

---

## EasySave

*Génie Logiciel,  
Livrable 1: UML*

**7 mai 2025**

Groupe 1

---

Akuwa Mawunu Nathalie

Leclerc Appéré Loan

Cavalca Kyllian



## Sommaire

<b>Introduction.....</b>	<b>3</b>
<b>UML.....</b>	<b>4</b>
<b>Use-Case.....</b>	<b>5</b>
<b>Séquence .....</b>	<b>6</b>
<b>Classe .....</b>	<b>7</b>

# Introduction

L'entreprise Prosoft lance le développement d'un nouvel outil dédié à la création de backup : EasySave. Ce logiciel a pour but de s'intégrer dans la suite applicative de l'entreprise et de proposer une solution simple, performante et évolutive dans la gestion des sauvegardes de fichiers.

Ce projet est découpé en plusieurs phases successives de développement, et donc plusieurs versions du logiciel (1.0, 2.0, 3.0, ...). Chacune de ces versions répondra à un ensemble d'exigences fonctionnelles et techniques en fonction des différents cahiers des charges fournis tout au long du projet. Etant donné que le logiciel est destiné à être commercialisé, celui-ci s'inscrit dans la politique tarifaire de Prosoft et donc il doit répondre aux attentes de qualité, de maintenabilité et de réutilisabilité imposées.

Nous sommes, l'équipe de développement, donc en charge de la conception, la programmation, la documentation, tant utilisateur que support, la gestion des versions et l'historique du projet.

Ce livrable présente la conception logique du projet grâce aux différents diagrammes UML, Use-Case, séquence et classe de l'application EasySave.

# UML

Dans le cadre du développement du logiciel EasySave, nous avons utilisé le langage UML (Unified Modeling Language) afin de formaliser et de structurer la conception du projet de manière claire. L'UML est largement utilisé dans l'ingénierie logicielle puisqu'il permet de représenter les différents aspects d'un système informatique, tant sur le plan fonctionnel que technique.

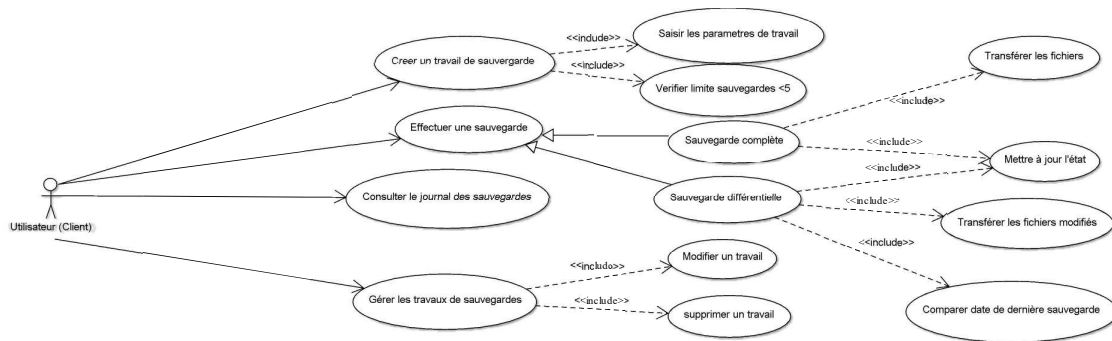
L'UML à deux principaux objectifs :

- Faciliter la compréhension d'un système par l'ensemble des acteurs du projet
- Servir de base solide pour le développement, en clarifiant les besoins exprimés dans le cahier des charges tout en les transformant en composants logiciels concrets et cohérents.

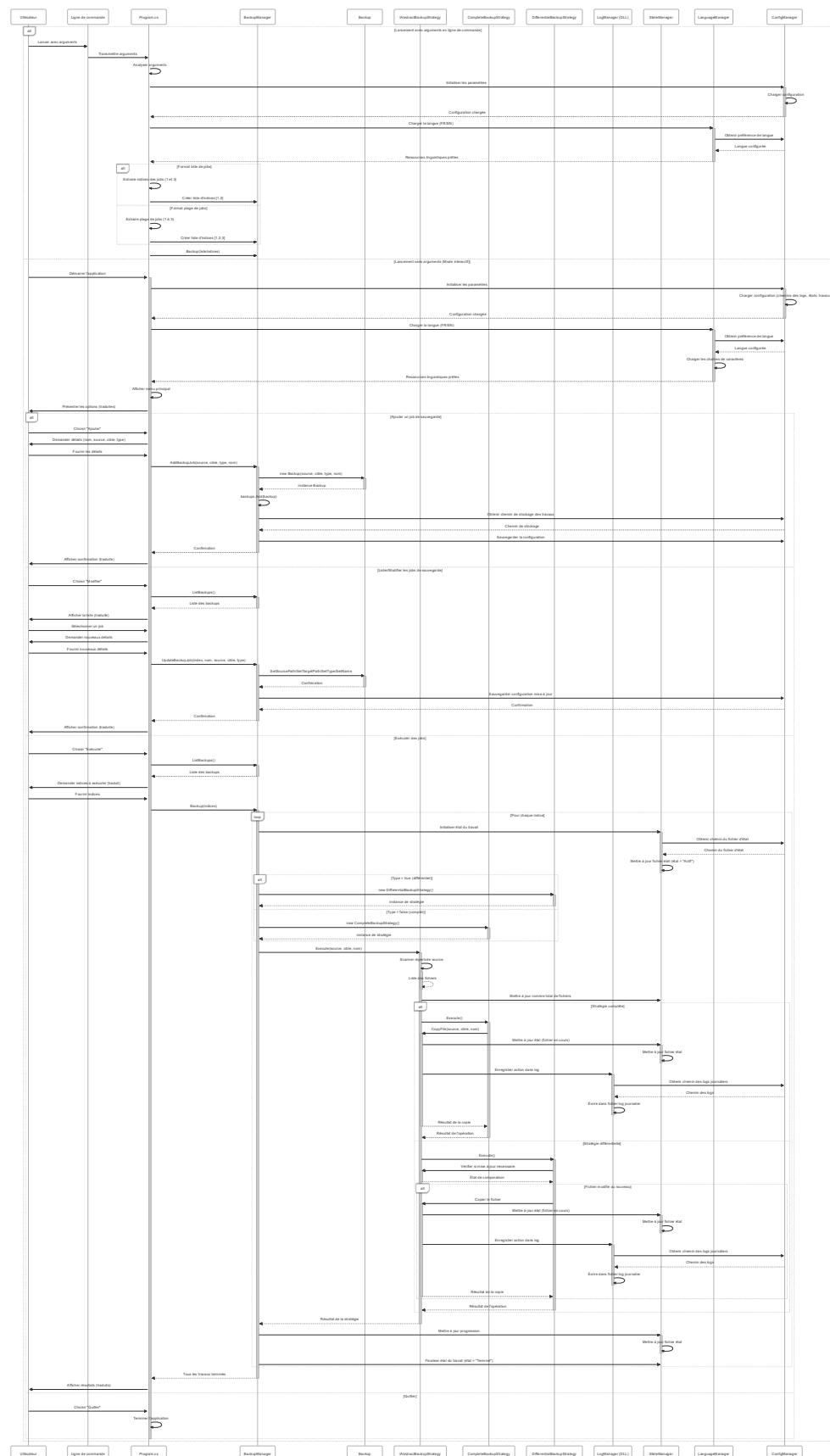
Dans le cadre de ce projet, nous avons réalisés plusieurs types de diagrammes UML afin de décrire le comportement du logiciel, son architecture et les interactions entre les différents éléments qui le composent :

- **Diagramme de cas d'utilisation (Use Case)** : il permet de représenter les différentes fonctionnalités du logiciel du point de vue de l'utilisateur final et de définir les interactions entre les acteurs et le système.
- **Diagrammes de séquence** : ils illustrent les échanges dynamiques entre les objets du système dans le temps, pour une fonctionnalité donnée.
- **Diagramme de classes** : il modélise la structure statique du système, en représentant les différentes classes, leurs attributs, leurs méthodes et les relations entre elles.

# Use-Case



## Séquence



# Classe

