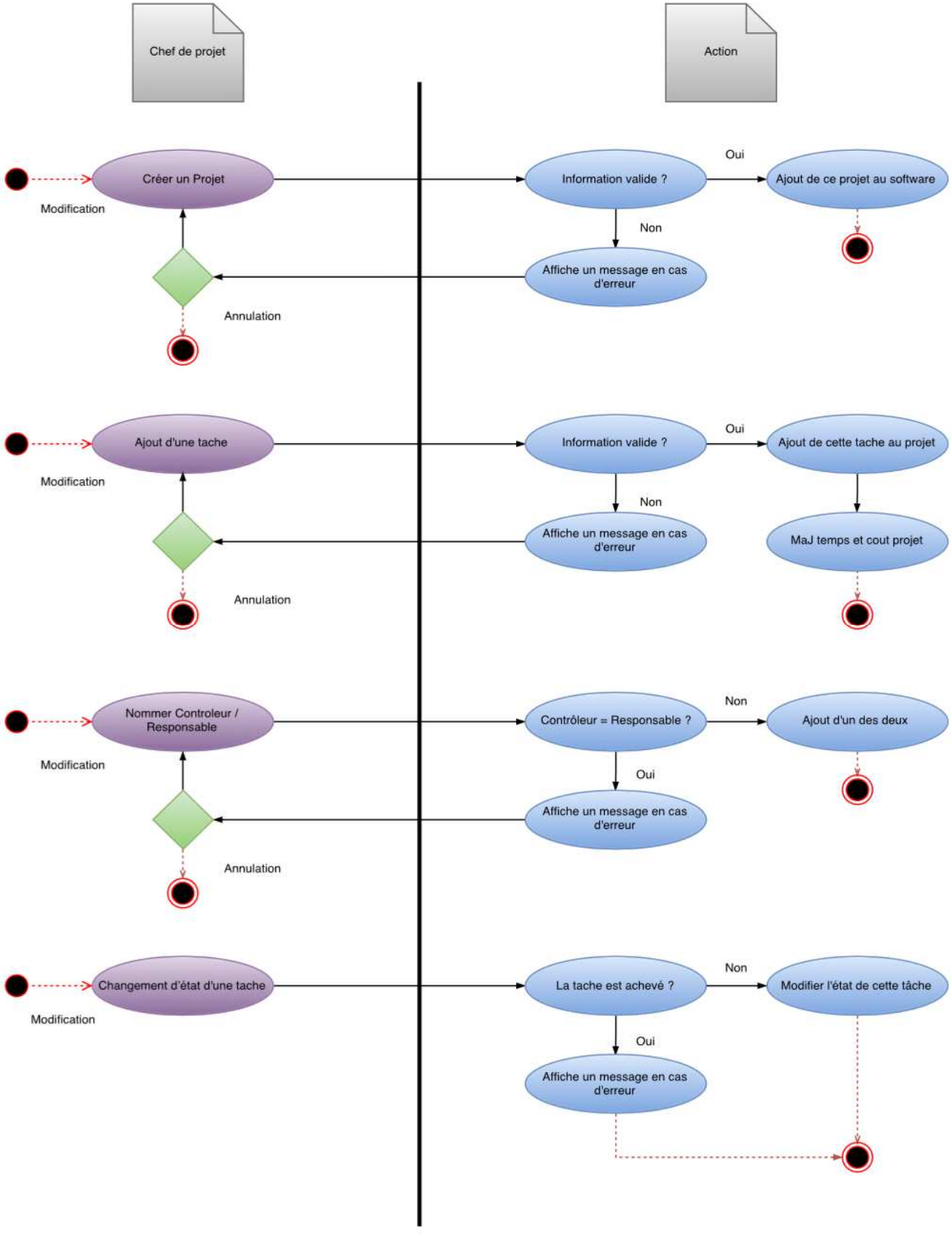
Ce diagramme d'objet permet de représenter les instances des classes.

Il exprime les relations qui existent entre les objets, mais aussi l'état des objets, ce qui permet d'exprimer des contextes d'exécution.

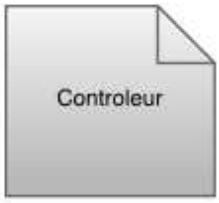
Ici nous avons donc un Chef de projet qui est en charge d'un projet composé de deux tâches qui on respectivement un responsable ainsi qu'un ensemble d'intervenant et au minimum un contrôleur de tâches chacune. On remarque qu'un responsable souhaite extraire l'emploi du temps au format csv.

Diagramme d'activité :

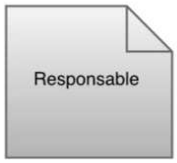
Chef de Projet

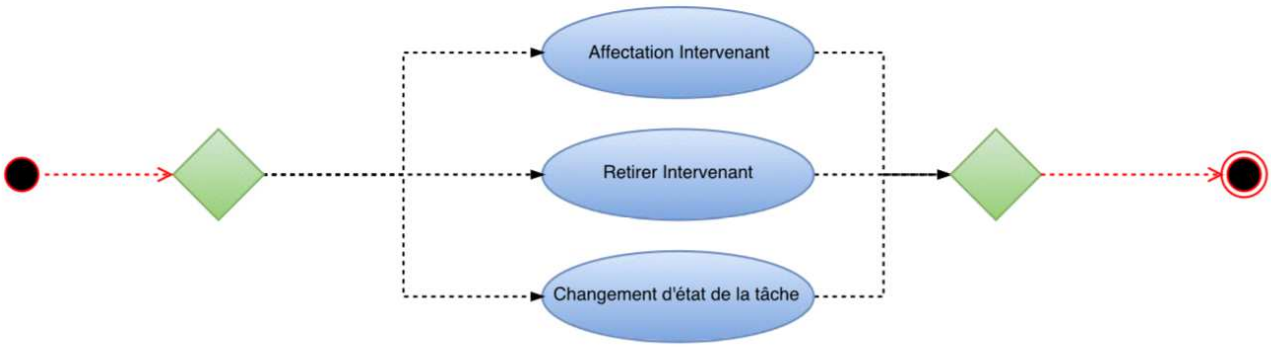


Contrôleur



Responsable



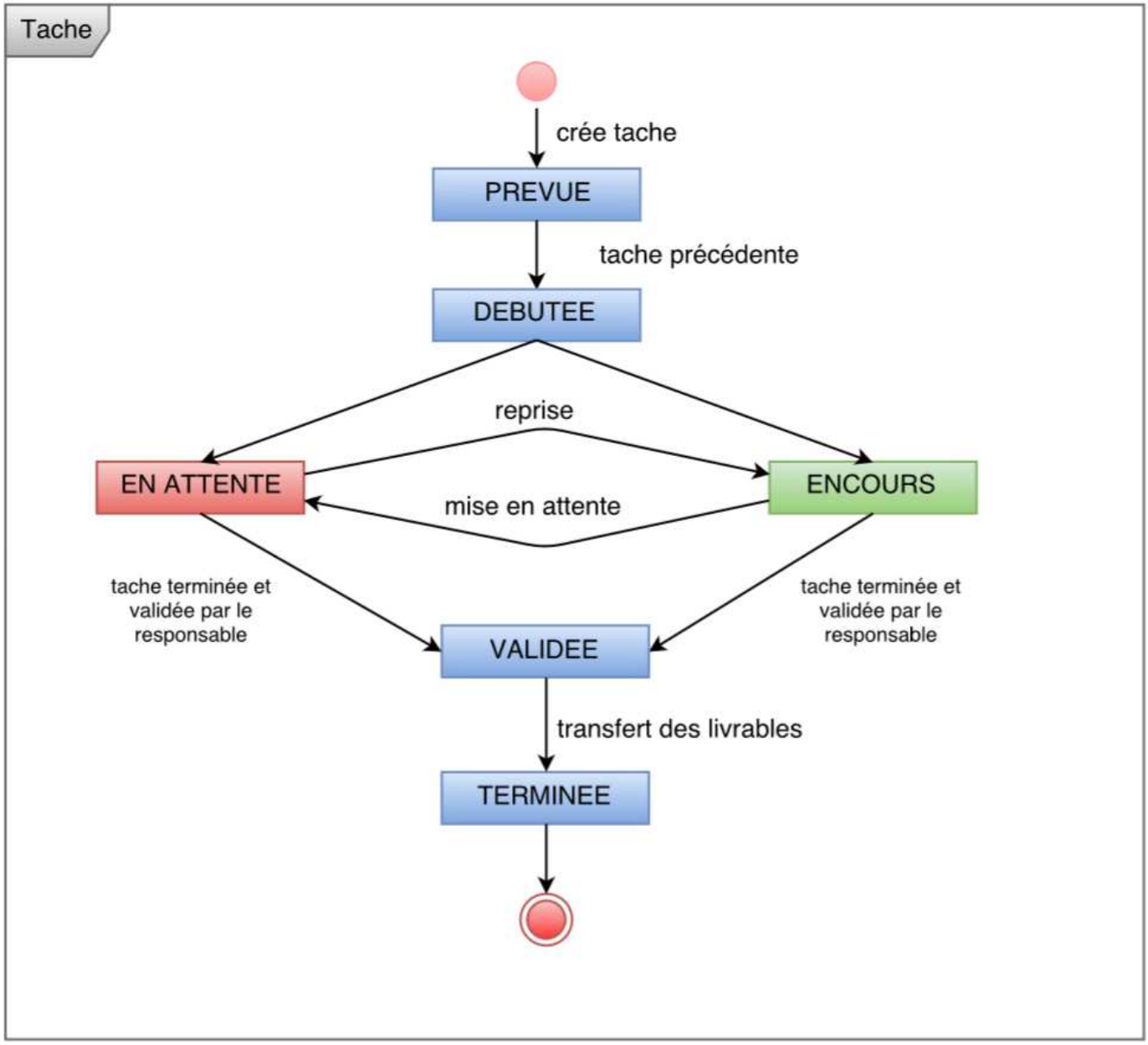


Ces diagrammes d'activité permettent de modéliser le comportement du système, dont leurs actions et leurs conditions d'exécution.

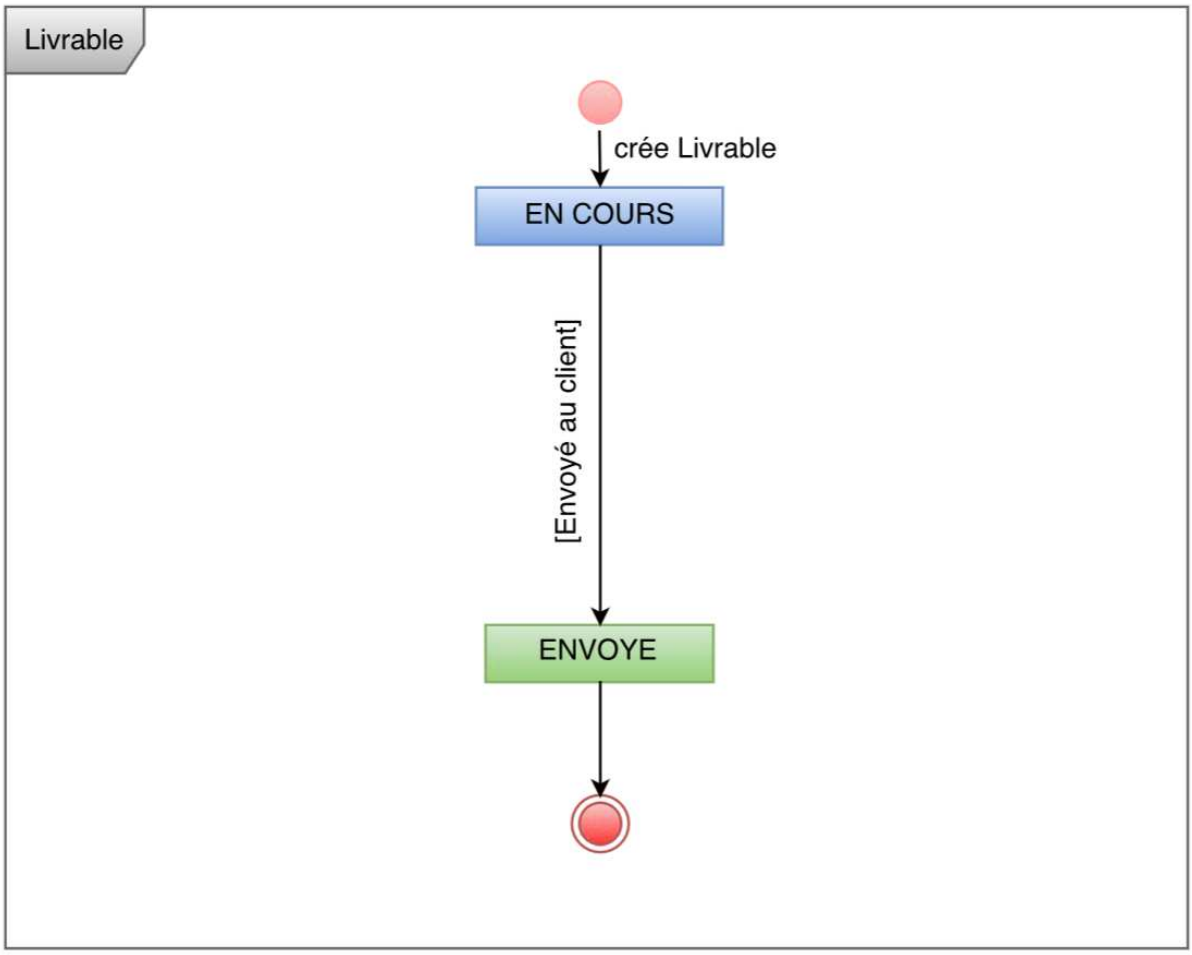
Ces diagrammes d'activités permettent donc de grouper et de dissocier des actions des chefs de projet, contrôleurs et responsables.

Diagramme d'états :

Tâche



Livrable



Ces deux diagrammes d'états montrent les différents états des objet taches et livrable ainsi que les transitions entre ceux-ci.

Conclusion:

La partie du projet s’intéressant à la génération du code a été traitée entièrement, il resterait à traiter l'interface graphique ou textuelle. Nous n'avons pas rencontrer de difficulté particulière durant la conception de ce projet. Cela nous à cependant permis d'apprendre à générer du code à partir d'un UML et inversement. Et à mieux comprendre toute les étapes obligatoire pour la conception d'un logiciel complet depuis le début.