



Rapport de stage

Développement d'interface logiciel et développement web

Réalisé par :

Dorian BARNIER

Sous la supervision de :

M. Jean-Marie BARTHELEMY (CEO)

et

M. Victor POUPET

Diplôme de BUT Informatique

Année universitaire 2024-2025

Remerciements :

À l'issue de ces huit semaines de stage, je tiens à exprimer ma profonde gratitude à toutes les personnes qui m'ont accompagné et soutenu tout au long de cette expérience.

Tout d'abord, un immense merci à M. Jean-Marie BARTHELEMY, mon tuteur de stage, pour son accueil, ses précieux conseils et son encadrement bienveillant, qui m'ont permis de progresser et de mieux appréhender le monde professionnel.

Je remercie également M. Leavan VICENTE, M. Guillaume ROBLES et M. Sam HOWELLS pour leur aide, leur patience et leur disponibilité, qui ont rendu mon intégration plus facile et mon travail plus enrichissant.

Enfin, je tiens à exprimer ma reconnaissance envers l'ensemble du corps enseignant de l'IUT Montpellier-Sète, dont l'enseignement et le soutien ont grandement contribué à ma formation et à ma préparation à cette expérience professionnelle, ainsi que M. Victor POUPET pour son suivi professionnel.

Résumé

Ce stage au sein de l'entreprise ECOSM s'intègre tout d'abord dans un projet de développement d'un nouveau logiciel utile au sous-titrage et l'édition de médias nommé «CreasySubs Editor». Mon objectif est d'assurer son accessibilité et sa flexibilité pour les utilisateurs expérimentés, notamment grâce à la réalisation d'un système de traduction du logiciel, ainsi qu'une gestion des raccourcis clavier, tous deux gérés via leur interface respective. Je travaille aussi sur la conception de plusieurs rubriques d'un site web nommé «Web MAM» visant à aider les clients pour la gestion de leurs médias.

L'application «CreasySubs Editor» est développée sous C# et WPF*, et le site Web MAM» est réalisé à l'aide des langages HTML, CSS, JavaScript et PHP.

Mots-clés : C#WPF, HTML, CSS, JavaScript, PHP, application, site web, accessibilité, ergonomie, traduction.

Abstract

My internship at ECOSM is part of a project to develop new software for subtitling and media editing called «CreasySubs Editor». My objective is to ensure its accessibility and flexibility for experienced users, especially through the realization of a translation system of the software, as well as a management of keyboard shortcuts, both managed via their respective interface. I am also working on the design of several sections of a website called «Web MAM» to help clients manage their media.

The application «CreasySubs Editor» is developed under C# and WPF, and the website «MAM» is built using HTML, CSS, JavaScript and PHP.

Keywords : C#WPF, HTML, CSS, JavaScript, PHP, application, website, accessibility, ergonomics, translation.

Table des annexes

Annexe 1 : Créateur de traductions.....	33
Annexe 2 : Créateur de traductions.....	33
Annexe 3 : Gestionnaire de raccourcis.....	34
Annexe 4 : Page Options Interface.....	34
Annexe 5 : Web MAM layout Grid, boutons switch de layout en light mode.....	35
Annexe 6 : Web MAM layout Grid, boutons switch de layout en dark mode.....	35

Table des figures

Figure 1 : Choix de l'utilisateur dans une ComboBox.....	14
Figure 2 : Valeur du choix de l'utilisateur.....	14
Figure 3 : Traduction sélectionnée par l'utilisateur dans une ListBox.....	15
Figure 4 : Valeur de la traduction sélectionnée par l'utilisateur.....	15
Figure 5 : DataGridView pour modifier les traductions.....	15
Figure 6 : Objet contenant les valeurs de traductions.....	16
Figure 7 : Bouton Annuler et Sauvegarder.....	16
Figure 8 : Erreur lorsque l'utilisateur essaye de supprimer une traduction sans sélectionner....	16
Figure 9 : Vérification du choix de l'utilisateur lors d'une suppression.....	17
Figure 10 : Construction d'un objet ShortcutsModels.....	19
Figure 11 : Construction d'un raccourci dans le fichier .cseshortcut.....	19
Figure 12 : ComboBox déjà existante.....	20
Figure 13 : DataGridView déjà existante sans la gestion des doublons.....	20
Figure 14 : Setter et binding convertir appliqués à l'arrière-plan.....	20
Figure 15 : DataGridView avec gestion des doublons.....	21
Figure 16 : Boutons Enregistrer sous, importer et exporter.....	21
Figure 17 : Fenêtre pour enregistrer un fichier .cseshortcut sous un autre nom.....	21
Figure 18 : OpenFileDialog permettant d'importer.....	23
Figure 19 : SaveFileDialog permettant d'exporter.....	23
Figure 20 : Boutons de sauvegarde et de sauvegarde par défaut.....	23
Figure 21 : Bouton de suppression.....	24
Figure 22 : Vérification de suppression pour l'utilisateur.....	24
Figure 23 : Erreur lorsque le fichier par défaut essaye d'être supprimé.....	24
Figure 24 : Exemple des multiples icones que j'ai réalisé.....	26
Figure 25 : Comment changer les classes CSS d'un item avec JavaScript.....	26
Figure 26 : Layout en liste en light mode.....	27
Figure 27 : Layout en liste en dark mode.....	27
Figure 28 : Rubrique de téléchargements pour chaque médias.....	27
Figure 29 : Logiciel de test MediaTools avec affichage du premier plan d'une vidéo.....	29
Figure 30 : Logiciel de test MediaTools avec affichage du neuvième plan d'une vidéo.....	29
Figure 31 : Schéma réalisé pour le créateur de traductions logiciel.....	30
Figure 32 : Schéma réalisé pour le gestionnaire de raccourcis claviers.....	31
Figure 33 : Schéma réalisé pour le responsive d'une page du menu Options.....	31
Figure 34 : Schéma réalisé pour les rubriques de téléchargements sur le site Web MAM.....	31
Figure 35 : Schéma réalisé pour le layout en liste du site Web MAM.....	31

Sommaire :

Glossaire :	6
Bibliographie	9
Introduction	10
1. Présentation de l'entreprise	11
1.1 Contexte.....	11
1.2 ECOSM.....	11
II. Analyse des missions	12
2.1 Présentation du sujet et analyse de l'existant.....	12
2.2 Contraintes légales.....	12
III. Rapport technique	13
3.1. Créateur de traduction logiciel.....	13
3.1.1 Cahier des charges.....	13
3.1.2 Implémentation technique.....	14
3.2. Gestionnaire de raccourcis clavier.....	18
3.2.1 Cahier des charges.....	18
3.2.2 Implémentation technique.....	19
3.3. Téléchargements et changement de layout.....	25
3.3.1 Cahier des charges.....	25
3.3.2 Implémentation techniques.....	26
3.4 Récupération des changements de plans.....	27
3.4.1 Cahier des charges.....	28
3.4.2 Intégration technique.....	28
IV. Méthodologie et organisation	30
Conclusion	32
Annexes	33

Glossaire :

Layout : Représente l'agencement des différents emplacements destinés au contenu d'une page web.

Plan : Unité de base du cinéma qui correspond à ce qui est filmé entre le moment où l'on déclenche la caméra et le moment où l'on arrête la caméra.

WPF : Windows Presentation Foundation (WPF) est la spécification graphique de Microsoft .NET 3.0. Il intègre le langage descriptif XAML qui permet de l'utiliser d'une manière proche d'une page HTML pour les développeurs.

PME : Entreprise de moins de 500 employés.

Cap Omega : Hotel d'entreprises.

Flux : On peut considérer un objet de flux comme un fichier intelligent qui agit comme une source et une destination pour les octets.

Flux de mémoire : Flux dont le magasin de stockage est la mémoire de l'ordinateur.

JSON : Format d'échange de données en texte lisible.

IDE : Une application logicielle qui aide les programmeurs à développer efficacement le code logiciel.

IA : Intelligence artificielle.

Intelligence artificielle : La capacité des machines à effectuer des tâches typiquement associées à l'intelligence humaine, comme l'apprentissage, le raisonnement, la résolution de problème, la perception ou la prise de décision.

Binding : Fournit un accès de haut niveau à la définition d'une liaison, qui connecte les propriétés d'objets cible de liaison.

ISO 639 : Norme internationale de l'Organisation internationale de normalisation (ISO) qui définit des codes pour la représentation des noms de langues.

DataGrid : Moyen flexible d'afficher une collection de données dans des lignes et des colonnes.

Workflow : Désigne une suite de tâches ou d'opérations qui doivent être réalisées par un individu ou un groupe d'individus selon un ordre spécifique.

Google Sheets : Application de tableur en ligne qui permet de créer et de mettre en forme des feuilles de calcul, et de les modifier en collaboration avec d'autres personnes.

Binding Converter : Propriété qui obtient ou définit l'objet de conversion appelé par le moteur de liaison pour modifier les données à mesure qu'elles sont passées entre la source et la cible, ou inversement.

Setter : Accesseur en modification.

Storage : Stockage.

Light mode : le « Light Mode » est une « polarité positive » : par exemple du texte sombre sur un fond clair.

Dark mode : Le « Dark Mode » est une « polarité négative » : un texte clair sur un fond sombre.

Refactoring : Le processus de restructuration d'un code, tout en ne modifiant pas sa fonctionnalité d'origine.

AJAX : Méthode ajoutée aux navigateurs web dont la particularité est de permettre d'effectuer des requêtes au serveur web.

Timecode : Référence temporelle utilisée dans les domaines du son et de l'image, pour la synchronisation et le marquage de matériaux enregistrés.

Timestamp : Horodatage

Byte : Unité d'information correspondant à un octet, soit 8 bits

Git : Système de contrôle de version distribué qui permet de suivre les modifications apportées aux fichiers et de coordonner le travail entre plusieurs personnes.

Semi-agile : Solution hybride qui permet d'utiliser les outils les plus adaptés au développement d'un projet Web, qu'ils soient traditionnels ou agiles

Bibliographie

[1] Disponible sur :

<https://www.yeschat.ai/gpts-9t557MrWFwS-JSON-Translator>. [Consulté le : 13-fevr-2025].

[2] Disponible sur : <https://www.photopea.com>.

[3] Disponible sur : <https://excalidraw.com>.

Introduction

Dans un contexte où les besoins en sous-titrage et en accessibilité des médias sont en constante évolution, les entreprises du secteur doivent développer des outils performants et adaptés. ECOSM, spécialisée dans la conception de logiciels pour les médias, a su se démarquer grâce à des solutions innovantes facilitant la gestion des contenus audiovisuels.

Ce stage s'inscrit donc dans un projet de développement consacré à un logiciel dédié à l'édition de sous-titres ainsi qu'un site web destiné à la gestion de médias. Ces projets m'ont permis d'explorer des défis techniques liés à l'optimisation, l'ergonomie mais notamment l'expérience du point de vue de l'utilisateur. Tout cela m'a permis de consolider mes compétences en développement informatique, mais aussi dans ma logique et la façon dont j'organise mes chemins de pensées.

Je vais donc présenter ce stage en structurant ce rapport en plusieurs parties. Tout d'abord, une présentation de l'entreprise ECOSM permettra de mieux comprendre son activité et ses enjeux. Ensuite, les différentes missions qui m'ont été confiées seront détaillées, avec une analyse du contexte et des solutions techniques mises en place. Puis une explication de la méthodologie et l'organisation utilisée au cours du stage. Enfin, une conclusion viendra synthétiser ce qui a été cité dans les parties précédentes avec un esprit critique sur les points à améliorer.

1. Présentation de l'entreprise

1.1 Contexte

Dans le cadre de ma seconde année de BUT Informatique, parcours RACDV (Réalisation d'application : conception, développement, validation), j'ai eu l'occasion de réaliser un stage d'une période de huit semaines au sein de l'entreprise ECOSM (European Cie Of Softwares For Medias).

1.2 ECOSM

ECOSM est une PME* de six employés basée à Cap Omega*.

Sous la direction de M. Jean-Marie Barthelemy, cette entreprise est spécialisée dans la conception de logiciels destinés aux médias via des méthodes semi-agile*. Sur les vingt dernières années, ECOSM a réussi à se démarquer dans le milieu du sous-titrage et de la diffusion via la réalisation de logiciels innovants.

II. Analyse des missions

Dans cette partie, nous aborderons le stage plus en profondeur avec en premier lieu la présentation du sujet de ce stage ainsi que l'analyse des problèmes techniques. Ensuite, nous passerons à l'analyse de l'existant tout en finissant par l'analyse des problèmes à résoudre.

2.1 Présentation du sujet et analyse de l'existant

Le logiciel principal sur lequel j'ai travaillé, «CreasySubs Editor», est une application développée en C# avec WPF, utilisée pour la gestion et l'édition des sous-titres. Les missions de ce stage comprenaient également des technologies web pour la refonte d'un site nommé «Web MAM».

Avant mon intervention, plusieurs limitations avaient été identifiées dans les solutions proposées par ECOSM. Le système de traduction de «CreasySubs Editor» était rigide, nécessitant des modifications manuelles des fichiers de langue, ce qui ralentissait le processus d'internationalisation du logiciel. De plus, la gestion des raccourcis claviers n'était pas personnalisable, ce qui limitait l'ergonomie et l'adaptabilité de l'application aux besoins des utilisateurs.

Par ailleurs, la structure du site web «Web MAM» nécessitait une refonte pour améliorer son accessibilité et son design. Enfin, dans le cadre des outils de gestion vidéo, il manquait une fonctionnalité permettant de récupérer et d'organiser les changements de plans dans des flux* de mémoire pour une analyse plus efficace.

2.2 Contraintes légales

Pour la création du logiciel «CreasySubs Editor», ECOSM doit appliquer les codes existants du monde du sous-titrage et de l'édition de vidéos.

Les sous-titres ont une durée minimum et une durée maximum d'affichage à l'écran.

III. Rapport technique

3.1. Créateur de traduction logiciel

3.1.1 Cahier des charges

L'un des objectifs premiers de ce stage a été l'amélioration du système de traduction du logiciel «CreasySubs Editor» afin de le rendre plus flexible et accessible aux utilisateurs. Avant mon intervention, la gestion des langues était limitée à des fichiers statiques, nécessitant des modifications manuelles dans le code source pour ajouter ou modifier des traductions. Cette approche posait plusieurs problèmes comme un manque de réactivité dans l'ajout de nouvelles langues, des difficultés pour les utilisateurs d'interagir avec le système et des risques d'erreur humaine lors de la modification des fichiers de langue.

Afin de rendre le système de traduction du logiciel plus efficace, il était nécessaire d'introduire une solution permettant une modification dynamique des langues et une gestion simplifiée des fichiers de traduction. Il a été décidé de réaliser une interface permettant d'ajouter, modifier et supprimer des traductions du logiciel directement depuis l'application, sans nécessité d'intervention sur les fichiers source.

Des fichiers JSON* déjà utilisés comme support principal pour stocker les traductions, il a été plus judicieux de garder ce format en raison de sa présence directe, mais aussi en raison de sa facilité à être manipulé.

Ma mission a été de développer cette interface utilisateur ([Annexe 1](#), [Annexe 2](#)). Celle-ci offre une visualisation en temps réel des modifications, permettant de valider immédiatement les changements apportés.

3.1.2 Implémentation technique

L'implémentation technique de cette interface s'est faite en plusieurs étapes.

Dans un premier temps, avant de passer aux fonctionnalités, il fallait quelques prérequis.

Tout d'abord, la réalisation de l'interface avec le langage C#WPF, pour cela, j'ai utilisé l'IDE* «Visual Studio» qui offre une utilisation de ces langages plus efficace, j'utilise aussi la classe «System.IO.File» pour toutes créations ou suppression de fichiers.

Comme il est possible d'ajouter n'importe quelle langue au logiciel, la création d'un fichier de traduction par défaut s'imposait.

Possédant déjà la traduction du logiciel en français et en espagnol, il me suffisait de créer un fichier de traduction par défaut avec toutes les traductions du logiciel en anglais à partir de la traduction française, ce processus étant très long pour un logiciel de cette envergure, j'ai décidé d'utiliser une IA[1]* capable de traduire des fichiers JSON tout en gardant la structure donnée.

Pour les fonctionnalités de l'interface, j'ai tout d'abord réalisé un bouton dédié à ajouter une traduction au logiciel.

Pour cela, je récupère la langue que l'utilisateur veut ajouter grâce à un binding*.



Figure 1 : Choix de l'utilisateur dans une ComboBox

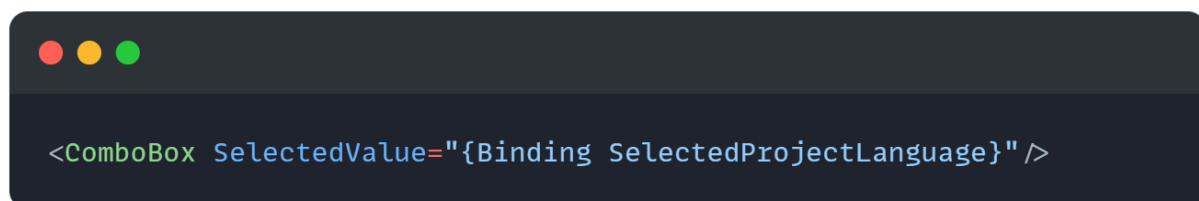


Figure 2 : Valeur du choix de l'utilisateur

Puis je copie le fichier de traduction par défaut disponible dans le projet et je l'ajoute directement à un répertoire localisé dans «C:\utilisateur\appdata\local» avec comme nom le code ISO 639* de la langue choisie, ce chemin-là est utilisé pour éviter toutes les erreurs de droits, car depuis l'application nous n'avons pas tous les droits sur la machine de l'utilisateur.

Le système pour le bouton dédié à supprimer une traduction est similaire sauf que celui-ci utilise un binding lié à la langue sélectionnée dans la liste des traductions en supprimant le fichier de traduction correspondant.

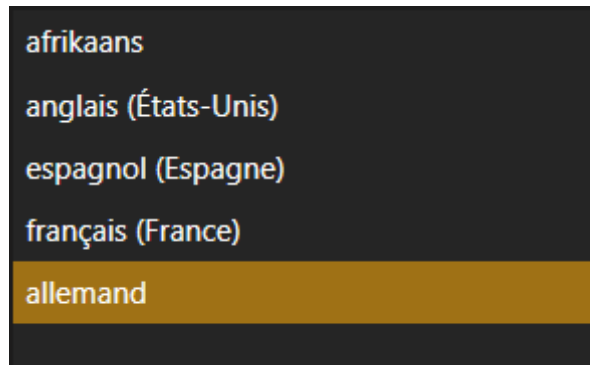


Figure 3 : Traduction sélectionnée par l'utilisateur dans une ListBox

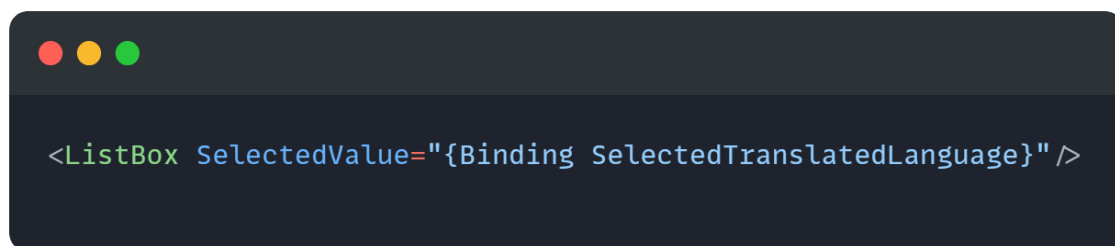


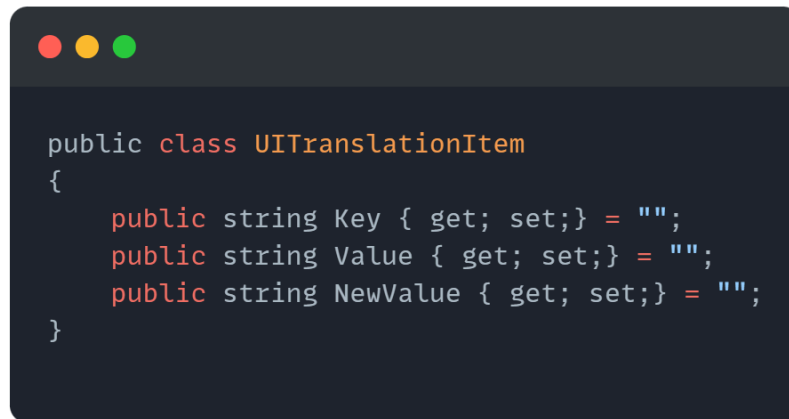
Figure 4 : Valeur de la traduction sélectionnée par l'utilisateur

Enfin, pour modifier les traductions, j'ai utilisé une DataGridView* contenant dans une première colonne les traductions présentes dans le fichier JSON, dans la deuxième un champ de texte pour inscrire les nouvelles traductions.

TRADUCTION PAR DÉFAUT	NOUVELLE TRADUCTION
New	<input type="text"/>
New Subtitle Project	<input type="text"/>
Open	<input type="text"/>
Open an existing subtitle project	<input type="text"/>
Save	<input type="text"/>
Save the currently opened project	<input type="text"/>
Save as...	<input type="text"/>

Figure 5 : DataGridView pour modifier les traductions

Les valeurs inscrites dans les champs de textes sont récupérées via un binding sur une collection d'objets contenant la clé de la traduction, l'ancienne valeur de la traduction ainsi que la nouvelle, c'est en récupérant cette nouvelle valeur que j'applique les modifications en parcourant le fichier de traduction et en modifiant les valeurs dont les clés correspondent à celles modifiées, j'applique cela via un bouton pour sauvegarder.

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains the following C# code:

```
public class UITranslationItem
{
    public string Key { get; set;} = "";
    public string Value { get; set;} = "";
    public string NewValue { get; set;} = "";
}
```

Figure 6 : Objet contenant les valeurs de traductions



Figure 7 : Bouton Annuler et Sauvegarder

Les fonctionnalités principales maintenant finies, j'ai pu utiliser mon temps pour réaliser la gestion des erreurs provenant du programme, mais aussi de l'utilisateur via des boîtes de dialogues.

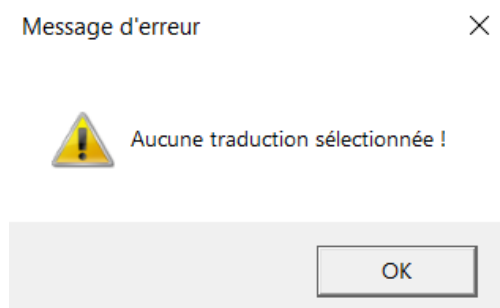


Figure 8 : Erreur lorsque l'utilisateur essaye de supprimer une traduction sans sélectionner

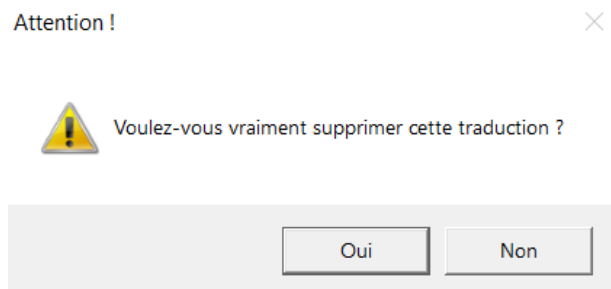


Figure 9 : Vérification du choix de l'utilisateur lors d'une suppression

Grâce à cette refonte, le système de traduction est devenu entièrement dynamique, permettant aux utilisateurs de modifier ou d'ajouter de nouvelles langues sans nécessité d'intervention sur les fichiers natifs. La nouvelle interface de gestion a rendu l'édition des traductions plus intuitive, réduisant ainsi les risques d'erreurs et offrant une meilleure expérience utilisateur.

Des améliorations futures sont possibles notamment via la réalisation d'un champ de recherche pour retrouver rapidement une clé de traduction.

3.2. Gestionnaire de raccourcis clavier

3.2.1 Cahier des charges

Comme deuxième mission de ce stage, j'ai été chargé de réaliser l'amélioration du gestionnaire de raccourcis clavier de «CreasySubs Editor» afin de le rendre plus flexible et personnalisable pour les utilisateurs. À l'origine, les raccourcis claviers étaient statiques et imposés par le logiciel, sans possibilité de modification par l'utilisateur. Cette rigidité limitait l'ergonomie du logiciel et contraignait les professionnels à adapter leur workflow* aux raccourcis existants, au lieu de pouvoir les configurer selon leurs habitudes.

Afin de répondre à ces enjeux, il était nécessaire de concevoir une interface ([Annexe 3](#)) de gestion des raccourcis clavier, offrant aux utilisateurs la possibilité d'attribuer librement des combinaisons de touches aux différentes actions du logiciel. Cette interface devait également intégrer un système de détection des conflits, permettant d'alerter l'utilisateur en cas de doublons, un moyen d'exporter ([Annexe 6](#)) des fichiers de raccourcis comme les importer ([Annexe 5](#)) et pouvoir les utiliser comme raccourcis par défaut.

Il a été décidé pour le stockage des raccourcis d'utiliser des fichiers «.cshorcut», qui sont en réalité des fichiers XML.

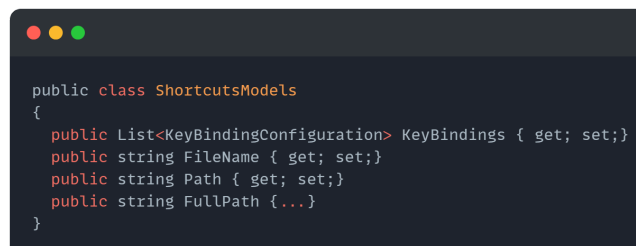
3.2.2 Implémentation technique

La réalisation de cette interface est aussi faite avec le langage C#WPF, j'utilise également la classe «System.IO.File» pour toute création ou suppression de fichiers.

L'implémentation technique de cette mission était déjà partiellement faite, l'interface était déjà présente néanmoins elle n'était pas complète et aucune fonctionnalités n'étaient implémentées.

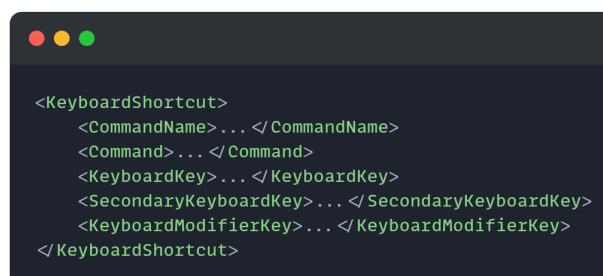
Une ComboBox contenant les différents fichiers, ainsi qu'une DataGrid affichant le contenu du fichier sélectionné était déjà présente sous forme d'objets «ShortcutsModels» contenus dans une liste qui sera mise à jour à chaque modifications, cependant les fichiers contenant les raccourcis n'étaient que partiellement fait avec seulement une vingtaine de raccourcis présents tous sans nom associés permettant d'expliquer l'action produite par le raccourci en question.

À l'aide d'un Google Sheets* listant tous les raccourcis qui devaient être présents, j'ai donc ajouté un par un, un total de 83 raccourcis au fichier contenant les raccourcis clavier, dans les fichiers de traductions du logiciel anglais, français et espagnol.



```
public class ShortcutsModels
{
    public List<KeyBindingConfiguration> KeyBindings { get; set; }
    public string FileName { get; set; }
    public string Path { get; set; }
    public string FullPath { ... }
}
```

Figure 10 : Construction d'un objet ShortcutsModels



```
<KeyboardShortcut>
  <CommandName>... </CommandName>
  <Command>... </Command>
  <KeyboardKey>... </KeyboardKey>
  <SecondaryKeyboardKey>... </SecondaryKeyboardKey>
  <KeyboardModifierKey>... </KeyboardModifierKey>
</KeyboardShortcut>
```

Figure 11 : Construction d'un raccourci dans le fichier .cseshortcut



Figure 12 : ComboBox déjà existante

NOM	COMMANDE	TOUCHE	TOUCHE SECONDAIRE	TOUCHE DE MODIFICATI
Ouvrir un fichier vidéo	OpenVideoCommand	X		Alt
Fermer un fichier vidéo	CloseVideoCommand	Cancel		
Jouer la vidéo	PlayCommand	Clear		
Arrêter la vidéo	StopCommand	FinalMode		
Jouer/arrêter la vidéo	TogglePlayPauseCommand	Return		Alt
Mettre en pause la vidéo	PauseCommand	K		Alt
Image précédente avec son	PrevFrameCommand	Delete		Alt
Prochaine image avec son	NextFrameCommand	0		Alt

Figure 13 : DataGrid déjà existante sans la gestion des doublons

Pour les fonctionnalités de la page, j'ai tout d'abord ajouté la gestion des doublons entre les différents raccourcis clavier, j'ai pour cela réalisé un binding `converter*` et un `setter*` sur le background de chaque rangée.

```
<Setter Property="Background">
  <MultiBinding Converter="{StaticResource DuplicateShortcutsToBrushConverter}">
</Setter>
```

Figure 14 : Setter et binding converter appliqués à l'arrière-plan

Le code de ce `converter` vérifie si le raccourci sur la rangée est déjà utilisé dans une autre rangée, si oui, il affiche l'arrière-plan de la rangée en rouge sinon il le laisse en transparent.

NOM	COMMANDE	TOUCHE	TOUCHE SECONDAIRE	TOUCHE DE MODIFICATI
Ouvrir un fichier vidéo	OpenVideoCommand	X		Alt
Fermer un fichier vidéo	CloseVideoCommand			
Jouer la vidéo	PlayCommand			
Arrêter la vidéo	StopCommand			
Jouer/arrêter la vidéo	TogglePlayPauseCommand	Return		Alt
Mettre en pause la vidéo	PauseCommand	K		Alt
Image précédente avec son	PrevFrameCommand	Left		Alt
Prochaine image avec son	NextFrameCommand	Right		Alt

Figure 15 : DataGridView avec gestion des doublons

J'ai ensuite réalisé un moyen de pouvoir enregistrer sous un autre nom, importer et exporter un fichier de raccourcis clavier.



Figure 16 : Boutons Enregistrer sous, importer et exporter

Pour enregistrer le fichier sous un autre nom, j'ai créé une petite interface contenant seulement un champ de texte pour le nom du fichier, celle-ci récupère la valeur de ce champ et crée un fichier contenant les raccourcis enregistrés directement dans le répertoire de fichiers de raccourcis dans «C:\utilisateur\appdata\local» pour éviter tous les problèmes de droits utilisateurs.

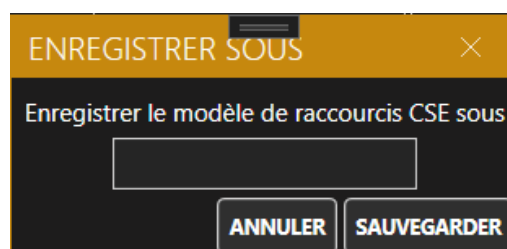


Figure 17 : Fenêtre pour enregistrer un fichier .cseshortcut sous un autre nom

Les boutons importer et exporter sont sur la base de classes déjà présentes dans la librairie «System.Windows.Forms» nommées «OpenFileDialog» et «SaveFileDialog». Celles-ci permettent d'ouvrir une fenêtre directement depuis l'explorateur Windows pour importer ou exporter un fichier, elles possèdent toutes les deux un chemin, pour importer je récupère les données au chemin

précisé tandis que l'exportation se fait par la création d'un fichier au chemin donné.

La classe «System.IO.File» ayant comme limite le remplacement de fichiers j'ai dû gérer moi-même si l'utilisateur veut remplacer un fichier déjà existant.

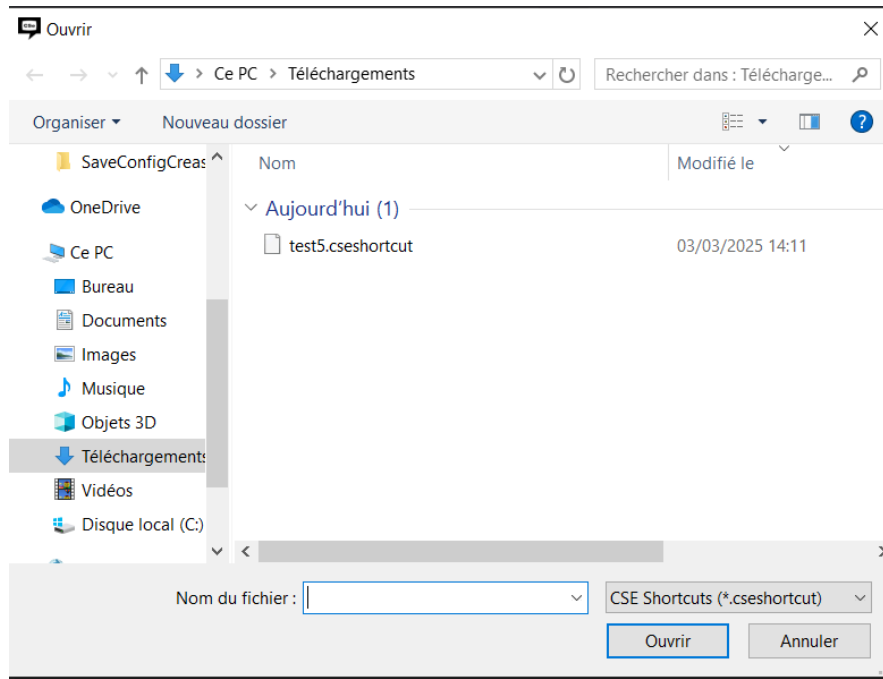


Figure 18 : OpenFileDialog permettant d'importer

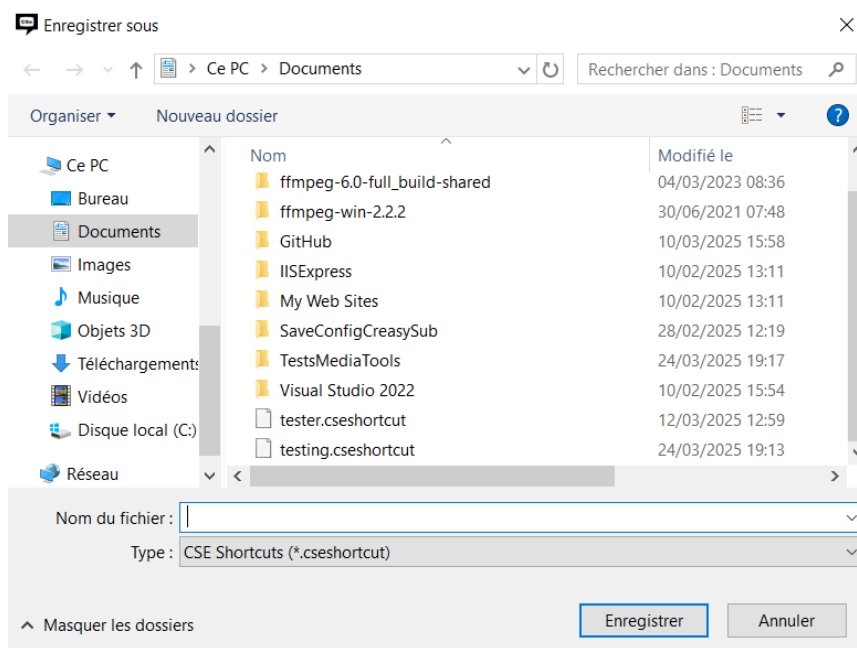


Figure 19 : SaveFileDialog permettant d'exporter

Par la suite, j'ai ajouté un moyen de supprimer, sauvegarder et sauvegarder les raccourcis dans le fichier par défaut du projet.

Le bouton «supprimer» supprime directement le fichier en cours et affiche directement celui par défaut, le fichier par défaut n'est pas supprimable. Le bouton de sauvegarde n'est pas très différent de celui pour sauvegarder sous un autre nom, il n'a juste pas besoin d'entrée de l'utilisateur.

Le bouton pour sauvegarder les raccourcis par défaut est légèrement différent car celui-ci ne sauvegarde pas les changements dans le fichier sélectionné, mais dans le fichier de configuration par défaut du projet, avec comme format «.csproj».



Figure 20 : Boutons de sauvegarde et de sauvegarde par défaut

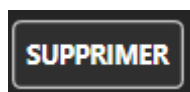


Figure 21 : Bouton de suppression

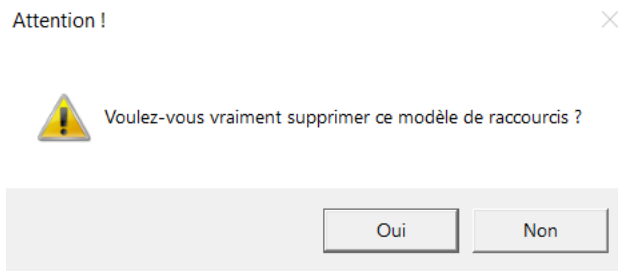


Figure 22 : Vérification de suppression pour l'utilisateur

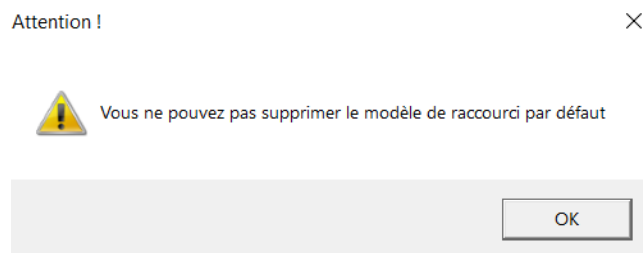


Figure 23 : Erreur lorsque le fichier par défaut essaye d'être supprimé

3.3. Téléchargements et changement de layout

3.3.1 Cahier des charges

Une mission plus légère de ce stage a été la refonte du layout* du site web «Web MAM» afin de moderniser son interface. Le site possédait un moyen de changer le layout cependant celui-ci était mal fait et décousu.

L'absence d'une mise en page moderne et adaptée aux nouveaux standards du web posait plusieurs problèmes : un manque d'ergonomie et une lisibilité pas du tout optimale.

Afin de répondre à ces problématiques, il était nécessaire d'effectuer une refonte complète du layout du site, en mettant l'accent sur une structure plus fluide, un design modernisé.

Mon rôle a été de concevoir et d'implémenter cette nouvelle interface ainsi que d'ajouter une nouvelle rubrique qui permet aux utilisateurs de télécharger les médias leur appartenant depuis n'importe quel storage*.

3.3.2 Implémentation techniques

La refonte du layout a impliqué une réécriture du code HTML et CSS, ainsi qu'une refonte des scripts en JavaScript. Le langage PHP a également été utilisé pour la gestion dynamique du contenu.

En premier lieu, la façon dont il fallait changer le layout du site web était faite par un bouton situé dans un menu déroulant, cependant cela n'était pas très moderne et ergonomique.

Même si cela n'était pas dans le but principal de ma mission, j'ai proposé à mon tuteur de me laisser m'inspirer d'applications plus modernes en créant deux boutons, un pour récupérer les médias dans une grille([Annexe 9](#), [Annexe 10](#)), un pour récupérer les médias dans une liste, l'idée lui a plu, il m'a donc laissé m'en occuper.

J'ai donc réalisé à l'aide de Photopea[\[2\]](#), un outil d'édition de photo, plusieurs images pour les boutons, en light* mode et dark* mode que j'ai relié à des boutons permettant de changer le layout du site grâce aux changements de classes CSS via JavaScript.

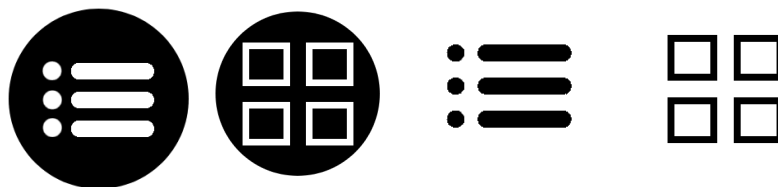


Figure 24 : Exemple des multiples icones que j'ai réalisé

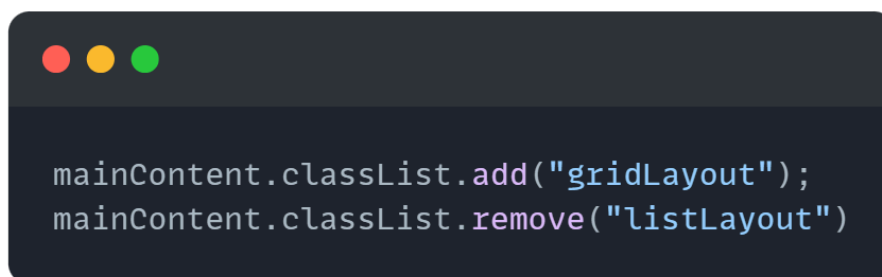


Figure 25 : Comment changer les classes CSS d'un item avec JavaScript

J'ai ensuite fait du refactoring* au niveau de tout le CSS du site web pour récupérer un layout en liste correct.



Figure 26 : Layout en liste en light mode

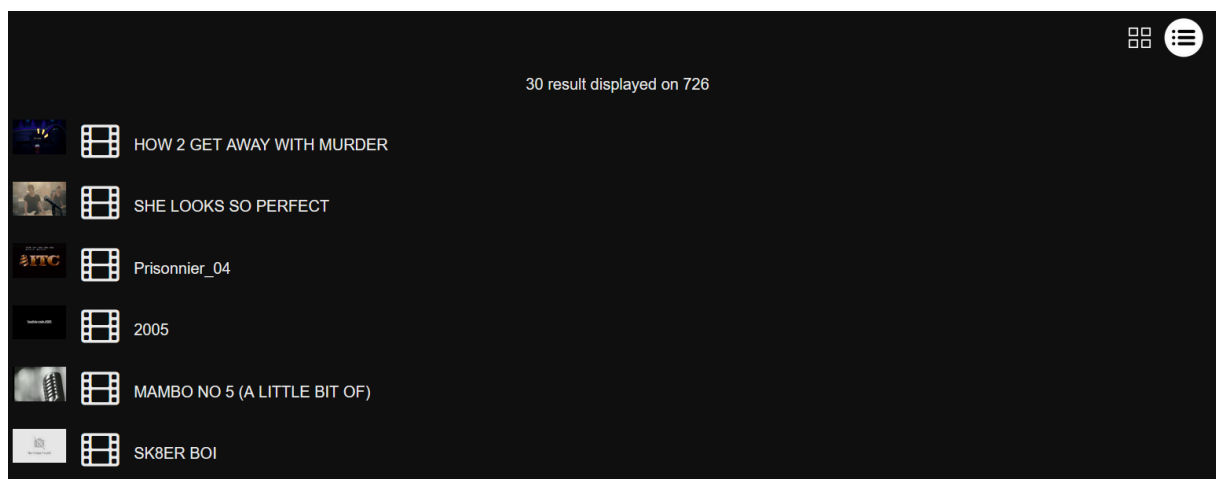


Figure 27 : Layout en liste en dark mode

Pour la rubrique de téléchargement de médias, j'ai simplement réalisé une grille avec le nom du fichier, son storage et un bouton pour télécharger.

Afin d'effectuer le téléchargement, j'utilise une requête AJAX* appelée directement depuis le script JavaScript relié à ma page.




General Info	Additional Fields	Segments	Credits	Press	Download
FILE NAME		STORAGE NAME		DOWNLOAD	
5_Seconds_of_Summer_-_She_Looks_So_Perfect.mp4		Uploads			
5_Seconds_of_Summer_-_She_Looks_So_Perfect.mp4		Uploads Low Res			
5_Seconds_of_Summer_-_She_Looks_So_Perfect.mp4		LowRes			

Figure 28 : Rubrique de téléchargements pour chaque médias

3.4 Récupération des changements de plans

3.4.1 Cahier des charges

Cette mission n'est pas finie alors elle ne sera pas complètement exploitée dans cette partie.

L'objectif de cette mission est de pouvoir récupérer les changements de plans des médias disponibles pour les utilisateurs du site «Web MAM» tout en les stockant dans des flux de mémoire.

Cela permettrait aux utilisateurs d'être plus efficaces dans la navigation de leurs médias en affichant directement les changements de plans, avec une fonctionnalité étant de cliquer sur n'importe quel plan pour arriver au timecode correspondant du média choisi.

J'ai donc été chargé de faire la récupération des images et des informations dans des flux de mémoire sur l'application de test nommée «MediaTools», puis récupérer ces données dans le site web.

3.4.2 Intégration technique

L'intégration technique s'est faite principalement à partir de code déjà présent.

Il m'a suffi d'initialiser deux flux de mémoire, un nommé «ShotStream» contenant les images des changements de plans et un nommé «Metadata» contenant le timestamp* de l'image sur 8 bytes*, la position de l'image dans le flux «ShotStream» sur 8 bytes ainsi que la taille de l'image sur 8 bytes, on obtient donc les informations de chaque image présente sur 24 bytes.

J'ai ensuite réalisé une conversion des bytes représentant les images en images grâce à la fonction «Image.FromStream(Stream metadata)» et je les ai fait afficher dans le logiciel de test.

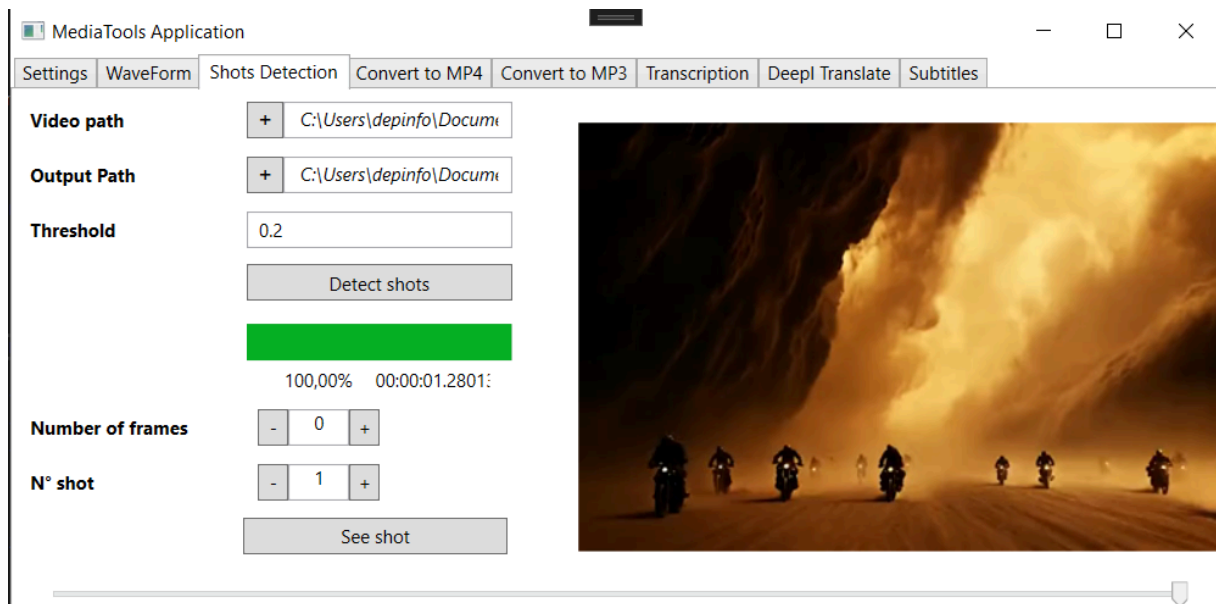


Figure 29 : Logiciel de test MediaTools avec affichage du premier plan d'une vidéo

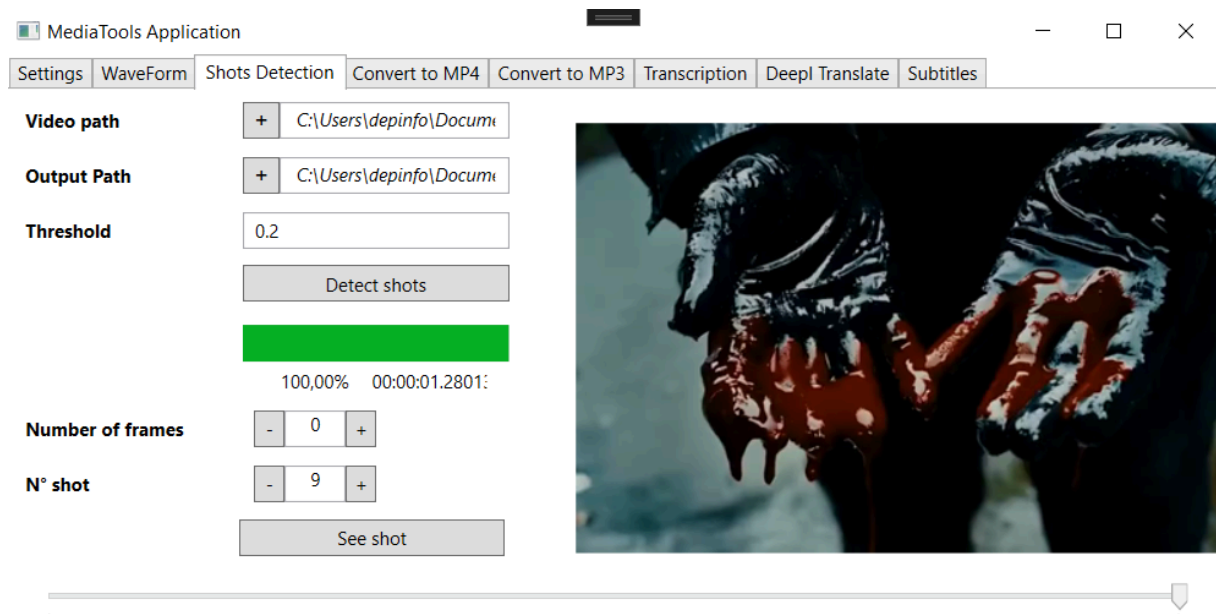


Figure 30 : Logiciel de test MediaTools avec affichage du neuvième plan d'une vidéo

IV. Méthodologie et organisation

L'organisation de travail qui m'a été attribuée était celle de travailler tout seul sur des petites fonctionnalités même si j'étais assigné à des projets en équipe. J'ai donc pu utiliser Git* sur ma propre branche lorsque je travaillais sur des logiciels.

Mon organisation, pour travailler seul s'est appuyée sur des schémas réalisés à l'aide du site Excalidraw[3]. Ces schémas m'ont permis de structurer mes idées, de planifier mes tâches et de suivre l'avancement des différentes missions. Ils ont servi de support visuel pour organiser chaque étape du développement et faciliter la compréhension des flux de travail. Ces schémas ont été régulièrement mis à jour afin d'adapter mon travail en fonction des évolutions du projet et des retours reçus.

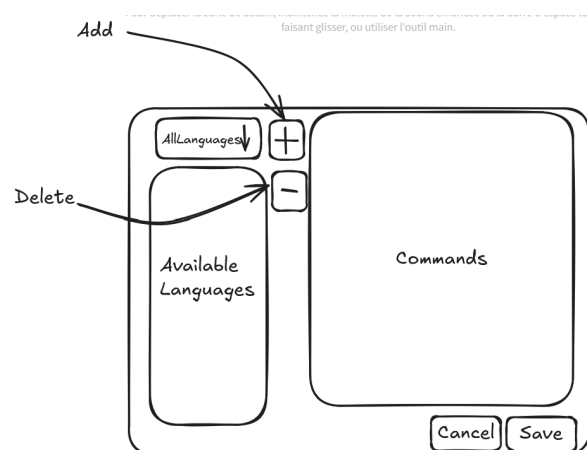


Figure 31 : Schéma réalisé pour le créateur de traductions logiciel

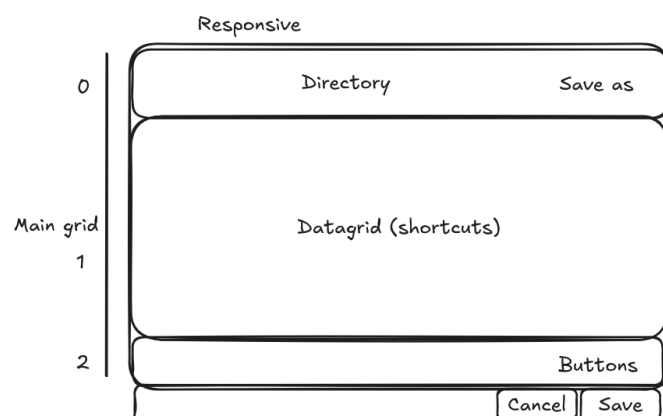


Figure 32 : Schéma réalisé pour le gestionnaire de raccourcis claviers

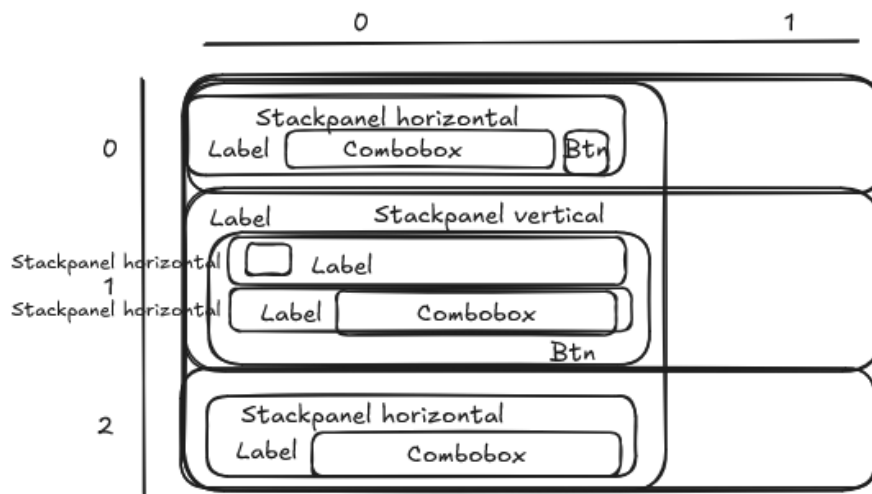


Figure 33 : Schéma réalisé pour le responsive d'une page du menu Options

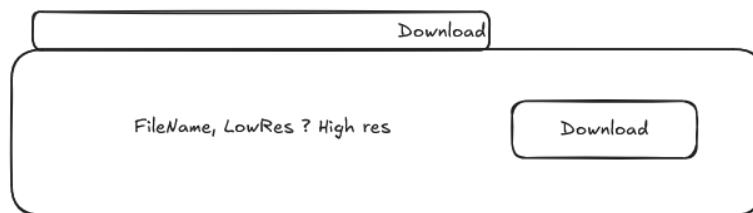


Figure 34 : Schéma réalisé pour les rubriques de téléchargements sur le site Web MAM

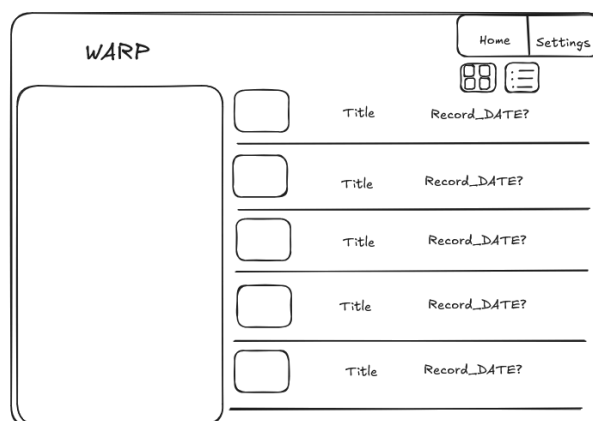


Figure 35 : Schéma réalisé pour le layout en liste du site Web MAM

Conclusion

Ce stage au sein d'ECOSM a été une expérience particulièrement formatrice qui m'a permis de développer un large éventail de compétences techniques et méthodologiques. En intervenant sur plusieurs aspects du développement logiciel, j'ai pu obtenir de nouvelles connaissances en C#WPF à travers l'amélioration du logiciel «CreasySubs Editor», accroître mes connaissances au niveau du système et du fonctionnement de la mémoire informatique avec les tests du logiciel MediaTools, mais aussi élargir mon expertise en développement web avec mon travail sur le «Web MAM».

L'intégration d'un système de traduction de logiciel dynamique et la mise en place d'une gestion avancée des raccourcis clavier ont permis d'améliorer efficacement l'ergonomie du logiciel «CreasySubs Editor». La conception de boutons capables de changer le layout d'un site web vise à une meilleure compréhension pour l'utilisateur.

Au-delà des compétences techniques acquises, ce stage m'a également permis de mieux comprendre le travail dans un environnement professionnel, ainsi que les méthodologies de développement utilisées dans l'industrie du logiciel.

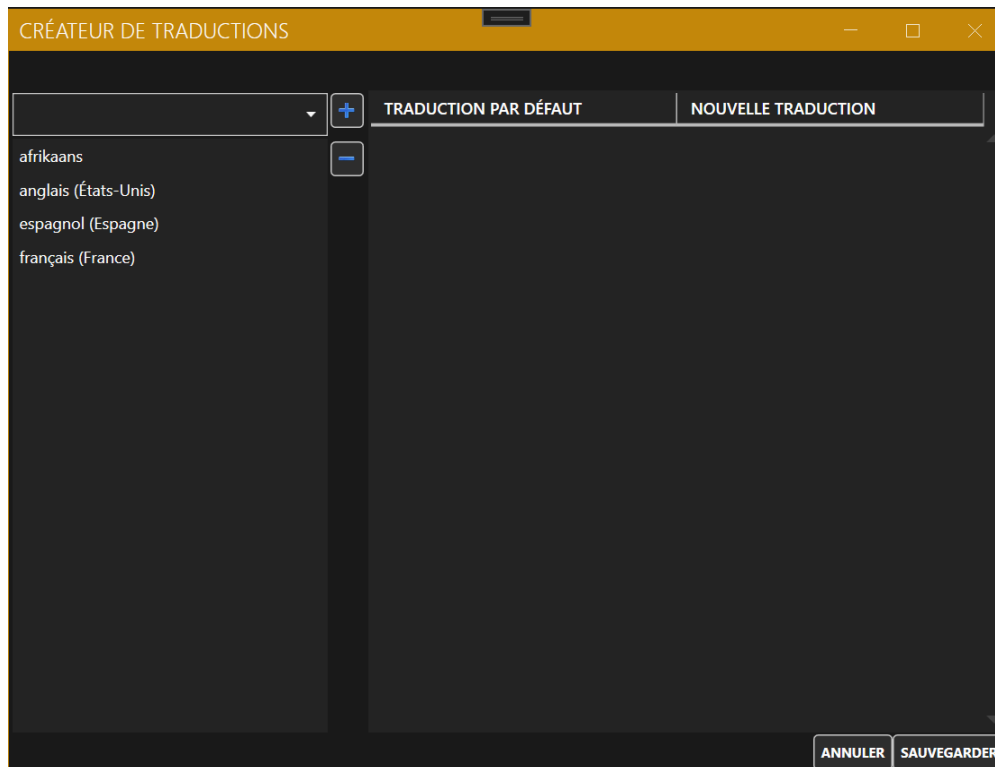
Cette expérience a été extrêmement formatrice et m'a permis de renforcer mon expertise technique tout en découvrant les exigences du développement logiciel appliqué aux médias. Elle constitue une étape importante dans mon parcours professionnel.

Vu le 25/03/2025
Jean-marie Barthelemy

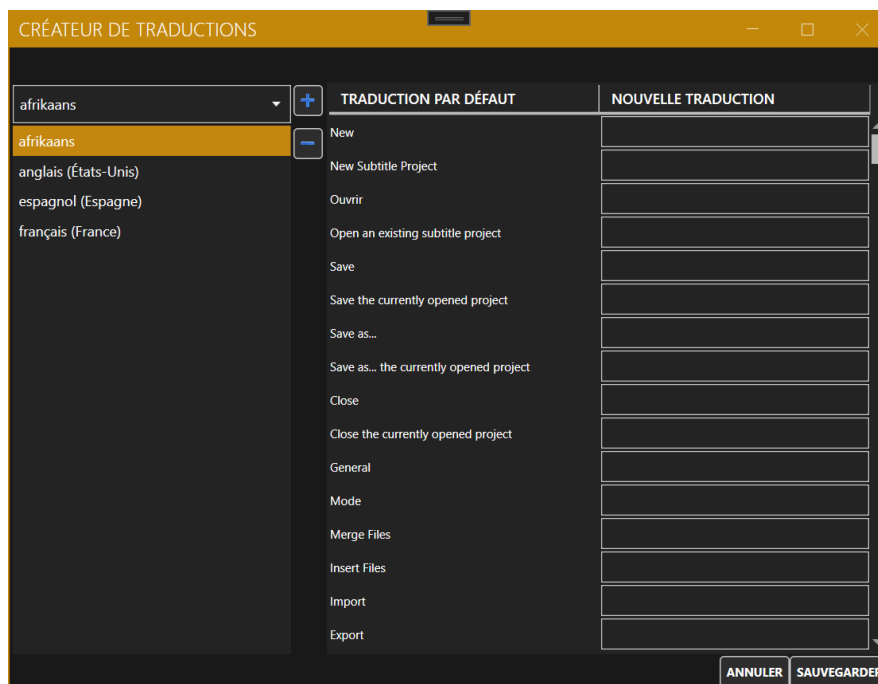


Annexes

Annexe 1 : Créateur de traductions



Annexe 2 : Créateur de traductions



Annexe 3 : Gestionnaire de raccourcis

OPTIONS ET PRÉFÉRENCES

Projet Time-Code Vidéo Mode en Direct Import Export Interface **Raccourcis Clavier**

Modèle de Raccourci: Default.cshortcut RAFRANCHIR

Default.cshortcut ENREGISTRER SOUS IMPORTER EXPORTER

NOM	COMMANDE	TOUCHE	TOUCHE SECONDAIRE	TOUCHE DE MODIFICATION
Ouvrir un fichier vidéo	OpenVideoCommand	X		Alt
Fermer un fichier vidéo	CloseVideoCommand			
Jouer la vidéo	PlayCommand			
Arrêter la vidéo	StopCommand			
Jouer/arrêter la vidéo	TogglePlayPauseCommand	Return		Alt
Mettre en pause la vidéo	PauseCommand	K		Alt
Image précédente avec son	PrevFrameCommand	Left		Alt
Prochaine image avec son	NextFrameCommand	Right		Alt

SUPPRIMER UTILISER PAR DÉFAUT SAUVEGARDER ANNULER SAUVEGARDER

Annexe 4 : Page Options Interface

OPTIONS ET PRÉFÉRENCES

Projet Time-Code Vidéo Mode en Direct Import Export **Interface** Raccourcis Clavier

Langue de l'interface Utilisateur: français (France) AJOUTER/MODIFIER

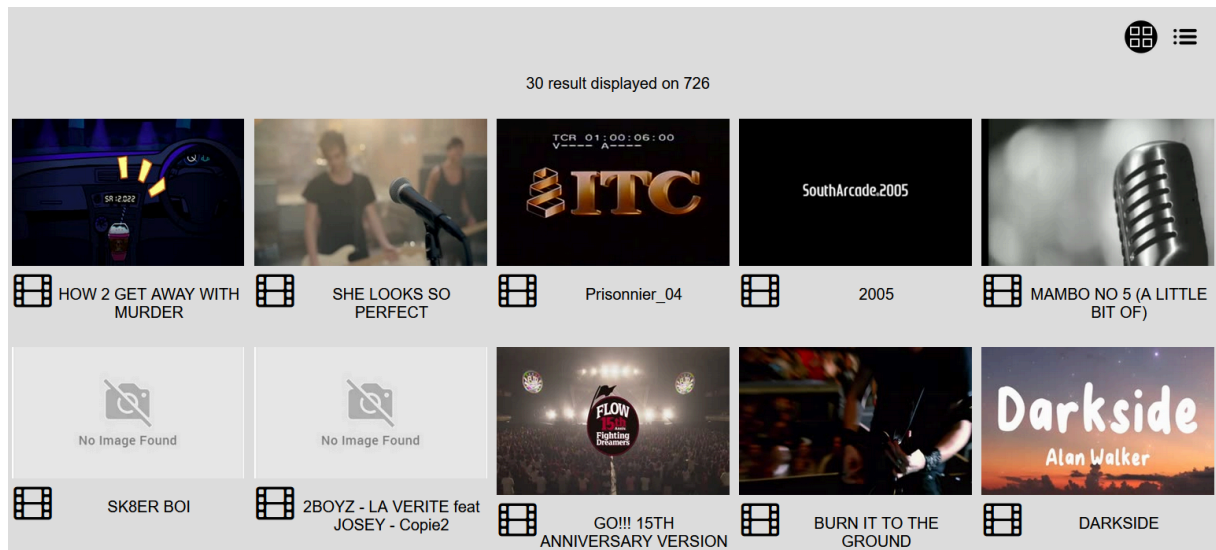
Mise en Page

☐ Utilisé lorsque CreasySubs est diffusé seul

Mise en Page par Defaut: RÉINITIALISER LA MISE EN PAGE PAR DÉFAUT

Couleur du Waveform: ANNULER SAUVEGARDER

Annexe 5 : Web MAM layout Grid, boutons switch de layout en light mode



Annexe 6 : Web MAM layout Grid, boutons switch de layout en dark mode

