



Figure 1 - Image titre

MusicSort

Tri, renommage et création de playlist de musique

Auteur : Dévaud Aurélien

Chef de projet : Lymberis Dimitrios

Expert 1 : Montemayor Ernesto

Expert 2 : Viret Loic

Du lundi 29 avril 2024 au mercredi 29 mai 2024

ETML -TPI

Table des matières

1	ANALYSE PRELIMINAIRE.....	4
1.1	INTRODUCTION	4
1.2	OBJECTIFS	4
1.3	PLANIFICATION.....	4
1.3.1	<i>Méthode des six pas IPDRCE</i>	<i>5</i>
1.3.2	<i>Planification détaillée.....</i>	<i>6</i>
2	ANALYSE	7
2.1	CONCEPT	7
2.1.1	<i>Environnement utilisé.....</i>	<i>7</i>
2.1.2	<i>Stratégie de conception.....</i>	<i>7</i>
2.1.3	<i>Structure du code</i>	<i>7</i>
2.1.4	<i>Installer.....</i>	<i>7</i>
2.2	STRATEGIE DE TEST.....	8
2.3	RISQUES	8
2.3.1	<i>Risques techniques</i>	<i>8</i>
2.3.2	<i>Manque de compétences</i>	<i>8</i>
3	CONCEPTION.....	8
3.1	MODELES DE CAS D'UTILISATION.....	9
3.2	MAQUETTES	12
3.2.1	<i>Maquette de base</i>	<i>12</i>
3.2.2	<i>Maquette des menus.....</i>	<i>13</i>
3.3	DIAGRAMME D'ACTIVITE	13
3.3.1	<i>Gestion des fichiers FLAC.....</i>	<i>14</i>
3.3.2	<i>Changement du mode d'application</i>	<i>14</i>
3.3.3	<i>Affichage des changements</i>	<i>15</i>
3.3.4	<i>Application des changements</i>	<i>16</i>
3.4	DIAGRAMME DE CLASSE	17
3.4.1	<i>Models.....</i>	<i>17</i>
3.4.2	<i>Views</i>	<i>18</i>
3.4.3	<i>Controllers.....</i>	<i>18</i>
3.5	LISTE DES TESTS	19

4	REALISATION.....	23
4.1	ENVIRONNEMENT	23
4.2	INTERFACE.....	24
4.3	STRUCTURE	26
4.3.1	<i>Changements par rapport à la conception.....</i>	<i>26</i>
4.4	COMPORTEMENT DU PROGRAMME	26
4.4.1	<i>Navigateur de dossiers.....</i>	<i>27</i>
4.4.2	<i>Gérance de la playlist</i>	<i>27</i>
4.4.3	<i>Changements généraux.....</i>	<i>28</i>
4.4.4	<i>Application des changements</i>	<i>28</i>
4.4.5	<i>Music Player</i>	<i>29</i>
4.5	INSTALLEUR	30
5	TESTS DES FONCTIONNALITES	32
5.1	AMELIORATIONS POSSIBLES.....	ERREUR ! SIGNET NON DEFINI.
5.2	LISTE DES DOCUMENTS FOURNIS	34
6	CONCLUSIONS.....	36
7	GLOSSAIRE	38
8	TABLE DES ILLUSTRATIONS.....	39
9	ANNEXES.....	43
9.1	RESUME DU RAPPORT DU TPI / VERSION SUCCINCTE DE LA DOCUMENTATION	43
9.2	SOURCES – BIBLIOGRAPHIE	43
9.3	ARCHIVES DU PROJET	43
9.4	MANUEL D'INSTALLATION	40
9.5	MANUEL D'UTILISATION.....	41

1 Analyse préliminaire

Dans cette partie, nous allons décrire les objectifs et détails du projet ainsi que les buts, tests et les moyens à disposition pour l'effectuer.

1.1 Introduction

L'objectif de ce projet est de développer une application en C# avec une interface utilisateur graphique (GUI) permettant à l'utilisateur d'explorer et de gérer des fichiers musicaux de différents formats tels que LFAC, WMA, MP3, etc.

L'application offrira des fonctionnalités telles que la navigation à travers les dossiers, la création de listes de lecture personnalisées par copie dans un dossier spécifique et le renommage de fichiers. Il est demandé également au candidat de mettre en place un package d'installation pour le programme avec son protocole d'utilisation.

Le projet commence par une planification initiale qui, selon les objectifs donnés dans le cahier des charges et la méthode des six étapes, essaie d'organiser en différentes tâches le déroulement du projet et leur charge de travail relationnelles (pourcentage).

La méthode des 6 étapes a été favorisée car les autres méthodes, comme la méthode agile et Scrum, sont plus adaptés au travail en équipe.

Les tests sont effectués à l'aide d'un tableau qui contient plusieurs colonnes qui permettent d'identifier le nom du test, la fonctionnalité testée, de décrire le test et de décrire le résultat demandé.

Pour la sauvegarde du projet, elle sera effectuée en enregistrant les fichiers dans Github de manière journalière.

1.2 Objectifs

L'application doit permettre à l'utilisateur de :

1. Parcourir et explorer les fichiers musicaux stockés dans différents dossiers sur son système.
2. Copier des fichiers musicaux sélectionnés dans d'autres dossiers pour créer des listes de fichiers de lecture personnalisées.
3. Renommer les fichiers musicaux selon ses préférences.
4. Prendre en charge plusieurs formats de fichiers musicaux courants, tels que FLAC, MP3, WMA avec possibilité d'écouter le morceau.
5. Fournir une interface utilisateur conviviale et intuitive pour faciliter la navigation et l'utilisation de l'application à travers une seule fenêtre Windows.

1.3 Planification

La planification initiale est la partie où une première structure est donnée au déroulement du projet. Ci-dessous, elle sera décrite.

1.3.1 Méthode des six pas IPDRCE

Pour la planification du projet, la méthode des six pas sera utilisée pour sa simplicité et son efficacité. Les estimations de temps incluent le temps passé sur la documentation.

Informer

Cette partie de la méthode des six pas est le moment où l'on s'informe sur les sujets impliqués à la mise en place du projet. Donc, les tâches suivantes seront effectuées :

1. Analyse du cahier des charges.
2. Recherche des thèmes impliqués et manques de compétences à combler
3. Recherche des risques et des solutions

Dû aux informations déjà possédée, elle représente ~2% du projet (~1h45).

Planifier

Cette partie de la méthode des six pas se centre sur la planification du projet et donc la répartition des tâches sur le temps. Donc, les tâches suivantes seront effectuées :

1. Sélection de la méthode de gestion de projet (IPDRCE).
2. Mise en place des tâches à effectuer durant le projet.
3. Répartition des tâches sur les jours

La planification prenant moins d'un jour, elle représente ~6% du projet (~5h16).

Décider

Cette partie est le moment où ce qui doit être réalisé est décidé. C'est la conception des modèles et maquettes.

1. Création du diagramme de cas d'utilisations.
2. Mise en place de la stratégie de test et création des tests,
3. Création de la maquette.
4. Création du diagramme d'activité.
5. Création du diagramme de classe.

Cette partie étant le moment où toute l'application est conceptualisée représente ~30% du projet (~26h24).

Réaliser

Cette partie est le moment où les éléments conceptualisés dans la partie 'Décider' sont réalisés.

1. Mise en place de l'environnement.
2. Création de l'interface principale.
3. Création de la structure du projet.
4. Remplissage des classes.
5. Débogage.
6. Création de l'installer.

Cette partie étant le moment où tout le projet est mis en place, mais déjà bien préparé, représente ~50% du projet (~44h).

Tester

Ceci est la partie où les fonctionnalités sont testées.

1. Test des fonctionnalités et commentaire des résultats si erreurs.

2. Description des erreurs, s'il y en a, et suggestions pour les résoudre.

Cette partie étant une simple utilisation de l'application et description des résultats représente ~4% du projet (~3h31).

Évaluer

Dans cette partie, le travail fait est évalué.

1. Comparaison du CDC avec le travail réalisé.
2. Corrections au projet si nécessaire ou si le temps le permet.
3. Discussion sur les résultats et améliorations possibles.

Cette partie représentant un grand temps de rédactions prendra donc le reste, ~8% du projet (~7h02).

1.3.2 Planification détaillée

La planification détaillée a été créée à l'aide des tâches et ratios présenté ci-dessus. Un fichier Excel a été utilisé qui décrit plus ou moins précisément les dates et durées des tâches sur les jours. Le fichier peut se trouver dans l'annexe du document.

2 Analyse

Le concept de ce projet est de créer une application permettant de sélectionner des fichiers de musique depuis des emplacements sur le PC et de les placer dans une playlist. Cette playlist doit pouvoir être triée et il doit être possible de renommer les fichiers et/ou de les déplacer ou copier dans un autre dossier.

Pour y arriver à bout, différentes technologies seront utilisées, telles que draw.io pour créer les modèles, et Windows Forms pour réaliser l'application. Étant une des versions les plus récente, donc qui contient plus du contenu, et pas la dernière, donc ayant moins de chance d'avoir des problèmes, la version 4.7.2 de .NET Framework sera utilisée.

Dans l'optique d'optimiser le temps, une aide sous la forme de ChatGPT sera utilisée pour débloquer des problèmes ou donner de l'inspiration.

2.1 Concept

Ici est analysé tous les concepts utilisés pour arriver à fin du projet tel que l'environnement dans lequel l'application serait codée, la stratégie utilisée pour concevoir le projet, la structure du code et l'installer.

2.1.1 Environnement utilisé

Le projet demandant la navigation dans des fichiers et leur manipulation, il a été décidé d'utiliser des bibliothèques capables d'afficher des formulaires. Pour ce cas, Windows Forms sera utilisé avec comme aide VisualStudio 2022 qui propose beaucoup d'options pour faciliter son utilisation.

Pour lire les fichiers de musique et pouvoir les écouter, le 'control' Windows Media Player sera employé en addition d'une bibliothèque, NAudio.Wave, pour convertir les fichiers FLAC, illisible par le 'control', en MP3.

2.1.2 Stratégie de conception

Pour concevoir l'application, un premier diagramme de cas d'utilisation pour identifier les fonctionnalités à implémenter et une maquette qui devra présenter l'apparence de l'interface qui devrait donner à l'utilisateur la possibilité d'exécuter toutes les actions décrites dans le premier diagramme seront créés.

Ensuite les processus seront pensés et présentés dans le diagramme d'activité et le programme structuré dans un diagramme de classe.

2.1.3 Structure du code

Dans le but de pouvoir bien s'organiser et se retrouver dans le code malgré les différences entre ce qui est affiché et ce qui est réel, on utilisera le model MVC pour séparer effectivement l'interface des données.

2.1.4 Installer

Pour que l'utilisateur puisse mettre en place l'application facilement et rapidement, un installer est nécessaire. L'installer de l'application ne sera pas celui de VisualStudio car de précédents cas ont prouvé qu'il était enclin à ne pas fonctionner correctement. Une alternative est l'application innosetup.

2.2 Stratégie de test

La stratégie de test utilisée sera une suite de test créée en fonction des actions identifiées dans le diagramme de cas d'utilisation. Chaque action possédera au moins un test dans le but de vérifier le bon fonctionnement de chaque fonctionnalité de l'application. Il y aura additionally un test pour l'installation.

Dû à la taille limitée de l'application, il a été décidé de ne pas utiliser des tests unitaires qui serait potentiellement trop volumineux pour des actions assez simples.

Un premier tableau utilisant 4 colonnes, qui sert à lister les tests à effectuer, a donc été utilisé avec la première colonne servant de nom et d'identifiant, la deuxième désignant quelle action elle représente, la troisième décrivant le test et la dernière montrant le résultat attendu.

Un nouveau tableau sera créé pour décrire les résultats des tests. Il possède 5 colonnes. La première l'identifie avec son par rapport au premier tableau des tests. La deuxième donne la date à laquelle le test a été effectué. Le troisième dit si le test a été réussi. La quatrième donne les conditions exactes dans lesquelles le test a été effectué. Et la dernière permet d'écrire des commentaires sur le résultat. Comme, par exemple, comment il a raté. Plusieurs tableaux pourraient être créés pour plusieurs séries de tests.

L'environnement des tests sera idéalement différent de celui utilisé durant le développement dans le but d'éviter de manquer des erreurs et les tests seront effectués au moins une fois par un camarade de classe avec des dossier remplis de données divers incluant des fichiers musiques.

2.3 Risques

Ce paragraphe décrit les différents risques techniques qui pourraient être rencontré durant le projet ainsi que les potentiels manques en compétences possibles.

2.3.1 Risques techniques

Ici, les risques techniques qui viennent des technologies utilisées sont décrits.

- Absence de Framework lors de l'installation.
 - Solution : Ajout du processus d'installation du Framework dans l'installer.
- Le 'control' Windows Forms utilisé ne supporte pas les fichiers FLAC.
 - Solution : Créer une copie du fichier convertie en MP3 pour être lu en utilisant la librairie NAudio.Wave (librairie proposée par ChatGPT).

2.3.2 Manque de compétences

- Manque de connaissance sur la lecture de fichier son.
 - Solution : lecture de la documentation Windows Forms pour trouver un 'control' utilisable. 'Control' Windows Media Player trouvé.

3 Conception

Dans ce paragraphe, le processus de conceptualisation du projet, les décisions effectuées pour la création du projet et la création des modèles pour la réalisation du produit final, est expliqué.

3.1 Modèles de cas d'utilisation

Ici, le modèle de cas d'utilisation qui affiche les différentes actions que l'utilisateur peut entreprendre est montré et expliqué.

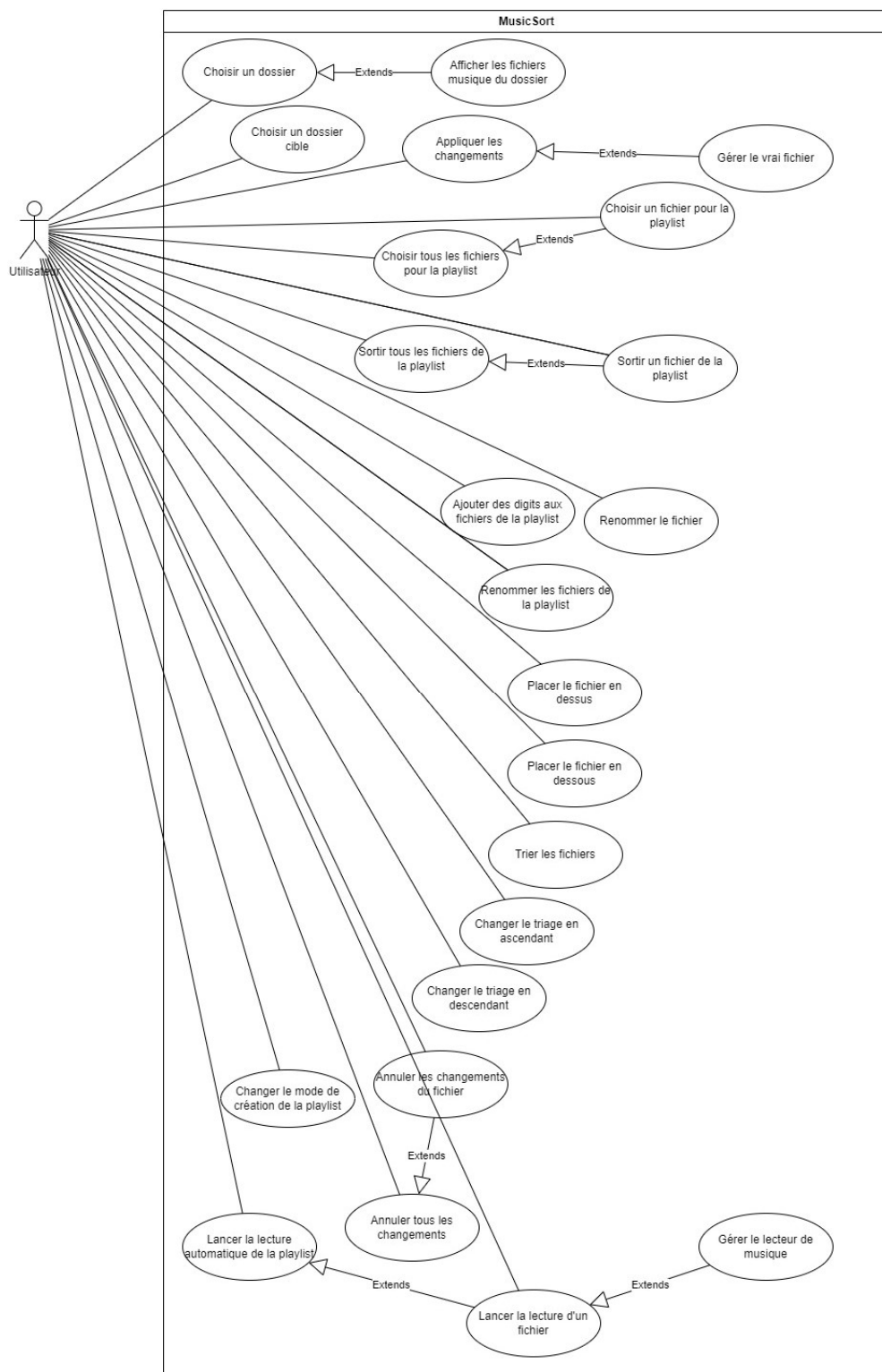


Figure 2 - Diagramme de cas d'utilisation

Choisir un dossier

Cette action permet à l'utilisateur de choisir un dossier dans lequel il pourra choisir des fichiers de musique à mettre dans la playlist

Afficher les fichiers musiques du dossier

Cette action affiche les fichiers musiques (FLAC, MP3, WMA) dans une fenêtre dans laquelle il sera possible de sélectionner ses fichiers et de les lire.

Choisir un fichier pour la playlist

Cette action permet à l'utilisateur de choisir un fichier d'un dossier et de le déplacer vers la playlist

Sortir un fichier de la playlist

Cette action permet de sortir un fichier de la playlist.

Choisir tous les fichiers pour la playlist

Cette action permet d'envoyer tous les fichiers du dossier sélectionné dans la playlist.

Sortir tous les fichiers de la playlist

Cette action permet de sortir tous les fichiers de la playlist.

Renommer le fichier

Cette action permet de renommer un fichier se trouvant dans la playlist.

Ajout de digit aux fichiers de la playlist

Cette action permet à l'utilisateur d'ajouter des chiffres comme préfixe ou suffixe au nom dans le but de garder l'ordre quand les fichiers seront sauvegardés dans le dossier. Elle donne la possibilité de placer un nombre de départ et de garder le nombre de digit donné dans le nombre de départ.

Renommer les fichiers de la playlist

Cette action permet à l'utilisateur de renommer les fichiers de la playlist avec un nom uniforme. Dans ce cas-là, il faudra obligatoirement attribuer un numéro pour ne pas avoir de doublon.

Placer le fichier en dessus

Cette action permet de placer le fichier sélectionné dans la playlist un cran au-dessus dans liste.

Placer le fichier en dessous

Cette action permet de placer le fichier sélectionné dans la playlist un cran en-dessous dans liste.

Trier les fichiers

Cette partie va trier les fichiers de la playlist dans l'ordre alphabétique.

Changer le triage en ascendant

Cette action va changer le triage en mode ascendant par rapport à l'ordre donné.

Changer le triage en descendant

Cette action va changer le triage en mode descendant par rapport à l'ordre donné.

Annuler les changements du fichier

Cette action va annuler les changements donner à ce fichier tel que son ordre personnalisé dans la playlist ainsi qu'un éventuel renommage.

Annuler tous les changements

Cette action va annuler tous les changements dans la playlist. Cela inclus le renommage général de la liste, les digits et les changements individuels des fichiers.

Changer le mode de création de la playlist

Cette action change le mode de création de playlist dans les données réelles. Par défaut, il est en renommage, il renomme tous les fichiers renommer sans changer leurs emplacements originaux. Le deuxième mode est 'Renommer et copier'. Dans ce cas, les fichiers originaux seront renommés et copier dans le dossier cible. Le troisième mode est 'Renommer et déplacer' qui déplace le fichier de son emplacement original au dossier cible tout en le renommant.

Gérer le lecteur de musique

Cette action gère le lecteur de musique. Elle fait en sorte que le fichier donné est lu sans avoir de double lecture.

Lancer la lecture d'un fichier

Cette action lance la lecture individuelle d'un fichier.

Lancer la lecture automatique de la playlist

Cette action lance la lecture automatique ordonnée de la playlist depuis le début.

Choisir un dossier de destination

Cette action permet de choisir un dossier qui sert de destination dans le cas où l'utilisateur déciderait de copier ou déplacer les fichiers de la playlist.

Appliquer les changements

Cette action applique les changements effectués aux éléments de la playlist.

Gérer le vrai fichier

Cette action renomme, copie et déplace le vrai fichier par rapport à son état dans la playlist.

Changer le mode de création de la playlist

Cette action change le mode de 'création' de la playlist entre 'Renommer', 'Renommer et copier' et 'Renommer et déplacer'.

Pour compléter les objectifs du CDC, il faut que l'utilisateur aie l'option d'interagir avec l'application de la manière décrite ci-dessus. En premier temps, l'utilisateur doit pouvoir choisir un dossier pour chercher les fichiers qu'il doit pouvoir mettre dans la playlist individuellement ou en groupe et faire la même chose avec les fichiers de la playlist vers leur dossier d'origine. Il doit pouvoir être possible de renommer les fichiers individuellement et de faire un renommage automatique de tout les membres de la playlist avec l'option d'ajouter un numéro. L'utilisateur peut aussi trier les fichiers et changer le mode de tri comme ascendant ou descendant. Les fichiers peuvent être placer en dessus ou en dessous des autres. Les changements des fichiers sont aussi annulable

individuellement ou totalement. L'utilisateur peut appliquer les changements de la playlist sur les fichiers réels et peut décider quel mode d'application et utilisé entre le renommage, renommage et copie et renommage et déplacement. Finalement, l'utilisateur peut lancer la lecture automatique ou individuelle de la playlist ou des fichiers.

3.2 Maquettes

Pour que l'utilisateur puisse interagir avec l'application, il a besoin d'une interface. Ce paragraphe montre et explique les maquettes utilisées qui décrivent comment l'interface devra être faite. Elles ne représentent pas la manière exacte dans laquelle l'application se présentera mais serviront de guides pour la création du programme.

3.2.1 Maquette de base

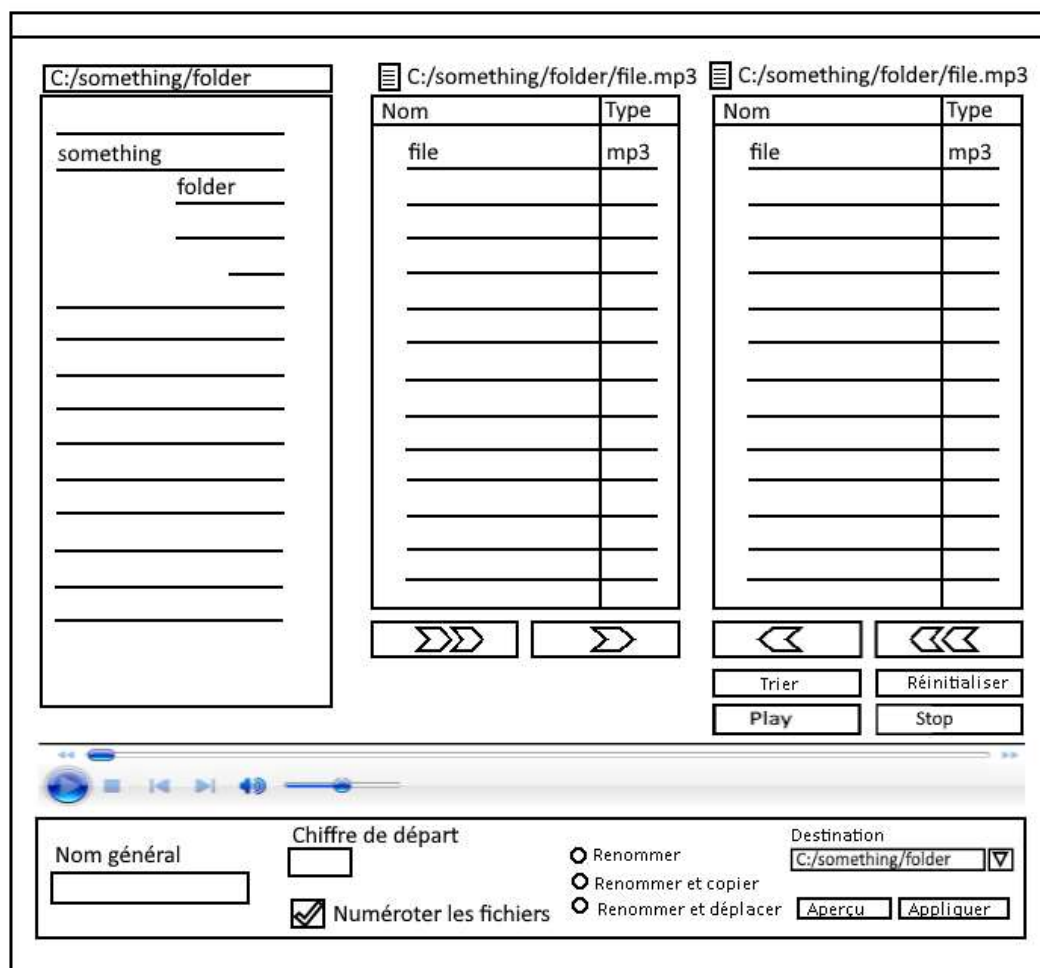


Figure 3 - Maquette de l'application

La maquette ci-dessus avec la maquette suivante permettent toutes les actions notées dans le diagramme de cas d'utilisation.

Tout à gauche se trouve le navigateur de dossiers, il permet à l'utilisateur de naviguer les dossiers du PC et d'en sélectionner un. En dessus de lui est noté le nom du dossier dans une boîte de texte. Il devrait être possible de donner son propre chemin.

Au milieu s'affiche les fichiers du dossier sélectionné. Au-dessus, le nom du fichier sélectionné et en-dessous, la flèche unique pour envoyer la sélection dans la playlist et la double flèche pour envoyer tout le dossier dans la playlist.

La fenêtre à droite représente les fichiers de la playlist avec le chemin du fichier sélectionné en-dessus. La flèche unique pour sortir la sélection. La flèche double pour tout sortir. Le bouton trier pour trier la liste alphabétiquement. Le bouton réinitialiser pour réinitialiser toutes les modifications apportées à la liste. Le bouton Play pour lancer la lecture automatique de la liste et le bouton Stop pour la stopper.

Sous les trois fenêtres, le lecteur de musique se présente et en dessous de lui le panel de gestion de la playlist.

Dans le panel, à gauche, la boîte de texte 'Nom général' permet de donner un nom à tous les fichiers de la playlist. 'Chiffre de départ' contient le numéro de commencement dans la numérotation. La boîte avec un vu, dessous, donne la possibilité d'activer ou de désactiver la numérotation et les trois petits points permettent de changer le mode l'application des changements.

À droite, on peut choisir le dossier de destination de la playlist. On peut choisir d'afficher les changements et de les appliquer dans les fichiers réels.

3.2.2 Maquette des menus

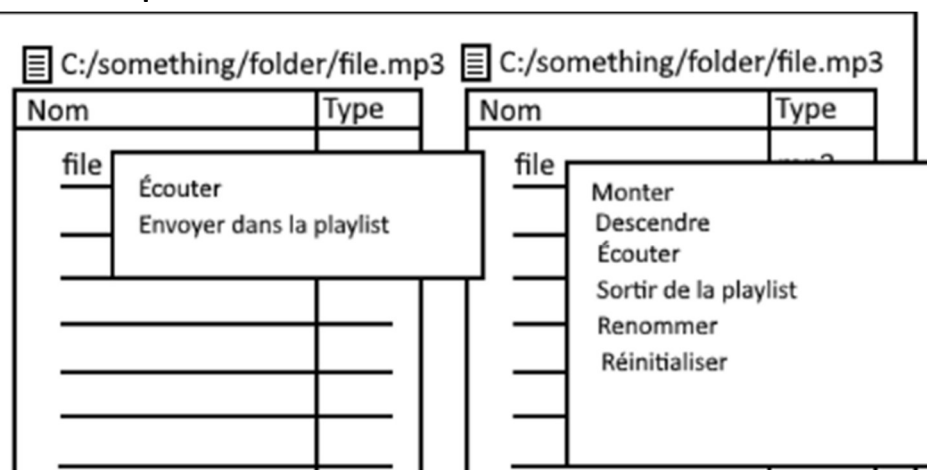


Figure 4 - Maquette des menus

Le menu à gauche est activé lorsque l'on fait un clic droit sur un élément des fichiers du dossier sélectionné. Il donne l'option à l'utilisateur d'écouter le fichier individuellement ou l'envoyer dans la playlist.

Le menu à droite est activé lorsque l'on fait un clic droit sur un élément de la playlist. Il donne l'option à l'utilisateur de monter et descendre le fichier dans la liste, de l'écouter, de le sortir de la playlist, de le renommer et de réinitialiser les changements appliqués sur lui.

3.3 Diagramme d'activité

Le diagramme d'activité décrit conceptuellement comment les processus du programme devraient se dérouler. Le diagramme étant trop volumineux, seul des parties limitées seront affichées et expliquées. Pour observer le diagramme en son intégralité, allez dans les annexes.

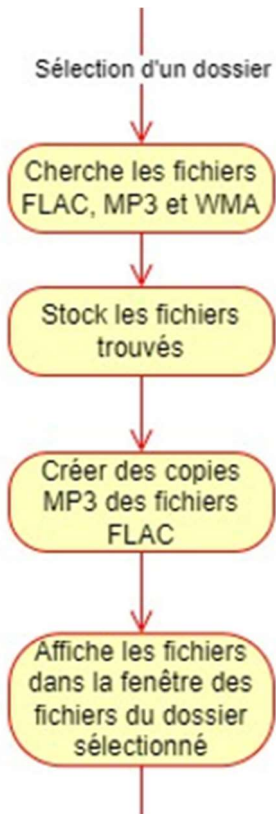


Figure 5 - Sélection d'un dossier

3.3.2 Changement du mode d'application

Lorsque l'on active le mode 'Renommer', une vérification de l'unicité des noms est entreprise. Quand le mode 'Renommer et déplacer' ou 'Renommer et copier' sont eux activés, il y a en plus une vérification que l'on a bien sélectionné un dossier et sinon, la fenêtre de navigation des dossiers s'affiche.

3.3.1 Gestion des fichiers FLAC

Lorsqu'un dossier est choisi, les fichiers MP3, WAV et FLAC sont sélectionnés et affichés pour pouvoir être lus. Malheureusement, le moyen de lecture utilisé n'est pas compatible avec les fichiers de type FLAC. Alors, une copie temporaire convertie en MP3 de ces fichiers dans un dossier est effectuée. Le lecteur utilisera ces copies durant l'écoute.

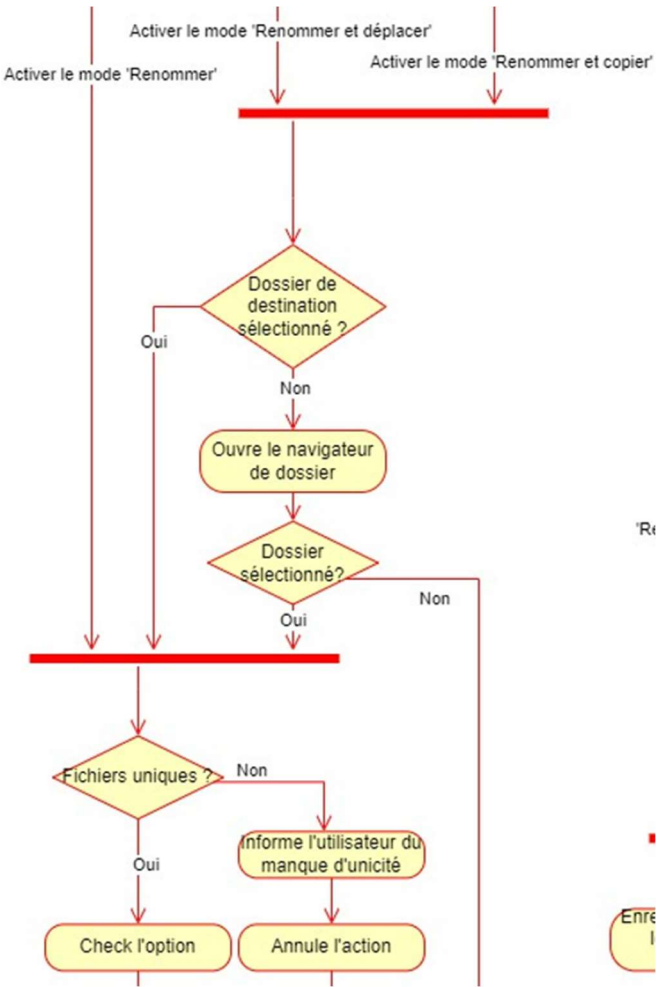


Figure 6 - Changement du mode d'application

3.3.3 Affichage des changements

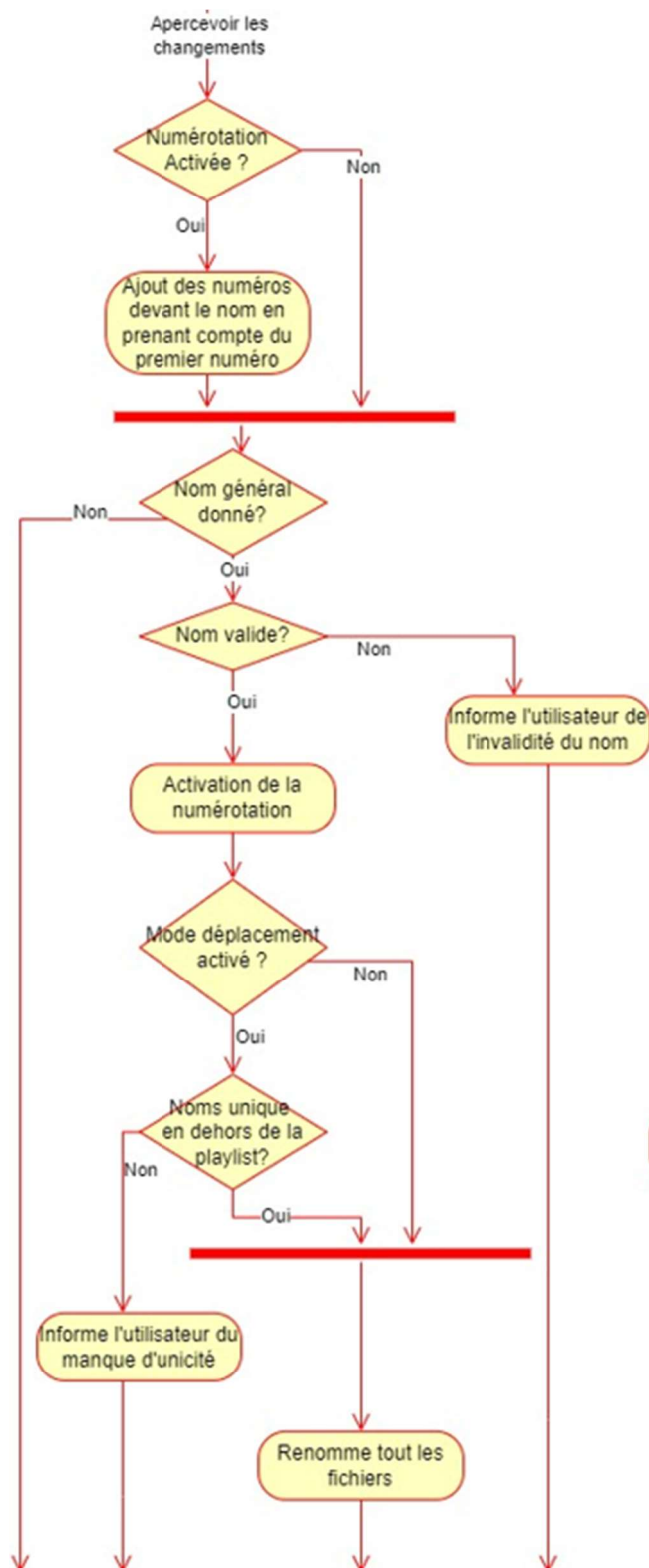


Figure 7 - Affichage des changements

L'affichage des changements est l'action qui affiche le résultat du renommage général et de la numérotation.

Si un nom a été donné, il sera vérifié s'il est valide pour des fichiers. Puis, la numérotation sera activée pour éviter les doublons. Dans le cas où les fichiers devraient se trouver en dehors de la playlist même avec l'application des changements, alors il sera testé si les noms seront uniques même dans ces conditions. Finalement, s'il n'y a aucune erreur détectée, les noms s'affichent.

3.3.4 Application des changements

L'application des changements dans les fichiers et dossiers réels se fait de la façon suivante.

En premier temps, la validité des noms est testée. Puis, le processus se divise en trois.

Dans le renommage, on test l'unicité des fichiers. On les renomme. Puis, on informe l'utilisateur du succès.

Dans 'Renommer et copier', on vérifie encore une fois l'unicité dans les dossiers originaux et dans la playlist. On renomme les fichiers et on copie les fichiers renommés dans le dossier cible. Finalement, on informe l'utilisateur du succès. Dans 'Renommer et déplacer', on vérifie seulement l'unicité dans la playlist et on déplace tout en renommant les fichiers dans le dossier cible. Puis, on informe l'utilisateur.

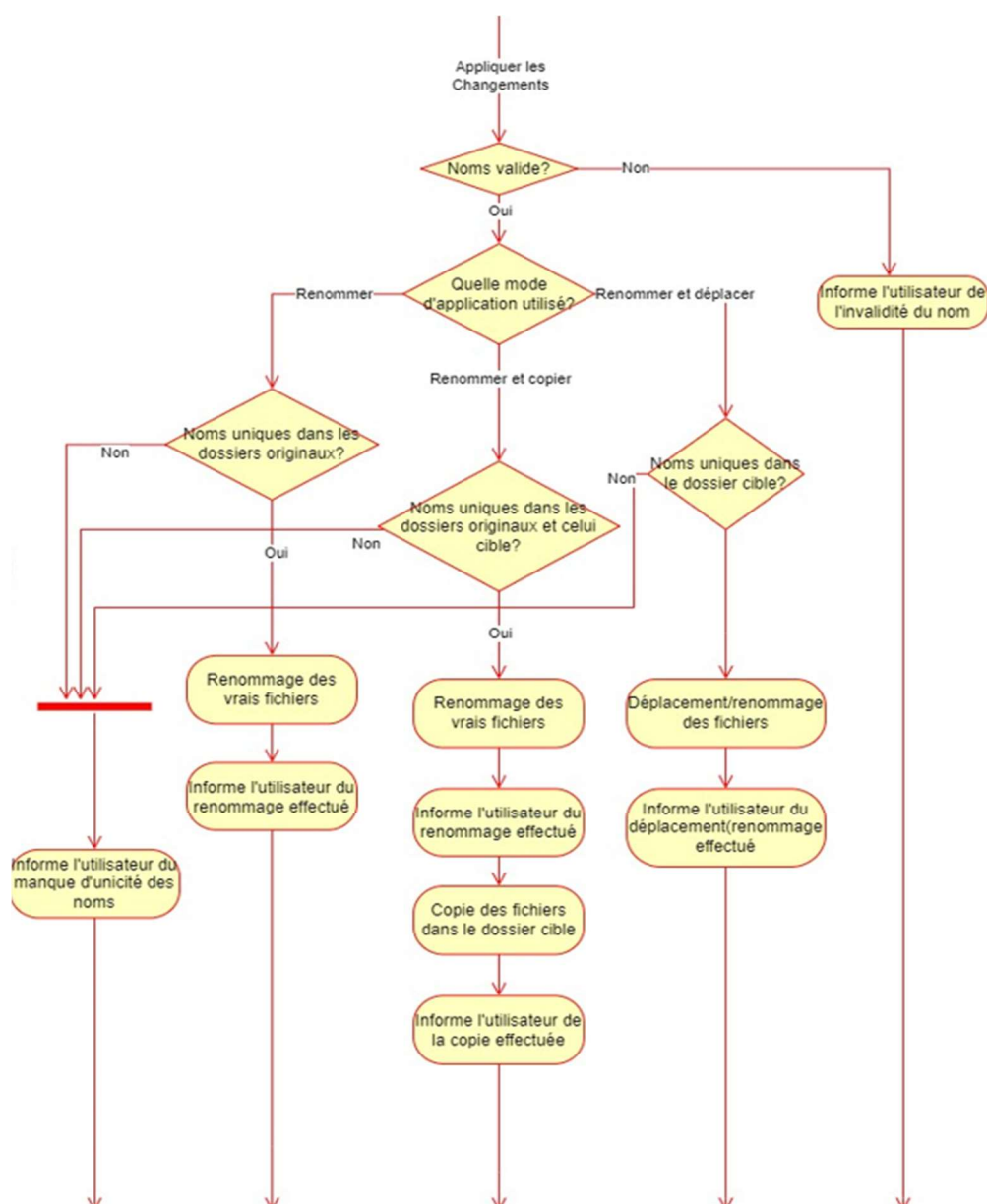


Figure 8 - Application des changements

3.4 Diagramme de classe

Dans ce diagramme de classe, on décrit les méthodes et propriétés nécessaires au fonctionnement de l’application. Bien sùre, des imprévus peuvent faire que le résultat final soit bien différent. Pour voir le diagramme en entier, allez dans les annexes.

3.4.1 Models

Le ‘model’ est la partie de l’application qui stock les informations et qui interagit avec les données extérieures pour les rendre utilisable dans le programme.
Dans ce ‘model’, il y a deux classes.
La première, Model, contient les informations générales de l’application tel que la playlist, les chemins des dossiers, le nom général, le nombre et le mode d’application. Il y a aussi les méthodes de vérification et de recherche dans les dossiers.

La deuxième classe est ‘File’. File permet de sauvegarder les changements et les appliquer dans les vrais fichiers.

Model	File
<div><<+get,+set>> Controller : Controller</div> <div><<+get,-set>> Playlist : List<File></div> <div><<+get,-set>> DestinationPath : string</div> <div><<+get,-set>> SelectedPath : string</div> <div><<+get,+set>> GeneralName : string</div> <div><<+get,+set>> NumberPrefix : string</div> <div><<+get>> NumberDigit : int</div> <div><<+get>> StartNumber : int</div> <div><<+get,+set>> IsPrefixActivated : bool</div> <div><<+get,-set>> ApplicationMode : ApplicationMode</div> <div>+ Model()</div> <div>+ SetNewDestination(path : string) : bool</div> <div>+ SetNewSelectedDirectory(path : string) : bool</div> <div>+ GetFilesFromDirectory(path : string) : File[]</div> <div>+ IsValidFileName(name : string) : bool</div> <div>+ TestMode(mode : ApplicationMode) : bool</div> <div>+ SetNewMode(mode : ApplicationMode) : void</div> <div>+ TestForApplication() : void</div> <div>+ SetPrefixFromIndex() : bool</div>	<div><<+get,-set>> FullRealPath : string</div> <div><<+get>> RealExtension : string</div> <div><<+get>> RealName: string</div> <div><<+get>> RealPath : string</div> <div><<+get>> DisplayName: string</div> <div><<+get>> FullCustomPath : string</div> <div><<+get,-set>> CustomName : string</div> <div><<+get,-set>> CustomPath : string</div> <div><<+get,-set>> Prefix : string</div> <div><<+get,+set>> IndexInPlaylist : int</div> <div>+ File(fullRealPath : string)</div> <div>+ WouldFileBeUnique(path : string, customName : string, prefix : string) : bool</div> <div>+ SetNewPrefix(name : string) : bool</div> <div>+ SetNewName(name : string) : bool</div> <div>+ SetNewPath(path : string) : bool</div> <div>+ ApplyChanges() : bool</div>

Figure 9 - File (MVC)

Figure 10 - Model (MVC)

3.4.2 Views

Les classes ‘Views’ servent comme interface entre l’utilisateur et l’application. C’est eux qui représentent et gèrent les formulaires et leurs contrôles.

Le formulaire principal est représenté par la classe ViewForm qui sert aussi à envoyer des pop-up à l’utilisateur pour l’informer de l’état de l’application ou pour lui demander des confirmations ou des dossiers. La classe contient des instances d’autres classes comme le lecteur de musique, le browser et les listes.

La classe FileItem représente les fichiers sous forme ajoutable dans les listes, ils contiennent des menus et sont inclus dans des ListView, comme la classe PlaylistView qui montre la playlist et tous ses fichiers.

3.4.3 Controllers

Controller
<<+get,-set>> Model : Model
<<+get,-set>> View : ViewForm
+ Controller(Model model, ViewForm view)
+ StartPlaylist() : void
+ StopPlaylist() : void
+ ListenToFile(file : File) : void
+ SwitchSortOrder() : void
+ SelectDirectory(path : string) : void
+ DirectorySelected(path : string) : void
+ SendFilesToPlaylist(files : File[]) : void
+ RemoveFilesFromPlaylist(files : File[]) : void
+ SortPlaylist() : void
+ PlaceFileHigher(file : File) : void
+ PlaceFileLower(file : File) : void
+ RenameFile(file : File) : void
+ ResetFile(file : File) : void
+ InputNewNumber(number : string) : void
+ InputNewGeneralName(name : string) : void
+ ActivateNumbering() : void
+ DeactivateNumbering() : void
+ SearchForDestinationDirectory() : void
+ DisplayGeneralChanges() : void
+ Apply() : void

Le ‘controller’ contrôle les processus du programme et sert d’intermédiaire et de liaison entre les données et l’interface.

Toutes les fonctions importantes et les opérations sont guidées par le contrôleur. Il gère comment les fichiers sont ordrés, comment les envoyer d’une liste à l’autre, comment afficher les changements ou encore comment écouter les fichiers.

Figure 11 - Controller (MVC)

3.5 Liste des tests

Ici se trouve le tableau des tests pour vérifier que le programme fonctionne correctement.

Nom du test	Fonctionnalité testée	Description	Condition de réussite
Choix du dossier	Choisir un dossier	Tester qu'il soit possible de parcourir les dossiers de la machine et de choisir un dossier.	Le dossier est bien sélectionné.
Affichage du contenu du dossier	Afficher les fichiers musiques du dossier	Tester que les fichiers du dossier sélectionné d'affiche dans une fenêtre.	Tous les fichiers du dossier sélectionné de type FLAC, MP3 et WMA devront être affichés.
Déplacement d'un fichier dans la playlist	Choisir un fichier pour la playlist	Tester qu'il soit possible d'envoyer un fichier individuel du dossier sélectionner dans la playlist.	Le fichier sélectionné doit se trouver dans la playlist.
Sortie d'un fichier de la playlist	Sortir un fichier de la playlist	Tester qu'il soit possible de sortir un fichier individuel de la playlist.	Le fichier ne se trouve plus dans la playlist.
Déplacement de tous les fichiers dans la playlist	Choisir tous les fichiers pour la playlist	Tester qu'il soit possible de déplacer tous les fichiers trouvés dans le dossier dans la playlist.	Tous les fichiers du dossier type FLAC, MP3 et WMA se trouvent dans la playlist.
Sortie de tous les fichiers	Sortir tous les fichiers de la playlist	Tester qu'il soit possible de sortir tous les fichiers se trouvant dans la playlist.	La playlist est vide.
Renommage individuel	Renommer le fichier	Tester qu'il soit possible de renommer un fichier de la playlist	Le fichier est renommé dans la playlist

Renommage de playlist	Renommer les fichiers de la playlist	Tester qu'il soit possible de donner un nom général à tous les fichiers de la playlist.	Les fichiers de la playlist ont tous le même nom donné sans compter l'extension et préfixe/suffixe.
Numérotation des fichiers	Ajout de digit aux fichiers de la playlist	Tester que les fichiers de la playlist sont numérotés selon l'ordre donné et le numéro de départ donné.	Les fichiers ont comme préfixe/suffixe leur position dans la playlist, le nombre de digit du nombre de départ est gardé et
Remontée du fichier	Placer le fichier en dessus	Tester que l'on puisse placer le fichier en dessus des autres dans la playlist.	Le fichier se trouve un cran au-dessus dans la playlist.
Décente du fichier	Placer le fichier en dessous	Tester que l'on puisse placer le fichier au-dessous des autres dans la playlist.	Le fichier se trouve un cran au-dessous dans la playlist.
Triage des fichiers	Trier les fichiers	Tester qu'il soit possible de donner un ordre alphabétique aux fichiers de la playlist.	Les fichiers sont triés de manière alphabétique.
Triage ascendant	Changer le triage en ascendant	Tester qu'il soit possible de trier les fichiers avec les premiers en haut et les derniers en bas.	Les premières lettres de l'alphabets et plus petits nombres sont au début de la liste.
Triage descendant	Changer le triage en descendant	Tester qu'il soit possible de trier les fichiers avec les derniers en haut et les premiers en bas.	Les dernières lettres de l'alphabets et plus grands nombres sont au début de la liste.
Annulation des changements d'un fichier	Annuler les changements du fichier	Tester qu'il soit possible d'annuler le renommage d'un fichier ainsi que sa	Le fichier recouvre son nom d'origine si le changement de nom de la playlist

		position personnalisée dans la liste.	n'est pas actif. Le digit reste s'il est activé.
Annulation des changements de tous les fichiers	Annuler tous les changements	Tester qu'il soit possible d'annuler tous les changements effectués tel que les noms personnalisés, le nom général de la playlist et le digit.	Les fichiers recouvrent leur nom d'origine.
Modes d'application : Renommer	Changer le mode de création de la playlist	Test qu'il soit possible de renommer les fichiers originaux.	Dans le mode 'Renommer', tous les fichiers ayant des noms qui diffèrent changent de nom dans leurs dossiers originaux.
Modes d'application : Renommer et Copier	Changer le mode de création de la playlist	Test qu'il soit possible de renommer les fichiers originaux et de les copier dans le dossier cible.	Dans le mode 'Renommer et copier', tous les fichiers ayant des noms qui diffèrent changent de nom dans leurs dossiers originaux et sont copiés dans le dossier cible.
Modes d'application : Renommer et Déplacer	Changer le mode de création de la playlist	Test qu'il soit possible de déplacer les fichiers dans le dossier cible et de les renommer.	Dans le mode 'Renommer et déplacer', tous les fichiers originaux sont déplacés dans le dossier cible et renommés.
Lecture de musique	Gérer le lecteur de musique	Test que les fichiers donnés sont bien lus et peuvent être écoutés.	Des fichiers des types FLAC, MP3 et WMA peuvent tous être lus.
Lecture d'un fichier	Lancer la lecture d'un fichier	Tester qu'il soit possible de lire les fichiers de la playlist	Quand un fichier est double-cliqué, cela lance sa lecture.

		et du dossier sélectionné.	
Lecture automatique	Lancer la lecture automatique de la playlist	Tester qu'il soit possible de lancer la lecture automatique de la playlist.	Tous les fichiers de la playlist sont lus un par un de haut en bas.
Dossier de destination	Choisir un dossier de destination	Tester qu'il soit possible de choisir un dossier de destination.	Lorsque l'on applique le changement des fichiers avec l'option 'Renommer et copier', les fichiers sont copiés dans le dossier cible choisi.
Application des changements	Appliquer les changements	Tester qu'il soit possible d'appliquer les changements effectués dans l'application.	Les changements sont appliqués de la manière choisie.
Modification de fichier	Gérer le vrai fichier	Tester que les fichiers sont bien changés en fonction de ce qui est fait dans l'application.	Ne change rien tant que rien n'est appliqué. Change le vrai fichier quand les changements sont appliqués.

4 Réalisation

Bien que beaucoup de pensées ont été mises dans la conception, la réalisation pourra toujours différer quand il est temps de mettre les modèles en pratique. Cette partie va expliquer comment l'environnement a été créé. Comment l'interface et la structure ont terminé et comment l'application a été programmée.

4.1 Environnement

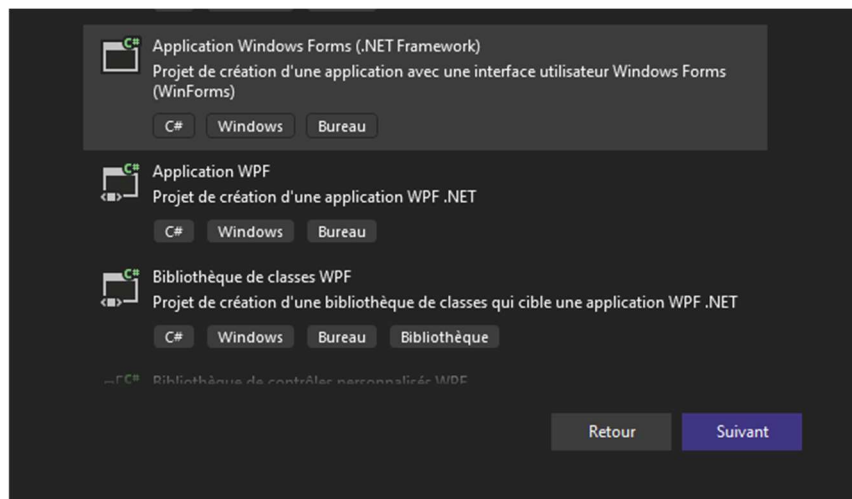


Figure 12 - Choix du modèle de projet

La première chose à faire est de préparer l'environnement dans lequel l'application sera créée. Alors, dans Visual Studio 2022, j'ai créé un projet en utilisant le modèle « Application Windows Forms (.NET Framework) » qui est un modèle permettant l'utilisation de formulaires dans un milieu .NET Framework.

Le projet a reçu le nom MusicSort comme le titre donné dans le CDC et la version du Framework a bien été choisie comme 4.7.2.

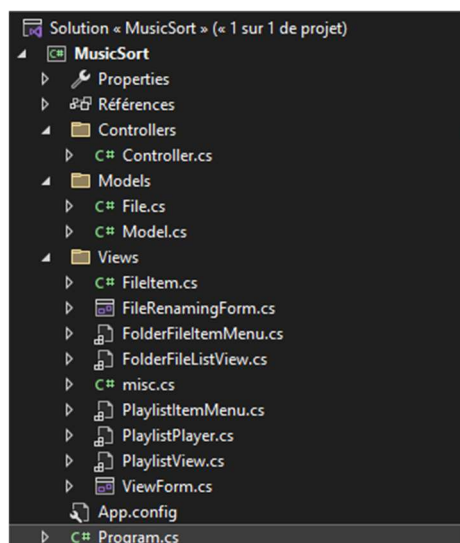


Figure 13 - Fichiers

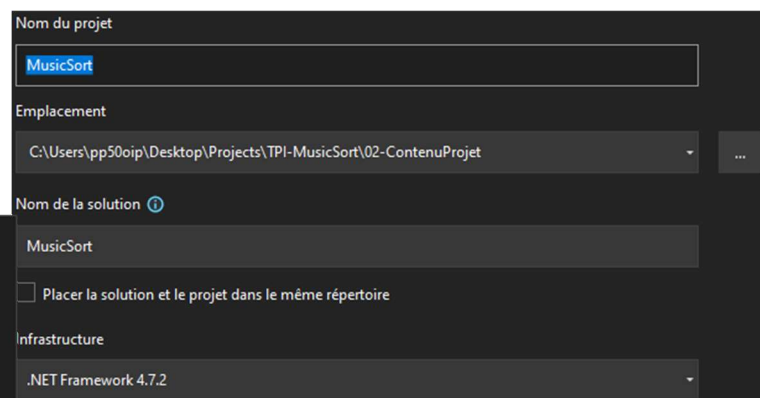


Figure 14 - Création du projet

Un dossier a été créé pour chaque partie du MVC ; Models, Views, Controllers. Pour chaque classe, excepté le 'Comparer', un fichier a été dédié. Les autres éléments Views ont été placés dans le fichier misc et l'énumérateur 'ApplicationMode' dans la classe Model.

4.2 Interface

L'interface permet à l'utilisateur d'interagir avec l'application. La première chose faite a été de préparer les 'Controls' personnalisés à ajouter dans le formulaire. Puis, les 'Controls' de base ont été ajouté sur le formulaire principal.

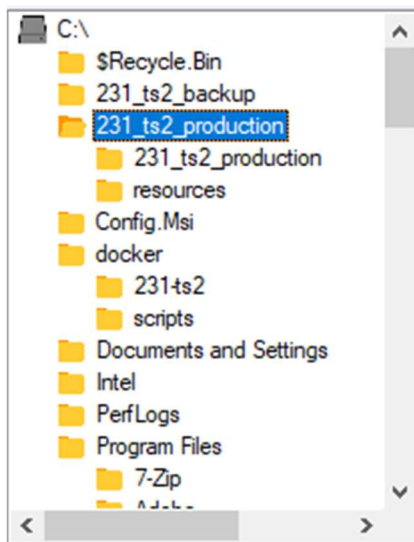


Figure 15 - Navigateur de dossier

Pendant l'ajout, j'ai rencontré mon premier problème. Je suis parti du principe qu'un explorateur de dossier sans 'Dialog' existait déjà. Ce qui n'est pas le cas. Il a donc été décidé d'utiliser un 'TreeView' à la place pour naviguer les dossiers.

Les classes créées pour remédier à ça s'appelle FolderBrowser et permet de mettre en place un dossier de base avec 'SetBaseDirectory()' dans lequel il est possible de naviguer dans ses enfants et d'en sélectionner un. La détection de la sélection se fait avec un event personnalisé ; 'FolderSelectedEvent'.

L'interface général a été créée en prenant comme modèle la maquette faite précédemment. La sélection du dossier de sélection se fait avec le FolderBrowser parler ci-dessus tandis que les fichiers du dossier de sélection et les fichiers de la Playlist sont affichés dans une ListView qui permettent de les lister effectivement. Ces ListView ont deux colonnes qui décrivent le nom et l'extension de chaque fichier.

L'outil qui permet de lire les musiques quant à lui est un Control nommé Windows Media Player qui est accessible en cochant « Windows Media Player » dans l'onglet « Composants COM » de l'option « Choisir les éléments » de la « boîte à outils » de Windows Forms.

Les modes d'application sont choisis à l'aide de controls « RadioButton » qui sont regroupés dans un control GroupBox.

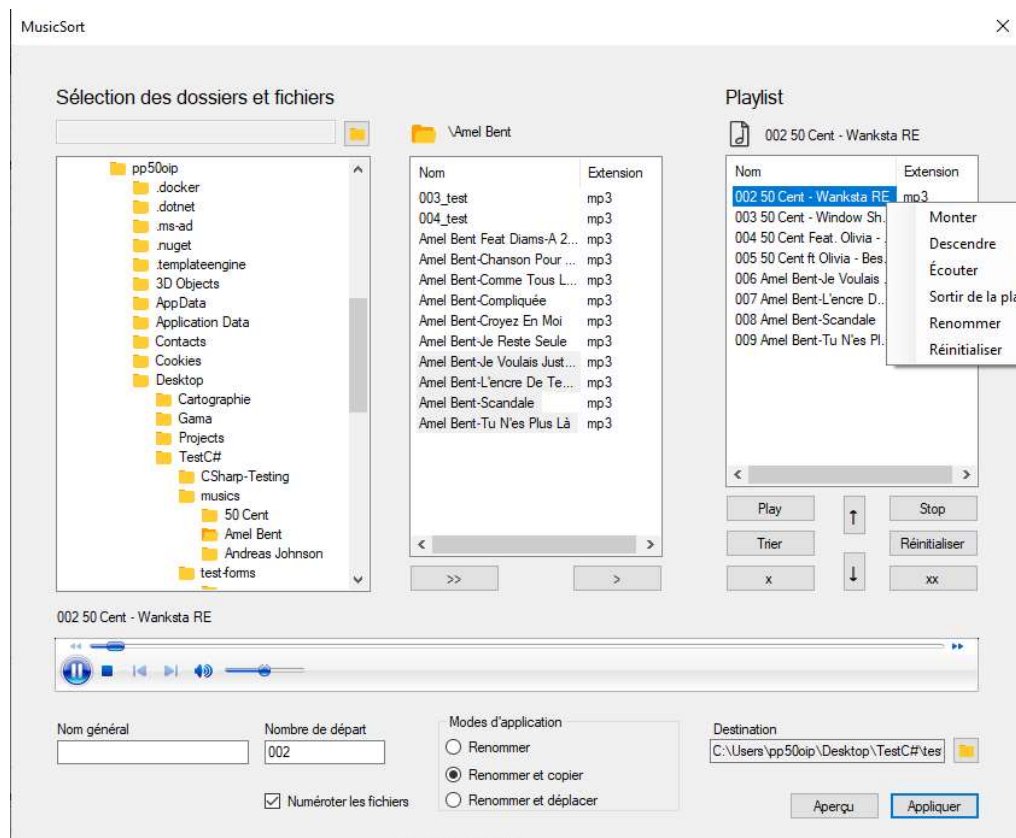


Figure 16 - Interface

4.3 Structure

La structure du code suit le diagramme de classe UML créé dans la conception ainsi que quelques conventions de nommage et des commentaires.

En premier temps, chaque page directement créée ou modifiée est munie d'un entête décrivant l'auteur, la date de création et la description brève du code.

En second, chaque méthode, champ et propriété est commenté et suit la convention de nommage suivante ;

- Les champs sont en « LowerCamelCase » avec un « _ » au début.
- Les propriétés sont en « UpperCamelCase ».
- Les méthodes sont en « UpperCamelCase » avec un verbe au début du nom.
- Les méthodes qui répondent à un événement sont en « UpperCamelCase » avec le nom de l'objet qui lance l'événement au début et le nom de l'événement à la fin séparés par un « _ ».
- Les méthodes qui répondent à un délégué sont en « UpperCamelCase » avec le nom de l'action.

```
///ETML  
///Auteur : Aurélien Devaud  
///Date : 08.05.2024  
///Description :
```

Figure 17 - Résumé du fichier

4.3.1 Changements par rapport à la conception

Pour la copie du diagramme, quelques changements ont été effectués ;

La méthode « SelectDirectory » n'accepte plus de chemin mais lance le processus pour en trouver un.

« NumberPrefix » perd aussi son « setter » et son « getter » crée le préfixe par lui-même. « SetPrefixFromIndex » lui rend une liste de chaque fichiers et leur message d'erreurs.

Pour la classe « File », le constructeur contient maintenant aussi la possibilité de donner un index directement au fichier. Un délégué « FileInfoChangedEventHandler » a aussi été ajouté pour gérer les événements d'informations changées sur le fichier « FileInfoChangedEvent ».

Le constructeur de « FolderFileItemMenu » a été corrigé pour accepter le fichier et les autres délégués.

Dans la classe « FolderFileListView », il a été décidé de ne pas passer les délégués directement mais de les donner après l'initialisation de l'objet. Ce qui explique les « getters ».

Dans « ViewForm », la méthode « SendFileItemsToPlaylist » n'est pas ajoutée car une méthode de « PlaylistView » se charge déjà de sa tâche.

La méthode de triage alphabétique a été placée dans la classe Model pour ne pas la confondre avec la méthode de triage du « ListView » qui trie en fonction de l'index.

Finalement, tous les événements des interactions avec l'interface ont été ajoutés.

4.4 Comportement du programme

Ici sera décrit comment les différentes fonctions du programme se comportent.

4.4.1 Navigateur de dossiers

Pour choisir un dossier, il faut utiliser le navigateur de dossier (FolderBrowser). Par défaut, le navigateur affiche les différents ‘drives’.

Depuis là, en appuyant sur ces répertoires, les répertoires enfants sont affichés et tous les fichiers des types autorisés (‘mp3’, ‘wma’, ‘flac’, ‘aac’, ‘wav’, ‘midi’, ‘asf’) sont récupérés en temps que File et affichés dans des objets ‘FileItem’s dans le fenêtre à gauche, le ‘FolderFileListView’.

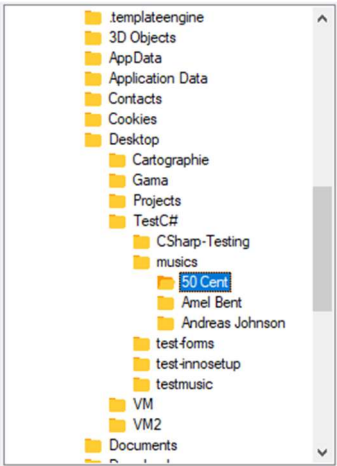


Figure 18 - Navigateur de dossier

4.4.2 Gérance de la playlist

La playlist permet de créer une liste de fichiers se trouvant dans plusieurs dossiers différents et de les écouter dans l’ordre donner.

Pour ajouter des fichiers dans la liste, il y a 3 manières. La première est d’appuyer sur le bouton des doubles flèches pour tout envoyer et la deuxième est le bouton de la flèche unique pour envoyer la sélection. Finalement, il est possible d’envoyer individuellement un fichier en utilisant le menu contextuel.

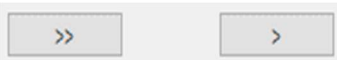


Figure 19 - Envoyer

Pour sortir les fichiers de la playlist, les mêmes fonctions sont disponibles mais avec des croix.



Figure 20 - Suppression



Figure 21 -
Changement de
position

Le tri quant à lui se fait de la manière suivante. En premier temps, les fichiers sont ordrés selon leur ordre d’ajout. Puis, lorsque l’utilisateur appuie sur le bouton du tri, tous les éléments de la playlist sont ordrés de manière alphabétique. Il est alors possible de faire monter ou descendre la sélection avec les flèches ascendante et descendante. Additionnellement, il est possible d’inverser l’ordre un appuyant sur l’entête du nom.



Figure 22 - Trier

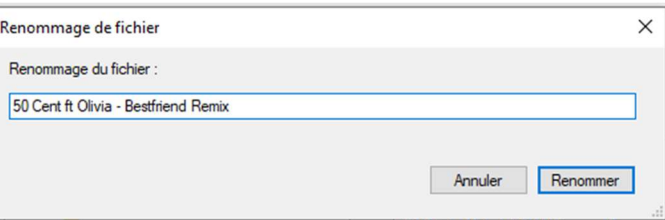


Figure 24 - Formulaire de renommage

potentiels erreurs.

Le renommage des fichiers se fait en allant dans l’option ‘Renommer’ du menu contextuel des fichiers. Il est effectué à l’aide d’un formulaire secondaire qui va vérifier si le nom est juste et informer l’utilisateur de

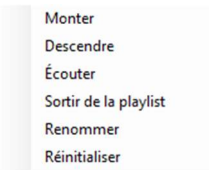


Figure 23 - Menu
Contextuel

Chaque fichier peut être réinitialisé individuellement via le menu, cela veut dire que son nom re-devient celui originel. Sinon, il est possible, avec le bouton ‘Réinitialiser’, de remettre à zéro le nom, le chemin, le préfixe, le mode d’application et l’ordre de tri de tous les fichiers.

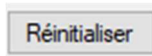


Figure 25 - Tout réinitialiser

Quand une action est entreprise qui change le nom de l’objet du fichier, un événement (FileInfoChangedEvent) est déclenché qui met à jour tous les objets FileItems liés servant à afficher l’objet.

4.4.3 Changements généraux

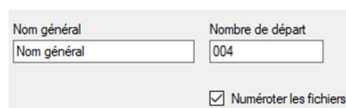


Figure 26 - Changements généraux

Une fonctionnalité du programme est la numérotation des objets de la playlist et la possibilité de leur donner un nom général. Quand la numérotation est activée lorsque l’utilisateur coche l’option, la possibilité de donner un nombre de départ est proposée en réactivant les boîtes de texte du nombre de départ et du nom général. Le programme prendra en compte le premier nombre et commencera la numérotation en utilisant comme repère la propriété IndexInPlaylist des Files. De plus, le nombre de chiffres total sera respecté tant que le nombre de fichiers ne dépasse pas le nombre de chiffres donnés.

Pour donner un nom général, il est impératif de donner un nombre pour garder l’unicité des noms des éléments de la playlist.

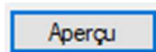


Figure 28 - Bouton aperçu

Finalement, aucun de ces changements ne sont mis en place avant d’avoir appuyé sur le bouton ‘Aperçu’ qui lance la méthode DisplayGeneralChanges() qui applique les préfixes et le nom général sur les objets si un nombre est donné.

Nom
004 Nom général
005 Nom général
006 Nom général
007 Nom général
008 Nom général
009 Nom général
010 Nom général

Figure 27 - Renommage général

4.4.4 Application des changements

Par défaut, tous les changements amenés aux éléments de la playlist ne sont pas appliqués sur les fichiers qu’ils représentent. Pour les appliquer, il suffit d’appuyer sur le bouton ‘Appliquer’.

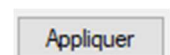


Figure 29 - Appliquer

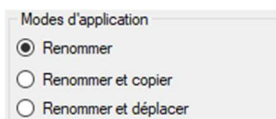


Figure 30 - Modes d’application

Mais d’abord, il faut choisir un mode d’application et il y a en a trois ;

Le premier mode, celui du renommage qui renomme les fichiers et ne prend pas en compte le dossier de destination. Donc lorsqu’il est choisi, tous les objets des fichiers courants et futurs sont donnés leurs noms personnalisés tandis que leurs chemins restent conformes à la place des fichiers réels. Quand l’utilisateur demande l’application des changements, le programme vérifie si les nouveaux noms pourraient causer des conflits avec des fichiers du même nom déjà existant et si c’est le cas, l’utilisateur sera demandé s’il veut écraser ces doublons.

Le deuxième mode est le mode ‘Renommer et Copier’ qui copie le fichier dans un dossier de destination avec son nom personnalisé. Donc, lorsqu’il est choisi, tous les objets des fichiers courants et futurs sont donnés leurs noms personnalisés et le chemin du dossier de destination. Quand



Figure 31 - Destination

l'utilisateur demande l'application des changements, le programme vérifie si les nouveaux emplacements des fichiers possède déjà des fichiers du même nom et si c'est le cas, l'utilisateur sera demandé s'il veut les écraser.

Le troisième mode est le mode 'Renommer et Déplacer' se comporte exactement de la même manière que 'Renommer et Copier' à l'exception que dans ce mode, le fichier original est supprimé.

Si des problèmes d'autorisation se produisent, les actions concernées sont annulées.

L'application se fait dans l'ordre suivant ; En premier temps, la présence du dossier de destination est testée si elle est nécessaire. Puis, si des changements généraux ont été fait, ils sont affichés. Ensuite, le programme demande à l'utilisateur de confirmer l'action. Si l'application des changements est confirmée, les changements seront testé et les résultats enregistrés. Une nouvelle confirmation sera demandée dans le cas où des complications tels qu'un écrasement de fichier ou un manque de permission aura été détecté.

Après ces multiples premières questions à l'utilisateur, les changements sont appliqués fichier par fichier. Si les fichier demande un écrasement, alors une confirmation est demandée à l'utilisateur. Si des permissions sont manquantes, alors l'application des changements du fichier est annulée.

Finalement, le dossier de destination est réinitialisé et la liste des fichiers du dossier sélectionné (FolderFileListView) est mise à jour.

4.4.5 Music Player

L'objet du lecteur de musique (MediaPlayer de la classe PlaylistPlayer) est utilisé pour lire les musiques de la playlist. Il hérite du 'control' AxWindowsMediaPlayer (ou MediaPlayer) de la librairie AxWMPLib.

```
using AxWMPLib;
PlaylistPlayer : AxWindowsMediaPlayer
```

Figure 32 - PlaylistPlayer Librairie

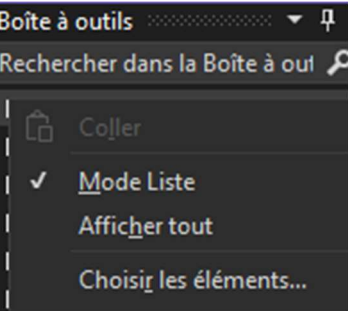


Figure 35 - Choisir les éléments

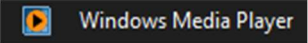


Figure 33 - Windows Media Player

Pour accéder au 'Media Player', il faut : aller dans la boîte à outils, puis ouvrir le menu contextuel et appuyé sur l'option 'Choisir les éléments'. Dans l'onglet 'Composants COM' de la fenêtre qui s'ouvre il faut finalement cocher l'option 'Windows Media Player' pour pouvoir utiliser le composant.

Choisir des éléments de boîte à outils

Composants .NET Framework		Composants COM	
Nom		Che	
<input type="checkbox"/>	Tabular Data Control	C:\	
<input type="checkbox"/>	TaskSymbol Class	C:\	
<input type="checkbox"/>	UMRDPProtocolManager Class	C:\	
<input type="checkbox"/>	VLC ActiveX Plugin and IE Web Plugin ...	C:\	
<input type="checkbox"/>	VLC ActiveX Plugin and IE Web Plugin ...	C:\	
<input type="checkbox"/>	VSTO FormRegionsHostX	C:\	
<input type="checkbox"/>	VSTO WinFormsHost Control	C:\	
<input type="checkbox"/>	Windows Mail Mime Editor	C:\	
<input checked="" type="checkbox"/>	Windows Media Player	C:\	

Figure 34 - Composants COM

Par défaut l'objet ne peut lire que les fichiers musiques de type 'mp3', 'wma', 'aac', 'wav', 'midi', 'asf' mais dans le Cahier des Charges, il est demandé de pouvoir aussi lire les fichiers de type 'flac'. Pour régler ce problème, j'ai décidé de créer une copie temporaire du fichier en MP3 quand sa lecture est demandée. Pour parvenir à cela, il a fallu en premier mettre en place un dossier 'temp_files' pour ces fichiers qui se crée automatiquement au lancement de l'application. Ce dossier est nettoyé au lancement et à la fermeture de l'application.

Pour convertir les fichiers Flac en MP3, j'ai suivi les suggestions de ChatGPT et de StackOverflow et utilisé la librairie NAudio.Lame et NAudio.Wave qui, avec le lecteur de données audios AudioFileReader et le rédacteur LameMP3FileWriter, donnent la possibilité de lire le fichier FLAC et de l'écrire sous format MP3 dans un fichier temporaire.

```
//create a reader of the file
using (var reader = new AudioFileReader(FullRealPath))
{
    //create the writer of the file
    using (var writer = new LameMP3FileWriter(TempFilePath, reader.WaveFormat, LAMEPreset.VBR_90))
    {
        //copy the file to mp3
        reader.CopyTo(writer);
    }
}
```

Figure 36 - Conversion FLAC -> MP3

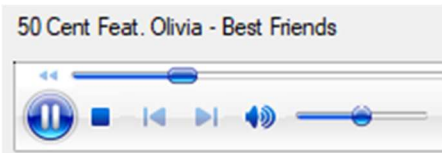


Figure 38 - PlaylistPlayer

Il y a deux façon de lire des fichiers. La première est de double-clicker le fichier ou d'utiliser le menu contextuel pour lire un fichier unique. Cela va vider la liste des fichiers en attente et lancera la lecture du fichier sélectionné. La deuxième façon est d'appuyer sur le bouton 'Play' qui mettra en attente en ordre tous les fichiers de la playlist dans le lecteur pour les écouter. Pour arrêter la lecture et vider la liste d'attente il suffit d'appuyer sur le bouton 'Stop'. Un label sur le 'control' qui décrit le fichier en court de lecture se met à jour à chaque nouvelle musique jouée.

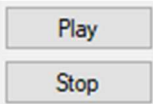


Figure 37 - Play Stop

4.5 Installleur

temp_files	22.05.2024 09:00	Dossier de fichiers	
AxInterop.WMPLib.dll	22.05.2024 09:00	Extension de l'app...	53 Ko
Interop.WMPLib.dll	22.05.2024 09:00	Extension de l'app...	323 Ko
libmp3lame.32.dll	22.07.2020 08:18	Extension de l'app...	681 Ko
libmp3lame.64.dll	22.07.2020 08:18	Extension de l'app...	1 057 Ko
Microsoft.Win32.Registry.dll	15.11.2019 09:40	Extension de l'app...	23 Ko
Microsoft.Win32.Registry.xml	16.10.2019 16:43	Microsoft Edge H...	93 Ko
MusicSort.exe	22.05.2024 09:00	Application	64 Ko
MusicSort.exe.config	17.05.2024 09:24	XML Configuratio...	1 Ko
MusicSort.pdb	22.05.2024 09:00	Program Debug D...	172 Ko
NAudio.Asio.dll	04.09.2023 21:49	Extension de l'app...	34 Ko
NAudio.Asio.xml	04.09.2023 21:49	Microsoft Edge H...	48 Ko
NAudio.Core.dll	04.09.2023 21:49	Extension de l'app...	184 Ko
NAudio.Core.xml	04.09.2023 21:49	Microsoft Edge H...	325 Ko
NAudio.dll	04.09.2023 21:49	Extension de l'app...	8 Ko
NAudio.Lame.dll	17.01.2022 06:15	Extension de l'app...	90 Ko
NAudio.Lame.pdb	17.01.2022 06:15	Program Debug D...	17 Ko
NAudio.Midi.dll	04.09.2023 21:49	Extension de l'app...	46 Ko
NAudio.Midi.xml	04.09.2023 21:49	Microsoft Edge H...	69 Ko
NAudio.Wasapi.dll	04.09.2023 21:49	Extension de l'app...	175 Ko
NAudio.Wasapi.xml	04.09.2023 21:49	Microsoft Edge H...	495 Ko
NAudio.WinForms.dll	04.09.2023 21:49	Extension de l'app...	40 Ko
NAudio.WinForms.xml	04.09.2023 21:49	Microsoft Edge H...	30 Ko
NAudio.WinFormsMM.dll	04.09.2023 21:49	Extension de l'app...	56 Ko
NAudio.WinFormsMM.xml	04.09.2023 21:49	Microsoft Edge H...	111 Ko
NAudio.xml	04.09.2023 21:49	Microsoft Edge H...	5 Ko
System.Security.AccessControl.dll	15.11.2019 09:41	Extension de l'app...	31 Ko
System.Security.AccessControl.xml	16.10.2019 16:43	Microsoft Edge H...	216 Ko
System.Security.Principal.Windows.dll	15.11.2019 09:41	Extension de l'app...	15 Ko
System.Security.Principal.Windows.xml	16.10.2019 16:44	Microsoft Edge H...	87 Ko

Figure 39 - Fichiers à installer

Le générateur d'installleur utilisé, InnoSetup, permet de créer un scripte qui, après une configuration, peut générer le fichier d'installation de l'application.

Pour commencer, il a fallu choisir l'environnement à installer. C'est-à-dire, tous les fichiers du dossier de release de la solution dans le but d'inclure les diverses librairies comme NAudio et autres.

Ensuite, avec l'aide du GUI de InnoSetup, j'ai créé un scripte 'MusicSort-Setup.iss' générant un installleur 'MusicSort-Setup.exe' pour l'application.

Le script prend donc tous les fichiers choisis dans son onglet '[Files]' et les déploient dans le dossier des programmes de l'utilisateur courant.

En outre, le script prend en compte si le Framework .NET 4.7.2 est présent au moment de l'installation de l'application avec la fonction NeedsDotNetFramework. S'il ne l'est pas, il est installé à l'aide de l'installateur 'ndp472-devpack-fra.exe' se trouvant dans le même dossier que le script.

```
#define MyAppName "MusicSort"
#define MyAppVersion "1.5"
#define MyAppPublisher "Aurélien Dévaud, ETML"
#define MyAppExeName "MusicSort.exe"
#define InputDirectory "C:\Users\pp50oip\Desktop\Projects\TPI-MusicSort\02-ContenuProjet\MusicSort\MusicSort\bin\Release"
#define OutputDirectory "C:\Users\pp50oip\Desktop\Projects\TPI-MusicSort\02-ContenuProjet\Installateur"
#define ScriptDirectory "C:\Users\pp50oip\Desktop\Projects\TPI-MusicSort\02-ContenuProjet\Installateur"
#define Icon "C:\Users\pp50oip\Desktop\Projects\TPI-MusicSort\02-ContenuProjet\MusicSort\MusicSort\Images\music-file.ico"
#define FrameworkInstaller "ndp472-devpack-fra.exe"
```

Figure 40 - Scripte, define

Dans le but de pouvoir facilement utiliser le scripte même si des modifications sont apportées à ses éléments, des variables '#define' ont été utilisées.

- InputDirectory est le dossier où l'application et ses dépendances se trouvent.
- OutputDirectory est le dossier où l'installateur est placé après la compilation du scripte
- ScriptDirectory est le dossier où le script se trouve (utilisé pour trouver l'installateur du Framework).
- Icon est l'icône de l'application.
- FrameworkInstaller est l'installateur du Framework.



Finalement, après avoir configuré les variables ci-dessus, il suffit de compiler le scripte pour générer l'installateur.

5 Tests des fonctionnalités

Ici, les fonctionnalités du projet sont testées par rapport au tests créé dans la liste de tests de la conception. Ce premier tableau consiste aux tests effectués par moi-même, Aurélien Devaud.

Nom du test	Date de passation	Test passé ?	Conditions	Commentaire
Choix du dossier	22.05.2024	✓	PC, Windows 10	
Affichage du contenu du dossier	22.05.2024	✓	Dossiers avec fichier des types demandés et sans.	Tous les fichiers des types demandés sont affichés.
Déplacement d'un fichier dans la playlist	22.05.2024	✓	Fichiers présents dans le dossier.	
Sortie d'un fichier de la playlist	22.05.2024	✓	Fichiers présents dans la playlist.	
Déplacement de tous les fichiers dans la playlist	22.05.2024	✓	Fichiers présents dans le dossier.	
Sortie de tous les fichiers	22.05.2024	✓	Fichiers présents dans la playlist.	
Renommage individuel	22.05.2024	✓	Fichier dans la playlist.	
Renommage de playlist	22.05.2024	✓	Plusieurs fichiers dans la playlist.	
Numérotation des fichiers	22.05.2024	✓	Plusieurs fichiers dans la playlist.	Il n'y a pas d'option suffixe.
Remontée du fichier	22.05.2024	✓	Plusieurs fichiers dans la playlist.	Le déplacement des fichiers supporte le déplacement de groupe non-liés d'éléments.
Décente du fichier	22.05.2024	✓	Plusieurs fichiers dans la playlist.	Le déplacement des fichiers supporte le déplacement de groupe non-liés d'éléments.

Triage des fichiers	22.05.2024		Plusieurs fichiers dans la playlist.	
Triage ascendant	22.05.2024		Plusieurs fichiers dans la playlist.	Il est possible d'inverser l'ordre de la playlist. Qu'il soit trié ou non.
Triage descendant	22.05.2024		Plusieurs fichiers dans la playlist.	Il est possible d'inverser l'ordre de la playlist. Qu'il soit trié ou non.
Annulation des changements d'un fichier	22.05.2024		Un fichier renommé avec un préfixe.	
Annulation des changements de tous les fichiers	22.05.2024		Plusieurs fichiers renommés avec le renommage général et le préfixe.	
Modes d'application : Renommer	22.05.2024		Des fichiers de plusieurs dossiers dans la playlist renommés.	
Modes d'application : Renommer et Copier	22.05.2024		Des fichiers de plusieurs dossiers dans la playlist renommés et un dossier de destination choisi.	
Modes d'application : Renommer et Déplacer	22.05.2024		Des fichiers de plusieurs dossiers dans la playlist renommés et un dossier de destination choisi.	
Lecture de musique	22.05.2024		Des fichiers des types FLAC, MP3 et WMA sont disponibles à la lecture.	
Lecture d'un fichier	22.05.2024		Des fichiers dans le dossier et la playlist.	

Lecture automatique	22.05.2024		La playlist possède plusieurs éléments.	
Dossier de destination	22.05.2024		Un dossier cible est choisi.	
Application des changements	22.05.2024		Des fichiers sont ajoutés dans la playlist et renommés.	
Modification de fichier	22.05.2024		Des fichiers sont ajoutés dans la playlist et renommés.	

Pour chaque partie testée de votre projet, il faut décrire :

- Les conditions exactes de chaque test
- Les preuves de test (papier ou fichier)
- Tests sans preuve : fournir au moins une description

Chaque test contient (exemple) :

- Nom du test
- Mise en place / prérequis
- Étape (ce qu'il faut faire pour que l'action doive s'exécuter)
- Résultats attendus
- Résultats (sous tous les navigateurs par exemple)
- Commentaires

Possiblement faire dans un fichier Excel

5.1 Liste des documents fournis

- Fichiers initiaux :
 - TPI-MusicSort\01-Fichiers-Initiaux
 - Cahier des charges :
D-tpi-Cdc-DLS-Devaud_Aurelien-MusicSort.docx
 - Installeur d'InnoSetup :
innosetup-6.2.2.exe

-
- Contenu du projet :
 - TPI-MusicSort\02-ContenuProjet
 - Installeur :
02-ContenuProjet\Installeur
 - Dossiers des versions :
Installeur\Versions
 - Scripte :
MusicSort-Setup.iss
 - Projet :
02-ContenuProjet\MusicSort
 - Solution :
MusicSort.sln
 - Documentation :
TPI-MusicSort\03-Documentation
 - Planification/Journal de travail sous format Excel :
T-TPI-MusicSort-planification-AurelienDevaud.xlsm
 - Rapport :
TPI-MusicSort-AurelienDevaud-Documentation.docx
 - Maquette :
TPI-MusicSort-AurelienDevaud-Maquette.png
 - Maquette avec menu :
TPI-MusicSort-AurelienDevaud-Maquette-Menu.png
 - Diagrammes : 03-Documentation\UML-Diagrams
 - Diagramme UML de cas d'utilisation :
D-TPI-MusicSort-AurelienDevaud-UseCaseDiagram.drawio
D-TPI-MusicSort-AurelienDevaud-UseCaseDiagram.jpg
 - Diagramme UML d'activité :
D-TPI-MusicSort-AurelienDevaud-ActivityDiagram.drawio
D-TPI-MusicSort-AurelienDevaud-ActivityDiagram.jpg
 - Diagramme UML de classe initial :
D-TPI-MusicSort-AurelienDevaud-ClassDiagram.drawio
D-TPI-MusicSort-AurelienDevaud-ClassDiagram.jpg

6 Conclusions

Ici, les fonctionnalités sont résumées. La planification et le journal de travail comparés. Les différences entre la conception et la réalisation expliquées et mon avis personnel donné.

6.1 Bilan des fonctionnalités

Pour ce projet toutes les fonctionnalités demandées dans les points techniques évalués ont été terminées.

Il est possible de naviguer dans des dossiers et de donner une base pour la navigation. Les fichiers des dossiers disponibles pour la lecture sont tous affichés. Ils peuvent être ajoutés et supprimés de la liste de lecture ainsi qu'être réorganisés et triés avec l'option de déplacer les fichiers dans la liste en groupe ou seul de haut en bas.

Les fichiers musicaux peuvent être lus avec un double clic ou avec le menu contextuel (clic droit). Les formats MP3, WMA, AAC, WAV, MIDI, ASF sont supportés et il est possible de lancer la lecture de la liste entière. Dans ce cas la navigation entre les musiques est possible. Un bouton permet de stopper la lecture.

Il est possible de renommer individuellement et de donner un nom général aux fichiers de la liste. Il est possible d'activer la numérotation dans tel cas elle respectera le nombre de chiffres et le nombre de départ donnés. Toutes ces changements peuvent être réinitialisés.

Il est possible de renommer les fichiers d'origines. L'application cause une validation et chaque écrasement de fichier avec un nom en conflit cause aussi une validation. La fin de l'application des nouveaux noms lance un message d'avertissement.

Il est possible de copier et de renommer les fichiers originaux. Les écrasements causent des validations. Les fichiers originaux existent toujours et la fin du processus lance un message d'avertissement.

Il est possible de déplacer et renommer les fichiers originaux. Les écrasements causent des validations et la fin du processus cause un avertissement.

Finalement, un installateur est généré à l'aide d'un scripte qui permet d'installer l'application sur Windows 10 et 11. L'application fonctionne avec 64 bits et le nom est pertinent.

6.2 Bilan de la planification

La planification a été assez bien faite car le projet s'est déroulé d'une manière assez similaire à elle. Les différences sont assez marginales et compensées par d'autres tâches.

La plus grande différence est le temps supplémentaire qu'a pris la création de l'interface. Cela provient du navigateur sans 'Dialog' de dossier que j'ai dû faire moi-même.

6.3 Analyse post-projet

Ici sont expliquées quelques différences entre la conception et la réalisation ainsi que quelques améliorations possibles.

En premier temps l'interface diffère de la maquette, des titres ont été ajoutés pour clarifier les listes et des positions ont été réajustées pour aligner les contrôles.

En deuxième temps, les FileItems sont créés dans la classe ViewForm contrairement au modèle de classe où ils sont créés dans FolderFileListView pour être plus tard transférés dans PlaylistView. Cela a été changé car les FileItems ne peuvent pas être dans plusieurs listes à la fois.

Un dernier changement fut comment les éléments de la playlist sont triés et gérés presque entièrement par le Model contrairement à la conception où c'est le View qui s'en charge. Le changement a été appliqué car l'ordre permet de donner les bons numéros aux éléments et c'est plus pratique si le model s'en charge.

Sinon, comme améliorations seraient de donner à l'utilisateur l'option de trier les fichiers avec leur propre filtre à la place d'avoir tous les types lisibles affichés. Une barre de recherche ou autre serait aussi une bonne idée.

6.4 Bilan personnel

Ce fut un projet intéressant pour ma part. J'ai pu enrichir mes connaissances dans les fichiers musicaux ainsi que la façon dont les lire et les convertir.

Je suis assez satisfait du résultat de mon travail.

7 Glossaire

Ci-dessous sont expliqués les termes utilisés pouvant porter à confusion. Les définitions sont pour la majeure partie générées par ChatGPT.

Sujet	Définition
#define	Un outil qui permet de créer des variables dans un scripte ISS qui s'applique pour la compilation.
Boîte à outils	Une fenêtre dans un environnement de développement intégré (IDE) qui contient des outils et des contrôles que les développeurs peuvent glisser-déposer dans leur interface utilisateur.
Control	Un élément graphique dans une interface utilisateur, tel qu'un bouton, une zone de texte, une liste déroulante, etc., qui permet à l'utilisateur d'interagir avec le programme.
Délégué	Un type de données qui représente une référence à une méthode avec une liste de paramètres spécifiée et un type de retour.
Dialog	Une fenêtre modale affichée à l'utilisateur pour obtenir des informations ou des instructions supplémentaires, ou pour lui demander de prendre une décision.
Drives	Les périphériques de stockage sur un ordinateur, tels que les disques durs, les lecteurs flash et les lecteurs optiques.
Événement	Une action ou un changement qui se produit dans un programme, auquel le programme peut répondre par l'exécution de code spécifique.
Framework	Un ensemble de bibliothèques, de composants et d'outils logiciels qui fournissent une structure de développement pour créer et gérer des applications logicielles.
Getter	Une méthode utilisée pour récupérer la valeur d'une propriété d'un objet.
GroupBox	Un conteneur d'interface utilisateur qui regroupe des contrôles associés et les affiche ensemble avec une bordure.
Innosetup	Un logiciel d'installation pour Windows qui permet de créer des programmes d'installation pour les applications développées sous Windows.
Installer	Un programme qui installe des logiciels sur un ordinateur en copiant les fichiers nécessaires, en configurant les paramètres, et en effectuant d'autres tâches nécessaires pour préparer le logiciel à l'exécution.
ListView	Un contrôle d'interface utilisateur qui affiche une liste d'éléments avec des colonnes pouvant être triées et personnalisées.
LowerCamelCase	Une convention de nommage où chaque mot dans un identifiant commence par une lettre minuscule, sauf le premier mot.
Objet	Une instance d'une classe dans la programmation orientée objet, contenant des données et des méthodes pour manipuler ces données.

RadioButton	Un type de contrôle d'interface utilisateur qui permet à l'utilisateur de sélectionner une option parmi un ensemble d'options mutuellement exclusives.
Setter	Une méthode utilisée pour attribuer une valeur à une propriété d'un objet.
SubItem	Un élément individuel dans un contrôle ListView qui correspond à une colonne spécifique pour un élément donné.
TreeView	Un contrôle d'interface utilisateur qui affiche une hiérarchie de données sous forme d'arbre, généralement utilisé pour représenter des structures hiérarchiques telles que les répertoires de fichiers.
UpperCamelCase	Une convention de nommage où chaque mot dans un identifiant commence par une lettre majuscule, y compris le premier mot.
Windows Forms	Une technologie de développement d'interface utilisateur pour les applications Windows utilisant le langage de programmation C#.

8 Table des illustrations

Figure 1 - Image titre	1
Figure 2 - Diagramme de cas d'utilisation	9
Figure 3 - Maquette de l'application	12
Figure 4 - Maquette des menus	13
Figure 5 - Sélection d'un dossier	14
Figure 6 - Changement du mode d'application	14
Figure 7 - Affichage des changements	15
Figure 8 - Application des changements	16
Figure 9 - File (MVC)	17
Figure 10 - Model (MVC)	17
Figure 11 - Controller (MVC)	18
Figure 12 - Choix du modèle de projet	23
Figure 13 - Fichiers	23
Figure 14 - Création du projet	23
Figure 15 - Navigateur de dossier	24
Figure 16 - Interface	25
Figure 17 - Résumé du fichier	26
Figure 18 - Navigateur de dossier	27
Figure 19 - Envoyer	27
Figure 20 - Suppression	27
Figure 21 - Changement de position	27
Figure 22 - Trier	27

Figure 23 - Menu Contextuel	27
Figure 24 - Formulaire de renommage	27
Figure 25 - Tout réinitialiser	28
Figure 26 - Changements généraux	28
Figure 27 - Renommage général	28
Figure 28 - Bouton aperçu.....	28
Figure 29 - Appliquer.....	28
Figure 30- Modes d'application.....	28
Figure 31 - Destination	28
Figure 32 - PlaylistPlayer Librairie	29
Figure 33 - Windows Media Player	29
Figure 34 - Composants COM.....	29
Figure 35 - Choisir les éléments	29
Figure 36 - Conversion FLAC -> MP3	30
Figure 37 - Play Stop.....	30
Figure 38 - PlaylistPlayer	30
Figure 39 - Fichiers à installer.....	30
Figure 40 - Scripte, define	31
Figure 41 - Interface numérotée	41

9 Manuels

Ce chapitre explique comment utiliser l'application.

9.1 Manuel d'Installation

Pour installer l'application, allez dans le dossier 'TPI-MusicSort\02-ContenuProjet\Installeur\Versions' et choisissez la version qui vous convient et lancez l'installateur. Il est conseillé d'utiliser la dernière version proposée car elle aura moins de chance d'avoir des bugs.

9.2 Manuel d'Utilisation

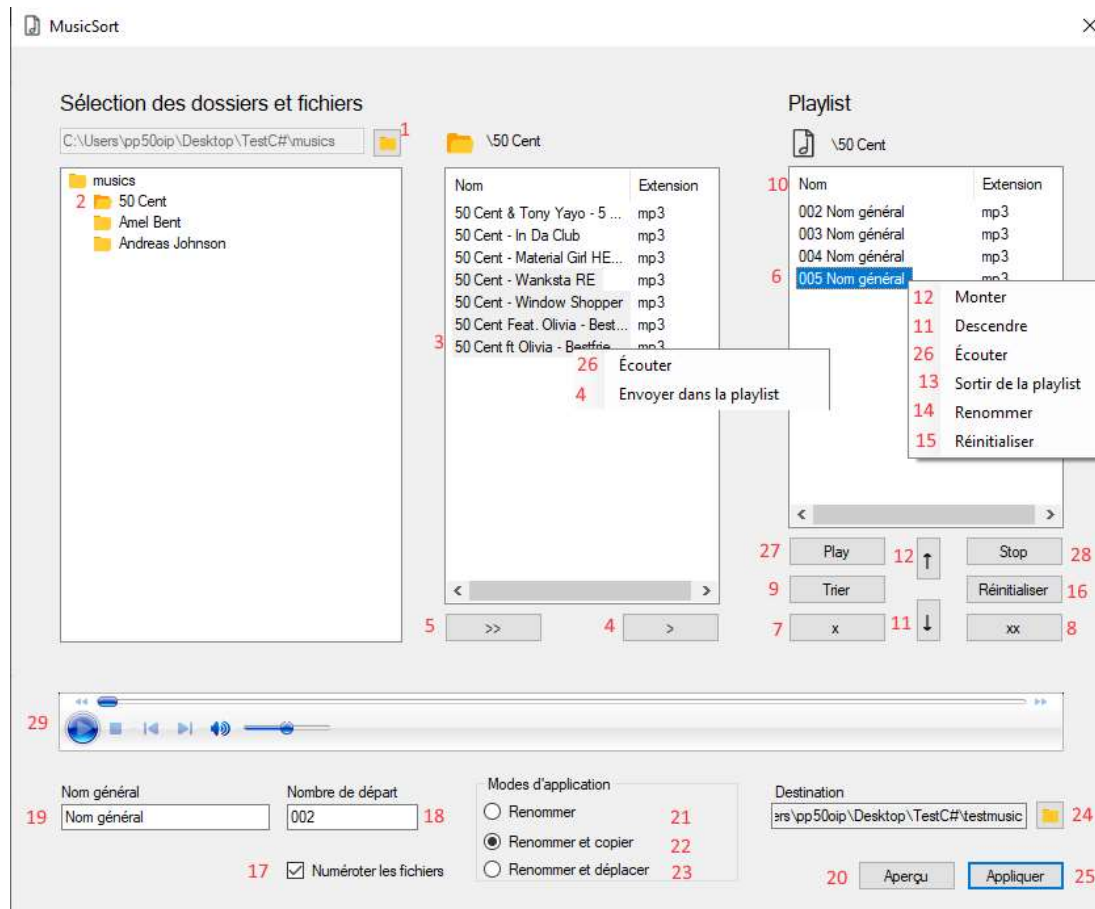


Figure 41 - Interface numérotée

1. Choix du dossier 'racine' du navigateur de dossier
2. En cliquant sur ces dossiers, leurs enfants sont affichés et les fichiers compatibles à la lecture sont affichés dans la liste à droite.
3. Fichier du dossier sélectionné. Un double clic démarre l'écoute du fichier.
4. Envoie les fichiers sélectionnés dans la playlist.
5. Envoie tous les fichiers dans la playlist.
6. Élément de la playlist. Un clic droit ouvre son menu contextuel. Un double clic lance sa lecture.
7. Sort les éléments sélectionnés de la playlist.
8. Sort tous les éléments de la playlist.
9. Trie les éléments de la playlist par ordre alphabétique
10. En cliquant sur nom, l'ordre est inversé.
11. Descend les fichiers sélectionnés d'un cran.
12. Monte les fichiers sélectionnés d'un cran.
13. Sort le fichier de la playlist.
14. Ouvre la fenêtre de renommage du fichier.
15. Réinitialise le nom du fichier.
16. Réinitialise tous les noms des fichiers ainsi que le renommage général.
17. Permet d'activer la numérotation

18. Nombre de départ utilisé pour donner un préfixe aux noms. Le nombre de chiffre total sera respecté.
19. Nom général à donner au fichier. Il est impératif de numéroté les fichiers si un nom général est donné.
20. Applique les changements généraux sur les éléments (pas les fichiers !).
21. Mode d'application des changements qui renomme les fichiers originaux en fonction du nom des éléments.
22. Mode d'application des changements qui copie et renomme les fichiers en fonction du noms des éléments et du dossier de destination choisi.
23. Mode d'application des changements qui déplace les fichiers dans le dossier de destination choisi et les renomme en fonction de leur nom.
24. Permet de choisir un dossier de destination.
25. Applique les changements. Chaque écrasement demandera une confirmation individuelle.
26. Lance la lecture individuelle du fichier.
27. Lance la lecture de la playlist dans l'ordre donné.
28. Stop la lecture de la playlist et vide la file d'attente d'écoute.
29. Controls pour la lecture de musique.

10 Annexes

10.1 Résumé du rapport du TPI / version succincte de la documentation

- *Le résumé est destiné à un public professionnel*
- *Le résumé contient 3 parties ; situation de départ, mise en œuvre, résultats*
- *Le résumé ne contient que des aspects essentiels au projet*
- *Pas plus d'une page A4*
- *Pas de graphiques ni d'images*

10.2 Sources – Bibliographie

Ici sont listées les sources des ressources utilisées durant le projet.

[Folder Icons, Logos, Symbols – Free Download PNG, SVG \(icons8.com\)](#)

[Drive Icons, Logos, Symbols – Free Download PNG, SVG \(icons8.com\)](#)

[Open folder Icons, Logos, Symbols – Free Download PNG, SVG \(icons8.com\)](#)

[Music file Icons, Logos, Symbols – Free Download PNG, SVG \(icons8.com\)](#)

10.3 Archives du projet

Le projet est stocké dans GitHub en privé sous ce lien : [AureDeva/TPI-MusicSort \(github.com\)](#)