CAMPUS
D'ENSEIGNEMENT SUPÉRIEUR
ET DE FORMATION PROFESSIONNELLE

Programmation Système

Diagrammes UML 1

CESI Saint-Nazaire | A3 FISE INFO

Table des matières

Introduction

La première étape lors du développement d'une application en groupe, est de vérifier que tout le monde a un commun objectif et une direction identique. Pour cela, il est intéressant de mettre à l'écrit les différentes fonctionnalités et spécifications de l'application.

C'est pourquoi, dans ce livrable, nous avons recueilli les différents diagrammes réalisés :

- Diagramme de classe
- Diagramme de cas d'utilisation
- Diagramme de séquence
- Diagramme d'activité

Chaque diagramme sera détaillé, afin que son utilité et son insertion dans le projet soit la plus explicite possible.

Diagramme de classe

Introduction

Le diagramme de classe est le premier qui vient à l'esprit quand il est fait mention d'UML. Ce fut la première étape de notre développement, et ce pour une bonne raison : le diagramme de classe nous permet de visualiser les relations entre les différentes classes que nous allons coder.

Notre code ayant pour but d'être la manière la plus simple de répondre à un problème complexe, et le C# étant un langage de programmation orienté objet, le diagramme de classe va nous permettre de visualiser la décomposition de notre programme en un ensemble de travaux simples.

Description de notre diagramme

Visible ci-dessous, notre diagramme de classe possède quelques caractéristiques notables :

- Deux Singleton sur DailyLog et StateLog
- Les différentes classes sont reliés par des « flèches » indiquant leurs relations entre-elles
- Les classes sont divisées en trois parties :
    - Le **nom** de la classe
        - Le nom est écrit normalement dans le cas d'une classe classique
        - Le nom est écrit *en italique* dans le cas d'une classe abstraite
    - Les **attributs** de la classe
        - + dans le cas d'un accès public
        - # dans le cas d'un accès protégé
        - ~ dans le cas d'un accès package
        - - dans le cas d'un accès privé
    - Les **méthodes** de la classe

Proposition de diagramme de classe

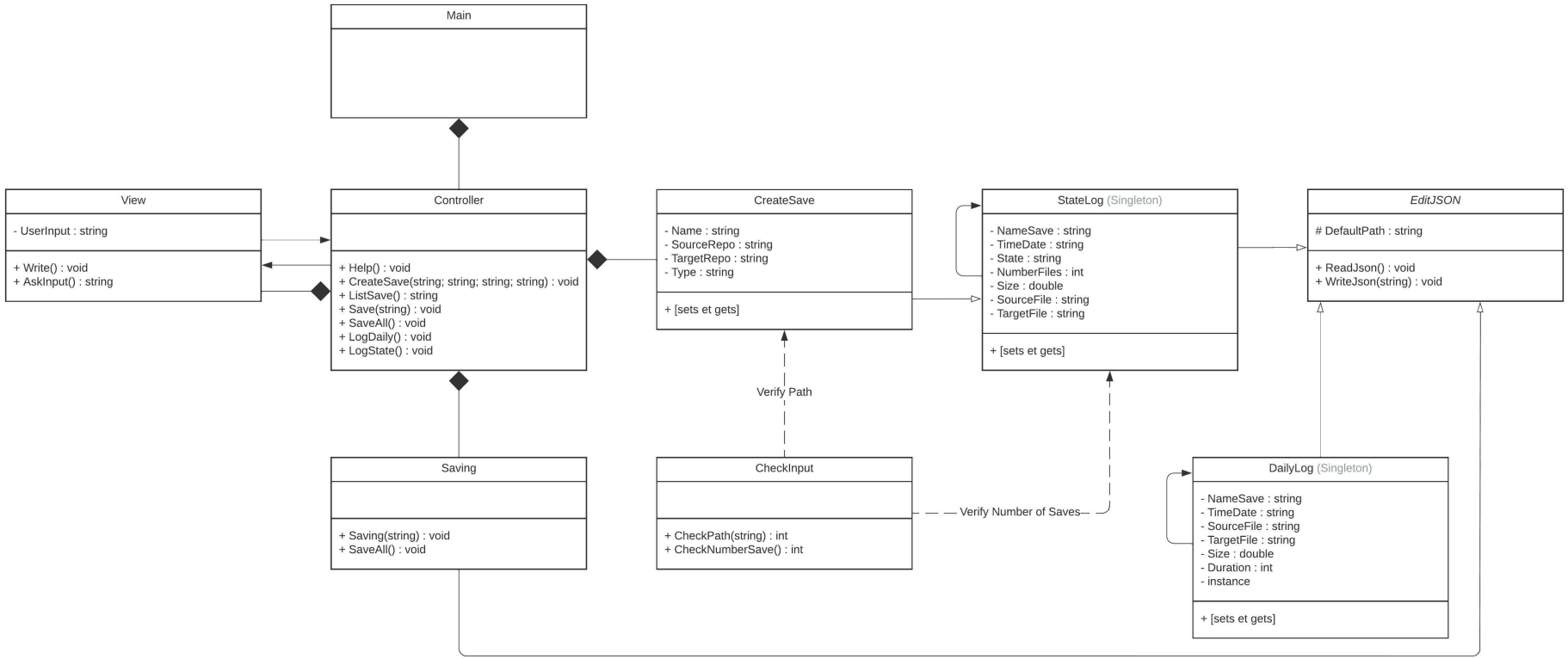

Diagramme de cas d'utilisation

Introduction

Les diagrammes de cas d'utilisation (DCU) sont une catégorie de diagrammes UML utilisées pour une représentation du comportement fonctionnel d'un système logiciel. Autrement dit, ils permettent de schématiser l'expression des besoins d'une application ou plus globalement d'un système. Ils sont utiles pour des présentations des besoins auprès de la direction ou encore auprès des acteurs d'un projet.
On comprend d'autant plus son intérêt au sein de ce projet. En effet, afin de garantir non seulement la compréhension du code et de l'application par la direction mais aussi sa réutilisabilité et la reprise du projet par de futures équipes, il est indispensable de mettre en place un diagramme cas d'utilisation. Il est à noter qu'il existe quatre différents types d'acteur :

- **L'initiateur** : qui peut être considéré comme étant un acteur qui active le système et déclenche un cas bien précis
- **Serveur** : peut être considéré comme étant un acteur aidant le système à assumer ses responsabilités
- **Receveur** : peut être considéré comme étant un acteur recevant les informations du système
- **Facilitateur** : acteur dont les actions sont effectuées au bénéfice d'un autre acteur.

Description de notre diagramme

Sur le diagramme cas d'utilisation ci-dessous, on peut constater que l'on a notre utilisateur (User) et notre « user's computer » qui représentent les différents acteurs de notre programme.
En effet, de façon générale, notre utilisateur crée un fichier (texte par exemple) et par ricochet crée une sauvegarde de ce fichier. Cette sauvegarde devra être stockée au soit sur des disques locaux, soit sur des disques externes ou encore sur des lecteurs réseaux.
L'ensemble de ces trois répertoires est représenté par le « user's computer ». Mais avant d'être stockée, le user's computer se rassurera tout d'abord :

- L'existence de la sauvegarde
- Du fait qu'il ne pourra stocker au plus que cinq sauvegardes, il vérifiera s'il en existe cinq, plus ou moins.

Par ailleurs, une fois la sauvegarde créée, il est à noter que l'utilisateur pourra lire les différents Log :

- Soit les « Daily Log » où sont stockées les différentes sauvegardes journalières
- Soit les « State Log » lieu où l'on pourra voir l'état d'avancement des différentes sauvegardes (tel que le pourcentage d'achèvement de la sauvegarde…)

Une fois que l'utilisateur les aura lus, il pourra les écrire et les stocker au sein de « user's computer ».

On parle beaucoup de stockage mais il est Indispensable de mentionner qu'une liste aura au préalable été créée au sein de « user's computer » afin de stocker les différentes sauvegardes journalières et/ou l'état de celles-ci.

Proposition de diagramme de cas d'utilisation

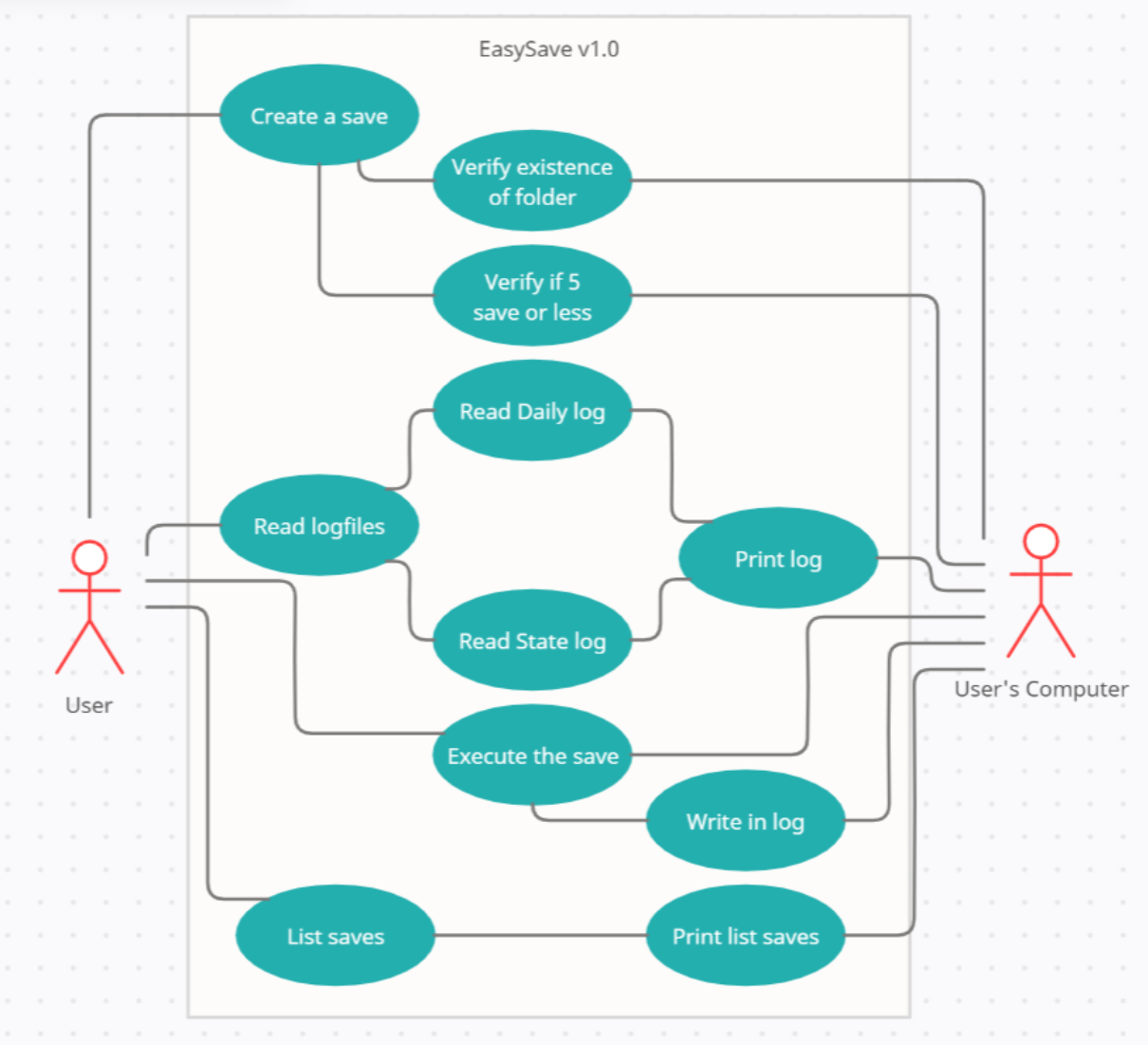

Diagramme de séquence

Introduction

Les diagrammes de séquence sont une catégorie de diagrammes UML permettant de montrer et mettre en évidence les différentes interactions entre les objets de notre application. Autrement dit, c'est la schématisation temporelle des messages (Appelle d'une méthode permettant de faire interagir les différents objets) entre les différents objets. Il est aussi considéré comme étant le cycle de vie d'un ensemble d'objets interagissant entre eux dans un cas d'utilisation précis.

Plusieurs types de messages peuvent transiter entre les différents objets et les acteurs à savoir :

- **Message simple** : le message n'a pas de spécificité particulière d'envoi et de réception.
- **Message avec durée de vie** : l'expéditeur attend une réponse du récepteur pendant un certain temps et reprend ses activités si aucune réponse n'a lieu dans un délai prévu.
- **Message synchrone** : l'expéditeur est bloqué jusqu'au signal de prise en compte par le destinataire. Les messages synchrones sont symbolisés par des flèches barrées.
- **Message asynchrone** : le message est envoyé, l'expéditeur continue son activité que le message soit parvenu ou pris en compte ou non. Les messages asynchrones sont symbolisés par des demi-flèches.
- **Message dérobant** : le message est mis en attente dans une liste d'attente de traitement chez le récepteur.

Tout comme le DCU, on comprend très vite son intérêt dans le cadre de la réalisation d'un projet ; ceci du fait qu'il permet de lister les différentes méthodes publiques, leur classe et leur rôle.

Proposition de diagramme de séquence

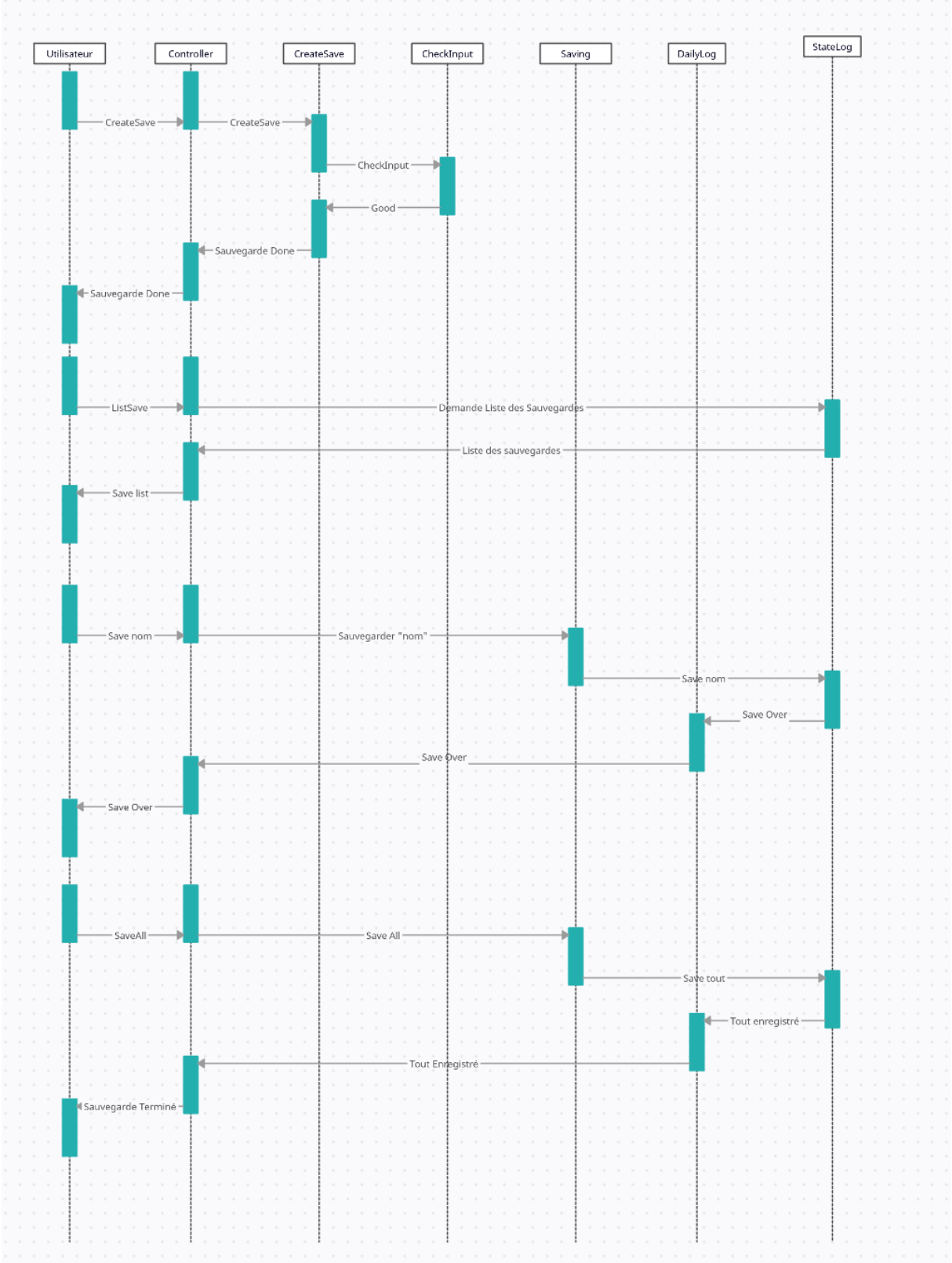

Diagramme d'activité

Introduction

Autre diagramme d'UML, le diagramme d'activité a pour but de diviser la représentation de notre application de manière macroscopique. On voit donc l'utilisateur et les actions de celui-ci lors de l'utilisation de l'application. Il montre le comportement dynamique de notre système.

Proposition de diagramme de classe