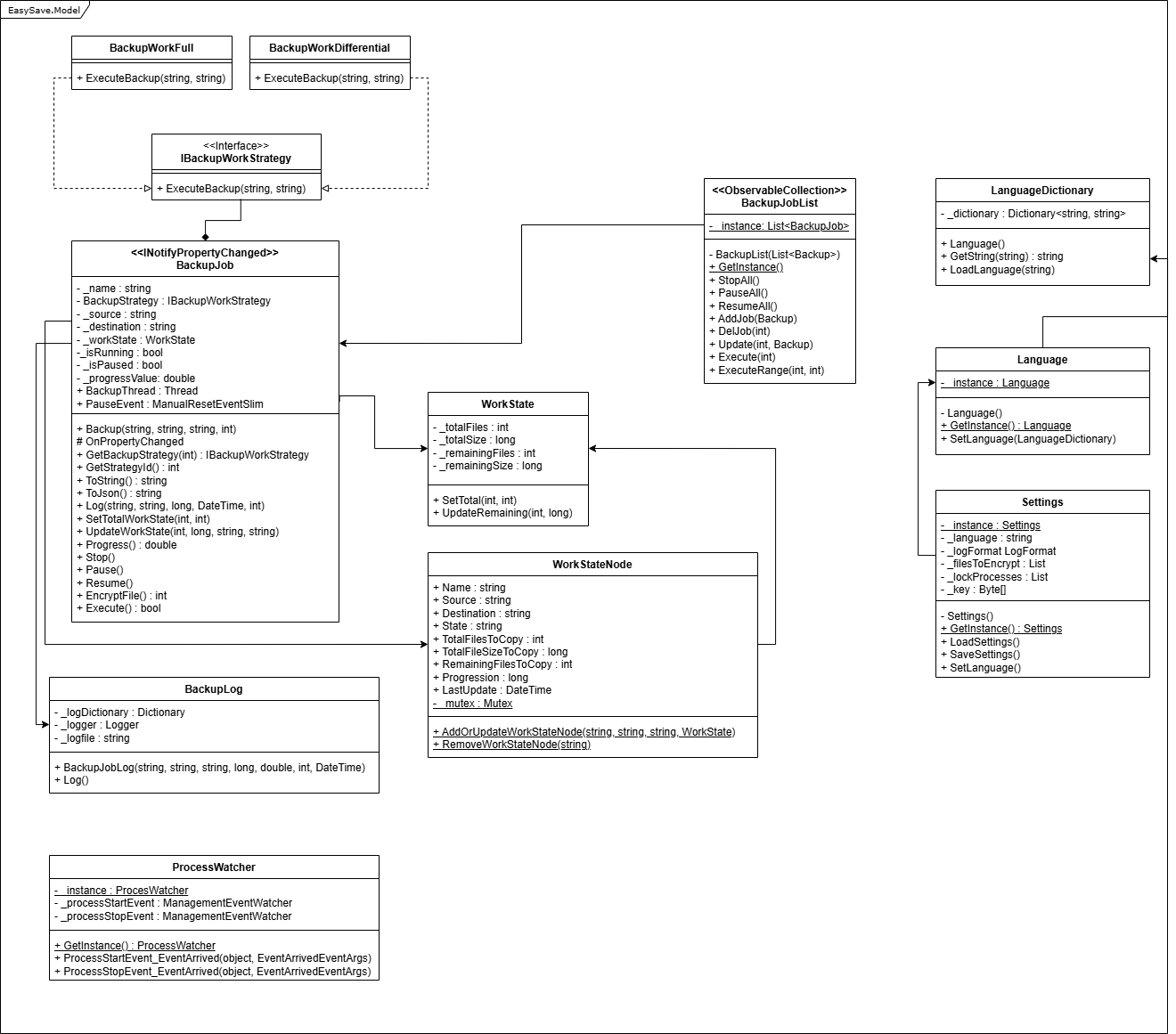
EasySave

# UML

## Diagramme de classe

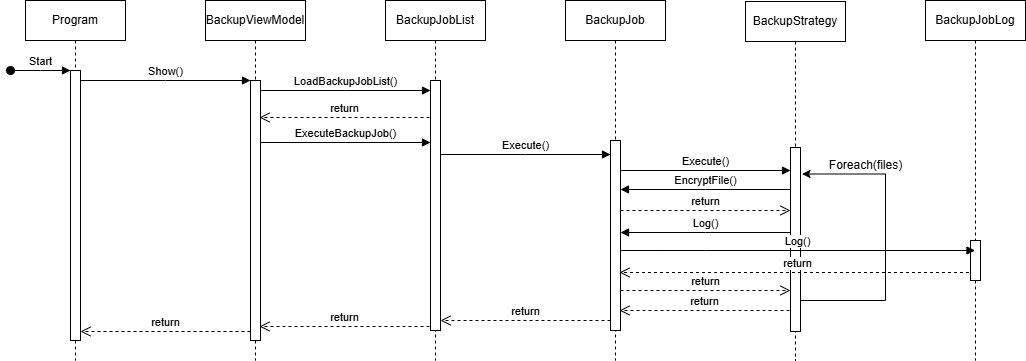


Nous avons utilisé un design pattern Strategy pour gérer les différents types de sauvegarde (complète, différentielle) avec la méthode commune Execute qui lance la sauvegarde.

Nous avons aussi utilisé un design pattern Singleton pour la langue utilisé par le programme pour s’assurer de n’avoir qu’une seule langue lors de l’exécution du programme, et de même pour les Settings, le ProcessWatcher et la BackupList.

Le tout fonctionne sur un modèle MVVM avec WPF (non visible sur le diagramme car indigeste et n’explique pas les fonctions) afin de bien séparer les différentes couches et d’avoir un code modulable.

## Diagramme de séquence



Voici l’exécution d’un travail de sauvegarde, du chargement de la liste de travaux jusqu’à la copie de chaque fichier, leur chiffrement et l’écriture des logs.