

MusicSort-Tri, renommage et création de  
playlist de musique

## Table des matières

<b>1</b>	<b>ANALYSE PRELIMINAIRE.....</b>	<b>3</b>
1.1	INTRODUCTION .....	3
1.2	OBJECTIFS .....	4
1.3	PLANIFICATION.....	4
1.3.1	Méthode des six pas IPDRCE .....	4
1.3.2	Planification détaillée.....	5
<b>2</b>	<b>ANALYSE .....</b>	<b>6</b>
2.1	CONCEPT .....	6
2.1.1	Environnement utilisé.....	6
2.1.2	Stratégie de conception.....	6
2.1.3	Structure du code .....	6
2.1.4	Installer.....	6
2.2	STRATEGIE DE TEST.....	7
2.3	RISQUES .....	7
2.3.1	Risques techniques .....	7
2.3.2	Manque de compétences .....	7
<b>3</b>	<b>CONCEPTION.....</b>	<b>7</b>
3.1	MODELES DE CAS D'UTILISATION .....	8
3.2	MAQUETTES .....	11
3.2.1	Maquette de base .....	11
3.2.2	Maquette des menus.....	12
3.3	DIAGRAMME D'ACTIVITE .....	13
3.3.1	Gestion des fichiers FLAC.....	13
3.3.2	Changement du mode d'application .....	13
3.3.3	Affichage des changements .....	14
3.3.4	Application des changements .....	15
3.4	DIAGRAMME DE CLASSE .....	16
3.4.1	Models.....	16
3.4.2	Views .....	17
3.4.3	Controllers .....	17
3.5	LISTE DES TESTS .....	18
<b>4</b>	<b>REALISATION.....</b>	<b>22</b>
4.1	ENVIRONNEMENT .....	22
4.2	INTERFACE.....	23
4.3	STRUCTURE .....	25

---

4.3.1	Changements par rapport à la conception.....	25
4.4	PROGRAMMATION.....	25
4.5	DESCRIPTION DES TESTS EFFECTUES .....	26
4.6	ERREURS RESTANTES .....	26
4.7	LISTE DES DOCUMENTS FOURNIS .....	26
5	CONCLUSIONS.....	27
6	GLOSSAIRE .....	27
7	TABLE DES ILLUSTRATIONS.....	27
8	ANNEXES.....	28
8.1	RESUME DU RAPPORT DU TPI / VERSION SUCCINCTE DE LA DOCUMENTATION .....	28
8.2	SOURCES – BIBLIOGRAPHIE.....	28
8.2.1	Icônes .....	28
8.3	JOURNAL DE TRAVAIL.....	28
8.4	MANUEL D'INSTALLATION .....	28
8.5	MANUEL D'UTILISATION.....	28
8.6	ARCHIVES DU PROJET.....	29

## 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.

## 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 additionnellement 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.

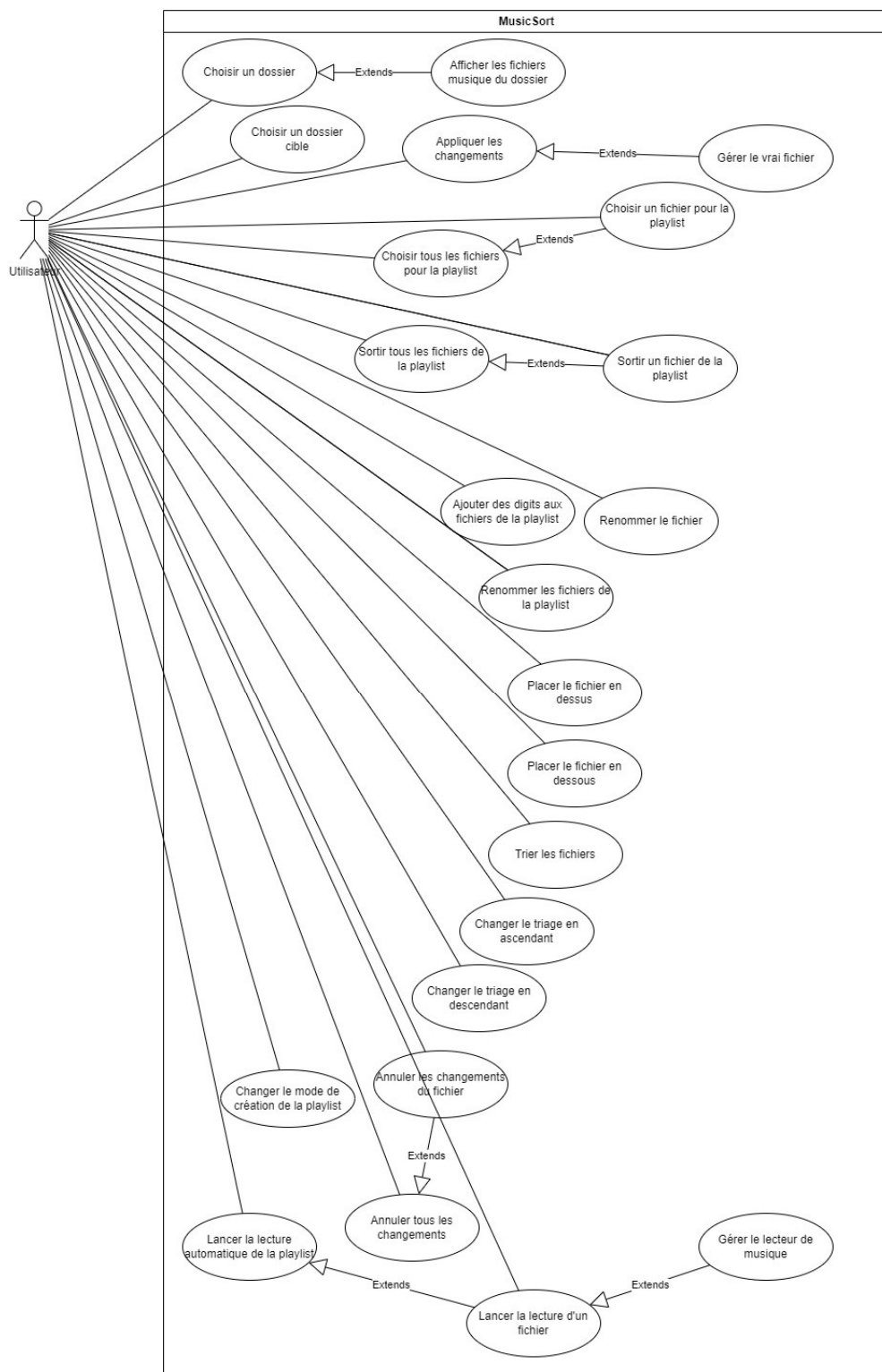


Figure 1 - Diagramme de cas d'utilisation



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 est utilisé entre le renommage, renommage et copie et renommage et déplacement. Finalement, l'utilisateur peut lancer la lecture du 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

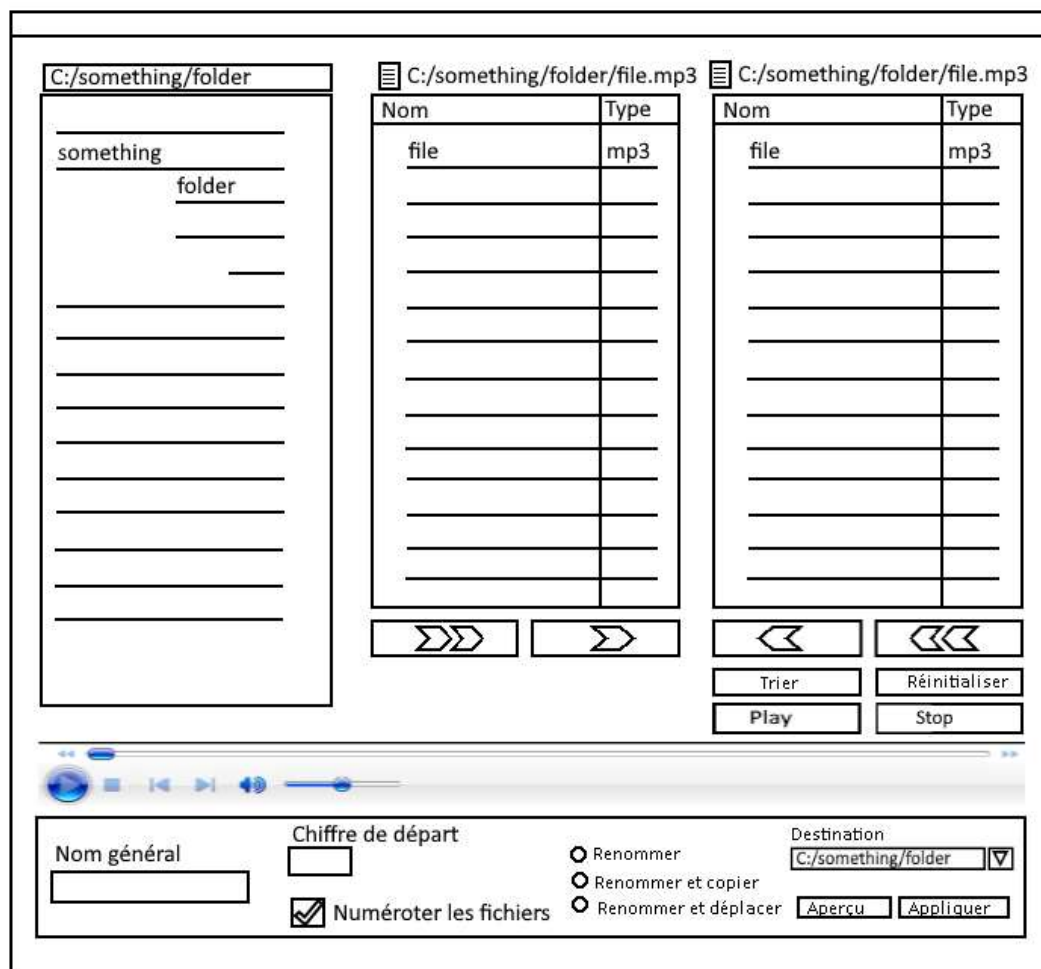


Figure 2 - Maquette de l'application

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

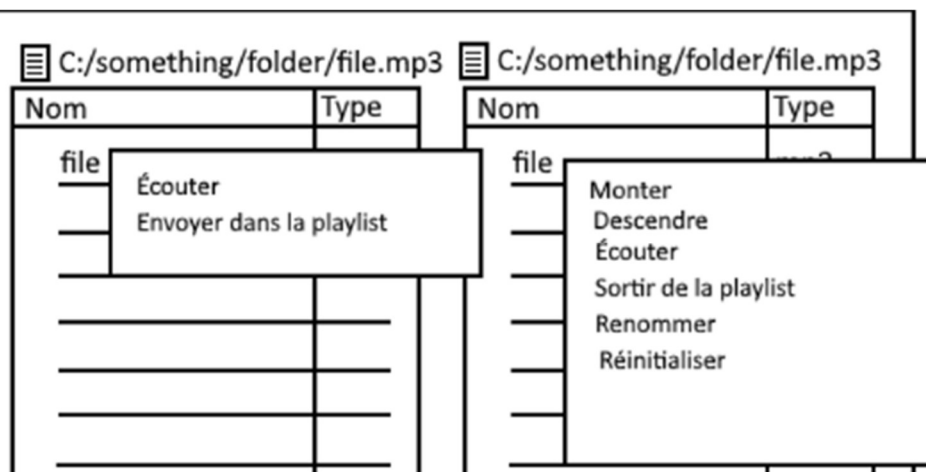


Figure 3 - Maquette des menus

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.

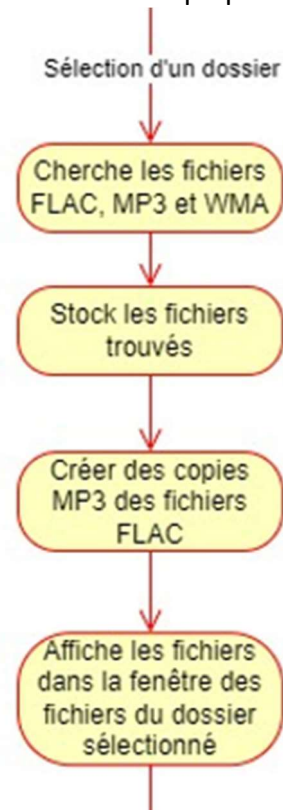


Figure 4 - Sélection d'un dossier

#### 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.

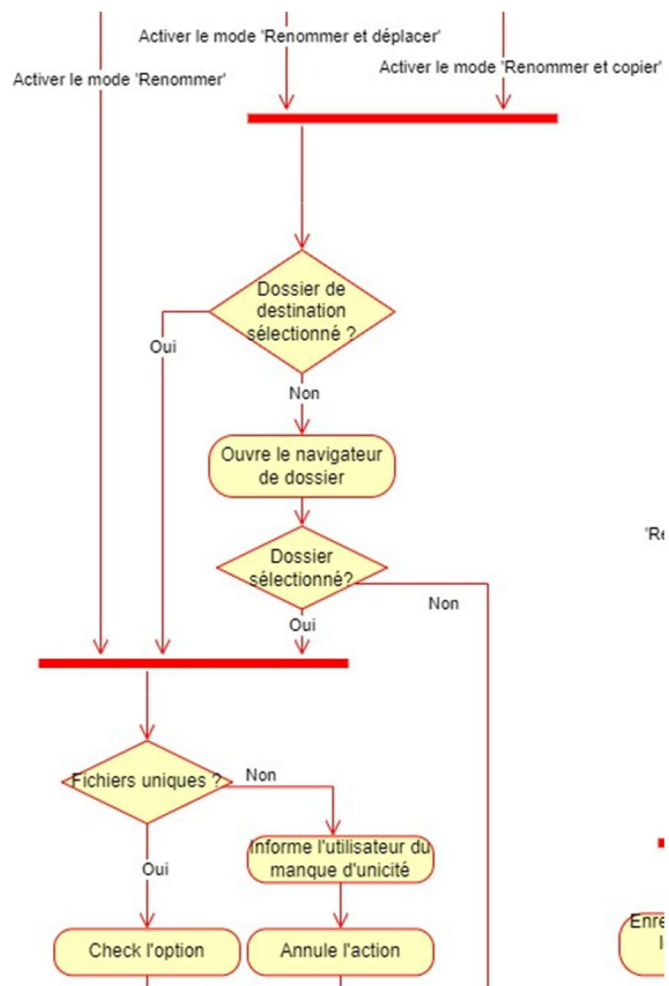


Figure 5 - Changement du mode d'application

### 3.3.3 Affichage des changements

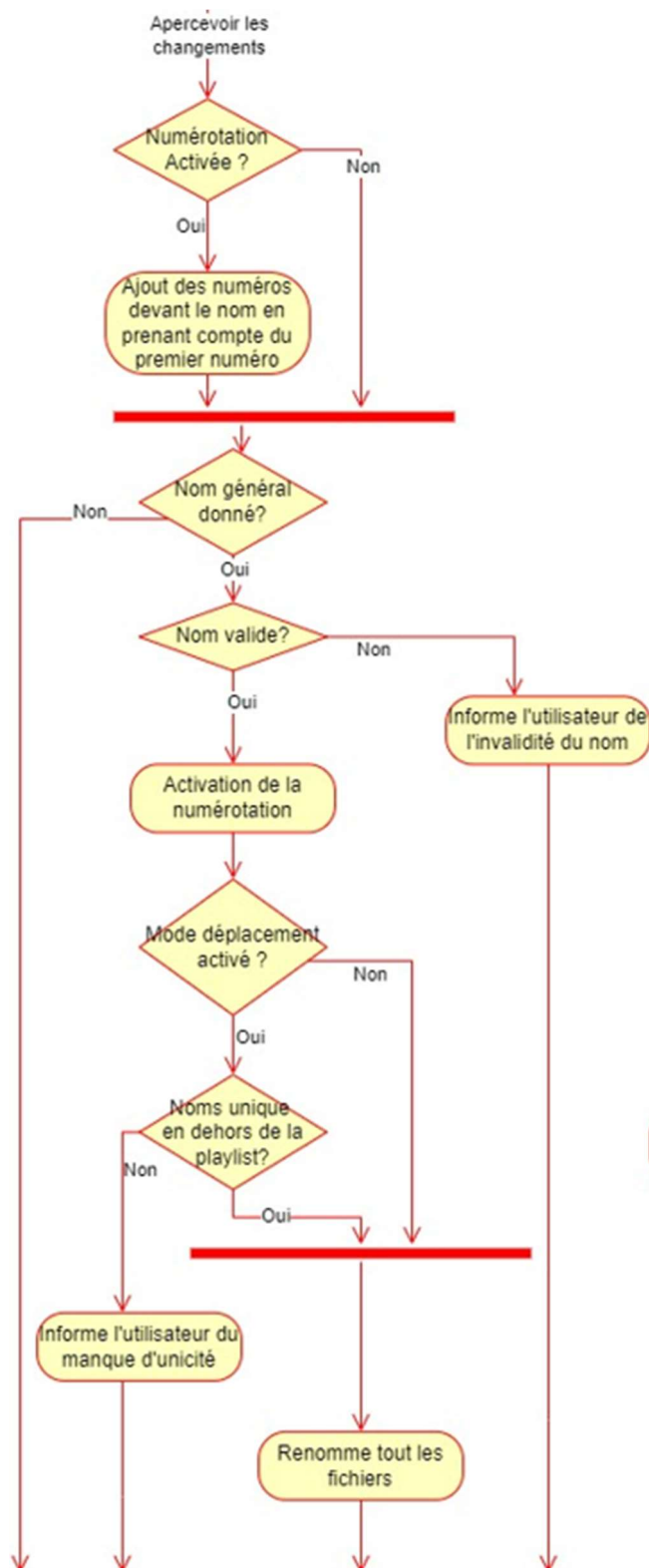


Figure 6 - Affichage des changements

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.

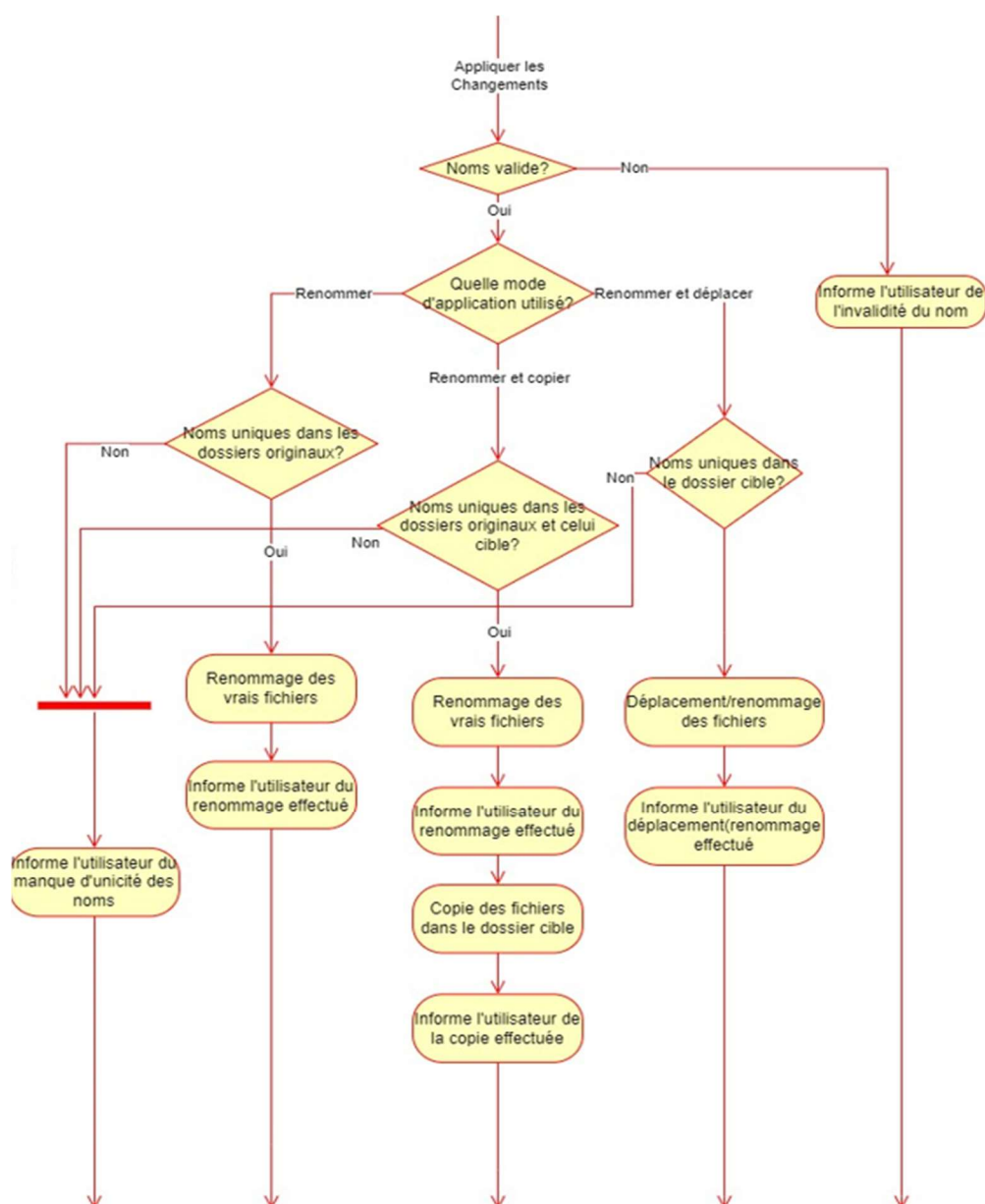


Figure 7 - Application des changements

3.4 Diagramme de classe

Dans ce diagramme de classe, on décrit les méthodes et paramètres 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

Model
<<+get,+set>> Controller : Controller
<<+get,-set>> Playlist : List<File>
<<+get,-set>> DestinationPath : string
<<+get,-set>> SelectedPath : string
<<+get,+set>> GeneralName : string
<<+get,+set>> NumberPrefix : string
<<+get>> NumberDigit : int
<<+get>> StartNumber : int
<<+get,+set>> IsPrefixActivated : bool
<<+get,-set>> ApplicationMode : ApplicationMode
+ Model()
+ SetNewDestination(path : string) : bool
+ SetNewSelectedDirectory(path : string) : bool
+ GetFilesFromDirectory(path : string) : File[]
+ IsValidFileName(name : string) : bool
+ TestMode(mode : ApplicationMode) : bool
+ SetNewMode(mode : ApplicationMode) : void
+ TestForApplication() : void
+ SetPrefixFromIndex() : bool

Figure 8 - Model (MVC)

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.

File
<<+get,-set>> FullRealPath : string
<<+get>> RealExtension : string
<<+get>> RealName: string
<<+get>> RealPath : string
<<+get>> DisplayName: string
<<+get>> FullCustomPath : string
<<+get,-set>> CustomName : string
<<+get,-set>> CustomPath : string
<<+get,-set>> Prefix : string
<<+get,+set>> IndexInPlaylist : int
+ File(fullRealPath : string)
+ WouldFileBeUnique(path : string, customName : string, prefix : string) : bool
+ SetNewPrefix(name : string) : bool
+ SetNewName(name : string) : bool
+ SetNewPath(path : string) : bool
+ ApplyChanges() : bool

Figure 9 - File (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 10 - 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
<b>Choix du dossier</b>	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.
<b>Affichage du contenu du dossier</b>	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.
<b>Déplacement d'un fichier dans la playlist</b>	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.
<b>Sortie d'un fichier de la playlist</b>	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.
<b>Déplacement de tous les fichiers dans la playlist</b>	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.
<b>Sortie de tous les fichiers</b>	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.
<b>Renommage individuel</b>	Renommer le fichier	Tester qu'il soit possible de renommer un fichier de la playlist	Le fichier est renommé dans la playlist

<b>Renommage de playlist</b>	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.
<b>Numérotation des fichiers</b>	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
<b>Remontée du fichier</b>	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. Si la playlist est en mode descendant, il descend.
<b>Décente du fichier</b>	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. Si la playlist est en mode descendant, il monte.
<b>Triage des fichiers</b>	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.
<b>Triage ascendant</b>	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.
<b>Triage descendant</b>	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.

<b>Annulation des changements d'un fichier</b>	Annuler les changements du fichier	Tester qu'il soit possible d'annuler le renommage d'un fichier ainsi que sa position personnalisée dans la liste.	Le fichier recouvre son nom d'origine si le changement de nom de la playlist n'est pas actif. Le digit reste s'il est activé mais la position revient à celle précédente.
<b>Annulation des changements de tous les fichiers</b>	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, le digit et la position des fichiers dans la liste.	La liste revient à son état de départ avant que l'on ait renommé des fichiers, ajouté un nom général, ajouté des digits ou changé les positions dans la liste.
<b>Modes d'application : Renommer</b>	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.
<b>Modes d'application : Renommer et Copier</b>	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.
<b>Modes d'application : Renommer et Déplacer</b>	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.
<b>Lecture de musique</b>	Gérer le lecteur de musique	Test que les fichiers donnés sont bien lu et peuvent être écoutés.	Des fichiers des types FLAC, MP3 et WMA peuvent tous être lu.
<b>Lecture d'un fichier</b>	Lancer la lecture d'un fichier	Tester qu'il soit possible de lire les fichiers de la playlist et du dossier sélectionné.	Quand un fichier est double-cliqué, cela lance sa lecture.
<b>Lecture automatique</b>	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.
<b>Dossier de destination</b>	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.
<b>Application des changements</b>	Appliquer les changements	Tester qu'il soit possible d'appliquer les changements effectués dans l'application.	Les changements sont appliqués de la manière choisie.
<b>Modification de fichier</b>	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

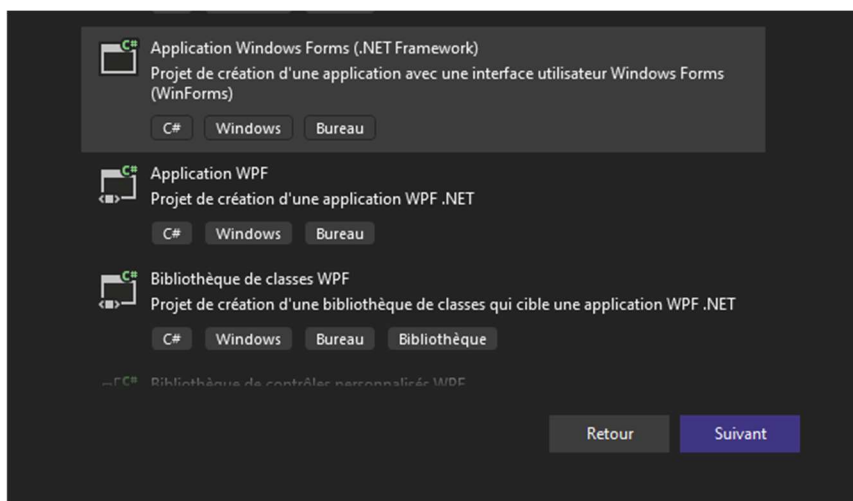


Figure 11 - Choix du modèle de projet

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.

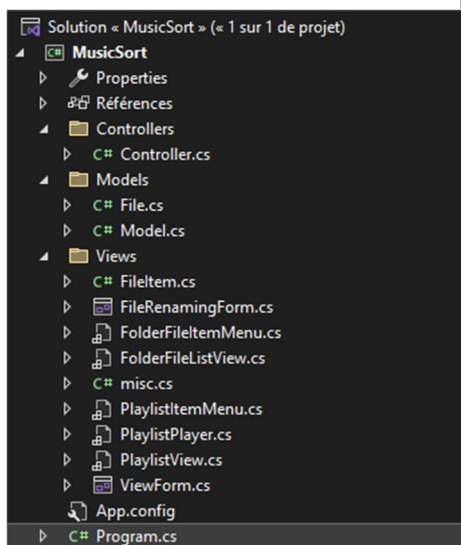


Figure 12 - Fichiers

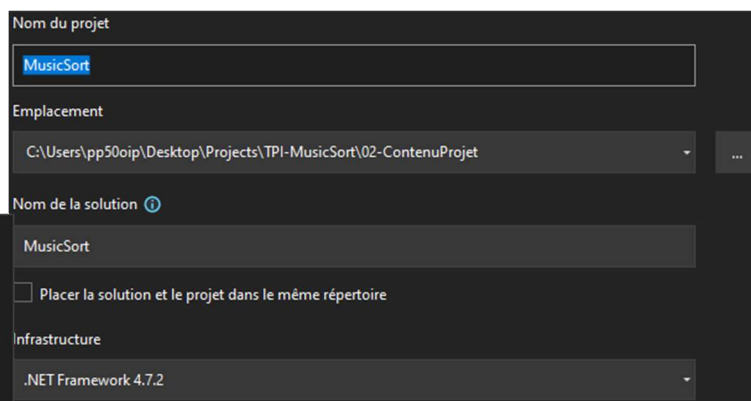


Figure 13 - Création du projet

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 ajoutés sur le formulaire principal.

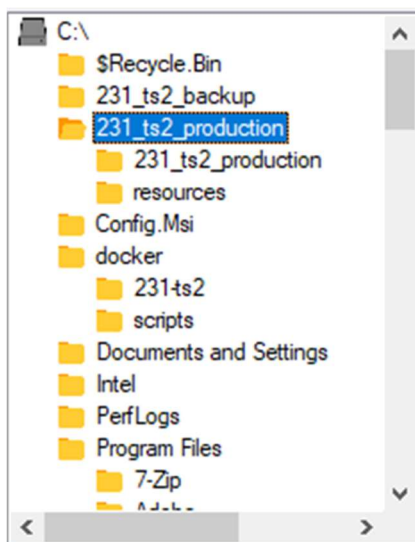


Figure 14 - Navigateur de dossier

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'appellent FolderBrowser et permettent 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 événement 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.

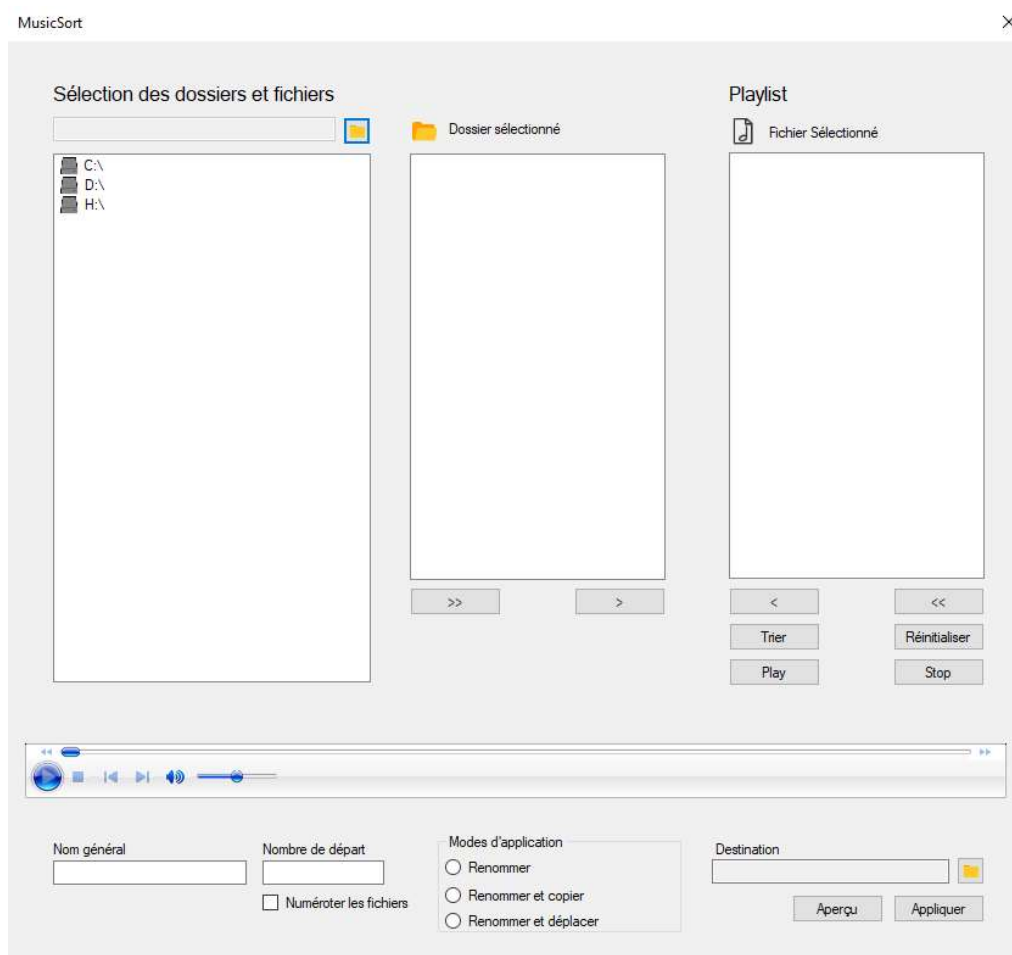


Figure 15 - Interface



### 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 paramètre est commenté et suit la convention de nommage suivante ;

- Les champs sont en « LowerCamelCase » avec un « \_ » au début.
- Les paramètres 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 16 - 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.

Dans la classe « Model », le « setter » du préfixe sera utilisé pour lancer les changements par lui-même, donc il nécessite un champ pour garder en mémoire l'information. « 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.

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

### 4.4 Programmation

La programmation des méthodes s'est effectuée de haut en bas.

Les menus ont donc été les premiers terminés. Vu que les délégués ont directement été passés aux lignes des menus, les paramètres les stockant sont devenu obsolètes et donc ont été supprimés. Le « FileItem » a eu un paramètre « SubItem » ajouté pour afficher les extensions. La méthode de triage alphabétique a été placée dans la classe Model pour ne pas la confondre avec la méthode de triage du « ListView »

Dans le but de ne pas créer de doublons, les méthodes qui gèrent la création de « FileItem » ont été déplacé dans la classe « ViewForm ».

Les fichiers réinitialisés ne reprennent que leur noms car la récupération de son emplacement précédent n'aurait pas beaucoup de sens.

## **4.5 Description des tests effectués**

## **4.6 Erreurs restantes**

## **4.7 Liste des documents fournis**

5 Conclusions

6 Glossaire

Sujet	Définition
Control	
Controls	
Dialog	
Famework	
Innosetup	
Installer	
TreeView	
ListView	
Boîte à outils	
Windows Forms	
RadioButton	
GroupBox	
LowerCamelCase	
UpperCamelCase	
Événement	
Objet	
Délégué	
Setter	
Getter	
SubItem	

Glossaire trié alphabétiquement avec les termes spécifiques du projet

7 Table des illustrations

Figure 1 - Diagramme de cas d'utilisation..... 8

Figure 2 - Maquette de l'application..... 11

Figure 3 - Maquette des menus ..... 12

---

Figure 4 - Sélection d'un dossier .....	13
Figure 5 - Changement du mode d'application.....	13
Figure 6 - Affichage des changements .....	14
Figure 7 - Application des changements .....	15
Figure 8 - Model (MVC) .....	16
Figure 9 - File (MVC) .....	16
Figure 10 - Controller (MVC) .....	17
Figure 11 - Choix du modèle de projet.....	22
Figure 12 - Fichiers .....	22
Figure 13 - Création du projet.....	22
Figure 14 - Résumé du fichier.....	25

## 8 Annexes

### 8.1 Résumé du rapport du TPI / version succincte de la documentation

### 8.2 Sources – Bibliographie

#### 8.2.1 Icônes

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

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

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

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

### 8.3 Journal de travail

### 8.4 Manuel d'Installation

### 8.5 Manuel d'Utilisation

## **8.6 Archives du projet**