

|  |
| --- |
| Livrable 2 : EasySave V1.1 et V2.0 |
|  |
| 19 FÉVRIER  Service : DSI ProSoft  Créé par : Hatice Yildiz, Lisa Jung, Aya Salem, Ayse Deniz |

Une image contenant capture d’écran, Police, Graphique, texte

Description générée automatiquement

Table des matières

[1. CONTEXTE : 3](#_Toc190875968)

[1.1 DSI ProSoft : 4](#_Toc190875969)

[2. INTRODUCTION : 5](#_Toc190875970)

[Version 1.1 : 6](#_Toc190875971)

[3. DIAGRAMMES : 6](#_Toc190875972)

[3.1 Diagramme Use Case : 6](#_Toc190875973)

[3.2 Diagramme d’activité : 8](#_Toc190875974)

[3.3 Diagramme de classes : 11](#_Toc190875975)

[3.4 Diagramme de séquence : 14](#_Toc190875976)

[4. Documentation de la Dynamic Link Library V1.1 : 28](#_Toc190875977)

[5. Documentation Utilisateur V1.1: 28](#_Toc190875978)

[6. Documentation Technique V1.1 : 28](#_Toc190875979)

[7. Release note V1.1 : 28](#_Toc190875980)

[Version 2.0: 29](#_Toc190875981)

[8. DIAGRAMMES : 29](#_Toc190875982)

[8.1 Diagramme Use Case : 29](#_Toc190875983)

[8.2 Diagramme d’activité : 31](#_Toc190875984)

[8.3 Diagramme de classes : 35](#_Toc190875985)

[8.4 Diagramme de séquence : 38](#_Toc190875986)

[9. Documentation CryptoSoft V2.0: 57](#_Toc190875987)

[10. Documentation de la Dynamic Link Library V2.0: 57](#_Toc190875988)

[11. Documentation Utilisateur V2.0: 57](#_Toc190875989)

[12. Documentation Technique V2.0 : 57](#_Toc190875990)

[13. Release note V2.0 : 57](#_Toc190875991)

[14. Conclusion 58](#_Toc190875992)

1. CONTEXTE :

Le projet EasySave, développé dans le cadre de l’éditeur de logiciels ProSoft, vise à concevoir une solution performante et évolutive pour la gestion et l’automatisation des sauvegardes de fichiers. Ce logiciel, structuré en plusieurs versions successives, repose sur une architecture modulaire favorisant la maintenance, l’évolutivité et la compatibilité avec les futures améliorations. En intégrant des fonctionnalités telles que la sauvegarde complète et différentielle, un suivi détaillé via des logs journaliers et une interface utilisateur optimisée, EasySave répond aux besoins des entreprises en matière de protection des données. Le développement suit les bonnes pratiques du génie logiciel, avec une gestion rigoureuse des versions, l’utilisation de C# et .NET 8.0, ainsi qu’une documentation garantissant une prise en main rapide et efficace par les utilisateurs et le support technique.

## DSI ProSoft :

Nous sommes une équipe de 4 membres, représentant la DSI ProSoft, qui vise à concevoir une solution performante et évolutive pour la gestion et l’automatisation des sauvegardes de fichiers autour d’un nouveau logiciel qui sera implémenté « EasySave ».

Lisa JUNG

Une image contenant personne, Visage humain, sourire, mur

Description générée automatiquement

Ayse DENIZ

Une image contenant personne, Visage humain, sourire, habits

Description générée automatiquement

Hatice YILDIZ

Une image contenant Visage humain, personne, habits, sourire

Description générée automatiquement

Aya SALEM

Une image contenant personne, Visage humain, habits, verres

Le contenu généré par l’IA peut être incorrect.

1. INTRODUCTION :

Le livrable 2.0 d’EasySave intègre les améliorations apportées par la version 1.1, qui permet aux utilisateurs de choisir le format des logs entre JSON et XML. Cette mise à jour répond à une demande client tout en maintenant l’architecture existante. La version 2.0 marque une évolution majeure avec l’introduction d’une interface graphique (WPF), la suppression de la limite de travaux de sauvegarde, l’intégration de CryptoSoft pour le cryptage des fichiers, ainsi qu’une meilleure gestion des logiciels métiers qui peuvent bloquer les sauvegardes en cours. De plus, le fichier de log journalier évolue en intégrant des données sur le temps de cryptage et les erreurs. Ces améliorations rendent EasySave plus ergonomique, performant et adapté aux besoins professionnels des utilisateurs.

# Version 1.1 :

1. DIAGRAMMES :

## Diagramme Use Case :

Un diagramme de cas d'utilisation (Use Case Diagram) est une représentation graphique des interactions entre les utilisateurs (acteurs) et un système. Il illustre les fonctionnalités principales offertes par le système sous forme de cas d'utilisation, permettant ainsi de comprendre qui utilise le système et comment.

Ce type de diagramme est essentiel en phase de conception, car il permet de clarifier les besoins et de définir les interactions entre le logiciel et ses utilisateurs sans entrer dans les détails techniques.

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

Explication :

Ce diagramme de cas d'utilisation illustre les principales fonctionnalités du logiciel EasySave et les interactions possibles entre l'utilisateur et le système. Elle est composée de :

Acteur :

Utilisateur : L’acteur représenté par un bonhomme est celui qui interagit avec le logiciel pour effectuer différentes opérations de sauvegarde.

Cas d’utilisation principaux :

* Réaliser une sauvegarde

Fonctionnalité centrale du logiciel, permettant à l’utilisateur de gérer ses sauvegardes.

* Créer un travail de sauvegarde

Avant d’exécuter une sauvegarde, l’utilisateur doit créer un travail de sauvegarde, définissant :

* Un nom
* Un répertoire source
* Un répertoire cible
* Un type de sauvegarde (Complète ou Différentielle)
* Exécuter une ou plusieurs sauvegardes

Une fois le travail de sauvegarde configuré, l’utilisateur peut lancer manuellement une ou plusieurs sauvegardes.

Il peut choisir de lancer toutes les sauvegardes en séquence ou de sélectionner des sauvegardes spécifiques.

* Consulter l’état temps réel des sauvegardes

Permet à l’utilisateur de suivre en direct la progression des sauvegardes.

Les informations incluent :

* Nom de la sauvegarde
* Source
* Destination
* Le nombre total de fichiers
* La taille totale des données
* Le nombre de fichiers restants
* Le progrès de la sauvegarde
* Consulter le fichier log journalier

L’utilisateur peut consulter les logs journaliers générés par le logiciel.

Ces fichiers enregistrent toutes les actions de sauvegarde et leurs détails (date, heure, état de la sauvegarde, erreurs éventuelles, temps, source, destination).

## Diagramme d’activité :

Un diagramme d'activité est une représentation graphique des flux de contrôle et des processus d'un système. Il illustre la succession des actions et des décisions effectuées au sein du logiciel, souvent sous la forme d'un workflow.

Ce type de diagramme est utilisé pour modéliser le comportement dynamique d’un système en mettant en évidence les différentes étapes d'un processus, les choix possibles, ainsi que les conditions de transition entre les actions. Il est particulièrement utile pour visualiser les algorithmes, les flux de travail métier, ou encore les enchaînements logiques d'exécution.

Une image contenant texte, diagramme, ligne, Parallèle

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme activité :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqVV91u2zYUfhWCQJEEiJPKdtvYFxuCprtYkwZD1mGYHQicRNvcZNLlT5vM8bv0cnkOv1gPKZmibCpOgiSWyHO-88PvO7KWOBM5xUM8KcS3bEakRr9fjDmCn4v1499GH5YfR6jT-QldEp7ROeU6JYtFwTKimeAjtypRcRAs3pYYUQcH9X4m2F1aED41dGRvFAMIgsqVyj00cl5kMmHZjMoUMI13I1kQsvz_6hX6zTCtqSzvG44OqtoeVZ_R9Dd71v4Xxg_h72grynu5fiyryg8Mp0gR85VOicxpW2Dvkda2I7sIgZoImy5EHBzSDbHlp1zMQ6hyFaAQbKA8Ahj1CxGVMDKjUVCAc5tPAe-4h9g5VZrxnQYEAQKLp6LEgTy5rIm-XzSq2BAGemO3ougtvg73j_WjZJOKI-mCSDJf_68lVctLqlBwD03iusMKhTLBJ0LOYennVUidVigb6OHasAd0RZUiU5pmkfNPrz-OzitqoazmYFgQuv54-5xgnwSvg9Wb6ZzwL4ZwXQcy_DTfKhQO4dQWixjPhJQ006ohxD1F7Kq66RZJp1VEY-5V-eFu_ZiZF8mydgkZU62W0nTF15tqe-K4cYVyg-yJE21PoxBTRL7arIsD6iOUHtGIwXAsUVKAWPoBST02RIEKmJ1cYroK52Xt547315vrTw_oUkxVai9HY_xZs4KpijE-W7s5xrdPIP15dVkBwVUrDux5GN-aBgNdY6AdTGnK7SiJHpHP2LXk3KRzwbhKk5BAlEs6BRhJ6fKD0h0A-2IOiEHOFiWhGoCpyNvDtPWC3FT0vDhhYXuNd_VFTAbFhhQbY6-vISBmzV40UvZ9bUVr01NUGdQ-cLfb_rya3IAqSRIIIl1_T-lGMcFT_bRoCgetvyNv12BcG1goUShsV6Gxgm6DiXD-jzA61CYBfti0gNwwxeCBQOcLhWyXi4Y8G9FKilgoWHYeVhxeWFWQBvSh3TvyJ_cS1FJlUVDYOtpSWUtaJbhjhGUMg8eRqmxcsT73DQltEMbLNkFXt5uzXVA8771Rq9qeHTRW8J6i2rSwJ62YWz3iITdT6M0YK4-j9ZteYOw6M6pWqoLt0ob926ZBAwGyGvVNstW984-C8onjutcc5y-A3zmZGPrukG_P9elziISPOeBjPKfADpbDy8rSAoyxnsG7xRgP4TIn8t8xHvMV2BGjxc09z_BQS0OPsVnkRNMLRqbwXWKzSHOmhbwqX37cO9AxXhCOh0t8h4fJoH9y9qbXf3s2gN_e26R_jO_xsJMMBif9pN8fnCWvu0n_rJesjvF_QgBs9yRJ-q_779503_WSXrfXBRcpzHSGhxNSKOrg_3Km7n71A3772og)

Explication :

L’application EasySave 1.1 démarre en affichant un menu interactif permettant à l’utilisateur de choisir une langue ou de quitter l’application.

* Si l’utilisateur choisit une langue, il accède alors à un second menu interactif.
* Si l’utilisateur choisit de quitter, l’application se ferme immédiatement.

Dans ce second menu, l’utilisateur peut :  
1. Créer une nouvelle sauvegarde  
2. Exécuter une ou plusieurs sauvegardes enregistrées  
3. Consulter le fichier de log journalier  
4. Quitter l’application

Création d’une sauvegarde

Vérification du nombre de sauvegardes enregistrées :

* Si 5 sauvegardes existent déjà, un message informe l’utilisateur qu’il ne peut pas en ajouter de nouvelles.
* Sinon, l’utilisateur est invité à entrer les paramètres de la sauvegarde :
  + Nom de la sauvegarde
  + Source
  + Destination
  + Type de sauvegarde

Vérification des paramètres :

* Si un paramètre est incorrect, un message d’erreur est affiché, et l’utilisateur doit réessayer.
* Si tout est valide, la sauvegarde est enregistrée et un message confirme sa création.

À la fin de cette action, le menu principal est de nouveau affiché.

Exécution d’une ou plusieurs sauvegardes

Vérification des sauvegardes existantes :

* Choisir le format du fichier des logs. (JSON ou XML)
* Si aucune sauvegarde n’existe, un message informe l’utilisateur et lui propose de créer une sauvegarde.
* Sinon, l’utilisateur sélectionne la/les sauvegardes qu’il souhaite exécuter.

Exécution des sauvegardes sélectionnées :

* Affichage de l’état en temps réel pendant l’exécution en fonction du format choisis.
* Une fois terminé, le menu principal est réaffiché.

Consultation du fichier de logs

Vérification de l’existence du fichier log du jour :

* S’il n’existe pas, cela signifie qu’aucune sauvegarde n’a été lancée ce jour-là.
  + L’utilisateur est alors redirigé vers l’exécution d’une sauvegarde.
* S’il existe, le contenu du log est affiché à l’écran en fonction du format.

À la fin de la consultation, le menu principal réapparaît.

Quitter l’application

À tout moment, depuis les menus interactifs, l’utilisateur peut quitter l’application, ce qui met fin à l’exécution du programme.

## Diagramme de classes :

Un diagramme de classe est une représentation graphique des structures statiques d’un système logiciel. Il décrit les classes, leurs attributs, leurs méthodes, ainsi que les relations qu’elles entretiennent entre elles (association, héritage, agrégation, composition, etc.).

Ce type de diagramme est essentiel en programmation orientée objet car il permet de visualiser l’architecture du code avant son implémentation. Il aide à mieux structurer le projet, favorise la réutilisabilité et assure une bonne compréhension du modèle de données du logiciel.

Une image contenant texte, diagramme, Plan, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de classes :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNq1WVtT2zgU_iseLw8BEgayaWAznc5Q6IVOaDsN7e7sssMotjAqtpRKNkvKwm9fSbZ80cUOhM1LiPSdiz6dc3Qk7vyAhNCf-EEMGDtGIKIgOcce_2CQQLYAAfRmILuBEaAhvMunxEcKeK9BcJ0tZikFKYyWb0GQErqsocRn2zuikM83sb1NTXirkro_x7qhI5IsYqiUrGWo1c4xuryEFOIUgfj_ttWEGgbe3MIgUxZ6LKUIR30v_970bggKtzrUM9hq4hdvSqIIUm9S_KFPz6X0NwT_4ZDX5Y-1HBWfAaeZQknrm1vE0l4dq2svdgHFUFPdNPGI6FmD8UfGzrNYKhjjij5dTgFLrUKVsACeoQQ2XG14XGX2GQVhFqSIYCOzT_gcZjFX5lzEGaARTKcARxmIII_93IX2qHzLCQquVtXVquoNjmLEnqar9vXocndEcEpJHEN6p-9SDpjymBZJxb8e8pFTXmXjBx19YS2fZa5p44b0GulpT7V6SeO0AbbEAWeTXevAz4Ay-IHMT3AIb0uVcr0Ipw9rhp7auRaeE8GnoLiASn510E3OjMLYuDFN9Roq-w1pO2sfwQ2KOG8KyXr2DD5cLOKlAs1gDOXqOXhOSLwKZYdSgBl8nUKcObnaLrkSMAtP24onMW_jiClX5W5zIN9iHdN0oVea6pda7dydAszJKBbWGnJHVwQxRN8SmoCUH1ct4IEK-Vxvj_tbBz8Ivo1cfA9wGBexL_Mg3xu3jbpAbs8t8aQ0UAW4OVfkPsIsBZgrmNRxBiwmwTWH8PWEBMdLj8y_873c0Mm9YEWp4Viz8Ovok8I2X2zNtqF0BtNSb8_Uag-IElZWldrG5UMrlZfOIi7D887M5URQqh8YxYJIRgP3dH7suKeXC7dszSfHwf6IPkSdHpzFpyxQtFqfQXrVsdBOmHTArYSkIBY62Bk5Ioulva5UqBn6KZTFxFT1BSYAYW5DAu2KPlMSUciM2Xb6PnBCMIiN9GNiskZAmV_5Wjd0vEjDT_PvnZk44MMM0htImeofTj4VIw9Ggh2GoZrslbA8Pgy9H0mKLpcKxHq1RVoltmXPnIl0rZLRjMWaGkPB10UIlHy3uRyttqnRphiLmRIQ5no3vXqXJcceDPgM3BRuGFjDgLuivOeyhKIfmVlR3sVkDuJvgCIw5xFoXLPitmvWthg1kmnDrBCsAyFm87W1Y85gsmBfoOwJDNhaDZu9ppZ9kQppo4yr80L0GTBUeEuqPqExsvtUSHR4dKH3-2WfXQ090yFv5c55zpsH-DMf9GscrKLZM_j-yuvOIbtutEeOq1HjVla_wmNGYmjXLmuHWWSeKaytNo8RW8RgWTX99UDqe7LnXNWHtvhtM676ZotpdyNdyJ5gXt15uU0W7c8vRzEEdBZQCHFv9RW1REl5Uq28jVtui7lKeWqApKZQ3CwQLhb2198e71hYnZSNFcr8xfGSD6NgivD1FM0paDwCFiEiC_qd-7x3HPNxo-Rr7cMq54MFwx3p2ffwHUwLHaL9Kx9EjOrwO0X8GsvvVrP8mG9-yd6r74Uk44dcX95aN7k_NoNK0R-nU6XL07-b2mTQTjri6-VLDoP0ku_Tq1dtjyRtQIUAc1ELg1QC9JdSmxpXvdRQKryV0_Y3np0du1_Wd24n2vla7ZSwe_Py38HAbrpTyumCTTIXcdFtjtscW03G_RbrlNsaDBoNmgOiP7RZYHXyq1uEbYNXEnGvpVUsf2nN6XDFbvGE2g4yuqCKhtp7UgvKmqMt-Dq5GqhtwVaw8TpiPvl5ROy8-ZLYgay8tACtppuvZFKZ9ijnQFSmNIAwo90-nDjDndqAJL8VQDoDpL4hObwqhQ6Utps5TluPkZclZbWI0fvxZlstvXGuDlbK2hcoX9q7A1CDGSlSzhselYSpule2vKp2lAN26uoAWzqpWOoyb1VedFly0pIttunKsm1W37Q6RssLfcquuBn0ft9PIE0ACv2JL1u0cz-9ggk89yf8zxDQ63P_HN9zHMhSMuO9oT9JaQb7fiY70eLf4WoQhohXr9Pi_-Xiq-8vAPYnd_6tPxns7e2_2BmPX4z)

Différents types de relations entre deux classes ont été utilisées dans le diagramme de classes :

*  : une flèche vide avec un trait en pointillés représente une dépendance. C’est-à-dire que la classe d’où part la flèche utilise la classe vers laquelle la flèche est dirigée.
*  : une flèche vide avec un trait plein représente un héritage. Cela signifie que la classe vers laquelle est dirigée la flèche est une fille de la classe d’où la flèche part. Dans ce type de relation la classe fille est un type plus spécifique de sa classe mère et implémente les mêmes attributs et méthodes que celle-ci.
*  : un losange vide avec un trait plein représente une agrégation. Cela veut dire que la classe vers laquelle le losange est dirigé contient un ou plusieurs objets de la classe d’où part le losange. Mais les instances de la classe d’où part le losange peuvent vivre une fois l’autre classe détruite.
*  : un losange plein avec un trait plein représente une composition. Cela veut dire que la classe vers laquelle le losange est dirigé contient un ou plusieurs objets de la classe d’où part le losange. Mais les instances de la classe d’où part le losange ne peuvent pas vivre une fois l’autre classe détruite.

Explication :

Dans le diagramme de classes de l’application EasySave nous trouvons plusieurs patrons de conception.

En premier lieu, nous avons un patron de conception nommé usine. Il permet de centraliser la création d’objet dans le but de découpler l’instanciation de classes spécifiques du reste du code.

Dans notre cas, le patron de conception usine est formé par l’interface BackupStrategy et ses deux classes héritières CompleteBackupStrategy et DifferentialBackupStrategy qui permettent d’exécuter une sauvegarde en fonction de son type (complète ou différentielle). De plus, il y a l’interface BackupStrategyFactory et ses deux classes filles CompleteBackupFactory et DifferentialBackupFactory dont l’objectif est de créer des objets de type CompleteBackupStrategy ou DifferentialBackupStrategy.

L’implémentation du patron de conception usine respecte le principe SOLID ouvert/fermé. En effet, si nous devons ajouter un type de sauvegarde il nous faudra uniquement ajouter une classe héritière de BackupStrategy et une classe fille de BackupStrategyFactory. Il n’y aura alors pas nécessité de modifier tout le code.

En second lieu, nous avons aussi un patron de conception appelé stratégie. Il permet de changer dynamiquement le comportement d’un objet en déléguant une partie de son comportement à des objets appelés stratégies.

Dans notre cas, le patron de conception Stratégie est formé par l’interface ITranslateStrategy ainsi que ses deux classes héritières French et English. Ces dernières permettent de déterminer la langue dans laquelle l’application est traduite (Français ou Anglais). Il y a encore la classe Translation qui implémente une instance de type ITranslateStrategy. Elle permet de traduire du texte dans la langue déterminée par ITranslateStrategy.

L’implémentation du patron de conception stratégie respecte, lui aussi, le principe SOLID ouvert/fermé. Effectivement, si nous devons ajouter une langue pour l’application il nous faudra uniquement ajouter une classe héritière de ITranslateStrategy. Il n’y aura alors pas nécessité de modifier tout le code.

En troisième lieu, nous avons un patron de conception singleton. Il garantit qu’une classe ne peut avoir qu’une seule instance et fournit un point d’accès global à l’instance en question.

Dans notre cas, le patron de conception singleton est mis en place dans la classe Translation. Pour ce faire, le constructeur de la classe est redéfini. Plutôt que de créer une nouvelle instance de la classe Translation son constructeur vérifie tout d’abord s’il existe déjà un objet de cette classe. Si tel est le cas l’objet existant est retourné, sinon une nouvelle instance est créée. De cette manière, il y aura toujours une seule et unique instance de la classe Translation dans le code.

L’implémentation du patron de conception singleton limite le nombre d’instance d’une classe à 1. Cela peut être utile dans certains cas, notamment lorsque l’instanciation d’une classe est couteuse en mémoire, il est alors préférable de garantir l’unicité de son instance.

En dernier lieu, nous avons encore un patron de conception nommé observateur. Il permet à un objet (le diffuseur) de notifier automatiquement plusieurs autres objets (les souscripteurs) lorsqu’un événement se produit.

Dans notre cas, le patron de conception observateur est formé par la classe BackupStateJournal qui a le rôle de diffuseur. Ce dernier est également relié à l’interface IObserver ainsi que sa classe héritière ConsoleView. Cette classe implémente une méthode qui retourne les informations comprises dans les notifications de la classe BackupStateJournal. En outre, le diffuseur implémente plusieurs méthodes qui ont pour objectif, entre autres, d’ajouter un objet à la liste des souscripteurs ou encore de notifier tous les souscripteurs.

L’implémentation du patron de conception observateur permet d’ajouter facilement des souscripteurs sans modifier tout le code. Effectivement, il nous suffit d’appeler la méthode adéquate du diffuseur. Un nouveau souscripteur sera alors ajouté à la liste de diffusion et notifié comme tous les autres souscripteurs. Le patron de conception observateur permet également d’automatiser les réactions dans le code sans devoir le modifier.

Enfin, l’utilisation de ces différents patrons de conception dans le code a encore pour objectif la réutilisabilité des différentes parties du code sans devoir le modifier.

* 1. Diagramme de séquence :

Le diagramme de séquence UML est un outil graphique permettant de représenter l’interaction entre plusieurs composants d’un système au fil du temps. Il est utilisé principalement en modélisation logicielle pour visualiser le flux d’exécution d’un programme.

Dans le projet EasySave, le diagramme de séquence est essentiel pour représenter le fonctionnement du processus de sauvegarde.

Un diagramme de séquence fait partie des diagrammes UML. Il permet de :

* Décrire l’ordre chronologique des interactions entre les composants.
* Visualiser la communication entre objets et acteurs d’un système.
* Aider à comprendre le flux d'exécution d'une fonctionnalité.
* Servir de référence pour le développement et la documentation technique.

Composé essentiellement de :

* Acteurs :

Représentés par un bonhomme   
Ils symbolisent un utilisateur ou un système externe qui interagit avec l’application.   
*Exemple concret :* Un utilisateur qui déclenche une sauvegarde.

* Classes/Objets et participants :

 Représentés par des rectangles en haut du diagramme   
Ils correspondent aux différents composants logiciels qui participent à l’exécution du scénario.

*Exemple concret :* Une chaîne de restaurants :

* L’acteur serait le client qui passe commande.
* Les classes/objets seraient les différents services du restaurant :
* Serveur (BackupController) qui prend la commande et la transmet à la cuisine.
* Chef cuisinier (LogController) qui prépare la commande et note les plats réalisés.
* Responsable de salle (MenuController) qui organise le bon déroulement du service.

* Messages et interactions :

Flèches dirigées entre les objets   
Elles indiquent les appels de méthodes et les retours de valeurs.

Il existe différents types de flèches :

*  : ce type de flèches représente des messages synchrones dont l’expéditeur attend une réponse de la part du destinataire avant de pouvoir continuer son exécution.
*  : ce type de flèches représente des messages asynchrones dont l’expéditeur n’attend pas de réponse de la part du destinataire avant de pouvoir continuer son exécution.
*  : ce type de flèches représente des messages asynchrones qui sont utilisés lorsque le destinataire d’un message synchrone répond à son expéditeur.
* Une image contenant blanc, Police, clé, ligne

  Le contenu généré par l’IA peut être incorrect. : ce type de flèches représente des messages asynchrones qui sont utilisés lorsqu’un nouvel objet d’une classe est créé.
* Une image contenant symbole, ligne, Symétrie, conception

  Le contenu généré par l’IA peut être incorrect. : un message avec une croix au bout de la flèche est utilisé lorsqu’un objet d’une classe est détruit.
* Barres d'activation :

Représentées par des rectangles verticaux sur les objets   
Elles indiquent qu’un objet est actif et en train d’exécuter une tâche.

* Boucles et conditions :

 Représentées par des blocs alt, loop, opt en UML   
Elles permettent d’ajouter des conditions et des répétitions.

Dans notre projet, nous nous sommes appuyés sur le diagramme de cas d'utilisation (Use Case Diagram). En effet, chaque bulle dans ce type de diagramme représente une action, ce qui nous a permis de structurer notre approche.

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

Ainsi, nous avons conçu cinq diagrammes de séquence, chacun illustrant un processus spécifique. Voici ces diagrammes accompagnés de leurs explications détaillées :

1. Diagramme de séquence pour le choix de la langue et de l’action :

Une image contenant texte, diagramme, Parallèle, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le [diagramme de séquence pour le choix de la langue et de l’action](https://mermaid.live/edit#pako:eNqtVltv2jAU_iuWX9ZqUBVouUQVEuplRRvdtK57mJAmKzkES7kwx-nKqv6erfsb_LEdO45JSEL7MB7A2N_5fPydi_1I3dgD6tAEfqQQuXDBmS9YOI8IfpgrY0HuEhCEJeRO8oAnTEIqsuUVE5K7fMUiSabXM4WZRhLEgrlAruMwBDJj7pJHUMV_YJGfMh_O40iKOAhANGNm6GLQvPyVw8_q6hfBoiRAb_ftMPk0VV6rn3dx7AewNauCrwQKtFR4M3LI1MJvpcBvf101u4x8lE3b5cPXGc4gSvc5r9YbpFFLmSw2kPwet9GxzKZ0VNuH5C1GziEXm-eQCYFTwRu2WgXcZZLHxlrFViGrMXPIDbvnPjLna8nBYWZUBZP2eLwl0Z475PPm2U1Xm2e9NSTqFCzc_JECx3NqWeeUgMSJWwjAleDlC3Na3k2zkjZ6267zVqsegkT6_7Fb8WwFfZTyKClPVgFbb4WxoxbZJd7RTBFkh9DBmSwW3F1qgUiIkSUeuhsoLLpaCqaGny9jnnBEMwMi7B5cfd5FsPmLTEkpri8KZXncZZxw2KdBHdcE88nKkJ0cUyvPExZI6_EV7rr5zbjxr2kb3CerQIecnbkCMP3G48wEggQsnSm3F8kMroYt8vYGvKbHOBhb-T0xRX1QrXNzbA9sTTa1wrzsyn3AwZ4aIXqiVbTlVgbpUtu1K5baHBu-CcQ08uAhz-xdGtx_l6aYGq-lUYVv29WuJyw7SdG2UMb7t6-x3YlR3gptQebCGdMWKR-hoGdWhuh7bZztZM5kLGvAmQB4y2grL-V1vqtCkGpVT5OP77d5q-4nJUazH1VBMi4utyy1jpkAZwq9mmqrzv4mlVNYw9pGlUZgkFugSf590cebKpPK1TxQdq-cBrs8lw_gptIU0UFtBuhWUgjITV1E2i-GBPBWzR9MjVEwglgNQ0gSrPGKcUm-KxBh_YVt-1ahyzS-hwqYuhdHYXn7eChMoiu0RdGRkHEPH5KPCjGncgkh3pcODj1YsDSQKs-fEMpSGd-uI5c6UqTQoiJO_SV1Fgz1btF05SGreYXmEHzTfIvj0ILwP3Ue6QN12p2jk06vP-ifHg_6_eHpqHfaomvq9EZH3eN-tz_qdfoD_AyfWvSXpugeDY4Hw97JybDbGXVHo97TP_rmxMc).

Explication :

Ce diagramme de séquence illustre le processus par lequel l’utilisateur peut choisir la langue d’affichage du logiciel, ainsi qu’une action à réaliser.

Au démarrage de l’application, un menu interactif est affiché sur l’Interface Homme Machine pour permettre à l’utilisateur de choisir la langue dans laquelle il souhaite continuer sa navigation.

Une fois que l’utilisateur a choisi sa langue préférée, un objet de la classe French ou de la classe English est créé. Celui-ci servira pour les différentes traductions nécessaires au sein de l’application.

Ensuite, un autre menu interactif est affiché sur l’Interface Homme Machine afin que l’utilisateur puisse choisir une action à effectuer. Bien sûre, avant d’être affichée, chaque option du menu est traduite dans la langue définie plus tôt. Mais si une erreur survient lors de la tentative de traduction, un message d’erreur s’affiche à la place du menu interactif.

1. Diagramme de séquence pour la création d’une sauvegarde :

Une image contenant texte, diagramme, Parallèle, Rectangle

Le contenu généré par l’IA peut être incorrect.

Lien vers le : [diagramme de séquence pour la création d’une sauvegarde](https://mermaid.live/edit#pako:eNq1mMFy2jAQhl9Fo1OYAAOEQuJDZijpNDkw05k0PXS4aOwFlNgSleQ0aSbv0mN5Dl6sK-MEgw3YxvElwdr9Vvr1W2Z5oa70gDpUw68QhAtXnE0VC8aC4MVcIxW506AI0-TOcJ9rZiBUq-E5U4a7fM6EITfXIxtzIwyoCXOBXMsgADJi7owLSMePQIRDKYySvg8ZvM_MfQjnhyNGOH1_1-APDr_TY98VE9rHdeyjD77d2PXYP1-lnPqwThuLd3X4I36OBNq6tbNGJGajRk5RMIcMZ5JrrsiYDtVygSOhAKJZ-AhTpjwY01WSFdfmbIrmrKoEYIyCqLQUxI2IseCb8RaxTbhmwvMhFlsBs4yT2q7s0-1dwRXYpBjwlujBuw5Z28x8Qz4lVqkJBhBvubhf_iXLhVkuiItqoB56lWCv7cqkcXlJTjNkdtban4zpZh2dKARCwZRrY3XXzTGtrWtlUFfl0A0R3gs5Kl6EvoZbRzVQy8buucc7WoRPTDQrc2ARtm56D6srmt6lGomMPphMuDsDdQQffA0kkFxoNFgVBnr3sz0mnOjJHOiHk9p2jh3OabcveBv-EDwthAzsNH2WmCdxytnsMLWcvQ5zC9tqJeXxZZLKb1iITcsi307eCHbLonMXs9cB8Sm7ayEbsaWdgfOUocL3Y9Xm2As-zh970ZVapGClnC4pSM0yygqQ2yvb4UfYBY83w0X0Yv4AzxyiH2ucQ_yK3VO8XG4LFUdn-ShByW2mzJzCjhrKYO4v_-HX11KWSaSX80QCUMmm7-UVVueKTybLhQJhOOBoOY1SkHJKpTCV6JWDWtxTUbOh9epVbJ7n1b3WcqJLejEfvBqflq114GT6GK5tH58imt5xkL21rDYm9xG2Gbw5g9NEI5_6xlWPX6X15ClY3-Il-s3tvn9dLuKv5xiXy267d6Vm9E6DexmaWHkhkYC3k1uAHQmO-NjVwNHN7O0ay2xZnHrJxjWbVO5hymZV2JDmLZDV4TVSz00eGggv9UsGglL37INB6zQAFTDuUYe-2JAxNTMI8IR18F8PJiz0jRX3FUNZaOTts3CpY1QIdapkOJ1RZ8Kwua3TcO4hNv4J7v3unImfUm58ps4LfaJOp91tnnUvLs67_X6r1el3O3X6TJ12s9_pts465xe9frfV7_XOXuv0T4RoN1vx1em1L1rds9f_5TTHlQ).

Explication :

Ce diagramme de séquence illustre le processus par lequel une sauvegarde peut être créée et enregistrée dans le logiciel.

Lorsque l’utilisateur choisit l’action de créer une sauvegarde, plusieurs scénarios sont possibles.

Tout d’abord, l’application doit permettre d’enregistrer au maximum 5 sauvegardes. Une vérification est alors effectuée pour contrôler le nombre de sauvegardes déjà en mémoire.

Dans le cas où 5 sauvegardes sont déjà enregistrées, un message pour en informer l’utilisateur est traduit puis afficher sur l’Interface Homme Machine.

Dans le cas contraire, si moins de 5 sauvegardes sont enregistrées, l’utilisateur peut créer une nouvelle sauvegarde.

Nous savons qu’une sauvegarde est définie par un nom, une source, une destination, ainsi qu’un type. Un message va alors être traduit pour demander à l’utilisateur de saisir le nom de la sauvegarde. Un autre message affiché sur l’Interface Homme Machine, préalablement traduit, demandera à l’utilisateur d’entrer la source de la nouvelle sauvegarde. Un dernier message, traduit puis affiché sur l’interface, permettra à l’utilisateur de saisir la destination de la sauvegarde à créer. Enfin, un menu interactif, traduit avant d’être affiché sur l’Interface Homme Machine, permettra à l’utilisateur de choisir le type de sauvegarde qu’il souhaite créer parmi une sauvegarde complète ou différentielle.

Ensuite, une sauvegarde sera créée avec les différents paramètres entrés par l’utilisateur. Cette nouvelle sauvegarde sera également ajoutée à la liste des sauvegardes enregistrées dans le logiciel.

Pour finir, un message informera l’utilisateur de la création de la nouvelle sauvegarde.

1. Diagramme de séquence pour l’exécution des sauvegardes :

Une image contenant texte, diagramme, Parallèle, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour l’exécution des sauvegardes :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNrNWdtuGzcQ_RViXywBlmM5tuzqIYCbOJfCdoy4SYtCL9TuSGayIrck17ES5AP6F32MvkM_1hmudrU3SoqLAtWDLtzhcK5nZqivQagiCIaBgT9TkCG8EHyq-WwkGb54aJVm7w1oxg17b0UsDLeQ6uxxwrUVoUi4tOzN6yuiwY_msyuQ6XMlrVZxnLGqrrTv-CDgc05L35tUP_PwU5pUOdfXfLturUZNpvOXTsf5emvtQdt-A1XSbHN9tbnzuZolMdiW3e1PmhxeiMkENEgreNzk4n_a5HSpplXDVRb8VkN2v6hUSx6XTbZebYmMt2MMoPvslOJHm3WkUTHkbi_9bNK-itWYxx-4FnwcgyH62lKrxtNCVfrWeXF52R3JjNIFee_ZMxfIQ_b8TgkjNBsFv2ouTYwKsosHCFOLZIan9zDlOoKO6Y6CIlfEPZERp9pSkRXEvNethf-QXXHJp3CO5Kh0p7uBFkVbiTQDazW4M5RkoZMXGsK0JdrGh7XjWoR9zWUUryI1MwkKkAtd3477UQ8SOrOoMfCFxcDsPAEWpWwiwjuBhLGaMgQgdq9SQ28xkoV6uVgudK7TdslWTnup9IxbdLJPKJ9OuCXb2zmPolxBC9I4Bf0a1kKv4aKPKrUsAhbvQc4OQzYlpScihoTbu31G72N3ZvbdwiwxGiBmnV9u314_-f3qsuvVp456Q8QCk8R8fvGANrdCTjOKIriKGGjHy11OeAU2X0ylzRlHsDG8GpDda_FGOekuUXyMFMzxddYZ1jlhM_5Q5F7p0N30WYfl-YRCELOvcUaEFlRSODAJmEqfbGO1Fvo8DVMJJW4MydP75QIO2DuwiJTk_hlyOtgc3Q4MC2e-kUlqbzTWCtvwY7VKFntX4jnYuJAEXqlkMp1hXikKyT5b_s1OWEIiQY5vSKJSlsQpIkqqK1bZ3-v3nu7V6DGP-6TQU3yzCvNHYCqjdrbFP005t8XZDdcGq8v4jYzgoSPIBrsF8Y6xdrv2klkuYnCAKtFXj4-urIzUmAOiNJrN-DyOm5rKr2pO9qDzWL3bzIpFW9tr-Nzpsk54p5VULOYyXC66Pi6FgJUuYOial9S6RY-EzQahhVFvXX3fLRcJaKsEoic4BAO0JYm3CiuKT9AaKJGKBB4FBlmFwCTirNuTcON1ok8mj8aVphB11sDr7ZpX-ZaGsr0J7bUHAYYRES4XU0EwVcYVqpG0lqTCFHU1awY2qd4iEY8tK0Wr60iX3y1kT31x1d64Dj3ruYUqVtrU-_rD2TSOrKZK9vESy2tRuWu-MVtOLDrrXmv-VHI7zK3FAFvw0KYl-Gi4wH9wiWyTVSA2UPYV9f0I6NT4Awq3yXQYX_4pYbjhWavrtg0c9Kq36iRC1oC7FoJ6LnTSNZ9B6xEZ6frBqnf3ZsprTHylqQC5go4tlmGYI2GaoIWWC49T6qf8uOHw8-3kkhtbheoNeLAm2DDVefWkA5kaYzuZViLNA2L_bbo0BmI_kK2jNqpGbSlzmNkTMcORllEXVdCFFUXHiMCfcGbAzp2Ar1wyCP8R_0RWAbDwhgpXsWSMBRZhWooqbAv85PhNS7H87urOarRhE05MSZSwCYpem3uG_V2gEWTUGqW7pNsOSFOwj5VK2A21clhAKWPyaSxUicBc2QLB9SqadwKwagXQ3TNhXOOD3a_r4snEiVZTDYamIHo0cRMXy2ecsou3Vuwf6FOG7H0S8UK2JtjXO5G26F3-hWQMlaKuGac67cX4bfLWzuuuL0aG7FpZMZnnv00rlNTuUdx9QXHP0vOZYBRkNO4OzzAhyfjLhfGosfkQF-PFHU1uX0-Jr93l0Kt831NumdcT2XJB5sYYcbMwwTjEj4yPhr39EXKzis9dgqRigFFwReG-Co1ynJeF3q52OUHqaVHk7qP62fY5QCWlGWAipPBOAO2DDY6AM0EDE-P3EDKThuHye1v32QgDD0ainHmHkH0e_KYRzN2NDsF4bRHN0-nusxxrcFJGSAfremPuELzoBD5m9nEIb3LYj-KYdXDWlVN2wc38FrU4yI6oN_TlNmFT0f6fye9pdDb6FwlXeF2aNtx)

Explication :

Ce diagramme de séquence illustre le processus d'exécution d'une sauvegarde dans l'application EasySave. Il met en évidence les interactions entre l’utilisateur et les différentes classes impliquées, en détaillant chaque étape du traitement.

L’utilisateur sélectionne l’option "Exécuter une sauvegarde" dans l’interface utilisateur (IHM). Cette action est envoyée à MenuController, qui gère les interactions principales du menu.

MenuController demande ensuite le choix du format des logs (JSON/XML) à l'utilisateur.

L’utilisateur sélectionne le format souhaité, qui est ensuite appliqué par GlobalVariables (modification des chemins de fichiers de logs et de l’état des sauvegardes).

MenuController demande à BackupController d'afficher la liste des sauvegardes existantes. BackupController récupère les sauvegardes et les envoie au MenuView, qui les affiche à l’utilisateur. L’utilisateur est invité à sélectionner une ou plusieurs sauvegardes en entrant un numéro ou une plage de numéros (1-3 pour exécuter les sauvegardes 1 à 3).

Une fois l’index de la sauvegarde saisi, il est envoyé à BackupController, qui :

* Vérifie que la sauvegarde sélectionnée existe bien.
* Démarre le chronomètre pour mesurer le temps d’exécution.
* Contacte BackupStateJournal pour calculer l’état initial de la sauvegarde (nombre total de fichiers, taille totale, progression initiale).
* Demande à BackupStrategyFactory la stratégie de sauvegarde adaptée (Complète ou Différentielle) :

Cas 1 : Sauvegarde Complète

* Tous les fichiers sont copiés.
* BaseBackupStrategy exécute la sauvegarde en copiant tous les fichiers du répertoire source vers la destination.
* Une confirmation est renvoyée à BackupController une fois la sauvegarde terminée.

Cas 2 : Sauvegarde Différentielle

* Seuls les fichiers modifiés depuis la dernière sauvegarde sont copiés.
* BackupController récupère la date de la dernière sauvegarde à partir des logs (Logger).
* DifferentialBackupStrategy compare les fichiers pour identifier ceux à copier.
* BaseBackupStrategy exécute la sauvegarde en copiant uniquement les fichiers mis à jour.

Cas particulier : Si le répertoire cible n’existe pas

* Si aucune sauvegarde antérieure n’a été trouvée, une sauvegarde complète est automatiquement déclenchée.

À chaque fichier copié :

* BackupStateJournal met à jour l'état de la sauvegarde en fonction du format (JSON/XML)
* Les observateurs (Observer Pattern) sont informés de la progression.
* ConsoleView affiche l’avancement en temps réel.
* Une pause (Thread.Sleep) est ajoutée pour que la progression soit visible.

Une fois la sauvegarde terminée :

* BackupController arrête le chronomètre et enregistre le temps total d’exécution.
* BackupStateJournal met à jour l’état de la sauvegarde en COMPLETED.
* LogController enregistre les informations de la sauvegarde dans un fichier journalier (JSON/XML) via la DLL EasySave.Logger, en utilisant le format sélectionné : JSON writelog(), ou XML writeLogXML().

Les informations stockées incluent :

* Le nom de la sauvegarde
* La source et la destination des fichiers
* La taille totale des fichiers transférés
* Le temps de transfert

BackupController informe l’utilisateur que la sauvegarde s'est terminée avec succès. L'utilisateur est redirigé vers le menu principal.

1. Diagramme de séquence pour la consultation de l’historique :

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour la consultation de l’historique :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNp9VNtu2zAM_RXCL00AZ3v3Q4Gi3boNcTesazEMAQrGYmytsuTpErQo-kH5jv7YKF9ycYL6xTZFHvIcUnxJCiMoyRJH_wLpgq4klhbrhQZ-sPDGwp0jC-jgzkslHXoKtjtu0HpZyAa1h69f8ujDr-OznHS4NNpbo1QHdWg5jrhWZonqHq3EpSIXQ0am45i5KcsOvf-aXM3n086vpTCbtlVmcFkZ6aSFRfLLonaKKQFX44Ly7BbzKVM6-GuC1agkWbdItoLIdXSPgCPTlnpMwrkOOWaQo8aSLtidU02m7_hyiX1pNXlvqc1hNBRt3XRUzCk1R4icYyRgFmX6LBX9QF8N5WwhT6o97spsdn7ei53BNfke8AZrmkxhgk1DnBwEgUJYGd2xENQEyRLzgVJ9XkHvZ-47yiyGdJ-epPMsI9y_baxccZOAoinOcNc-EdoO9sRUOyBsla4xWu6g9-Ejn3Hb2ij-C-u3DUy-3X6_-fg7H-bqILgdr-xgqvYLgSUWj6F5YNvDMz-zPJ8J8WFAhIk2dQrk0afg2L-glHVxXmqMuqWglxa88aiioitZVEy6s-7-wWFYU4lWvG1S8CiZRAqVsSgtf3iqGy7ojJ6oCBF1OgwTKUcnOY31uAhF0JHYIMouajxzpxTZhe86ARiiQOKsCnJbjxanBzn2qEf9ST7qigFqdoLGSh13gRow9qbq1BXZO97e3T1bd8WTNKnJ1igFL8mX6LRIfEU138JYgkD7GNO9sh8Gb26fdZFk3gZKk9AIxukXapKtkCVOExKSl2rebd12-aYJC1lWWw_eZn-MGSJe_wMLa-pi)

Explication :

L’utilisateur sélectionne "Consulter les logs journaliers" depuis l’interface utilisateur (IHM). L’IHM transmet cette action à MenuController, qui gère l’affichage du menu et l’exécution de l’action demandée.

MenuController envoie une requête à GlobalVariables pour récupérer le chemin du fichier log. GlobalVariables appelle la fonction GetLogFileName() de la DLL EasySave.Logger pour déterminer le fichier de log du jour.

Logger exécute la méthode Exists() pour vérifier si un fichier de log est disponible pour la date demandée. Deux cas de figure sont possibles :

*Cas 1 : Les logs sont disponibles :*

1. Logger confirme que les logs existent et envoie cette information à MenuController.
2. Logger récupère les détails du fichier de log en format json ou xml, notamment :

* Nom du fichier (backup\_log\_yyyy-MM-dd.JSON/XML)
* État de la sauvegarde (Active, Complétée, Échec…)
* Source et destination des fichiers
* Nombre total de fichiers traités
* Nombre de fichiers sauvegardés
* Taille totale des fichiers transférés
* Horaire de début et de fin de la sauvegarde
* Temps d’exécution

1. Ces informations sont envoyées à IHM pour être affichées à l’utilisateur.

*Cas 2 : Aucun log disponible :*

1. Logger informe MenuController qu’aucun log n’a été trouvé pour la date demandée.
2. MenuController affiche un message à l’utilisateur :

"Aucun log disponible aujourd’hui."

1. L’utilisateur est alors redirigé automatiquement vers le menu principal.

Après l’affichage des logs (ou en cas d’absence de logs), MenuController envoie une instruction à l’IHM pour retourner au menu principal. L’utilisateur peut alors choisir une autre action.

1. Diagramme de séquence pour quitter l’application :

Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour quitter l’application :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqNU0tOwzAQvYrlDVQKPUB2FRSBRCVQ6QZlM3KmrYUzTv2RqKpKXAc2HKI34SSM0zYKbRFdWLHjN-_NvPGspLIlylx6XEQkhTcaZg4qUdD3-5dIC1SwTkw8OgFeTII22kPA6DqYGlzQStdAQdzfjRIwff5AjJDitaXgrDFb1oM_p8MenW1SY_xuWyHvNKV7I0oUU3QVhuiww9DZNjVc9VJquRhvPgyqoC0Rw-VT1CGgK2QHn0pg-EFyOaOfHZBnqcBS0JAINbfaa_yfIBcjIJjhoInzl72u1yaIh4vY8VhZmmquqgPar4O8TggN31DFsFNiIZfMIRFcPI9vZ3Mubltjyyjq1vxjjn2Prjg62Z3MGtR1XDKhT09GDlmg8bm2fF5sbRcGuHm0-UyGciO5am8N9vv9X36i8eck3moPiKKBpj1M2mgcPxCkkk8ykxVfgi55GFYF8buRYY4VNzTVUIJ7LWRBa8ZBDHa8JCXzZGQmY11yr3aDI_MpcJaZxFLz3Iy209UMWSadjbN5i-BH_WLtPmL9A4vmX78)

Explication :

Ce diagramme de séquence illustre le processus de fermeture de l’application EasySave. Il montre comment l’utilisateur interagit avec l’interface pour quitter l’application et comment le programme gère cette demande.

L’utilisateur sélectionne l’option "Quitter" dans l’interface utilisateur (IHM). L’IHM transmet cette action au MenuController, qui est responsable de la gestion du menu et des actions utilisateur.

MenuController appelle la fonction ManageActions() pour traiter la demande de l’utilisateur. Une boîte de dialogue ou un message de confirmation est affiché pour demander à l’utilisateur s’il veut vraiment fermer le programme. Deux scénarios possibles sont alors traités via une condition alt :

Cas 1 : L’utilisateur confirme la fermeture

* ExecuteAction() retourne true, validant la demande de fermeture.
* MenuController envoie une requête au Programme Principal pour exécuter la fermeture.
* Un message s’affiche pour indiquer à l’utilisateur d’appuyer sur "Entrée" afin de fermer définitivement la console.

Cas 2 : L’utilisateur annule la fermeture

* Si l’utilisateur choisit d’annuler, MenuController annule la demande de fermeture.
* Un message est affiché pour informer l’utilisateur que la fermeture a été annulée.
* L’IHM reste active et retourne au menu principal.

Si l’utilisateur confirme, le programme se termine et ferme l’application. L’interaction utilisateur prend fin.

1. Documentation de la Dynamic Link Library V1.1 :



1. Documentation Utilisateur V1.1:

1. Documentation Technique V1.1 :



1. Release note V1.1 :



# Version 2.0:

1. DIAGRAMMES :

## Diagramme Use Case :

Un diagramme de cas d'utilisation (Use Case Diagram) est une représentation graphique des interactions entre les utilisateurs (acteurs) et un système. Il illustre les fonctionnalités principales offertes par le système sous forme de cas d'utilisation, permettant ainsi de comprendre qui utilise le système et comment.

Ce type de diagramme est essentiel en phase de conception, car il permet de clarifier les besoins et de définir les interactions entre le logiciel et ses utilisateurs sans entrer dans les détails techniques.

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

Explication :

Ce diagramme de cas d'utilisation illustre les principales fonctionnalités du logiciel EasySave et les interactions possibles entre l'utilisateur et le système. Elle est composée de :

Acteur :

Utilisateur : L’acteur représenté par un bonhomme est celui qui interagit avec le logiciel pour effectuer différentes opérations de sauvegarde.

Cas d’utilisation principaux :

* Réaliser une sauvegarde

Fonctionnalité centrale du logiciel, permettant à l’utilisateur de gérer ses sauvegardes.

* Créer un travail de sauvegarde

Avant d’exécuter une sauvegarde, l’utilisateur doit créer un travail de sauvegarde, définissant :

* Un nom
* Un répertoire source
* Un répertoire cible
* Un type de sauvegarde (Complète ou Différentielle)
* Exécuter une ou plusieurs sauvegardes

Une fois le travail de sauvegarde configuré, l’utilisateur peut lancer manuellement une ou plusieurs sauvegardes.

Il peut choisir de lancer toutes les sauvegardes en séquence ou de sélectionner des sauvegardes spécifiques.

* Consulter l’état temps réel des sauvegardes

Permet à l’utilisateur de suivre en direct la progression des sauvegardes.

Les informations incluent :

* Nom de la sauvegarde
* Source
* Destination
* Le nombre total de fichiers
* La taille totale des données
* Le nombre de fichiers restants
* Le progrès de la sauvegarde
* Consulter le fichier log journalier

L’utilisateur peut consulter les logs journaliers générés par le logiciel.

Ces fichiers enregistrent toutes les actions de sauvegarde et leurs détails (date, heure, état de la sauvegarde, erreurs éventuelles, temps, source, destination).

## Diagramme d’activité :

Un diagramme d'activité est une représentation graphique des flux de contrôle et des processus d'un système. Il illustre la succession des actions et des décisions effectuées au sein du logiciel, souvent sous la forme d'un workflow.

Ce type de diagramme est utilisé pour modéliser le comportement dynamique d’un système en mettant en évidence les différentes étapes d'un processus, les choix possibles, ainsi que les conditions de transition entre les actions. Il est particulièrement utile pour visualiser les algorithmes, les flux de travail métier, ou encore les enchaînements logiques d'exécution.

Une image contenant texte, diagramme, ligne, Plan

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme activité :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqlWOtOIzcUfhXL0gqQkuXSbhfyoxWC7Y8uLKroVlUTNDIzTnB3YoexTaGBd9mfTV8jL9bjy3g8k5kktJEgGfuc7_jcPp9kjlORUTzA41z8md6RQqFfzkccwet8ubjVate97aF-_3t0QXhKp5SrhMxmOUuJYoIP7WqB8p1o8cZhtCpYqLM7wR6TnPCJpkPzIBlAEORWvHosZLXIeMzSO1okgKmDGkkjk-7_mzfoZ82UooV7rilaKL899O-txy_3jPyPjO_C317DylmxXDivsh3NKZJEP9AJKTLaZThoJJXs0CyCoTpCGYUWBYt0TYz7CRfTGMqtAhSCDZS1ALbqxYhS6CKlraAAZzfXAa-ox9gZlYrxlQBEBiKJdVbagUJxGRH1NKt5URYMxMZstaJ36FrcX5eLgo19jSQzUpDp8m9VUDm_oBJFzxAkrvoslygVfCyKKSz98BKXTieUMfR8pdkzuqRSkglN0pb8J1cfh6e-tFBa1WDsELr6eLONsU-CV8aqzWRK-L0mXFWGNN_PGo5CEvaNs4jxVBQFTZWsNeIGJ1a7uq7WcpzOJhrx0JUfHpeLVL-qLSuVuGL8qmtN63y1KZuMY-kKZRqZjBNlspGLCSIP5tT5Dg0WnEarxYgcHUoCEPNAkDRggxXwgBnmEpOXmC8rPZven66vPj2jCzGRifk4HOHPiuVM-ooJpzWbI3yzBum3ywsPBJ86cWAvwITQ1CrQBgbCwaSi3FBJa4rCiW1ITnUyFYzL5DAuIMoLOgGYgtL5B6n6AHavd4hGVhYdxt0AlYqCPLBtaMjSo-3sxI5tFF7tL6JTcDYusREO_TUAxLQei9qRQ1w70br6qTMR0pQPSxkFrpouFwoKSr7GOctUdX4pAcG8gZt_5sFGacIkA-gRnAOC1IWEEqi6Yy1TNsCrM5xWTrvYNCTrcb6uInabi3tt6kGyOO6as3uYOmD4SM2m0IjALMAVdCBzFUzqafKeGJTlAlQ5WM2hYWeaSXQr0hwOhfotofCaIbmbPGnLcSvzNc64bThtvToCiMguWX5NaMmG0cS2n9dJES2_oiBXo4H1kJ6EqWdEsx2LsjxnU6YgFXIYpFCDkFEkdTMKNLsFput9G1MTcbMLoZHDsFS2ijT5UnQ6k8h0JHSNpSgoD2CQWgJjHH9nPc2UWfXELefngFk-oEywB-qnhuU_0GBwr4PCciHtVDSGgMcFJ76EPgkBbhqousNRNZSLlRHXYqyG5Rqq1tAMKtHZNRdLfLzUZlzerDNlK6eTfVrO4ML-hzAFkNioJmmJDHErLywv4eOe-cjYrBCg-bxMza6R3wuN9B8Muotta3sgvrd62W1wqFFrDAZE6WVtSQWvS7oy9hh31yvUcbMEm26vP0G46NYcwEdha_vtYQizU_N-2uT7_0Mw80cjHxvsdV2aG1TLSHZR8RmESueqdtPKzm-FkbCjHr_i42-Wyt5rikYhAUjfkXanpZTC2OimU5vM-uj3CviVQmlDXx0Iu8_alYpO86sKuIenFEqVZXiA50Z9hNUdncL0NICPGSm-jPCIv4Ac0cAKTzzFA1Vo2sN6lhFFzxmZwLeOcpFmTIni0v1MYn8t6eEZ4Xgwx494cPj-5O3x0dHhu5ODo-Nvjk8Ojnv4CQ8OXnr4LyEA5ODtiXu9--7bkyP4_76HC6End3gwJrmkFux3K2rsvfwLfkxjKg)

Explication :

Démarrage de l’application

Au lancement, l’utilisateur est accueilli par un menu interactif permettant de :

* Choisir une langue (Français ou Anglais)
* Quitter l’application immédiatement

Si l’utilisateur choisit une langue, il accède ensuite au menu principal où plusieurs options lui sont proposées.

Fonctionnalités disponibles

a) Création d’une sauvegarde

L’utilisateur peut créer une nouvelle sauvegarde en suivant ces étapes :

1. Saisie des paramètres :
   * Nom de la sauvegarde
   * Répertoire source
   * Répertoire de destination
   * Type de sauvegarde (Complète ou Différentielle)
2. Vérification des paramètres :
   * Si un paramètre est incorrect, un message d’erreur s’affiche et l’utilisateur doit recommencer.
   * Si tout est valide, la sauvegarde est enregistrée et un message confirme sa création.

Une fois l’action terminée, l’application revient automatiquement au menu principal.

Exécution d’une ou plusieurs sauvegardes

Avant d'exécuter une sauvegarde, plusieurs étapes sont réalisées :

1. Choix du format des logs
   * L’utilisateur choisit le format dans lequel seront enregistrées les informations des sauvegardes (JSON ou XML).
2. Vérification des sauvegardes existantes
   * Si aucune sauvegarde n’est enregistrée, un message avertit l’utilisateur et il est redirigé vers la création d’une sauvegarde.
   * Sinon, l’utilisateur peut sélectionner une ou plusieurs sauvegardes à exécuter.
3. Vérification des logiciels métiers
   * Si un logiciel métier est en cours d’exécution, la sauvegarde est :
     + Bloquée si une seule sauvegarde est lancée.
     + Mise en attente si plusieurs sauvegardes sont lancées en séquence.
4. Exécution des sauvegardes
   * L’exécution est illimitée en nombre de sauvegardes.
   * L’affichage des logs en temps réel (JSON ou XML) informe l’utilisateur de l’état d’avancement.
5. Cryptage des fichiers après la sauvegarde
   * Une fois la sauvegarde terminée, une vérification est effectuée pour savoir si certains fichiers doivent être cryptés.
   * Si oui, le logiciel CryptoSoft est utilisé pour crypter les fichiers définis par l’utilisateur.
   * Une fois cryptés, le temps de cryptage est ajouté aux logs.

L’utilisateur est ensuite redirigé vers le menu principal.

c) Consultation des logs

L’utilisateur peut consulter les logs journaliers des sauvegardes :

1. Vérification de l’existence du fichier de logs du jour :
   * S’il n’existe pas, un message informe l’utilisateur qu’aucune sauvegarde n’a été lancée.
   * Il est alors redirigé vers l’exécution d’une sauvegarde.
2. Affichage du fichier de logs dans le format choisi (JSON ou XML).
3. Après consultation, l’application revient au menu principal.

d) Quitter l’application

À tout moment, l’utilisateur peut quitter l’application, ce qui entraîne l’arrêt immédiat du programme.

## Diagramme de classes :

Un diagramme de classe est une représentation graphique des structures statiques d’un système logiciel. Il décrit les classes, leurs attributs, leurs méthodes, ainsi que les relations qu’elles entretiennent entre elles (association, héritage, agrégation, composition, etc.).

Ce type de diagramme est essentiel en programmation orientée objet car il permet de visualiser l’architecture du code avant son implémentation. Il aide à mieux structurer le projet, favorise la réutilisabilité et assure une bonne compréhension du modèle de données du logiciel.

Une image contenant texte, diagramme, Plan, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de classes :



Différents types de relations entre deux classes ont été utilisées dans le diagramme de classes :

*  : une flèche vide avec un trait en pointillés représente une dépendance. C’est-à-dire que la classe d’où part la flèche utilise la classe vers laquelle la flèche est dirigée.
*  : une flèche vide avec un trait plein représente un héritage. Cela signifie que la classe vers laquelle est dirigée la flèche est une fille de la classe d’où la flèche part. Dans ce type de relation la classe fille est un type plus spécifique de sa classe mère et implémente les mêmes attributs et méthodes que celle-ci.
*  : un losange vide avec un trait plein représente une agrégation. Cela veut dire que la classe vers laquelle le losange est dirigé contient un ou plusieurs objets de la classe d’où part le losange. Mais les instances de la classe d’où part le losange peuvent vivre une fois l’autre classe détruite.
*  : un losange plein avec un trait plein représente une composition. Cela veut dire que la classe vers laquelle le losange est dirigé contient un ou plusieurs objets de la classe d’où part le losange. Mais les instances de la classe d’où part le losange ne peuvent pas vivre une fois l’autre classe détruite.

Explication :

Dans le diagramme de classes de l’application EasySave nous trouvons plusieurs patrons de conception.

En premier lieu, nous avons un patron de conception nommé usine. Il permet de centraliser la création d’objet dans le but de découpler l’instanciation de classes spécifiques du reste du code.

Dans notre cas, le patron de conception usine est formé par l’interface BackupStrategy et ses deux classes héritières CompleteBackupStrategy et DifferentialBackupStrategy qui permettent d’exécuter une sauvegarde en fonction de son type (complète ou différentielle). De plus, il y a l’interface BackupStrategyFactory et ses deux classes filles CompleteBackupFactory et DifferentialBackupFactory dont l’objectif est de créer des objets de type CompleteBackupStrategy ou DifferentialBackupStrategy.

L’implémentation du patron de conception usine respecte le principe SOLID ouvert/fermé. En effet, si nous devons ajouter un type de sauvegarde il nous faudra uniquement ajouter une classe héritière de BackupStrategy et une classe fille de BackupStrategyFactory. Il n’y aura alors pas nécessité de modifier tout le code.

En second lieu, nous avons aussi un patron de conception appelé stratégie. Il permet de changer dynamiquement le comportement d’un objet en déléguant une partie de son comportement à des objets appelés stratégies.

Dans notre cas, le patron de conception Stratégie est formé par l’interface ITranslateStrategy ainsi que ses deux classes héritières French et English. Ces dernières permettent de déterminer la langue dans laquelle l’application est traduite (Français ou Anglais). Il y a encore la classe Translation qui implémente une instance de type ITranslateStrategy. Elle permet de traduire du texte dans la langue déterminée par ITranslateStrategy.

L’implémentation du patron de conception stratégie respecte, lui aussi, le principe SOLID ouvert/fermé. Effectivement, si nous devons ajouter une langue pour l’application il nous faudra uniquement ajouter une classe héritière de ITranslateStrategy. Il n’y aura alors pas nécessité de modifier tout le code.

En troisième lieu, nous avons un patron de conception singleton. Il garantit qu’une classe ne peut avoir qu’une seule instance et fournit un point d’accès global à l’instance en question.

Dans notre cas, le patron de conception singleton est mis en place dans la classe Translation. Pour ce faire, le constructeur de la classe est redéfini. Plutôt que de créer une nouvelle instance de la classe Translation son constructeur vérifie tout d’abord s’il existe déjà un objet de cette classe. Si tel est le cas l’objet existant est retourné, sinon une nouvelle instance est créée. De cette manière, il y aura toujours une seule et unique instance de la classe Translation dans le code.

L’implémentation du patron de conception singleton limite le nombre d’instance d’une classe à 1. Cela peut être utile dans certains cas, notamment lorsque l’instanciation d’une classe est couteuse en mémoire, il est alors préférable de garantir l’unicité de son instance.

En dernier lieu, nous avons encore un patron de conception nommé observateur. Il permet à un objet (le diffuseur) de notifier automatiquement plusieurs autres objets (les souscripteurs) lorsqu’un événement se produit.

Dans notre cas, le patron de conception observateur est formé par la classe BackupStateJournal qui a le rôle de diffuseur. Ce dernier est également relié à l’interface IObserver ainsi que sa classe héritière ConsoleView. Cette classe implémente une méthode qui retourne les informations comprises dans les notifications de la classe BackupStateJournal. En outre, le diffuseur implémente plusieurs méthodes qui ont pour objectif, entre autres, d’ajouter un objet à la liste des souscripteurs ou encore de notifier tous les souscripteurs.

L’implémentation du patron de conception observateur permet d’ajouter facilement des souscripteurs sans modifier tout le code. Effectivement, il nous suffit d’appeler la méthode adéquate du diffuseur. Un nouveau souscripteur sera alors ajouté à la liste de diffusion et notifié comme tous les autres souscripteurs. Le patron de conception observateur permet également d’automatiser les réactions dans le code sans devoir le modifier.

Enfin, l’utilisation de ces différents patrons de conception dans le code a encore pour objectif la réutilisabilité des différentes parties du code sans devoir le modifier.

* 1. Diagramme de séquence :

Le diagramme de séquence UML est un outil graphique permettant de représenter l’interaction entre plusieurs composants d’un système au fil du temps. Il est utilisé principalement en modélisation logicielle pour visualiser le flux d’exécution d’un programme.

Dans le projet EasySave, le diagramme de séquence est essentiel pour représenter le fonctionnement du processus de sauvegarde.

Un diagramme de séquence fait partie des diagrammes UML. Il permet de :

* Décrire l’ordre chronologique des interactions entre les composants.
* Visualiser la communication entre objets et acteurs d’un système.
* Aider à comprendre le flux d'exécution d'une fonctionnalité.
* Servir de référence pour le développement et la documentation technique.

Composé essentiellement de :

* Acteurs :

Représentés par un bonhomme   
Ils symbolisent un utilisateur ou un système externe qui interagit avec l’application.   
*Exemple concret :* Un utilisateur qui déclenche une sauvegarde.

* Classes/Objets et participants :

 Représentés par des rectangles en haut du diagramme   
Ils correspondent aux différents composants logiciels qui participent à l’exécution du scénario.

*Exemple concret :* Une chaîne de restaurants :

* L’acteur serait le client qui passe commande.
* Les classes/objets seraient les différents services du restaurant :
* Serveur (BackupController) qui prend la commande et la transmet à la cuisine.
* Chef cuisinier (LogController) qui prépare la commande et note les plats réalisés.
* Responsable de salle (MenuController) qui organise le bon déroulement du service.

* Messages et interactions :

Flèches dirigées entre les objets   
Elles indiquent les appels de méthodes et les retours de valeurs.

Il existe différents types de flèches :

*  : ce type de flèches représente des messages synchrones dont l’expéditeur attend une réponse de la part du destinataire avant de pouvoir continuer son exécution.
*  : ce type de flèches représente des messages asynchrones dont l’expéditeur n’attend pas de réponse de la part du destinataire avant de pouvoir continuer son exécution.
*  : ce type de flèches représente des messages asynchrones qui sont utilisés lorsque le destinataire d’un message synchrone répond à son expéditeur.
* Une image contenant blanc, Police, clé, ligne

  Le contenu généré par l’IA peut être incorrect. : ce type de flèches représente des messages asynchrones qui sont utilisés lorsqu’un nouvel objet d’une classe est créé.
* Une image contenant symbole, ligne, Symétrie, conception

  Le contenu généré par l’IA peut être incorrect. : un message avec une croix au bout de la flèche est utilisé lorsqu’un objet d’une classe est détruit.
* Barres d'activation :

Représentées par des rectangles verticaux sur les objets   
Elles indiquent qu’un objet est actif et en train d’exécuter une tâche.

* Boucles et conditions :

 Représentées par des blocs alt, loop, opt en UML   
Elles permettent d’ajouter des conditions et des répétitions.

Dans notre projet, nous nous sommes appuyés sur le diagramme de cas d'utilisation (Use Case Diagram). En effet, chaque bulle dans ce type de diagramme représente une action, ce qui nous a permis de structurer notre approche.

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

Ainsi, nous avons conçu cinq diagrammes de séquence, chacun illustrant un processus spécifique. Voici ces diagrammes accompagnés de leurs explications détaillées :

* 1. Diagramme séquence pour choisir les paramétrages et format des logs :

Une image contenant reçu, ligne, texte, diagramme

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme : [Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqlV1FPIzcQ_iuWXy6RQtpAICUPJx3Qu6MHB7oUqKpIyPE6iWFj79neEIr4L30kvyN_rDN2NrvZXeChSBBiz3pmvu-b8ewT5ToStE-t-JkKxcWJZBPDZkNF4Idxpw25ssIQZsmVk7G0zInUhO2EGSe5TJhy5FpaOYJ994im-beq5TmT6kaqSD-g5enXc_LFsGQqwT9p3Fx-btY8IlR6rJUzOo5DLNsr1Sc-azNjDi3Df1WLUzziWooQRfal3ndm9rbVOSAZZ2b-S9XuS6xHLL5mRrJRLCxal5caA6f5PZuIGiB-XzihrNRqG42a5eqzAz12D8yIG-b4NDx3slo6wR08SM70BAxFDNHDoswOCH-9AnY-fixw1yfHsafMpoYM6UA4J9XE3o7asM7vh5Qk2u9cMpATHGkgoyHd6ErOQUj-4NJS7iJsFOSCEeTK6pPBVD9kjhtN8LXZXC1JJEj8gfnkKm7L8ozEq1tF982t_If003gs-RTyIlFKZkA6HGQRc0j4xRlhM8e51jyI6y_-iEQAW_BckkoLAUvlhBkzDsEzwrWy2u8SBWnqR3QlXEgtXVcjkucX8kcnWT1VEt8WcCUqL1oM64fgabJaGqA9C02QGe7ifxYsQC-hsmxGNZ9qwM6g5ThUH6IR64mtDaNQIAX4Szt5hBjiaRG5LMQMAUtQZVPtKcB4MvZzQgoKLLnMMVk3jlfcvRjh889d_UwlieHXQuUR5CR3HxhCCJmyBHjhUHKglLp4NtGUOtHpNgIhOoznYgQl79MDQUhlHVNc5liAHAsBTgEnCYdHWnEUS-6kCkcpgFL5ZQGET6h1wcyAGyEUFGAjV2nkRfKnWLjbkV4MabOUZrErv-3iRNokZo-fPJe2sRFII1dgs7nNXblTbVVqkHOqgEhpXYg06BWx-oB6EanbqJnr2Ux4FcMxx3o20keYTwW2LKE3GuZgtYxDt0XnlSohjT8GF99_-ev8rFmLy_aV1ydfmYpisUGhET5uMav2QKAnEZ2qSCyaNdVXvihKNyy2upK7Ib3a7jiAahQdMX6fJpv7B3oBmwuOzWu-vtBIezGLiU5J-86iWhXkrUJhQi2sMQhobxpFORrIvnRLelLvNBBVPoSwdN2eoBYvmZtCTDr1F5GbOjFL7A8B7YVWKq_kocLwe_u1sBYorKqycG1efGtjsWS9dM5iGQnfSzGrhe_6WLwsMasXS8QC2pHHMM7UA4qD-kB8twDHNN_QpMcwFIHISLTILtYLTAGWrP4l3DwmLkvp_Vng4tuGx20F14wp_XzRHk-ZmohGvnDrPChl_b467dRNRzuviOccOjPmdodwY_qrZZ6_T3i1xPks4A2boXJbZK6hKyRx6qGO4B5gEi5F3-Lxqg7jM7QMu1qGiTqcVn_31KbyBs5r0QvoX8i6n9j8ZQRUYTZAwFxIfwz933SVBsa85Cqeh_QotVIJa7NnkDdolGtRV2qtdHTYLQ-or1GXzcdY-pVQPD2-acN9DISMYs392GQdMasl1JX1w2E21NQR5tlOXQ5igbFK5BWLfKZ9p4Hj9Jd4Jgw2TKhYLGbuqwDDr5-b3uvcFcWciezowmRKcFwByAALGOvkWK6WNrRum3IOkn9lUKrqtGhQGN1pi84EtGUZwcvlE5oOqZtCXkOKQUXM3KOLZ7BjqdODR8Vp35lUtGiaRHDa-kWU9scstrAqIgkvo-fhbdW_tLYovNfQ_hNd0H6nu9vePdzfO-j81u10u3udbos-0v7O7uFhe3-v2zvsdg46vc5B77lF_9Eajt1t78PiXm__192DXm-_0-u1qNHpZLrxCKf_7U0xrOf_ANyCXVY)

Explication :

Ouverture du menu des paramètres

* L'utilisateur clique sur "Settings", ce qui déclenche ShowSettings().
* L'interface (Visibility) gère l'affichage.
* L'interface affiche les options disponibles.

Récupération des formats de logs

* MenuView demande à MenuModel la liste des formats disponibles.
* Format et IMenuView récupèrent les méthodes nécessaires pour afficher ces formats.

Sélection du format de logs

* L'utilisateur choisit JSON ou XML.
* MenuController met à jour GlobalVariables.LogFilePath pour que les sauvegardes futures utilisent ce format.
* L'interface affiche une confirmation (Settings\_OK.Text).

Ajout des extensions à crypter (voir plus en détails dans le diagramme séquence cryptage)

* L'utilisateur saisit une liste d'extensions dans Extensions\_t.Text et clique sur OK.
* ExtensionController met à jour GlobalVariables.CryptedFileExt pour que ces types de fichiers soient cryptés lors des sauvegardes.

Ajout des logiciels métiers à surveiller (voir plus en détails dans le diagramme séquence logiciels métiers)

* L'utilisateur ajoute un logiciel métier dans BusinessSoftwareTextBox.Text et clique sur OK.
* SoftwareWatcher enregistre le logiciel dans GlobalVariables, ce qui permet de bloquer la sauvegarde si ce logiciel est en cours d’exécution.

Application des modifications

* MenuController applique les changements.
* L'interface affiche une confirmation.
  1. Diagramme de séquence pour le choix de la langue :

Une image contenant texte, diagramme, ligne, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Lien vers le [diagramme de séquence pour le choix de la langue](https://mermaid.live/edit#pako:eNq1Vt1O2zAUfhXLN1BROmihpRFCQjAQ2tAmOpg0VZpMctpacu3MdqAd4l12t_EcfbEdO0mbNikdF8tF659zPh9_5zsneaKhioAG1MCPBGQI55wNNRv3JcGHhVZpcmtAE2bIreWCG2Yh0el2zLTlIY-ZtOTy9srZXGoWj3jIROp1JS3oAQuh7HDNuPzKZaQey3t33PB7UeH0kclhwoZwpqTVSgjQ622u8WJi_fYdh0cX8dI8IFfFedn7i2bSCOTgtQhyI65kefP0syfK_V0qNRSwwCwbX2jMycjZZyMMcG7esxp_h9Oy23s5xEx5v3xY7ThPM3_AFZ-zdMlnb7dGdjCvATkTHNWhiUk0wYDvVWKVJH3quQLyjoiMtD5dQVzN8mLu0BezgPRG6jHn3mzXyuYnJ2RnKbcBuZm9hEk8e9Hg40IK2Hj2x2rIY3NYeUxLvmQXj98tnu_ZGYN13hux1sXlRBOQc25iwaaL28xHddIDAaGFKF-qLQfnVViK7QbGseB4R0YwmUhrBCblHEOqDCirIJLyur10hezI3KR0GhLEB9xRusWQDn6fWOTAm2MDsNM-xfNJCNqikyGzFzF7GYO0pirFXj83IKOU1nCk-MS5u6uk6nkoFnuuu5Ls-vTTBxIrHD0wwaM04RVoEWZxztMmzc3byQgHsE50hXIOFsWaWxd2U3OsbG8WJdyLxw9Dvw-ShHgF02g0ciW5PuAzUD4lE-N6BGL9KbYikNfU_a-AVYksFtwW7sEK_ZgRwyt5Xynd08HA1cGrMGT2C4NdrZiqGqxVvBnwjDgW8ypMYfDKeeKYsOQMBWSwsC6QntlvxjMJFytyAehZTftwQI6PQw0og5OT1AWEgTlc1nQ3gmV2FWgyWu4LRcfaQpJL1-2B_W6y1r5d7vb_Q91WJVht2IwsTCz-uSRuMaQdPwEWr7_NGt-EkwvTvFHqb8StbF3YevEDRm_GwlcGdsVksgIawdq3YWELDyutuWZI63QMesx4hB9pT86kT-0IxlgGAQ4jGLBEWFcTz2jK8MXcm8qQBlYnUKdaJcMRDQYM1VmnSRwhbPaFN1_FL4ZvSi3NafBEJzTY3W8c7LfanfbhXqfdPjrstg7rdEqDVrfR3Gs3293WfruDz9Fznf70EM1GZ69z1Do4OGrud5vdbuv5Ly3ajOc).

Explication :

Ce diagramme de séquence illustre le processus par lequel l’utilisateur peut choisir la langue d’affichage de l’application.

Sur l’interface graphique de l’application, plusieurs boutons permettent à l’utilisateurs de choisir une action à effectuer. Un de ces boutons concerne le choix de la langue utilisée pour les affichages dans le logiciel.

Au clic sur le bouton, une liste déroulante avec les différentes langues disponibles ainsi qu’un bouton « OK » sont affichés sur l’Interface Homme Machine. L’utilisateur peut alors choisir la langue dans laquelle il souhaite continuer sa navigation, puis cliquer sur le bouton « OK ».

Une fois que l’utilisateur a cliqué sur le bouton de validation, un message le prévient que la traduction est en cours.

Ensuite, un objet de la classe French ou de la classe English est créé selon la langue choisie. Celui-ci servira pour les traductions des différents affichages au sein de l’application.

Enfin, l’ensemble des textes pouvant être affichés dans l’application est traduit.

* 1. Diagramme de séquence pour la création d’une sauvegarde :

Une image contenant texte, diagramme, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le [diagramme de séquence pour la création d’une sauvegarde](https://mermaid.live/edit#pako:eNqtVttOGzEQ_RXLL4AIUQgRgX1ASqmEUItalUKlKi_GO0lcvHbwJUAR_9JH8h37Yx17c9lsNiDU5iW2d-bMzPGZ2X2iXKdAE2rhzoPi8FGwoWFZXxH8Me60IVcWDGGWXDkhhWUOvCkej5lxgosxU46cXZ0HmzPDxiPBmSy8zpUDM2Ac1h0umFA_hEr1_fqza2HFjaxx-sD4rR-HQLNVQs6L1SbbU62c0VKC2WRxgQTImoKkvmHymhnBMBW7bvDdMGUl0vFaiN7XSEv4O9N6KGHp1lcLksUE95GxytHGGJHdvR2yi7wn5FSKOx_CWW9In56afMqc0IqkQCzzExgyk0KfVtCrV7DcB-jlLiGXI31_aiCCbu-sW5-ckN3ZpZHCejukUTj06cxjbrGH6HtleLwCMRBYkdxizhlx4x2WEc1Rcu6xT0MlHIxDJ0vyqcynGShn6_KOjHwDlRogkhG-JGPLqzIfZFKW2ZzRJaG4L_j88omMNa4mTIo0ZAk2XDLL8hdncG0ZAtnC5tWIb9E8k-wbRFeFjekGDyjO515Vq7JvEqvt2dtV47JJ7IqkEGAGeClAlM4axGKRHBp4H9YJFfNsEPc4hjJQdA7FrWAFWWIedZSsuAV1zPMsx0dqlUY_rGYDQNm3zM_7UMqc7ZB1tN4v7R3Eqy6JKf8TDnBAYmOlURTzR3Yz-G5lyqBw8yn343xqos4IH0EmUEieDAQfhRYJLe00v2XDtTgEHjA8zoF5yBQWvV47zt5dbw1zhC1y2wRabXd8NhAmg2q3wCql9eqvGYnJck7i2Pm8ci0Mp4XLpzFMeI1NgGNXc56_2OZiMNVgFsFwbkfw1AuUzvuwC-gw-SMBm_OeCfN96MTFrNwrBVRp_1_BVufXXhyYvUHQQNDkvyCX9Xp1vnYWhhZtUBROxkSKHy1PwaRPHTYJvt0SXKYwYF66wP4zmjLv9OWj4jRxxkODGu2HI5oMmLS48-MUYWdfPHMTfG3_1DpbGOGeJk_0gSbt_U7zoHN8fNTpdlutdrfTbtBHmuw3u-1O66B9dHzY7bS6h4cHzw36O0LsN1uzX_tw_7jVOXj-CyvhT-E)

Explication :

Ce diagramme de séquence illustre le processus par lequel une sauvegarde peut être créée et enregistrée dans l’application.

Sur l’interface graphique de l’application, plusieurs boutons permettent à l’utilisateurs de choisir une action à effectuer. Un de ces boutons concerne la création d’une sauvegarde qui sera enregistrée dans le logiciel.

Au clic sur le bouton, plusieurs éléments qui n’étaient pas visibles jusqu’alors sont affichés. Cela comprend trois champs pour la saisi du nom de la sauvegarde, sa source, sa destination, ainsi qu’une liste déroulante pour le choix du type de la sauvegarde. Un bouton « OK » permettant de valider les paramètres entrés est également affiché.

Au clic sur le bouton de validation, les différents paramètres saisis par l’utilisateur sont récupérés. Un objet de la classe BackupModel est alors créé avec ces paramètres. L’instance de BackupModel, correspondant à la sauvegarde, est transmise au contrôleur.

Ensuite, la nouvelle sauvegarde sera ajoutée à la liste des sauvegardes, ainsi que dans le fichier de contenant toutes les sauvegardes enregistrées dans le logiciel via le contrôleur.

Pour finir, un message informera l’utilisateur de la création de la nouvelle sauvegarde.

* 1. Diagramme de séquence pour le blocage de l’exécution de sauvegardes :

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le [diagramme de séquence pour le blocage de l’exécution de sauvegardes](https://mermaid.live/edit#pako:eNqdVNtuGjEQ_RXLL0kUgrgusA-RSBpFqI1aBaFKFS_OegCrXnvrCyFB_E_pb_BjHe8SCkuSqt2Hle2ZOT5zZsZLmmgONKYWfnhQCXwQbGpYOlYEP5Y4bcjIgiHMkpETUljmwJvCnDHjRCIyphy5HQ2Cz61h2UwkTBZRA-XATFgCxwF3TKivQnH9eGy7Ysl3n11r5YyWEvC6HR8xRwJHwX_25OJsbxeToUPcO60EZiLU9MpbocDaoZ64R2bg9OxvAJ804_05E5I9SCiH7QiFbIujPG_EOEdFYnItBepqiPWGSCAP2jutyJh-_jim5IvGU1hs1olHmYhXQCzzc5gyw-E1XjnkffDPNmsDAdIG4Vi6-ekMrvnJFk7gLfx9uMtLcl4WOiY3CwhsCsNLmmW3wOU4dGDL6t57pVD0nVrSkZEiUk-x0CBJulk7gWhgHQFFEpTjMIUi7C0GuRr9yUQkMzwY03-Dpu-Bl3qgb8zmVyiRPNTX7glsCzyQFkjfJ_4VMurkvzItkbnZNYw8JEBO51rkbcaLIU4h74HNuhhtvBb_RoUJk282CjZUaMtMehsMHLlj89ttCUHxYsFh1_vH01pyKE_rnglreHQWJohWaAomZYLj27QMLmPqZpDCmMa45DBhXrpQxBW6Mpyq4ZNKaOyMhwo12k9nNJ4wLEaF-owj7PZhe3HBd-ab1unOCfc0XtIFjS_q1Va9GXWidq0TRd12r9mu0CcaN3vVRi1qRL1mPerg111V6HMO0ah2ap1us9XqNuq9Rq_XXP0G4OndtQ).

Explication :

Ce diagramme de séquence illustre le processus par lequel une sauvegarde peut être bloquée dans l’application en raison d’un logiciel métier qui est en cours d’utilisation.

Sur l’interface graphique, l’utilisateur a la possibilité de définir des logiciels métiers qui ne seront pas compatible avec l’exécution d’une sauvegarde.

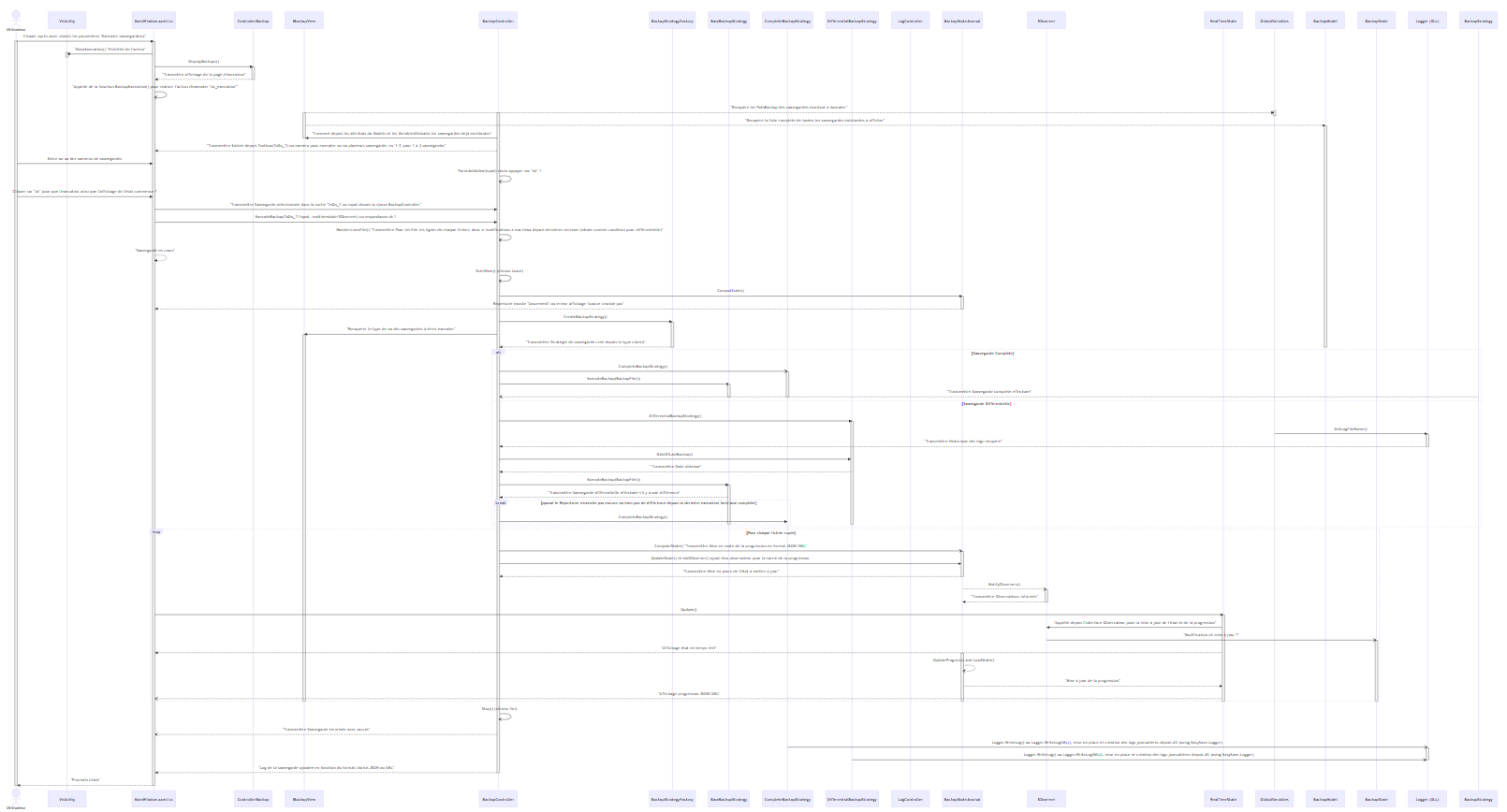
Ensuite, lorsque l’utilisateur clique sur le bouton de validation pour lancer l’exécution d’une ou plusieurs sauvegardes, une vérification est effectuée avant que la première sauvegarde ne se lance.

En effet, une méthode de la classe BackupController recherche les différents logiciels métier en cours d’exécution. Celui lui permet de vérifier si l’un des logiciels préalablement défini par l’utilisateur est en cours d’exécution.

Dans le cas où l’un des logiciels métier défini par l’utilisateur est en cours d’exécution, un message le lui fait savoir. L’exécution de sauvegarde sera arrêtée à cette étape.

Dans le cas contraire, la méthode permettent l’exécution de sauvegarde continue son exécution normalement. Le diagramme de séquence pour l’exécution des sauvegardes explique plus en détails le déroulement de l’exécution d’une sauvegarde.

* 1. Diagramme de séquence pour l’exécution des sauvegardes :



Lien vers le diagramme de séquence pour l’exécution des sauvegardes :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNrNWVtv2zgW_iscvcQG4rRJ0zTjhwG6TeeGpCkmaWexCDCgJdphI5NaUUrtLfpfBvtU79_wH5vvUDdKopxkgQU2D7HEy-G5n-9QX4JQRyKYBkb8MxcqFGeSL1K-vFEMfzzMdMo-GJEybtiHTMbS8EzkaTGd8DSToUy4ythHaeQM89maljZv_ZW__HxBSy64VL9LFenPByu-jJ-Fpr_0jVZZquNYpH_j4V2e0L7umOeAYuKjFJ9pg_PaX1vMNTRpQ3dsaNdVlkIZi_WPVk3rZmtnwrffiPbSYnN31KeRZRKLzLPbP9OncCbnc5EKlUke96kMz_YpnetFW3GtgWGtgdyvOk8Vj12VNaMei17O4IP3xSn1S39dKnicyaUwRI3W_oaBawxY8v31P8V6xuOPPJV8FgtDOzpDQ1JcIGoc9u3rTok7onq1uajVSE-js_Pz8Y0qVtoYnPzwg42eKXsTS4QrVifp9hv4vtcyZeGtRtgZRpKAMl9uv2Upnm-CtysR5hnWG57fiwVPIzEy45ugjnJ5TyzSIZ0hHFeM0Ll0fhPZU3Z1qz8XpKVWozEOqme3GxYJFu8RIa16J3XzQyQGp6qDu3E_ha-aJObr4tWMxp0z_Imil1ImpVanYP865WYpMmiN8flchrd8IawcHAqlx73tppK3kslh3X-iFWBcHfE6SQRWlFTnWlkFla7hKjNBMLB0u-HIubBcpUrwICpr3gT67g_hMFSy5CZAK17HqYmP37abME-23yAq-ct7nt2WConw2viJYWIFi2Tkots_WSk__KRr0l6arWe8IeWozTvfouDPyD0xnVi0IopaQs6gRRAKKUkiLKz-Mw1BinBx5YWIpcD08mfpCR6Je6HfKybWtRouSxdT5GNgIMnLYOXwODnLM8OinFl60Hpmp2qtFErysBttN5-2_26sBAX2PbNnnD6rTXJpmKRAeIsl242o-L0Wq2ymV6Nrfab_uB6zXDGVL7ebVBceWzsnJnTOkjg3EoDB5Xkfi9je4eTFXrHlkLT8whWqksDNelPLiSjpkuTluYZs6W4eEHDcG5vC7VODsjP7RUViNZIqybOxzaMrpNYkX1PKzItAuwnYdz6uqlTsrLNSYRBBW0cnA9wxshxtJxckFZQE8s0lIbDqmCrx9bluG-iqFp2Z7SYWNk0oazIsIt83GmVG0DZrNNKfFbX2Qc7CmBvjCbTgIV7K0lJMlF7xzFLfb5fjZ3XdhoZ1isKUaAUOIbC-q2R-nNXe5cuZSM-lEuZHGQtbeVyNvCf938M35FxS6qSokQslrKdA72QEMgAm91mEFMxgmaWOsDzkpD2EJBP5Crvof6mlSKRKUjoxDEIYu2wkI8rQojAe_qtIWmtbFwDBObggNEU5f9zSplN2HAsKBSIIl2rp4xQCSJFm78RnaGIU3qZaadhUhdvNeEdi8sCxqUWReWYHewV1CKl5CDnZBIUmEWkGgALpVsam4ZuA2BNw9wzhAncU8AcorImLm8CAFHxD7VFmw56E-xLbbp4GJG6hc8gMP-3i5kHhPcj-MWm_rkXkjixbJzb0y0TWqj6osf-xsO0Jpdbfj0x2ZA-jquxBm7abhRTtNMpCN-tXPJcQc9AQTjH0GMmjOyQIN4G9qepzAwJ8EeDvdaYD45UtO-hwuF0aDjzTO7Kd_4qfIimNvUDGPHBi3YxNvJE-mPobZCPQwYVZDutVZuqZY5iJFpod1hAQSuv0s3am26VG-ORwkzndMec140P9Kv11uzxioeixpuwnkeGRDPaOL4X3iGJpM1G2Z5PH1Oafkbp0SgDBBnqsF4YwPeWCDX4HDNQ98elKxO_l_JybrPTL8SCo7qtrxwXBoMzMFZpOZ3qWCZW3XHAgJ_9vY6p30TJ5IqbqFHEnvJjZkzFbAyvkylkXtoSeobjcAfZxFVEWdashlTZDqN0WN5R-wCJbE2YSMICGohZZB67VWIQ1AHPOpYXHTi5o2BjU_8CF0mNyqVCR130fE5OPSEc1-VjrpMB0bewGOROJIHogZ3cBQgVyRIlyOqa_QLtNMCyl7rBq_lO9QEkmzEdTc50uAdh_vbp89-z)

Explication :

Ce diagramme de séquence détaille le processus d'exécution d'une sauvegarde dans l'application EasySave, en mettant en évidence les interactions entre l'utilisateur et les différentes composantes du système.

1. Initiation de la sauvegarde :
   * L'utilisateur clique sur "Exécuter sauvegarde(s)" dans l'interface principale (MainWindow.xaml/cs).
   * L'interface appelle la méthode ShowExecution() de Visibility pour afficher la section d'exécution.
   * Ensuite, DisplayBackups() est invoqué sur ControllerBackup pour afficher les sauvegardes disponibles.
2. Sélection de la sauvegarde :
   * ControllerBackup transmet les informations des sauvegardes à l'interface pour affichage.
   * L'utilisateur saisit le numéro ou la plage de numéros des sauvegardes à exécuter.
   * Cette entrée est envoyée à BackupController via ExecuteBackup(input, realtimestate/IObserver).
3. Préparation de l'exécution :
   * BackupController analyse l'entrée avec ParseJobIndex(input) pour déterminer les sauvegardes sélectionnées.
   * Un chronomètre est démarré pour mesurer la durée de l'opération.
   * BackupStateJournal est sollicité pour calculer l'état initial de la sauvegarde via ComputeState().
4. Détermination de la stratégie de sauvegarde :
   * BackupStrategyFactory crée la stratégie appropriée (CompleteBackupStrategy ou DifferentialBackupStrategy) en fonction du type de sauvegarde requis.
   * Sauvegarde complète :
     + Tous les fichiers sont copiés.
     + BaseBackupStrategy exécute la sauvegarde en copiant l'intégralité des fichiers du répertoire source vers la destination.
   * Sauvegarde différentielle :
     + Seuls les fichiers modifiés depuis la dernière sauvegarde sont copiés.
     + BackupController récupère la date de la dernière sauvegarde à partir des journaux via Logger.
     + DifferentialBackupStrategy identifie les fichiers modifiés à copier.
     + BaseBackupStrategy exécute la sauvegarde en copiant uniquement ces fichiers.
     + Cas particulier : Si le répertoire cible n'existe pas ou si aucune différence n'est détectée, une sauvegarde complète est effectuée.
5. Exécution de la sauvegarde :
   * Pour chaque fichier copié :
     + BackupStateJournal met à jour l'état de la sauvegarde en temps réel.
     + Les observateurs (IObserver) sont notifiés des progrès.
     + L'interface (MainWindow.xaml/cs) affiche la progression en temps réel.
6. Finalisation :
   * Une fois la sauvegarde terminée, le chronomètre est arrêté.
   * BackupStateJournal met à jour l'état final de la sauvegarde en indiquant son achèvement.
   * Logger enregistre les détails de la sauvegarde (nom, source, destination, taille totale, durée) dans un fichier journal au format choisi voir diagramme paramétrage (JSON ou XML).
   * L'interface informe l'utilisateur du succès de l'opération et le redirige vers le menu principal.

* 1. Diagramme séquence pour le cryptage des fichiers choisis :

Une image contenant texte, diagramme, Parallèle, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme séquence : [Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNrVVt1u2zYUfhWCN7MBx3DjpnZ80YulaTYgboM6TYrNQ8FIJzZnidRIqk0W5F12WT2HXqyHpCzL-vGG3k2ADYnk-c4533d4yCcayBDojGr4KwURwBvOVorFS0HwYYGRinzUoAjT5KPhEdfMQKr8dMKU4QFPmDBkzri45SKUX-3SX3-ZkwvFkjVHVNK7vXrbb5qcPxgQmktxJoVRMoq8m5bhpu1FJO9YdMMUZ3cRaGtXH-otjAw2bAUtrn9mwSZN9v3Wx5pWb3kEcyYQ0hmcqcfEyIW8N8PKTNPsUq4Ki-Kt9-byEoPyKx27R69fVxickSVdMK65ITYR2BKiSf4PCaxXdENLifgX1MTh1IZ2iH6iopF12EK09VwO67M1Eyvo7QY-m-E1RtN3cawlRtiMo1O-NrltGDXdXPKFciTcT39JHecQWr4R7_c_lpTEXIMN6E-ZNqOpgfvZEA5He0CY84c8C1I0TAUQzdIvsGIqhAbuTo0a6fUqs5BgEf1Er49J3kiuSBKlmLsgYZ4ZxiNNQoYMRMiJ36Ex-s8zv2lJnnmQJgHNqvYLGjugPThkXCbcC3HPgzUHpYlGpgMkkx6AalH1Js8URxBmkO-6tNvKzrMdcIXPVh3rO_7oqN-awiXXBroces48YCRlQq4wO6xuZjtXkTIJkIQ884scvZEhC76DK8RhJCpdlWxVHJX29ql2kw7GPqCoSf5NAdnAI5ZGYkPDEnB4aLglqgzrIF0_5PhsDcHGWmzdKzeVKXCR2JkrZtaYMol-gjiJWAAxYNcDQ7DyNScqzxjSAnuRE-0zuUPuCqbqyTTLygZbCX5GrhXSfi9V7AIscAYkiPKs38FMo1N3sbLnaEk_vf9AzoUL3-r9Sao5mLUMe_163HWgtoq8Rqa05azkA3d6rMnC4LnxDr72-kfYA5MmeGVLdGZSnDJd0toGlkChX-K0S7HyV66BCpTKFm0PQbbaboVnuMB4M9OI_wJMYfKOxdAM_L_u1bnt5kiGKyS3kbYuyg6obbT6ADMHq__figxZ8wTOyK3iBvCj59L9vA1kQMpKq3kaVlnrKEAP3q5YV_9CcUAoWGFzwd10IPM6OETI5l6jEvgLJJ4ZSmCr_REyRv-n_EXoP9xL13GFbuuXr-3BjpWuYu6oah5K9ftVZap57NIBjRGL8RBv20_WYEnNGnvlklqXIVMb6-IZ17EUr5WPIqAzo1IY0DQJEbO4mdPZPUNVBxRCjrfzub--u1v8gOJ9k86e6AOdHU0mx8PpyfTkdDQdTcfT6asBfcThl-PRcDwZj8anxyP8G52-eh7Qv6VE3PFw8uL4eDQZTaan4_HkxcnLAVUyXa1Llwj_m1vqvp-_A6PsI4M)

Explication :

Le diagramme de séquence fourni illustre le processus par lequel un utilisateur configure les extensions de fichiers à chiffrer et exécute une sauvegarde où les fichiers correspondants sont chiffrés en conséquence. Voici une explication détaillée de chaque étape pour suivre la logique du diagramme :

1. Saisie des extensions à chiffrer :
   * Action : L'utilisateur saisit les extensions de fichiers qu'il souhaite chiffrer dans l'interface graphique principale (MainWindow) depuis le paramétrage (voir diagramme séquence).
   * But : Permettre à l'utilisateur de spécifier quels types de fichiers doivent être protégés lors des opérations de sauvegarde.
2. Transmission des extensions au contrôleur :
   * Action : MainWindow appelle la méthode ExtensionsChange du ExtensionController, en lui passant les extensions saisies (Extensions\_t.Text) une fois avoir cliqué sur « ok ».
   * But : Transférer les informations saisies par l'utilisateur au contrôleur responsable de la gestion des extensions à chiffrer.
3. Mise à jour des extensions chiffrées :
   * Action : ExtensionController met à jour la variable globale CryptedFileExt[] dans GlobalVariables avec les nouvelles extensions depuis l’action de paramétrage.
   * But : Stocker les extensions à chiffrer de manière accessible pour les opérations ultérieures de sauvegarde et de chiffrement.
4. Initiation de la sauvegarde :
   * Action : L'utilisateur déclenche une opération de sauvegarde via l'interface (MainWindow), depuis exécuter une sauvegarde.
   * But : Commencer le processus de sauvegarde des fichiers, incluant le chiffrement des fichiers avec les extensions spécifiées.
5. Appel au contrôleur de sauvegarde :
   * Action : MainWindow appelle la méthode ExecuteBackup() du BackupController.
   * But : Déléguer la gestion de la sauvegarde au contrôleur dédié, qui orchestrera les étapes nécessaires.
6. Copie des fichiers sources :
   * Action : BackupController commence par copier les fichiers sources vers l'emplacement de sauvegarde.
   * But : Créer des copies des fichiers à sauvegarder avant d'appliquer le chiffrement si nécessaire.

Remarque : voir la partie d’exécution plus en détails dans le diagramme séquence dédiée à celle-ci.

1. Vérification des extensions à chiffrer :
   * Action : BackupController consulte GlobalVariables pour obtenir la liste des extensions de fichiers à chiffrer.
   * But : Déterminer quels fichiers parmi ceux copiés nécessitent un chiffrement basé sur leur extension.
2. Traitement de chaque fichier copié :
   * Action : Pour chaque fichier copié, BackupController vérifie si son extension figure dans la liste des extensions à chiffrer.
   * But : Appliquer le chiffrement uniquement aux fichiers dont l'extension correspond à celles spécifiées par l'utilisateur.
3. Chiffrement des fichiers (si l'extension est à chiffrer) :
   * Action : Si le fichier a une extension à chiffrer :
     + a. FileManager récupère la clé de chiffrement depuis GlobalVariables.
     + b. FileManager utilise CheckFile() pour obtenir le chemin exact du fichier à chiffrer.
     + c. BackupController appelle FileManager.TransformFile(fichier, clé) pour chiffrer le fichier en utilisant une méthode XOR (XorMethod()).
     + d. Le temps de chiffrement est mesuré en millisecondes.
   * But : Assurer que les fichiers sensibles sont protégés en les chiffrant avant de finaliser la sauvegarde.
4. Journalisation du processus de chiffrement :
   * Action : Après le chiffrement, Logger enregistre le temps de chiffrement et le nom du fichier dans le fichier de log dont le chemin est spécifié dans GlobalVariables.LogFilePath.
   * But : Maintenir une trace des opérations de chiffrement pour des raisons de suivi et d'audit.
5. Journalisation des fichiers non chiffrés :
   * Action : Si le fichier n'a pas une extension à chiffrer, BackupController appelle Logger.WriteLog(0, fichier, GlobalVariables.LogFilePath) pour enregistrer que le fichier a été sauvegardé sans chiffrement.
   * But : Documenter toutes les opérations de sauvegarde, même pour les fichiers qui n'ont pas été chiffrés.
6. Finalisation de la sauvegarde :
   * Action : Une fois tous les fichiers traités, BackupController informe MainWindow que la sauvegarde est terminée.
   * But : Notifier l'utilisateur que le processus de sauvegarde (incluant le chiffrement des fichiers spécifiés) est achevé.
   1. Diagramme de séquence pour la consultation de l’historique :

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour la consultation de l’historique :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNp9VO1u2jAUfRXLfwpSoOWr0PyoNLVbt6l0k7p204RUXRJDvDl25tgUVvV92ufgxXbtEIIIHX-wr6_PPfecGz_RSMWMhjRnfyyTEbvkMNeQTiTBH0RGaXKXM00gJ3eGC56DYVYXxxlowyOegTTknud8iudm5VKrXT3z08exSxkDl9-5jNVjewmpOI7yeqpQ89zluv_66ZVQUxD3oDlMBfOJe6H6nWs1nxfdbFaNy-vr5kQWmb7T1vm55xiSC8FRFMzO9PoF8ReKaxIlCptDTlgRkSFdvxiN6wm9UDK3wuAFd-Y5062OfIHC-QJ7ISxVRFxNV7vSLiS3iXp0QI0m4m8P1q8kZkQcOQwla0X2xY_Zm0dlTVcjJJc8zwSsioJ7oJUF3pTWRqUQeX3TkKfMoAoEZjMeJTBnniCgQMUyKqSBXb47tCpwT6i572Po3PrABfsKJqlRO2j6_nB4woXnIbliZgN4AylDcRuQZUyIkvZMSS8tbjPrvSaxEPt1C7RaMwfpbKYNWys5vF_y3Dhj79evms84njIXch9hoXFsyS9Vfmsg_PBiFE1SklfQu_A7rvhko5Vd4Lg0Pt9-uTn-McZRpwduNXeslLkoLSkZkClEv232gLGHFf5a43ErjtslJmlIlQaEob0ByTE_YgEKkhsuveEBkVNNjDIgnL5uQrDbIlrtSQ52weag4_VrQAxwtCMgidLANS4MSzMkdMSWLLIOddsJEzn7X0_vbGSl66dUY1cCP2-H-q9uVYITsE6O-CixfFtdxrUJeGMwtl_6Tqx4EGhAU6ZT4DG-xE8uaUJNwlI2oY5VDPq3q_eMeWCNul3JiIZGWxZQm8WIs3m1aTgDVCOgLOb4co-Lp92_8AHF94-GT3RJw-5ZuzMcdEf9bn_QG3bOTgO6ouEQoyejzqB_2h31ev3O6Dmgf5VC0H67czoanYzOesOTYXc46AcUlZwn23II_dNn-v3zP4Q6Gx8)

Explication :

L’utilisateur initie la consultation des logs

L'utilisateur clique sur le bouton "Consulter les logs" après avoir choisi les paramètres. Cette action déclenche l’appel à la fonction Showlogs() de la classe Visibility, qui gère l’affichage des fenêtres de l’interface graphique.

Affichage de la page de consultation

La classe MainWindow.xaml.cs (IHM) envoie une requête à la classe logs en appelant la méthode DisplayLogs(). La classe logs exécute l'affichage de la page de consultation et retourne cette action à IHM.

Vérification du fichier log dans les variables globales

MainWindow.xaml.cs (IHM) interroge la classe GlobalVariables pour récupérer le chemin du fichier log en appelant LogFilePath(). GlobalVariables appelle ensuite la méthode GetLogFileName() depuis la DLL Logger pour déterminer le fichier log correspondant à la date du jour.

Vérification de l'existence du fichier log

Logger (DLL) exécute la fonction Exists() pour vérifier si un fichier log du jour est présent. Deux cas de figure peuvent se produire :

Cas 1 : Le fichier log existe

* + Logger informe IHM que le fichier log a été trouvé (format JSON ou XML).
  + Il transmet les logs contenant les informations suivantes :
    - Nom de la sauvegarde
    - État (en cours, terminé, annulé, etc.)
    - Source & destination
    - Nombre total de fichiers sauvegardés
    - Taille des fichiers
    - Horodatage de l'exécution
    - Temps d'exécution
    - Temps de cryptage (s’il y a eu cryptage de fichiers)

Cas 2 : Aucun fichier log disponible

* + Logger envoie un message indiquant qu’aucun log n’est disponible aujourd’hui.
  + IHM affiche un message à l’utilisateur :  
    *"Aucun log disponible aujourd’hui"*

Fin du processus

Logger, GlobalVariables, IHM et User terminent leur exécution et se désactivent. L’utilisateur peut revenir au menu principal.

* 1. Diagramme de séquence pour quitter l’application :

Une image contenant texte, capture d’écran, Police, diagramme

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour quitter l’application :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNptU21q20AQvcqwEGqDbGzLkmz9CJS0aQs1tIQQKIKyWU2iJfJK3Y80qfEBehSdQxfraNXGil390szOezNvPnZMVDmylBn84VAJfCf5vebbTAF9XNhKw7VBDdzAtZWlNNyi0_1zzbWVQtZcWdhwqW6kyqufXeinjxv4oHldSGKF0c2Xy_Ep5G1dl1JwKyvVYYYmATLVI87OoP1teY0whxQ-v3GHKkCUnt_Qb4lwWzlL2Ix9ddJa1BnrGXz9k_PzQY0pXBygAwCM3j9J-_12_KJfPlIqT3HkOpCdVrrwlbbNY9so3CJpRWNB8No63TaAFkoO27axBTUfupSjsQ_J20aUNIaibbAnHfR1Mn6twNMhlK8SkS7xcAol9YP-pkNjeuG0Juj0qnCWgtVofKooJEWXqLfoU-ZdVn6g6KOHA5wc99uDwVbO0KgM1LoSaAxZqEBUTpvTmklv1_eulYNcYBDuOrJ_Reb435kcPfUTZAHbEpbLnFZ-1wVlzBbUuYyl9Jtz_dBtzZ7iOC3T1bMSLLXaYcBcnRPP3_Ng6R0vDXkxl3Qim_6G_CkFjJabpTv2xNLJKpwmi3C-TJJlMp_PwiRgz-ReJOE0jNfxLF6vvTvaB-xXVRHvfBpHUTyLomgZrWbxahUFTFfuvnhJSfTffKi3938AC_JFUQ)

Explication :

L’utilisateur clique sur le bouton "Quitter"

* Acteur : User (Utilisateur)
* L’utilisateur appuie sur le bouton "Quitter" (Exit\_b) dans l’interface WPF.
* Cela déclenche un événement qui est capturé par la fenêtre principale (MainWindow.xaml.cs).

Capture de l'événement et appel de la méthode Exit()

* Classe impliquée : MainWindow
* La fenêtre principale (MainWindow) capture l’événement Click et exécute la méthode Exit().
* Ensuite, elle appelle la méthode système Application.Current.Shutdown() pour fermer l’application.

Fermeture de l'application

* Classe impliquée : Application
* L’application ferme tous les processus en cours, libère les ressources et stoppe l’exécution.
* Elle envoie un signal à MainWindow pour confirmer la fermeture.
* L’interface se ferme, et l’utilisateur voit l’application disparaître.

1. Documentation CryptoSoft V2.0:



1. Documentation de la Dynamic Link Library V2.0:



1. Documentation Utilisateur V2.0:

1. Documentation Technique V2.0 :



1. Release note V2.0 :



1. Conclusion

Le livrable 2 marque une évolution majeure d'EasySave avec deux versions distinctes mais complémentaires : EasySave 1.1 et EasySave 2.0. La version 1.1 améliore l'existant en intégrant le choix du format des logs (JSON/XML), garantissant ainsi une meilleure flexibilité pour les clients souhaitant rester sur la version console. De son côté, EasySave 2.0 introduit une interface graphique WPF, supprime la limite de 5 sauvegardes, intègre CryptoSoft pour le cryptage, optimise les logs et l’état en temps réel et gère la détection des logiciels métiers. Ces améliorations rendent l’application plus intuitive, performante et adaptée aux usages professionnels. Ce livrable répond aux attentes de ProSoft et de ses clients, en posant les bases pour les futures évolutions, notamment la version 3.0. EasySave est désormais une solution complète et évolutive, prête pour une adoption à grande échelle.