LIVRABLE 2

Version 1.1 et 2.0

19 FEVRIER 2024

CESI
Louis JAGUENEAU, Nathan LORIT, Julien DESPREZ, Paul PESCHEL

Table des matières

Table des figures

Contexte :

L'objectif principal est de fournir une application console en utilisant .NET Core pour gérer un ensemble de fonctionnalités concernant une gestion limitée de sauvegarde de données, ainsi qu'un accès aux fichiers de logs.

Ce projet est disponible sur la plate-forme de dépôt GitHub à l'adresse suivante : EasySave. Il est possible de soit le télécharger directement au format .zip, soit de cloner un projet avec le lien URL depuis le logiciel Visual Studio 2022. Pour ce faire, il est nécessaire d'installer les bibliothèques recommandées. De plus, si l'option pour le téléchargement au format .zip est choisie, il suffit d'ouvrir un projet ou une solution depuis Visual Studio 2022.

Objectif du livrable :

Ce livrable a pour but d'améliorer l'application que nous avons rendu au livrable 1. Cette amélioration se fait via l'ajout de nouvelles fonctionnalités :

- **Interface graphique :** Abandon du mode Console. L'application doit désormais être développée en WPF sous .Net Core
- **Sauvegarde illimitée :** Le nombre de travaux de sauvegarde est désormais illimité
- **Chiffrement :** Le logiciel devra être capable de crypter les fichiers en utilisant le logiciel Crypto Soft (réalisé durant le prosit 4). Seuls les fichiers dont les extensions ont été définies par l'utilisateur dans les paramètres généraux devront être cryptés.
- **Les logs :** Le fichier Log journalier doit contenir une information supplémentaire
    - 0 : pas de cryptage
    - >0 : temps de cryptage (en ms)
    - <0 : code erreur
- **Logiciel métier :** Si la présence d'un logiciel métier est détecté, le logiciel doit interdire le lancement d'un travail de sauvegarde. Dans le cas de travaux séquentiels, le logiciel doit terminer le travail en cours et s'arrêter avant de lancer le prochain travail. L'utilisateur pourra définir le logiciel métier dans les paramètres généraux du logiciel. (Remarque : l'application calculatrice peut substituer le logiciel métier lors des démonstrations)

Solution

Diagramme de cas utilisation :

Avant de passer à l'ajout des différentes fonctionnalités énumérées précédemment, nous avons mis à jour les diagrammes UML.

Le diagramme de cas d'utilisation que nous avons élaboré est un outil fondamental pour décrire les interactions entre les acteurs (utilisateurs ou systèmes externes) et le système en question. Son objectif principal est de représenter les différentes fonctionnalités offertes par le système du point de vue de l'utilisateur.

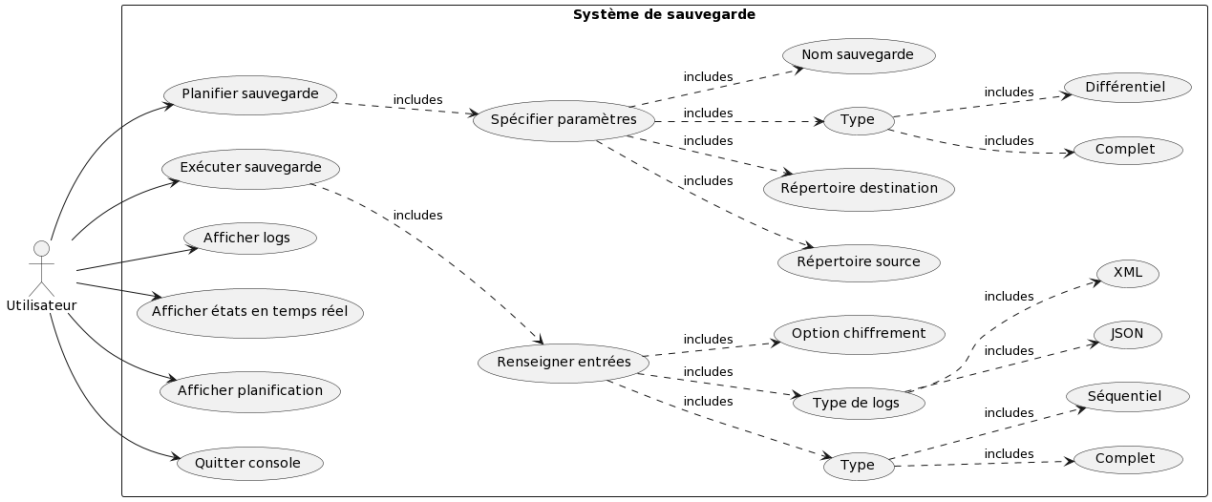

Figure 1 : Diagramme de cas utilisation

Diagramme de séquence :

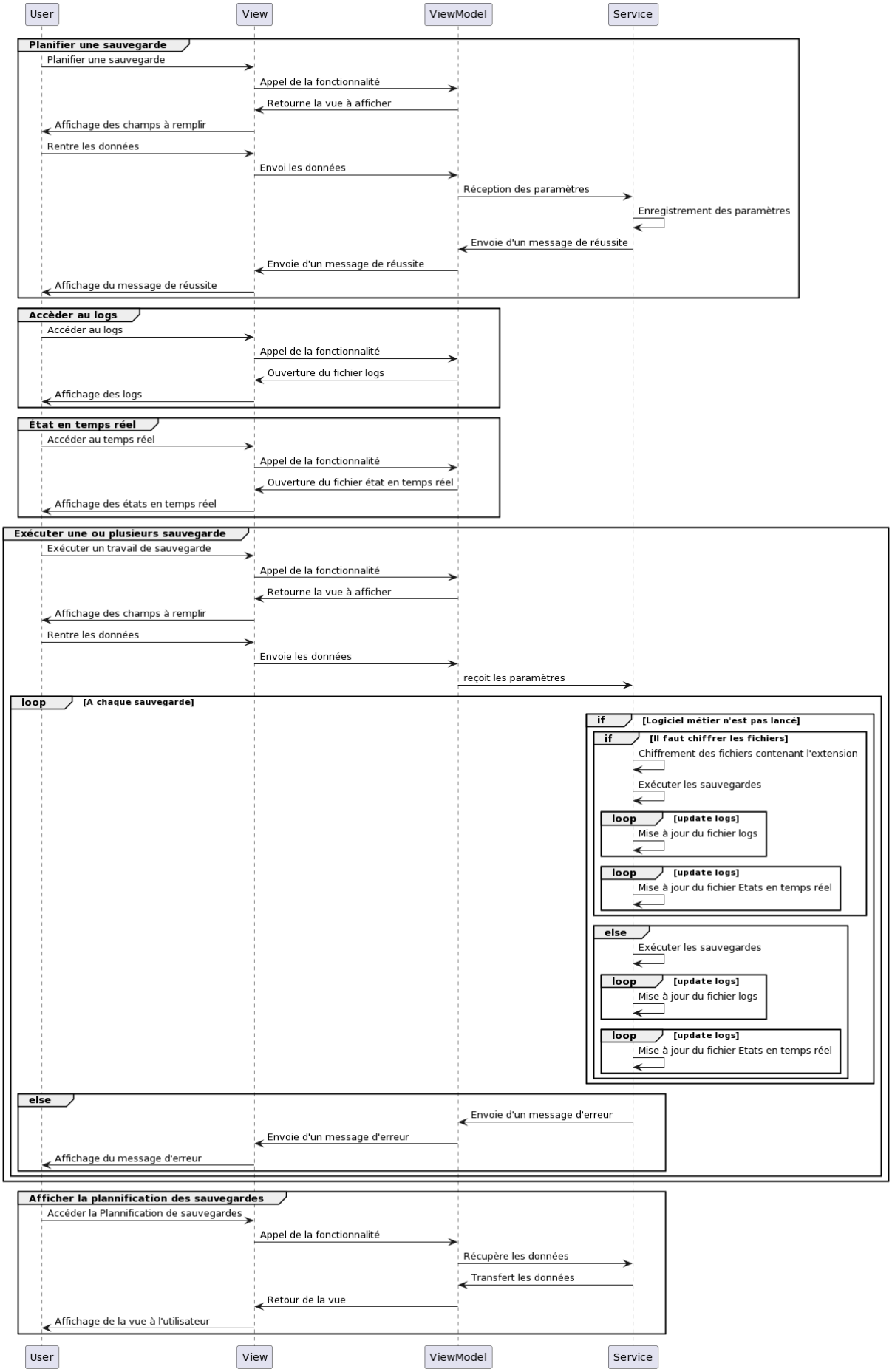

Figure 2: Diagramme de séquence

Diagramme d'activité :

Le diagramme d'activité que nous avons conçu est un outil utilisé pour modéliser les flux de contrôle et les comportements des processus ou des systèmes logiciels. Son objectif principal est de représenter graphiquement les différentes activités d'un processus, ainsi que les transitions entre ces activités.

Ce diagramme est particulièrement utile pour plusieurs raisons. Tout d'abord, il permet de visualiser de manière claire les étapes d'un processus, ainsi que les décisions prises à chaque étape. Cela facilite la compréhension du fonctionnement du processus et permet d'identifier les éventuels goulets d'étranglement ou les points d'amélioration.

De plus, le diagramme d'activité est un outil utile pour la documentation et la communication des processus. Il offre une représentation visuelle intuitive des étapes à suivre, ce qui facilite la transmission des connaissances et la formation des utilisateurs.

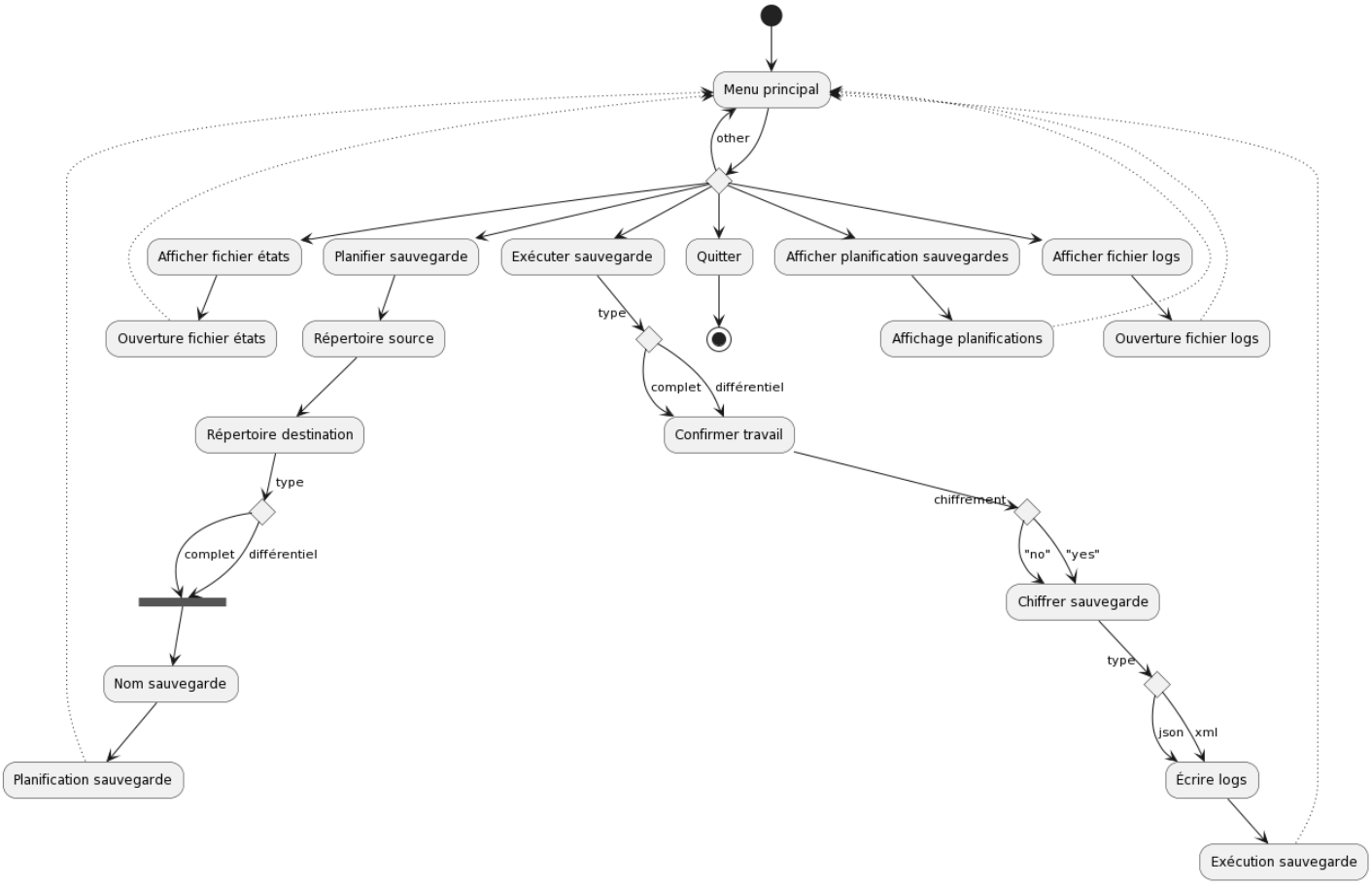

Figure 3 : Diagramme d'activité

Diagramme de classe :

Le diagramme de classe que nous avons réalisé est un outil essentiel pour représenter la structure statique d'un système logiciel. Son but principal est de visualiser les différentes classes du système, ainsi que leurs relations et attributs.

Ce diagramme offre plusieurs avantages. Tout d'abord, il permet une meilleure compréhension de l'architecture du logiciel en identifiant clairement les entités et leurs interactions. Cela facilite la communication entre les membres de l'équipe de développement et favorise une conception plus cohérente et modulaire du système.

En outre, le diagramme de classe joue un rôle dans la phase de conception du logiciel. Il aide les développeurs à organiser et à structurer leur code de manière efficace, en définissant les différentes classes, leurs attributs et méthodes, ainsi que les relations. Cela permet de garantir une meilleure évolutivité, maintenabilité et réutilisabilité du code.

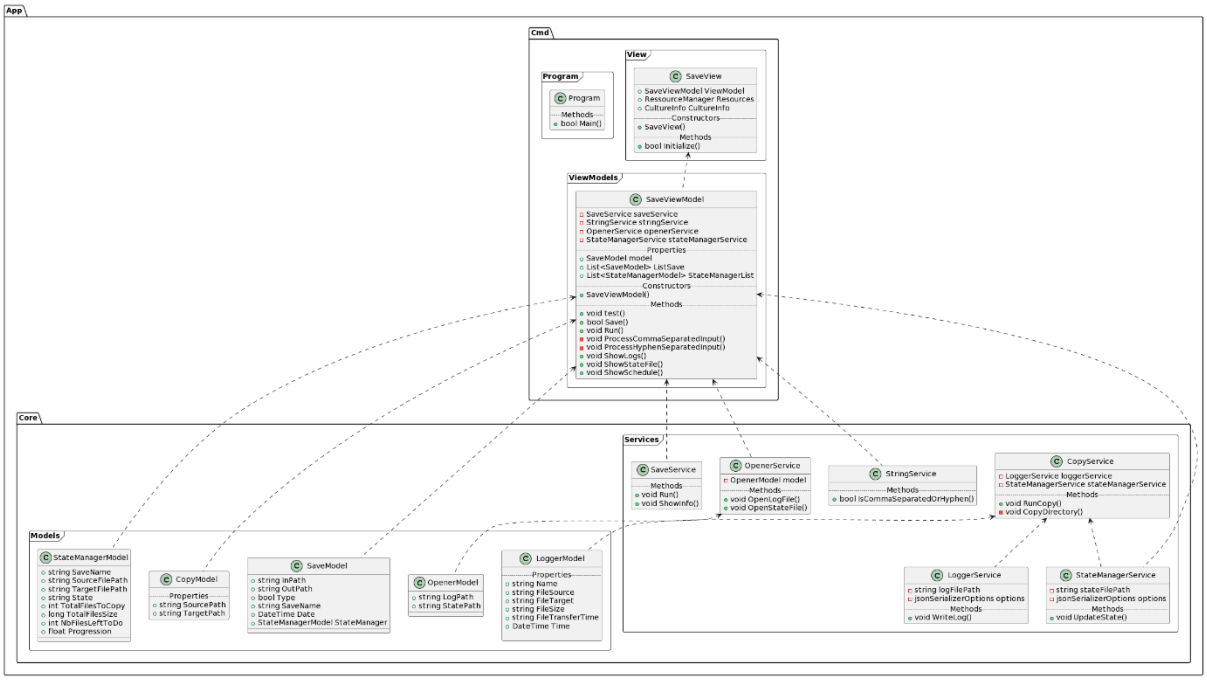

Figure 4 : Diagramme de classes

Conclusion :

Pour ce deuxième livrable, l'objectif était de mettre en place de nouvelles fonctionnalités afin d'améliorer l'application. Ces nouvelles fonctionnalités ont été mise en place et sont aujourd'hui fonctionnel.

L'objectif maintenant est de prendre connaissance d'un troisième cahier des charges, en vu d'une seconde amélioration de l'application.