

EasySave Groupe 1

Alban Clavo
Matheo Pinget
Evan Joasson
Jonas Mionnet



SOFTWAI'RE

Contexte	1
Gestion de projet	2
Versioning	3
Architecture	4
Diagrammes UML	5
Version 1	6
Version 2	7
Version 3	8
Documentation	9
Conclusion	10



Contexte

Objectifs du projet :

- **développer un logiciel de sauvegarde performant et maintenable**

Enjeux :

- **Intégration dans la suite ProSoft et respect des standards**
- **Facilité d'utilisation et expérience utilisateur optimisée**
- **Maintenance et évolutivité du logiciel**

Gestion de Projet

Projets / EasySave

Sprint 2

V1.1 Modif logs (choix utilisateur XML ou JSON) V2.0 Interface Graphique : Remplacez l'interface console par une interface graphique. WPF ou un autre framework.

Rechercher

PM JM EJ CA

A FAIRE 3

- Digramme classe
- SCRUM-67 PM
- diagramme sequence
- SCRUM-68 CA
- Interface Graphique Logiciel Métier
- SCRUM-71 EJ
- + Créer un ticket

EN COURS 1

- WPF Cryptage
- SCRUM-72 JM

FINI 8

- Migration MVC vers MVVM
- SCRUM-61 JM
- Chiffrage sauvegardes
- SCRUM-62 PM
- Logs choix utilisateur XML JSON
- SCRUM-63 JM
- Résolution des bugs
- SCRUM-64 JM
- Ajout_interface_graphique
- SCRUM-65 CA

1

Suivi structuré des tâches

- Organisation des tâches sous forme de tickets
- Attribution des tâches aux membres de l'équipe
- Définition des priorités et des échéances

2

Gestion Agile efficace

- Supporte les méthodes Scrum et Kanban
- Planification et suivi des sprints
- Adaptabilité aux évolutions du projet

3

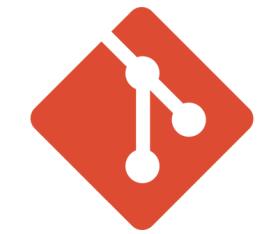
Collaboration et transparence

- Visibilité en temps réel de l'avancement des tâches
- Centralisation des informations et des discussions
- Intégration avec GitHub, Confluence et Slack

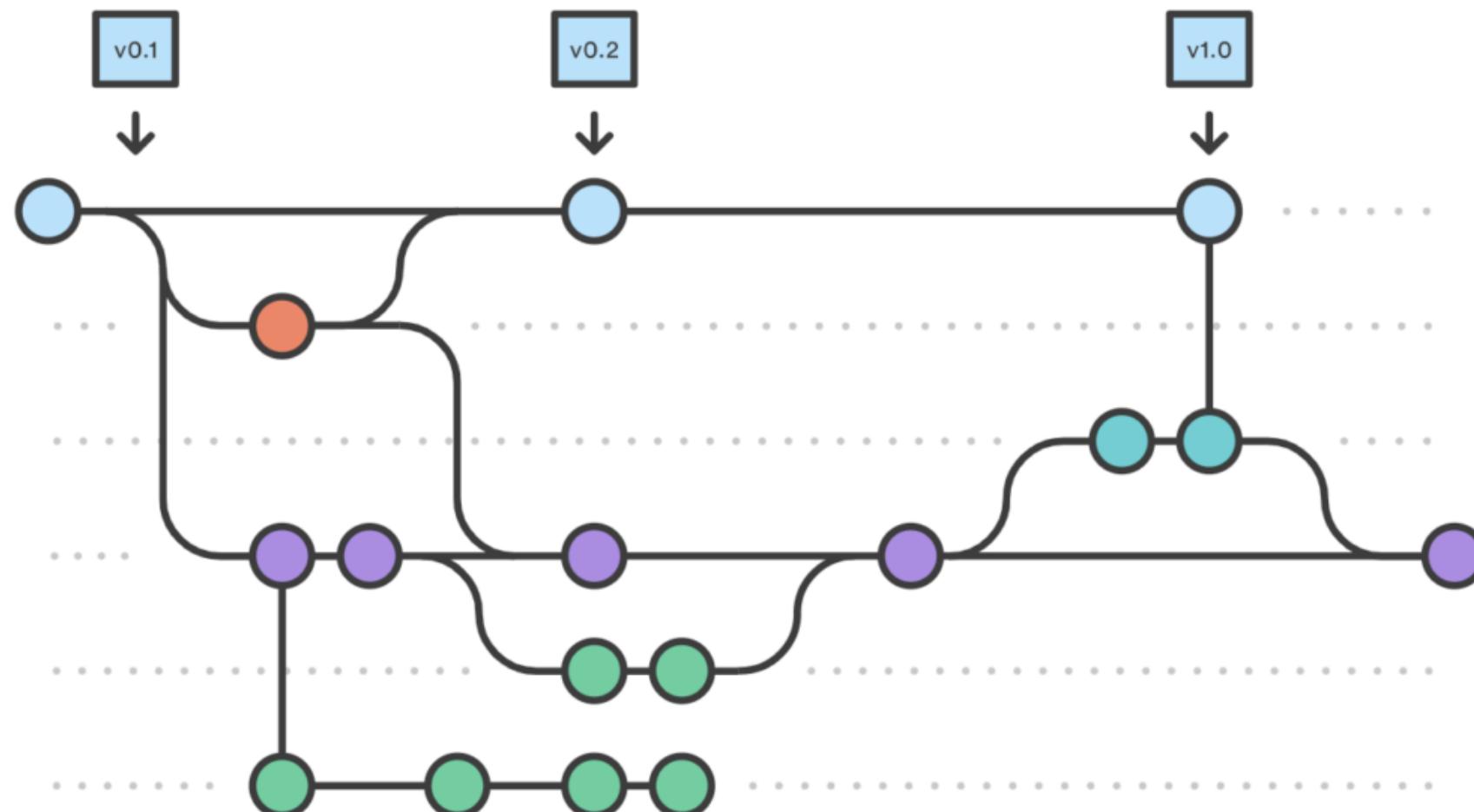
4

Suivi des bugs et évolutions

- Identification et traçabilité des anomalies
- Création de backlogs et roadmaps
- Génération de rapports et indicateurs de performance



Versioning



Organiser notre workflow de développement

- Création de branches dédiées pour chaque nouvelle fonctionnalité (feature branches)
- Utilisation de la branche dev pour centraliser le code à tester avant de le déployer en production

2

Suivre l'avancement du code au quotidien

- Intégration des modifications des développeurs via des merges dans les branches appropriées
- Gestion des versions grâce aux tags (par exemple v1.0) pour suivre les différentes étapes du développement

3

Faciliter la collaboration et la coordination

- Poussée des modifications dans des branches de feature spécifiques, permettant une meilleure séparation des tâches
- Utilisation des merges pour combiner les efforts de l'équipe et assurer la cohésion du code

4

Gérer les tests et la stabilité

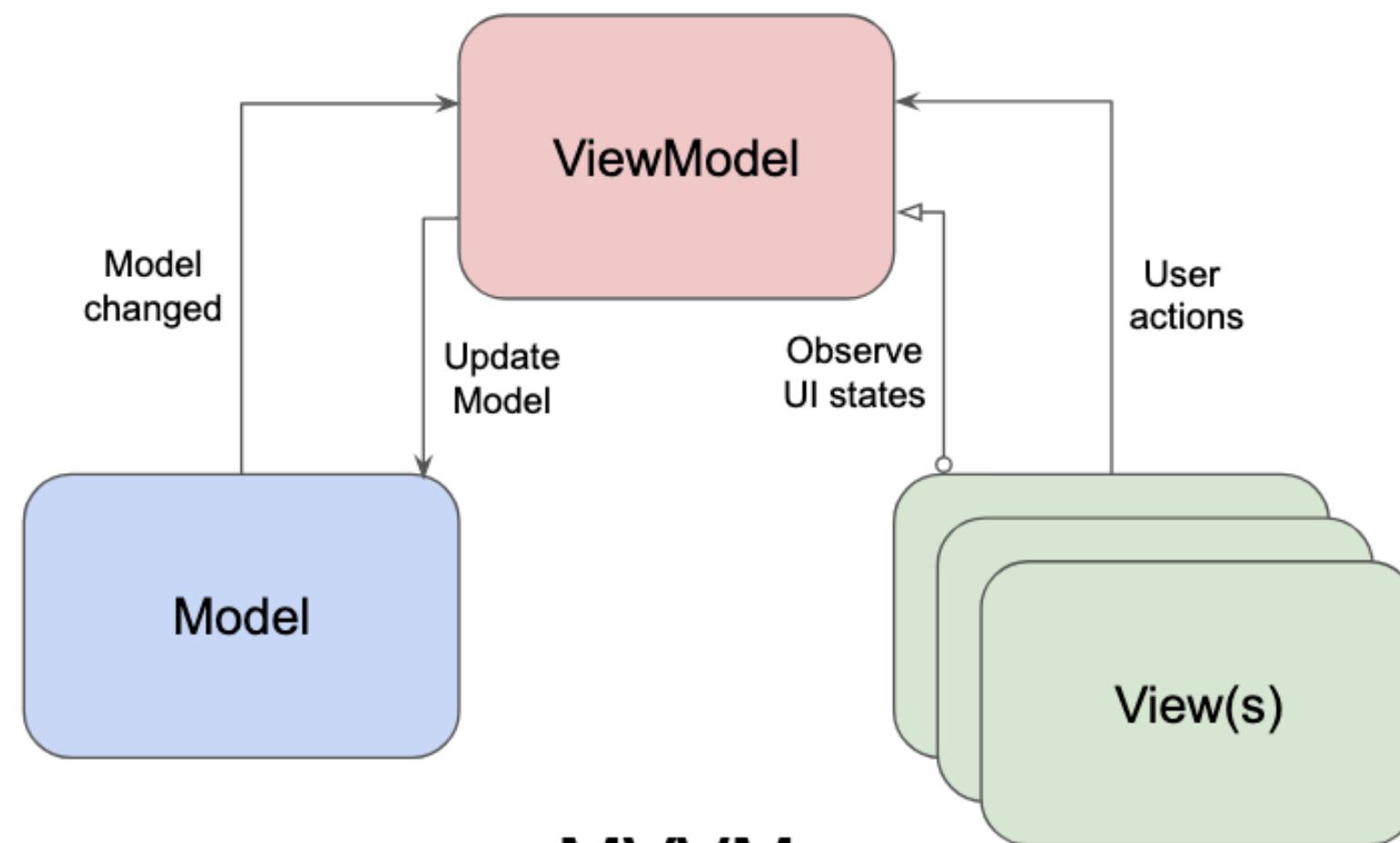
- Tests effectués dans la branche dev avant la mise en production
- Fusion des modifications dans la branche release pour préparer le code en vue d'un déploiement stable



Versioning

Figure 1 : Arborescence git

Architecture



MVVM

1

Séparation des responsabilités

- Divise le code en **Model (Données)**, **View (Interface)** et **ViewModel (Gestion des interactions)**.
- Facilite la **compréhension et la maintenance** du projet.

2

Facilite les tests et la maintenance

- La logique métier (Model) est **indépendante de l'interface graphique** (View), ce qui **simplifie les tests unitaires**.
- Le ViewModel permet de **modifier l'interface sans impacter la logique**.

3

Évolutivité et modularité

- Ajout de **nouvelles fonctionnalités** sans toucher au cœur du projet.
- Permet de **réutiliser du code facilement** entre plusieurs interfaces (View).

4

Gérer les tests et la stabilité

- Structure claire pour les **grands projets**.
- Séparation entre **les actions utilisateur et la gestion des données**.



Diagrammes UML

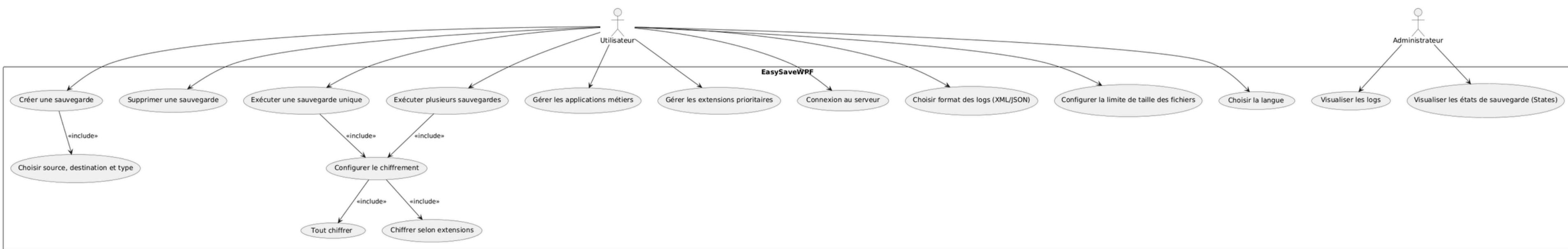


Figure 2 : Diagramme de cas d'utilisation

Diagrammes UML

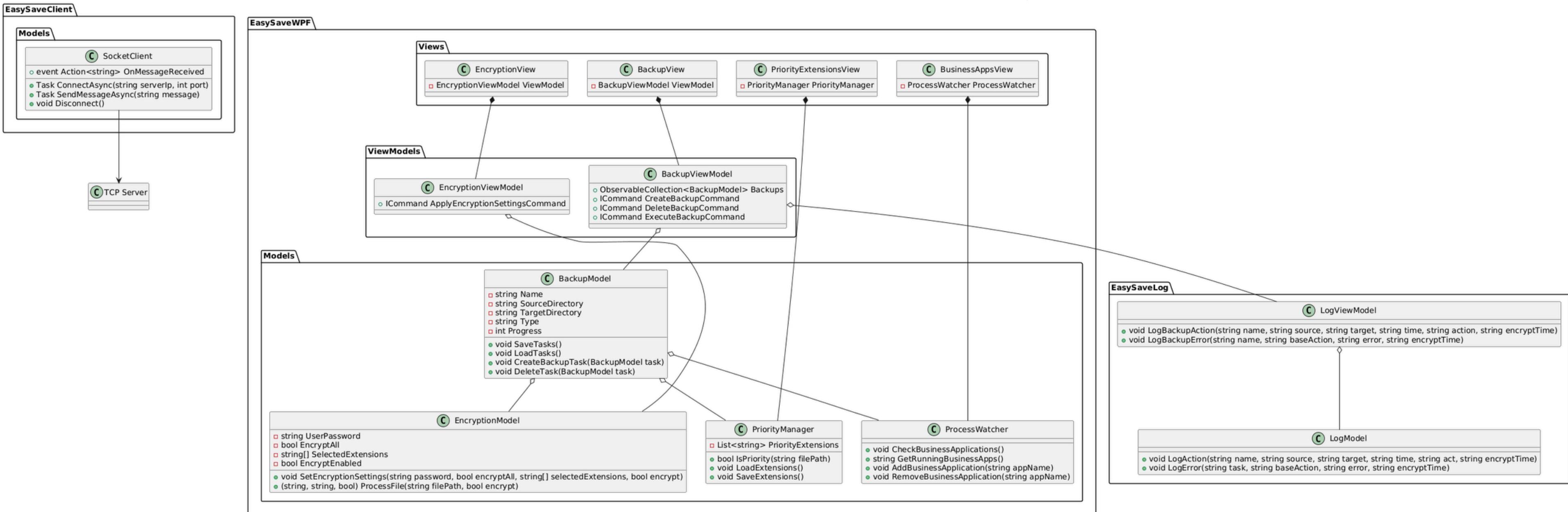


Figure 3 : Diagramme de classe

Version 1.x

```
$$$$$$$$$  
$$_--/_  
$$      /$$$$$$$  /$$$$$$$$$/ $$  /$$| $$ \_/_ /$$$$$$$ /$$ /$$$$$$$  
$$$$$ | ____ $$/ $$/_/_/| $$ | $$| $$$$/ | ____ $$/| $$ /$$//$$$_ $$_  
$$/_/ /$$$$$$$/| $$$_$/| $$ | $$ \____ $$/ /$$$$$$$$/ \ $$/$$/| $$$_$/$$$  
$$ /$$$_$/ \____ $$| $$ | $$ /$$ \ $$ /$$$_$/ \ $$ \ $$/| $$$_/_/  
$$$$$$$$/| $$$_$/ /$$$$$$$$/| $$$_$/$/| $$$_$/$/| $$$_$/ \$/| $$$_$/$/  
/_/_/_/_/ \____/_/| ____/_/ \____ $$/ \____/_/ \____/_/ \_/_/ \____/_/  
 /$$| $$  
| $$$$/$/  
 \____/_/
```

1. Creer une tache de sauvegarde
2. Executer une tache de sauvegarde
3. Executer toutes les taches de sauvegarde
4. Voir toutes les taches de sauvegarde
5. Supprimer une tache de sauvegarde
6. Changer le Format des Log
7. Quitter

Selectionnez une option: |

Contraintes

- C#
- .NET 8
- Persistance des données (JSON)
- DLL pour les logs

Attentes

- Interface CLI
- Sauvegarder un dossier
- Choix langue FR/EN
- Log
- State
- Format Logs (JSON/XML)(1.1)
- Actions sur les sauvegardes :
 - Créer
 - Supprimer
 - Lister
 - Exécuter

Version 2.x

The screenshot shows the software's configuration interface. On the left, there are several input fields and dropdown menus:

- Name: A text input field.
- Source: A text input field with a browse button (...).
- Destination: A text input field with a browse button (...).
- Type: A dropdown menu with options like "Add" and "View".

On the right, there are two more configuration sections:

- Language: A dropdown menu with an "Apply" button below it.
- Log file type: A dropdown menu with options "business software" and "Execute all".

Below these sections is a table view:

Name	Source	Destination	Type	
jonas	C:\Users\Claude\Desktop\AAAsource	C:\Users\Claude\Desktop\AAAdesti	Full	● ✖

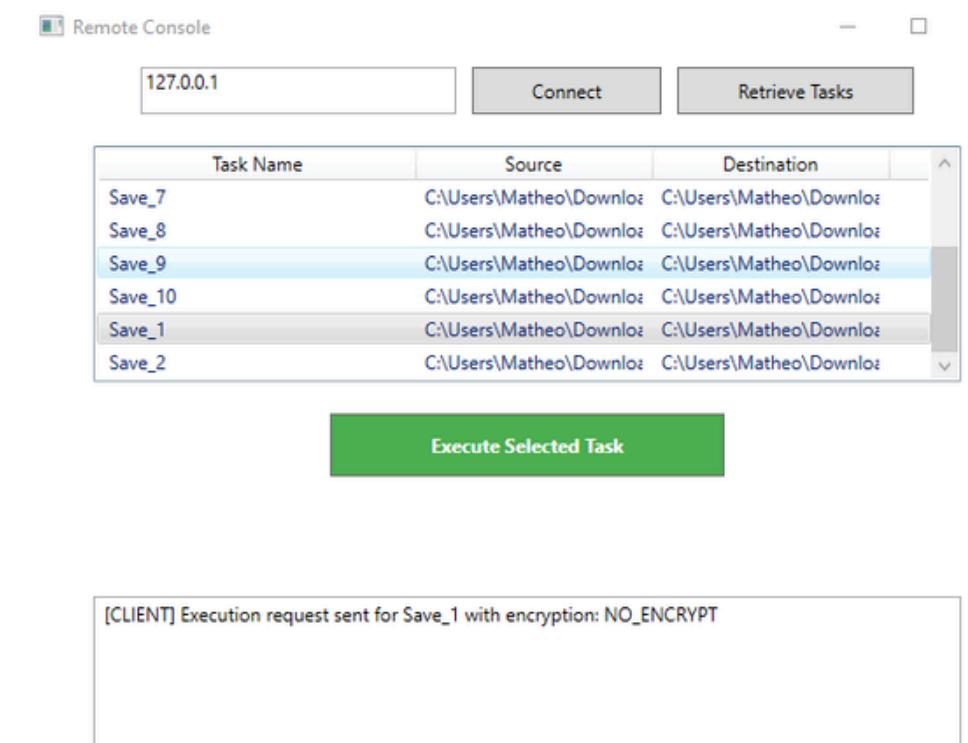
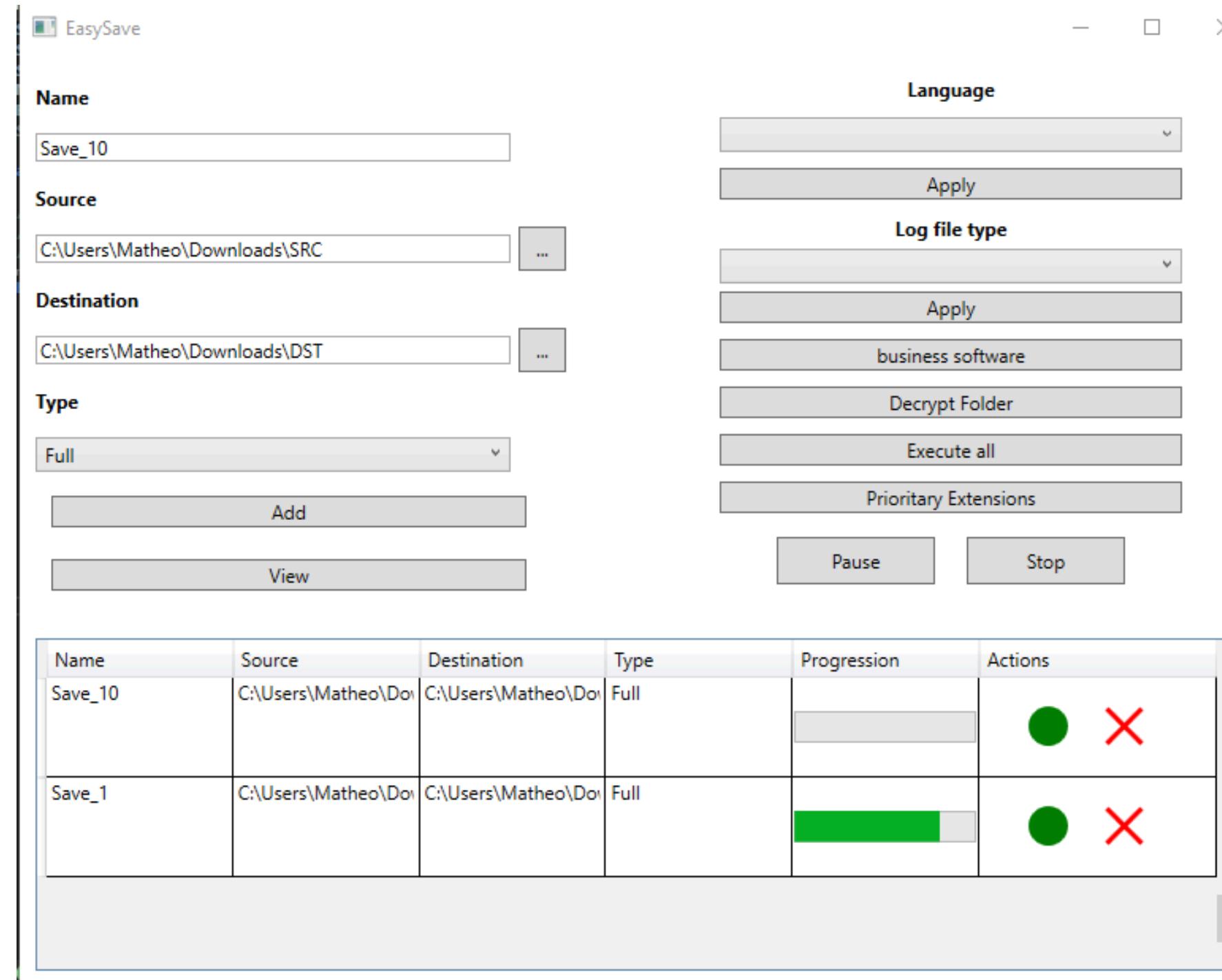
Attentes

- Interface graphique (WPF)
- Nombre de travaux illimités
- Chiffrement SHA256
- Détection de logiciel métier
 - Si un logiciel métier est détecté, les sauvegardes ne sont pas lancées

Version 3.x

Attentes

- Sauvegardes en parallèle
- Sauvegardes de fichiers prioritaires (extensions)
- Limiter la taille de fichiers transmis en même temps
- Suivi et intéraction :
 - Pause
 - Relance
 - Arrêt
- Pause temporaire si logiciel métier détecté
- Console déportée avec des sockets





Documentation

Documentation disponible sur : <https://groupe1-easysave.alwaysdata.net/index.html>

Documentation utilisateur : <https://github.com/Cesi-AlbanCalvo/EasySave#user-manual---version-20>



Points d'amélioration

- Gestion avancée de la bande passante
- Sauvegarde cloud et stockage hybride
- Interface utilisateur améliorée
- Sécurité renforcée
- Gestion des erreurs et résilience

CONCLUSION