ETUDE PREALABLE

Sujet

S22 : Réalisation d'une interface graphique

pour un logiciel de traitement vidéo

Tuteur :

OUNI Slim

Membres de l'équipe du projet :

DA SILVA CARMO Alexandre
PALMIERI Adrien
HUBLAU Alexandre
CHEVRIER Jean-Christophe

Groupe :
S3B

Ce document réunit :

La liste des fonctionnalités

Les maquettes

Les diagrammes de cas d'utilisation

Les diagrammes de classes

Déposé sur le dépôt bitbucket le 09/12/2018

TABLE DES MATIERES

---

3) DIAGRAMMES DE CAS D'UTILISATION

3.1) DIAGRAMME DE CAS D'UTILISATION D'OUVERTURE DU LOGICIEL

3.2) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LE MODE TRAITEMENT

3.2.1) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LES BIBLIOTHEQUES

3.2.2) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LES CHRONOLOGIES

3.3) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LE MODE CONVERSION

3.3.1) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER SUR UN FICHIER

3.3.2) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LA BIBLIOTHEQUE

---

4) PROGRAMMATION : DIAGRAMMES DE CLASSES ET FONCTIONNEMENT

4.1) PROGRAMMES FEDERATEURS

4.1.1) DIAGRAMME DE CLASSES

4.1.2) DESCRIPTION DU FONCTIONNEMENT

4.2) PACKAGE files

4.2.1) DIAGRAMME DE CLASSES

4.2.2) DESCRIPTION DU FONCTIONNEMENT

4.3) PACKAGE ffmpeg_tools

4.3.1) DIAGRAMME DE CLASSES

4.3.2) DESCRIPTION DU FONCTIONNEMENT

4.4) PACKAGE tools

4.4.1) DIAGRAMME DE CLASSES

4.4.2) DESCRIPTION DU FONCTIONNEMENT

4.5) PACKAGE gui_processing

4.5.1) DIAGRAMME DE CLASSES

4.6) PACKAGE gui_conversion

4.6.1) DIAGRAMME DE CLASSES

---

5) CONCLUSION

0) INTRODUCTION

0.1) SUJET DU PROJET

On rappelle que le sujet de notre projet consiste en la réalisation d'une interface graphique. Cette interface graphique a pour mission d'adapter FFmpeg, un logiciel de montage vidéo utilisable qu'en ligne de commande. Le but principal est de **rendre accessible FFmpeg aux utilisateurs non informaticiens**, peu coutumier de l'invite de commande sous Windows, du terminal sous Linux, ou même de tout type de **Shell** (= interpréteur en ligne de commandes).

0.2) CHOIX DE CONCEPTION

Nous avons choisi d'utiliser comme technologie le langage informatique JAVA. FFmpeg étant codé en C, nous ne pourrons pas communiquer directement avec ses librairies. Ce pourquoi nous aurons des classes java réalisant un **« interfaçage » entre notre logiciel et FFmpeg** (**Cf Package ffmpeg_tools**).

Notre logiciel sera découpé en plusieurs packages. Et dans chaque package, des **patrons de conception** seront utilisés. Ces patrons de conception seront adaptés aux fonctionnalités des classes en cause**.** Principal but recherché : **établir une architecture logicielle cohérente et logique, avoir un code factorisé, éviter toute redondance dans les programmes, temps d'exécution optimal, économie de la mémoire.**

Un choix a été réalisé dans notre équipe de projet, nous avons choisis de réaliser tout le code du logiciel en **ANGLAIS**. Enfin… On parle essentiellement ici **des noms des classes, des méthodes et des attributs**, qui seront donc écrits en anglais.

0.3) QUELQUES PRECISIONS

Des maquettes de l'interface graphique seront présentées plus tard dans le document, on tient à préciser que ce sont les **composants graphiques** qu'il faut retenir. En effet les thèmes, les couleurs choisis ici ne sont pas définitifs. **Les thèmes et autres petits détails seront peut-être modifiés plus tard, au gré de nos envies et de nos préférences**.

Ensuite, des diagrammes de classes seront présentés par la suite. Nous nous devons de préciser quelques détails. Il est possible lors de la programmation concrète du logiciel que nous rajoutions encore des **méthodes internes** (méthodes déclarées en private) pour factoriser le code, ou même que nous rajoutions des **classes**, ou des **exceptions personnalisées** (classes héritant d'Exception ou même de Throwable). Soyons honnête nous avons pu oublier certains détails, ou même certains

détails améliorant le codage du logiciel, c'est humain… **Il y aura donc peut-être au cours des étapes suivantes de l'étude du logiciel, des modifications minimes des diagrammes de classes**.

Les maquettes, les diagrammes de cas d'utilisation, et les diagrammes de classes sont – vous allez le voir - plutôt conséquents. Nous vous conseillons donc d'en plus de les regarder sur ce document de consulter les répertoires suivants du dépôt, qui les contiennent :

s3b_s22_chevrier_dasilvacarmo_hublau_palmieri\doc\etape_2\maquettes\

s3b_s22_chevrier_dasilvacarmo_hublau_palmieri\doc\etape_2\ diagrammes_de__cas_d_utilisation\

s3b_s22_chevrier_dasilvacarmo_hublau_palmieri\doc\etape_2\ diagrammes_de_classes\

s3b_s22_chevrier_dasilvacarmo_hublau_palmieri\doc\etape_2\autres\

Ainsi vous pourrez voir les images en grand format, au cas où vous les trouvez flou dans ce document.

Par ailleurs, nous nous sommes centrés sur la réalisation de diagrammes de classes, car ils nous semblaient être **les diagrammes les plus adaptés pour concrétiser où nous voulons allés avec ce projet**.

Dernière chose**, il n'y aura pas de description du fonctionnement des patrons de conception mvc mis en place dans les packages gui_processing, et gui_conversion**, car ce serait d'une grande complexité à rédiger, et beaucoup trop long. Et puis les diagrammes de classes présentés sont assez explicites.

LISTE DES FONCTIONNALITES

Ci-dessous la liste des fonctionnalités de l'interface, triées par groupes et sous-groupes de fonctionnalités. Nous tenons à préciser que cette liste est seulement **énumérative.** C'est-à-dire qu'elle se contente de nommer les fonctionnalités et ne les explique pas en détail. Concernant les détails du fonctionnement des fonctionnalités, cela sera approcher avec les diagrammes de cas d'utilisation, puis développer avec les diagrammes de classes auxquels seront ajoutés des descriptions. Pour le moment on se contente d'énumérer les fonctionnalités.

ATTENTION ! Dans la fenêtre de traitement, les fichiers acceptés sont les fichiers vidéo, les fichiers audio, et les fichiers image, tandis que dans la fenêtre de conversion, les fichiers image ne sont pas acceptés. Dans la fenêtre de conversion, on peut convertir dans d'autres format seulement des fichiers vidéo ou audio.

Petite précision : dans la fenêtre de traitement**, il y a 4 bibliothèques** (multimédia, vidéo, audio, image), tandis que dans la fenêtre de conversion, il n'y a **qu'une seule bibliothèque**, elle s'apparente à une liste.

Ensuite, autre détail important, **dans la fenêtre de conversion, la bibliothèque désigne directement les fichiers sur lesquels on travaille, tandis que dans la fenêtre de traitement un fichier devient un fichier de travail uniquement lorsqu'il est ajouté dans les chronologies**. Ce qui est une grande différence.

1.1) GROUPE DE FONCTIONNALITES : « VIA LA FENETRE D'OUVERTURE »

- afficher une fenêtre de chargement
- afficher une fenêtre de choix du mode d'utilisation
- démarrer une fenêtre de traitement
- démarrer une fenêtre de conversion

1.2) GROUPE DE FONCTIONNALITES : « VIA LA FENETRE DE TRAITEMENT »

1.2.1) MENUS

afficher les menus dans une barre de menus

Menu Fichier

- commencer un nouveau projet
- passer en en mode conversion
- quitter
- exporter un traitement

- exporter et convertir un traitement
- choisir un format de sortie
- enregistrer un traitement
- charger un traitement
- enregistrer une image extraite
- afficher **certains** fichiers (choisis par le système) récemment importés lors d'une précédente session de travail
- charger **un** fichier récemment importé lors d'une précédente session de travail

Menu Bibliothèques

- importer **un** fichier multimédia dans les bibliothèques
- importer **plusieurs fichiers multimédias d'un même dossier** dans les bibliothèques
- supprimer un ou plusieurs fichiers sélectionnés dans une bibliothèque
- vider toute la bibliothèque multimédia
- vider toute la bibliothèque audio
- vider toute la bibliothèque vidéo
- vider toute la bibliothèque image

Menu Chronologies

- ajouter **un ou des** fichiers sélectionné(s) d'une bibliothèque dans la ou les chronologie(s) correspondante(s)
- supprimer **un** fichier sélectionné dans la chronologie
- couper **un** fichier sélectionné dans la chronologie (dépend de la position du curseur)
- zoomer sur la chronologie
- dézoomer sur la chronologie

Menu Vidéo

- lire la vidéo
- mettre en pause la vidéo
- extraire l'image actuelle de la vidéo
- pivoter la vidéo de 90°
- pixeliser la vidéo

1.2.2) PANELS BIBLIOTHEQUES

- supprimer **un ou plusieurs** fichiers sélectionnés (touche suppr)
- défiler sur une des bibliothèques (barre de scroll)
- sélectionner un ou des fichier(s) multimédia (clic souris + touche control enfoncée)
- sélectionner un ou des fichier(s) vidéo (clic souris + touche control enfoncée)
- sélectionner un ou des fichier(s) audio (clic souris + touche control enfoncée)
- sélectionner un ou des fichier(s) image (clic souris + touche control enfoncée)
- changer de bibliothèque en changeant d'onglet (exemple : passer de l'onglet bibliothèque image à bibliothèque vidéo.

1.2.3) BIBLIOTHEQUES VERS CHRONOLOGIES

glisser un ou des fichier(s) importé(s) de la bibliothèque vers la chronologie (ce qui permet d'ajouter des fichiers dans la chronologie)

1.2.4) PANEL CHRONOLOGIES

- afficher les chronologies
- défiler sur la chronologie (barre de scroll)
- sélectionner **un** fichier et en même temps mettre à jour la position du curseur au point du clic
- supprimer un fichier de la chronologie (touche suppr)
- couper (raccourcir) un fichier (touche c (c pour “cut” : couper))

1.2.5) PANEL BARRE DE LECTURE

- afficher la barre de lecture et le bouton play/pause
- activer la lecture de la vidéo
- mettre en pause la vidéo
- se déplacer dans la vidéo (aller à un moment précis de la vidéo)

1.2.6) PANEL VIDEO

- afficher la vidéo
- lire la vidéo de manière fluide (environ 25 frames/secondes)
- lire la vidéo de manière ralentie (nombre de frames/secondes < 25)
- mettre en pause
- rogner une vidéo graphiquement dessus

1.3) GROUPE DE FONCTIONNALITES : « VIA LA FENETRE DE CONVERSION »

1.3.1) MENUS

afficher les menus dans une barre de menus

Menu Fichier

- importer un fichier
- importer plusieurs fichiers d’un même dossier
- afficher **certains** fichiers (choisis par le système) récemment importés lors d'une précédente session de travail
- réimporter des fichiers précédemment importés
- vider la bibliothèque
- passer en mode traitement
- quitter

Menu Profils

- charger un profil
- enregistrer un profil
- supprimer un profil

Menu Option

- choisir un répertoire de sortie

Bouton convertir une vidéo

1.3.2) PANEL BIBLIOTHEQUE

- afficher la bibliothèque des éléments à convertir sous forme de liste
- supprimer le fichier sélectionné (touche suppr)

1.3.3) PANEL « RESUME »

afficher les informations principales sur la vidéo sélectionnée dans la bibliothèque

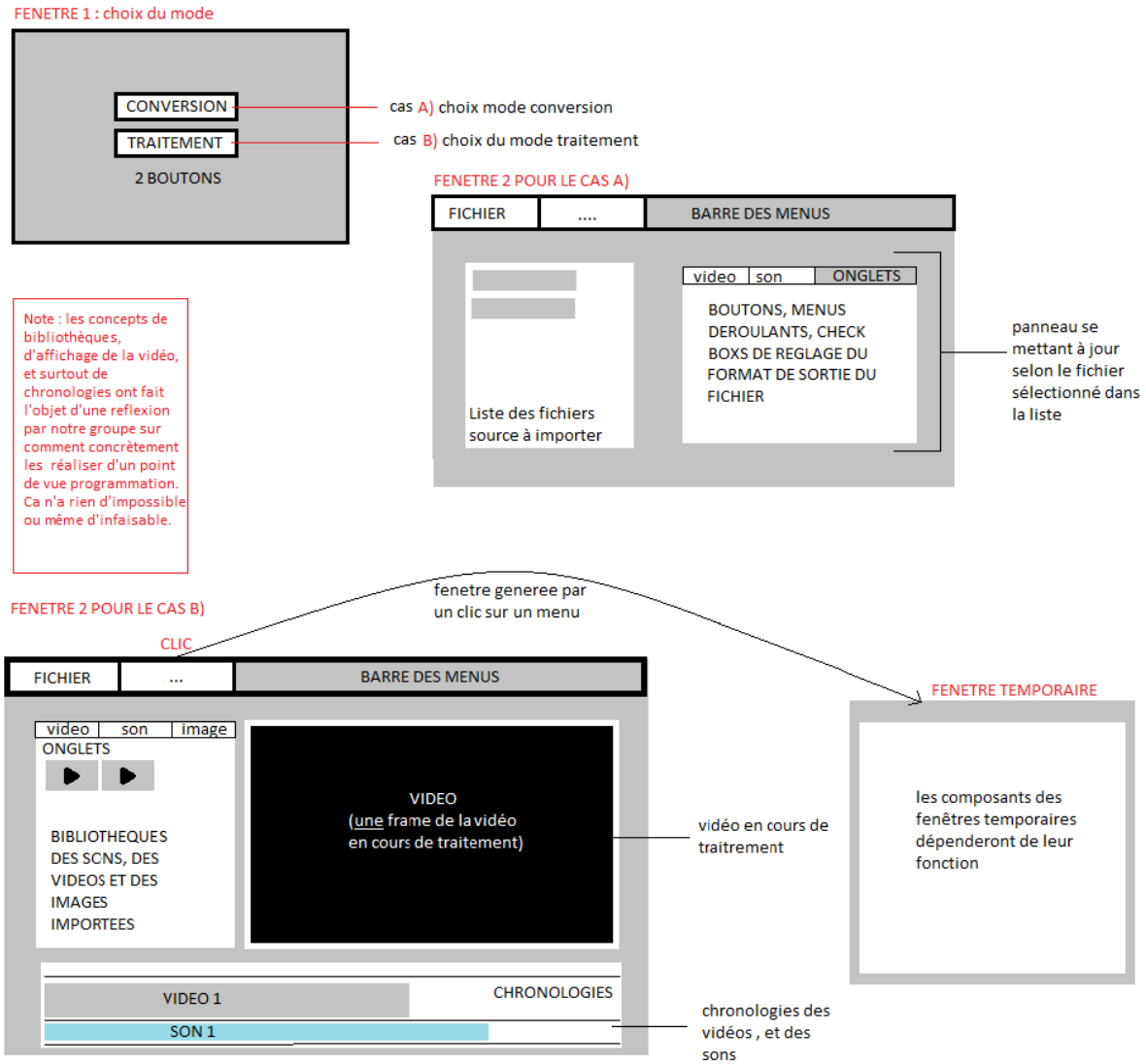
1.3.4) PANELS PARAMETRES DU FICHIER

- afficher les paramètres du fichier
- modifier les codecs audio/vidéo en sortie
- modifier la taille/qualité du média en sortie (Note : cela est fait en utilisant des profils qui modifient automatiquement les autres paramètres de sortie)
- modifier le bitrate audio/vidéo en sortie
- modifier le taux d'échantillonnage audio en sortie
- modifier le nombre d'images par secondes de la vidéo en sortie (FPS)k
- modifier le nombre de canaux audio en sortie
- modifier le volume audio en sortie
- changer la résolution d'une vidéo en sortie
- extraire le son d'une vidéo
- ajouter des sous-titres à la vidéo

2) MAQUETTES DES FENETRES DU LOGICIEL

2.1) PREMIER POTOTYPE ET REFLEXIONS

Avant d'arriver à des maquettes concrètes, nous avions réalisés une première maquette ambiguë de ce que nous attendions de notre interface. Elles montrent les éléments principaux que nous souhaitons mettre en place. La voici :

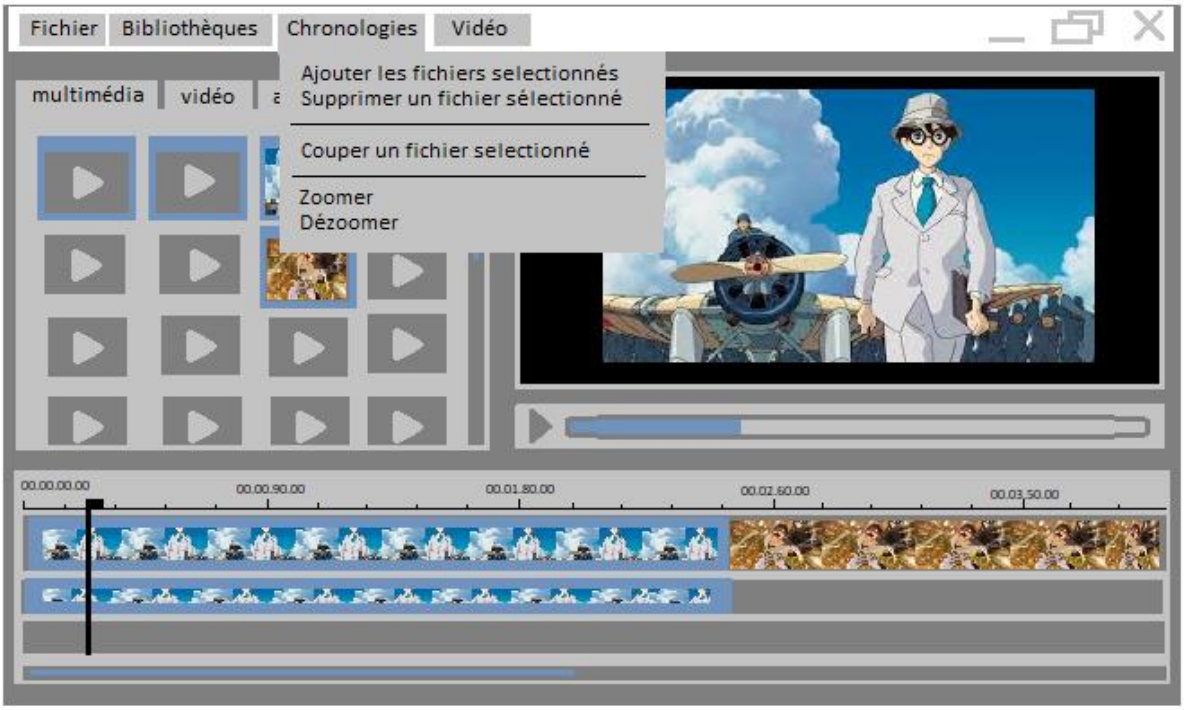


Il faut retenir de cette première maquette ambiguë une chose essentielle : notre programme génère au début une fenêtre d'**OUVERTURE** qui permet de choisir le mode d'utilisation. L'utilisateur choisit soit le mode **CONVERSION**, soit le mode **TRAITEMENT**. Selon le mode choisi, l'interface générée est différente. Le reste est sans importance pour cette maquette, car nos points de vue ont évolué depuis. Nous vous demandons donc de ne pas la considérer comme une maquette de référence, mais comme une simple première approche. Pour les détails des composants graphiques des fenêtres, cela est davantage visible sur les maquettes suivantes.

2.2) MAQUETTE DE LA FENETRE DE TRAITEMENT

Voici une suite de maquettes présentant la fenêtre de traitement. **On peut y voir aisément les composants graphiques et ce que propose chaque menu de la barre des menus**. Ces diagrammes permettent **de concevoir comment les fonctionnalités prévues seront traduites dans l'interface graphique**.


Fichier
Bibliothèques
Chronologies
Vidéo
Importer des fichiers
Supprimer les fichiers sélectionnés
Supprimer tous les fichiers importés
Vider la bibliothèque vidéo
Vider la bibliothèque audio
Vider la bibliothèque image
00.00.00.00
00.00.90.00
00.01.80.00
00.02.60.00
00.03.50.00

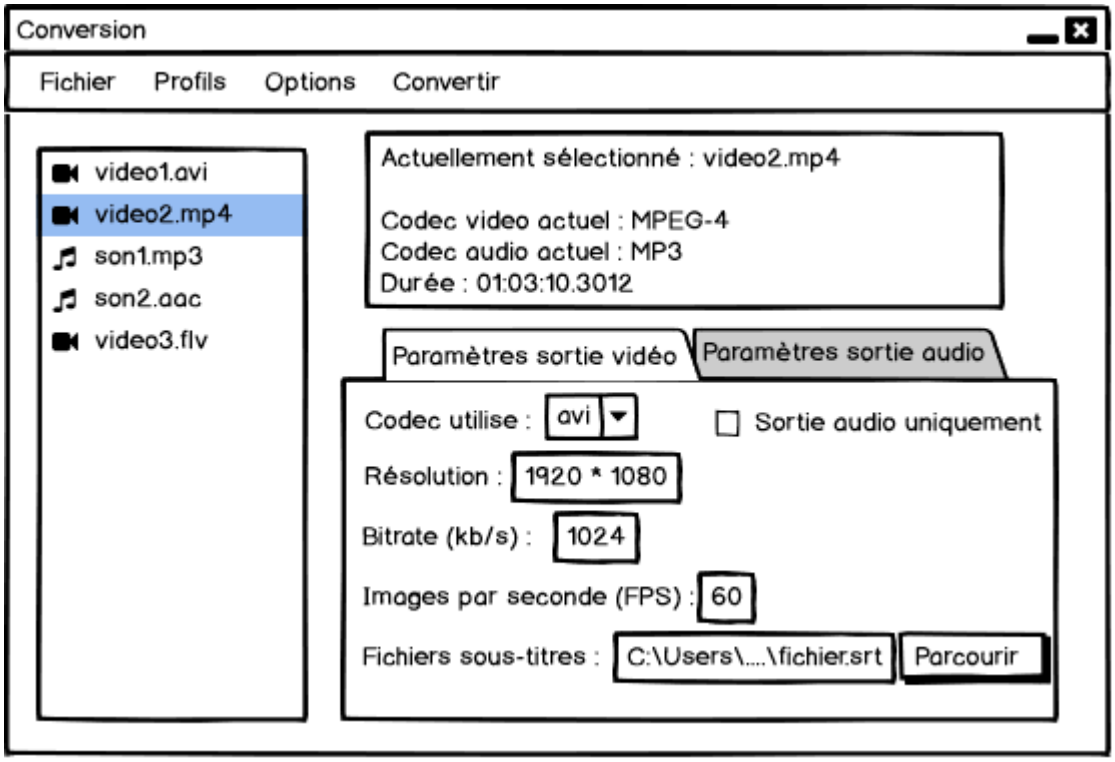

Fichier
Bibliothèques
Chronologies
Vidéo
multimédia
vidéo
Ajouter les fichiers selectionnés
Supprimer un fichier sélectionné
Couper un fichier selectionné
Zoomer
Dézoomer
00.00.00.00
00.00.90.00
00.01.80.00
00.02.60.00
00.03.50.00

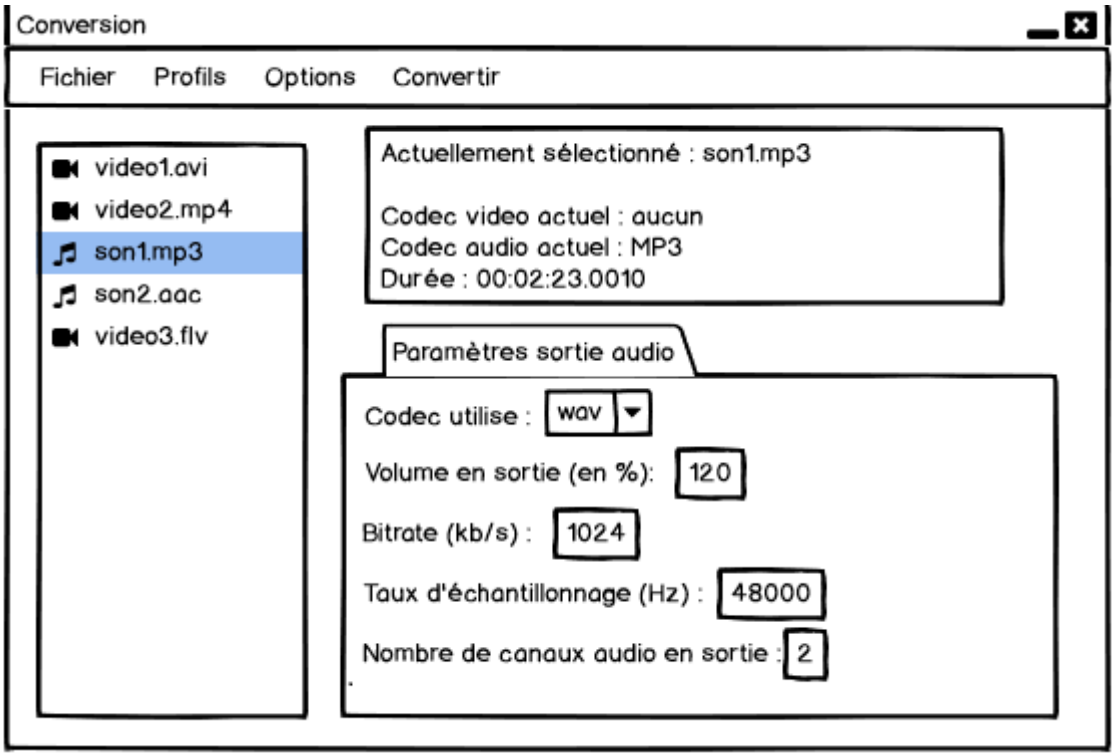
2.2) MAQUETTE DE LA FENETRE DE CONVERSION

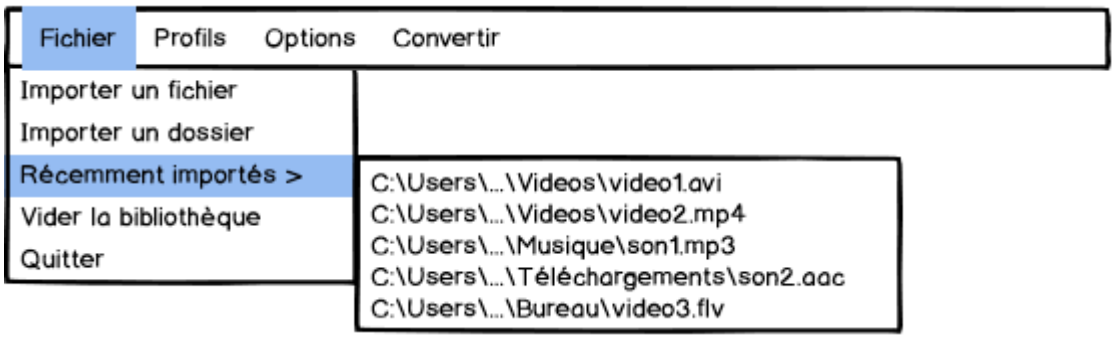
De même, pour la fenêtre de conversion, voici une suite de maquettes.

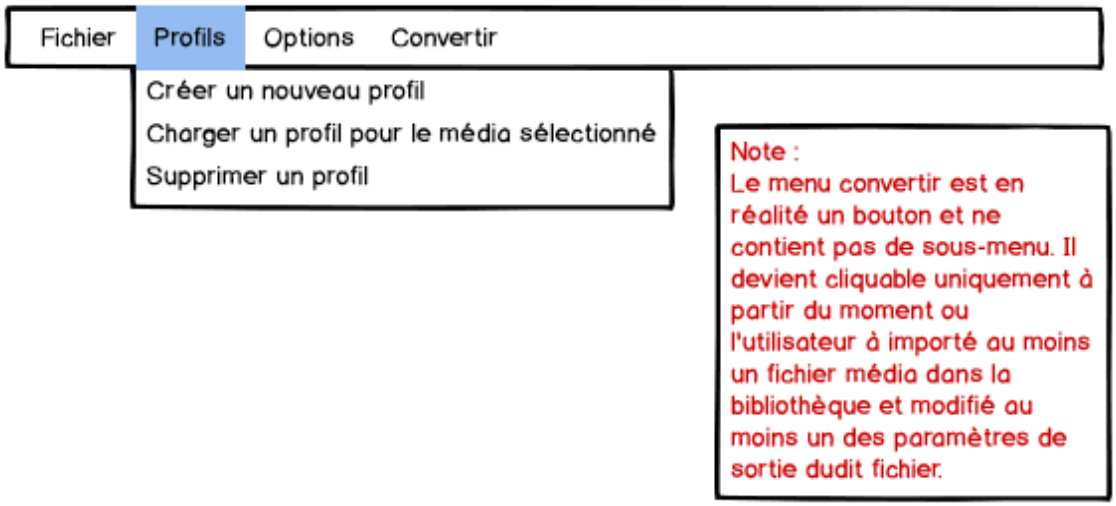
Note : nous avions tentés initialement de réaliser les maquettes de cette fenêtre dans le même style et avec les mêmes thèmes que la fenêtre de traitement mais nous avons eu des imprévus… Voici donc la fenêtre de conversion non accordée avec les thèmes de la fenêtre de traitement.

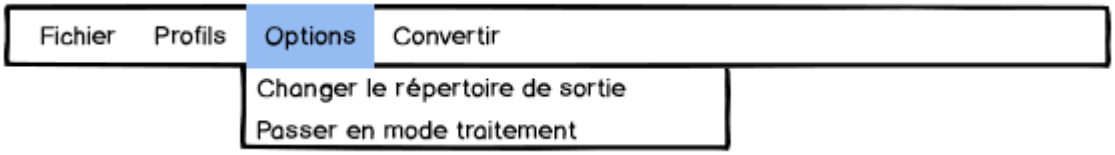
Ensuite il faut ajouter que dans le menu Fichier, normalement il y a aussi « Supprimer la vidéo sélectionnée ».


Conversion
Fichier
Profils
Options
Convertir
video1.avi
video2.mp4
son1.mp3
son2.aac
video3.flv
Actuellement sélectionné : video2.mp4
Codec video actuel : MPEG-4
Codec audio actuel : MP3
Durée : 01:03:10.3012
Paramètres sortie vidéo
Paramètres sortie audio
Codec utilise : avi
Sortie audio uniquement
Résolution : 1920 * 1080
Bitrate (kb/s) : 1024
Images par seconde (FPS) : 60
Fichiers sous-titres : C:\Users\....\fichier.srt
Parcourir


Conversion
Fichier
Profils
Options
Convertir
video1.avi
video2.mp4
son1.mp3
son2.aac
video3.flv
Actuellement sélectionné : son1.mp3
Codec video actuel : aucun
Codec audio actuel : MP3
Durée : 00:02:23.0010
Paramètres sortie audio
Codec utilise : wav
Volume en sortie (en %): 120
Bitrate (kb/s) : 1024
Taux d'échantillonnage (Hz) : 48000
Nombre de canaux audio en sortie : 2


Fichier
Profils
Options
Convertir
Importer un fichier
Importer un dossier
Récemment importés >
Vider la bibliothèque
Quitter
C:\Users\...\Videos\video1.avi
C:\Users\...\Videos\video2.mp4
C:\Users\...\Musique\son1.mp3
C:\Users\...\Téléchargements\son2.aac
C:\Users\...\Bureau\video3.flv


Fichier
Profils
Options
Convertir
Créer un nouveau profil
Charger un profil pour le média sélectionné
Supprimer un profil
Note :
Le menu convertir est en réalité un bouton et ne contient pas de sous-menu. Il devient cliquable uniquement à partir du moment ou l'utilisateur à importé au moins un fichier média dans la bibliothèque et modifié au moins un des paramètres de sortie dudit fichier.


Fichier
Profils
Options
Convertir
Changer le répertoire de sortie
Passer en mode traitement

3) DIAGRAMMES DE CAS D'UTILISATION

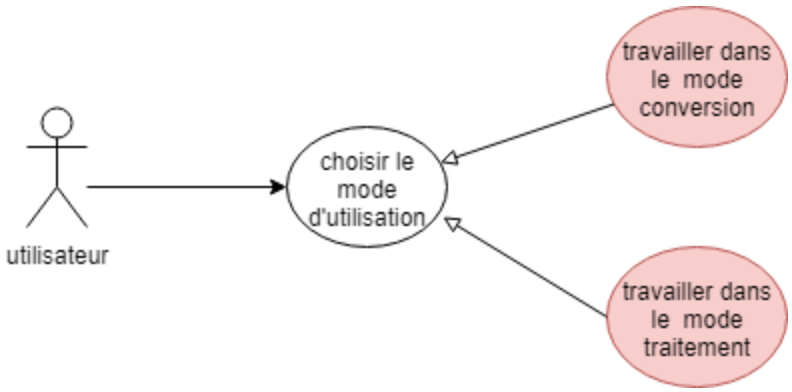
Le système d'information étudié ici est le logiciel vidéo. 2 acteurs sont visibles : **l'utilisateur du logiciel et le système**.

Il faut prendre en compte, **que certaines fonctionnalités n'avaient pas d'intérêt à être préciser dans les diagrammes de cas d'utilisation**. Par exemple défiler sur la bibliothèque, qui est une fonctionnalité qui décrit l'existence d'une barre de scroll dans la bibliothèque, n'a pas d'intérêt à être précisée dans les diagrammes de cas d'utilisation.

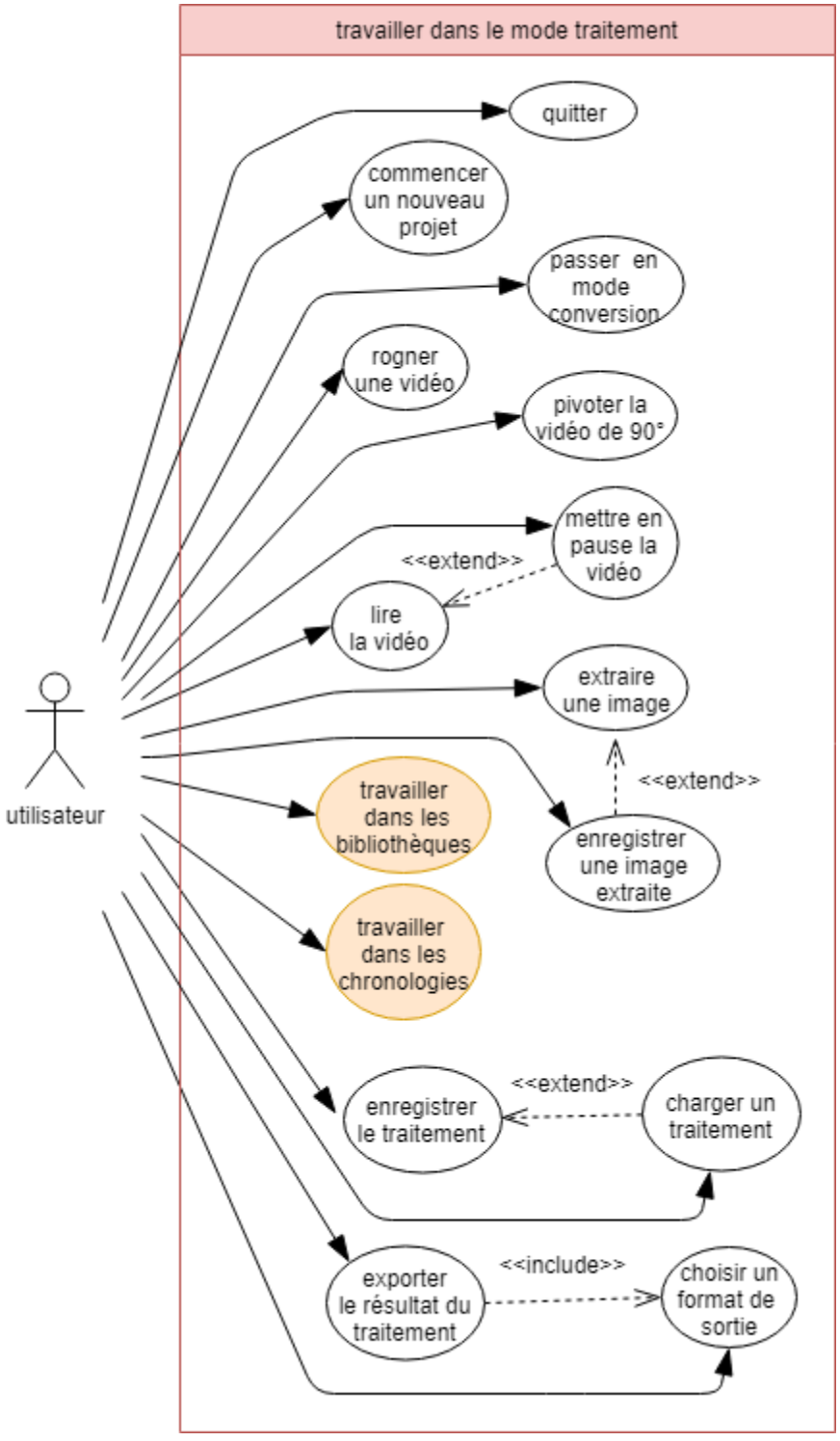
Ensuite notez que toutes les flèches, placées dans nos diagrammes de cas d'utilisation, de type : include, entend, **ont été placées avec une certaine réflexion**. Même si dans certains cas cela donne une surcharge de flèches dans le diagramme, **cette surcharge est volontaire et utile selon nous**.

Voici les diagrammes de cas d'utilisation.

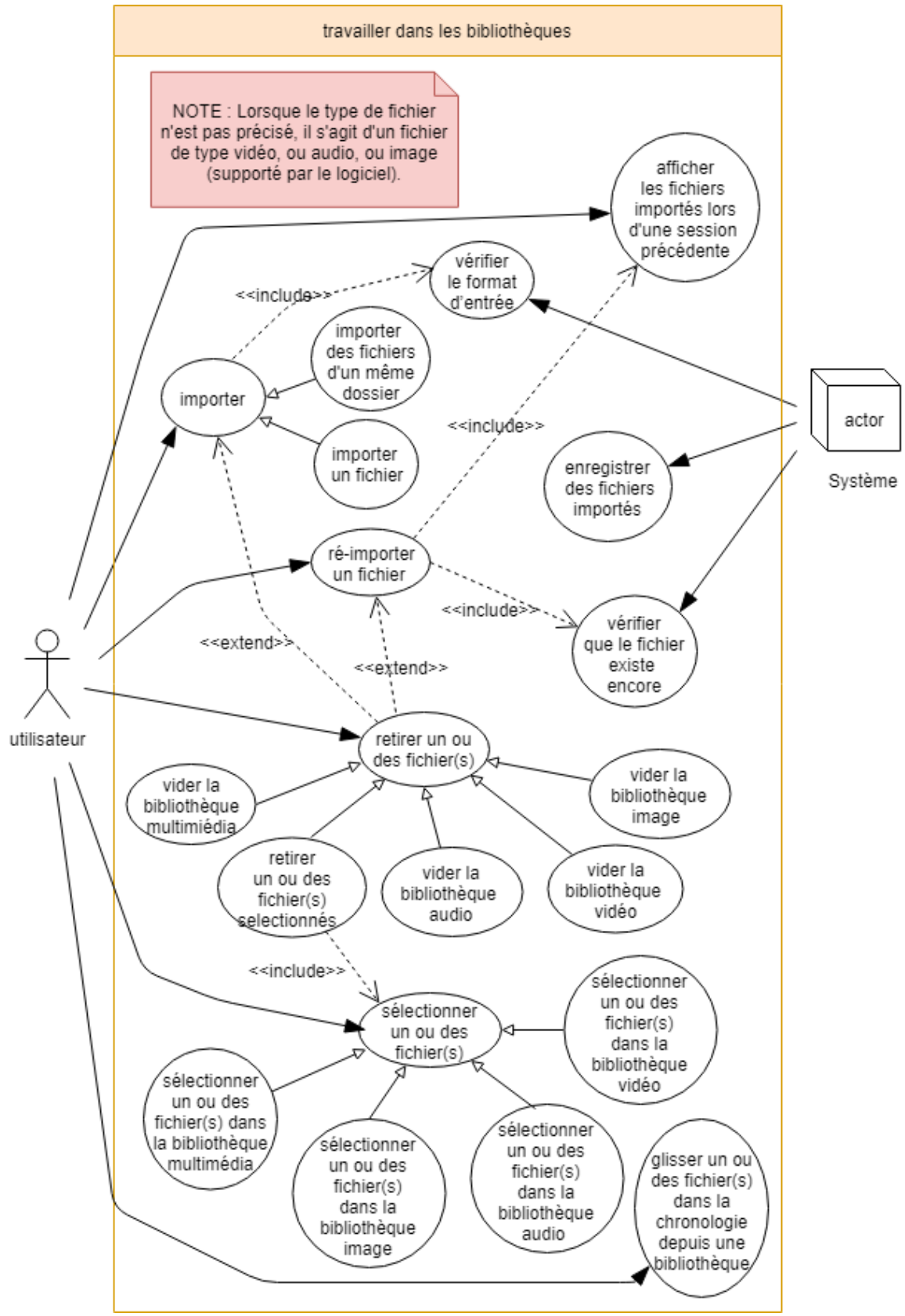
3.1) DIAGRAMME DE CAS D'UTILISATION D'OUVERTURE DU LOGICIEL



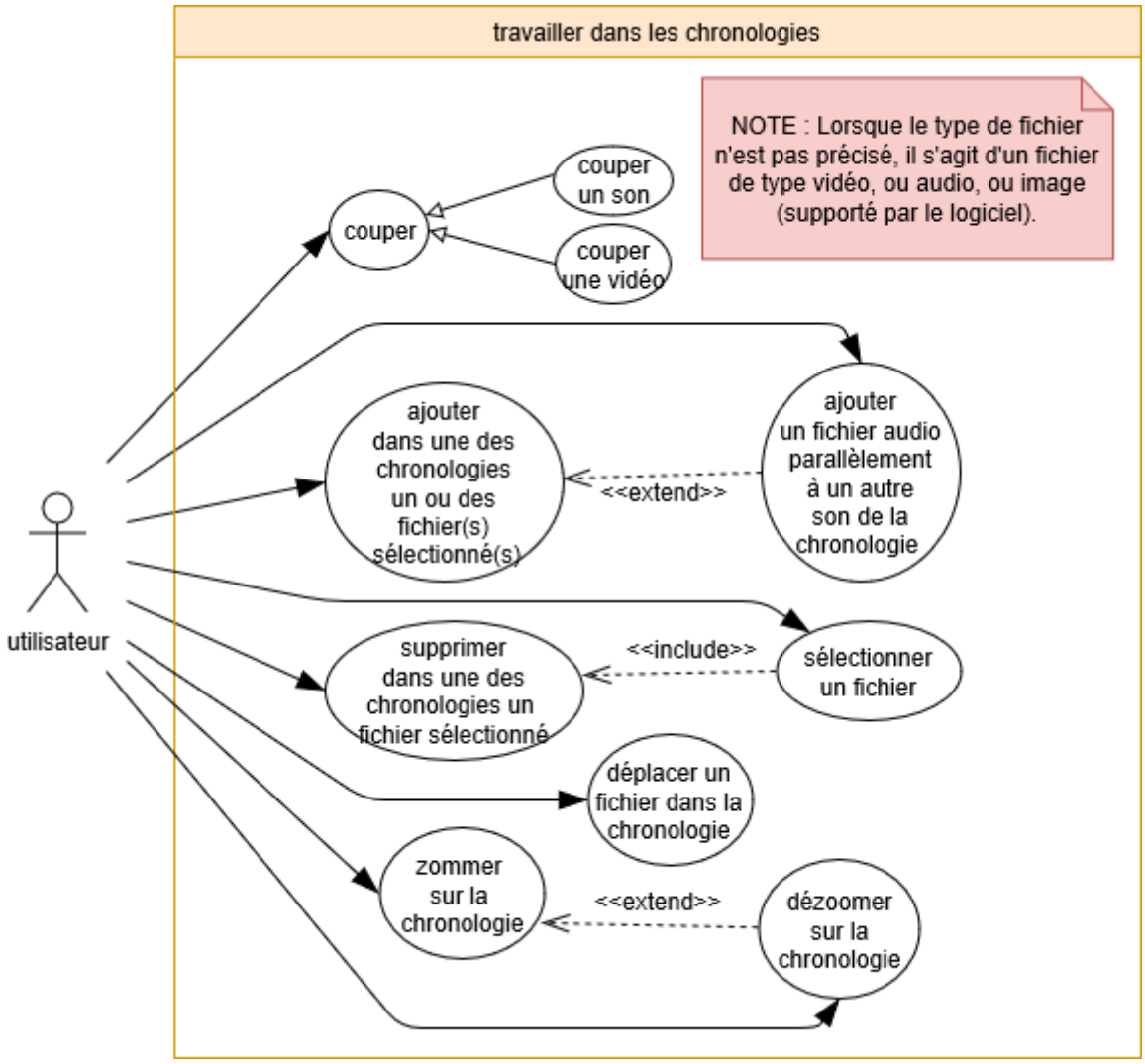
3.2) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LE MODE TRAITEMENT



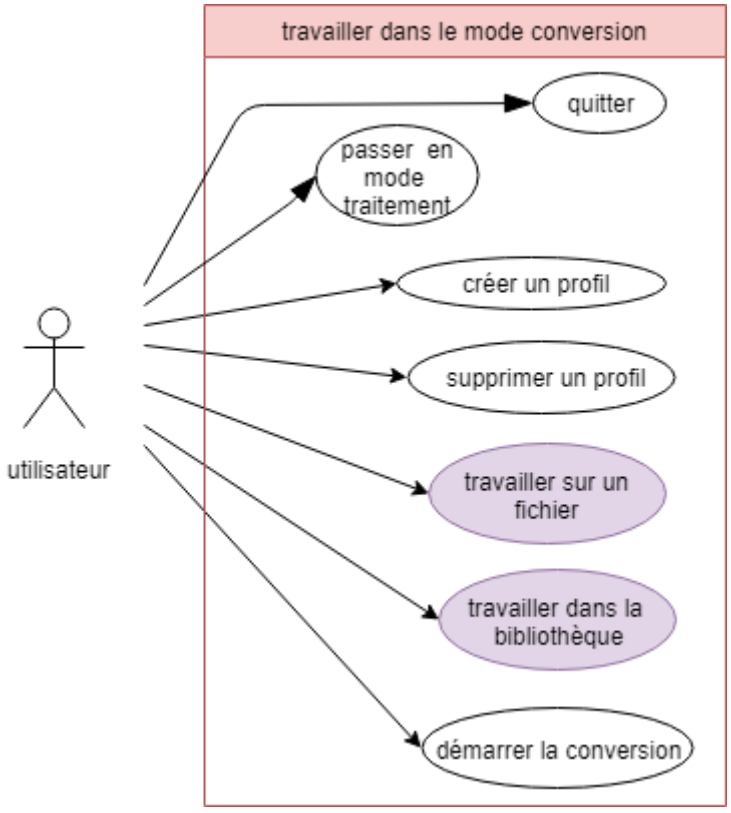
3.2.1) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LES BIBLIOTHEQUES



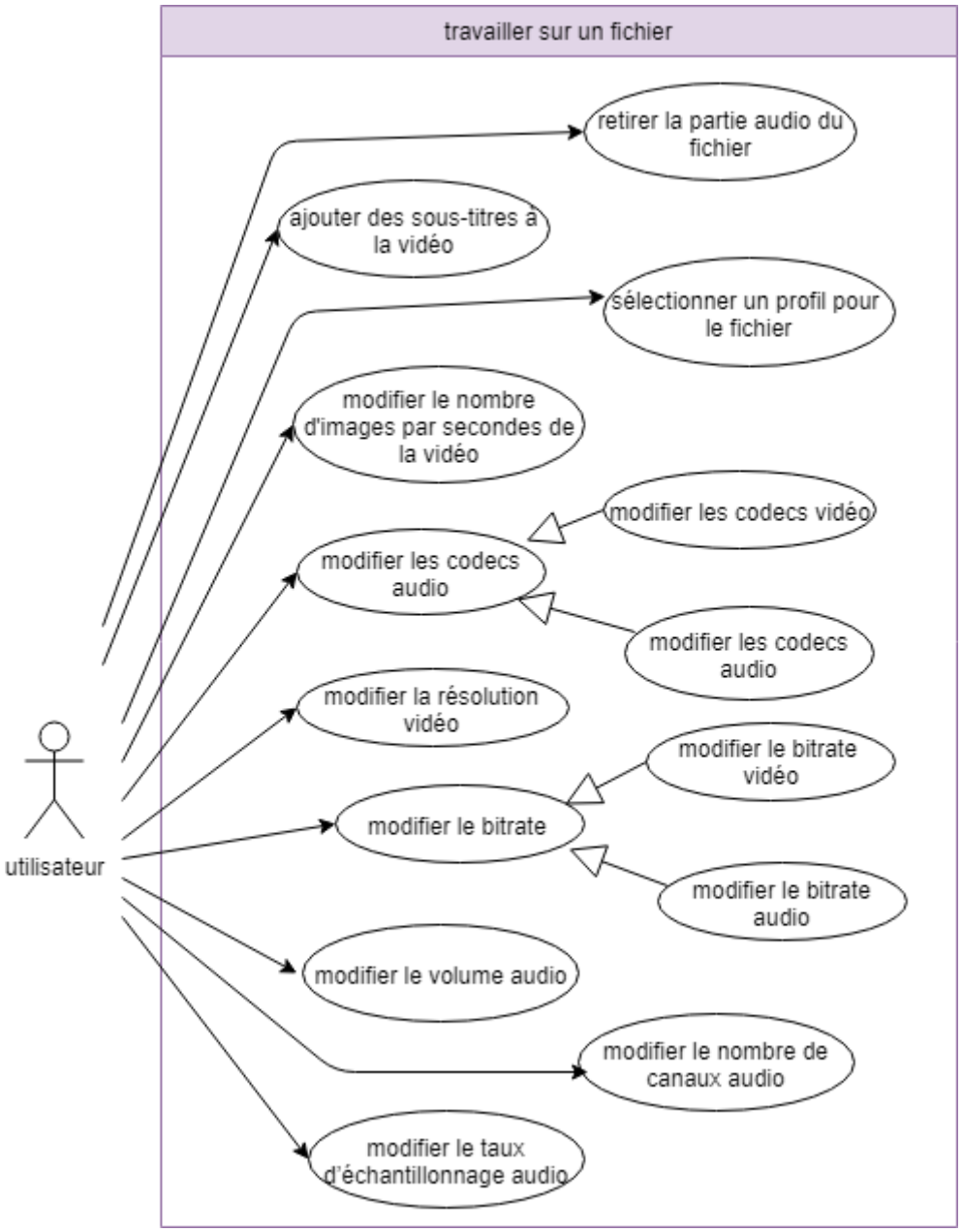
3.2.2) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LES CHRONOLOGIES



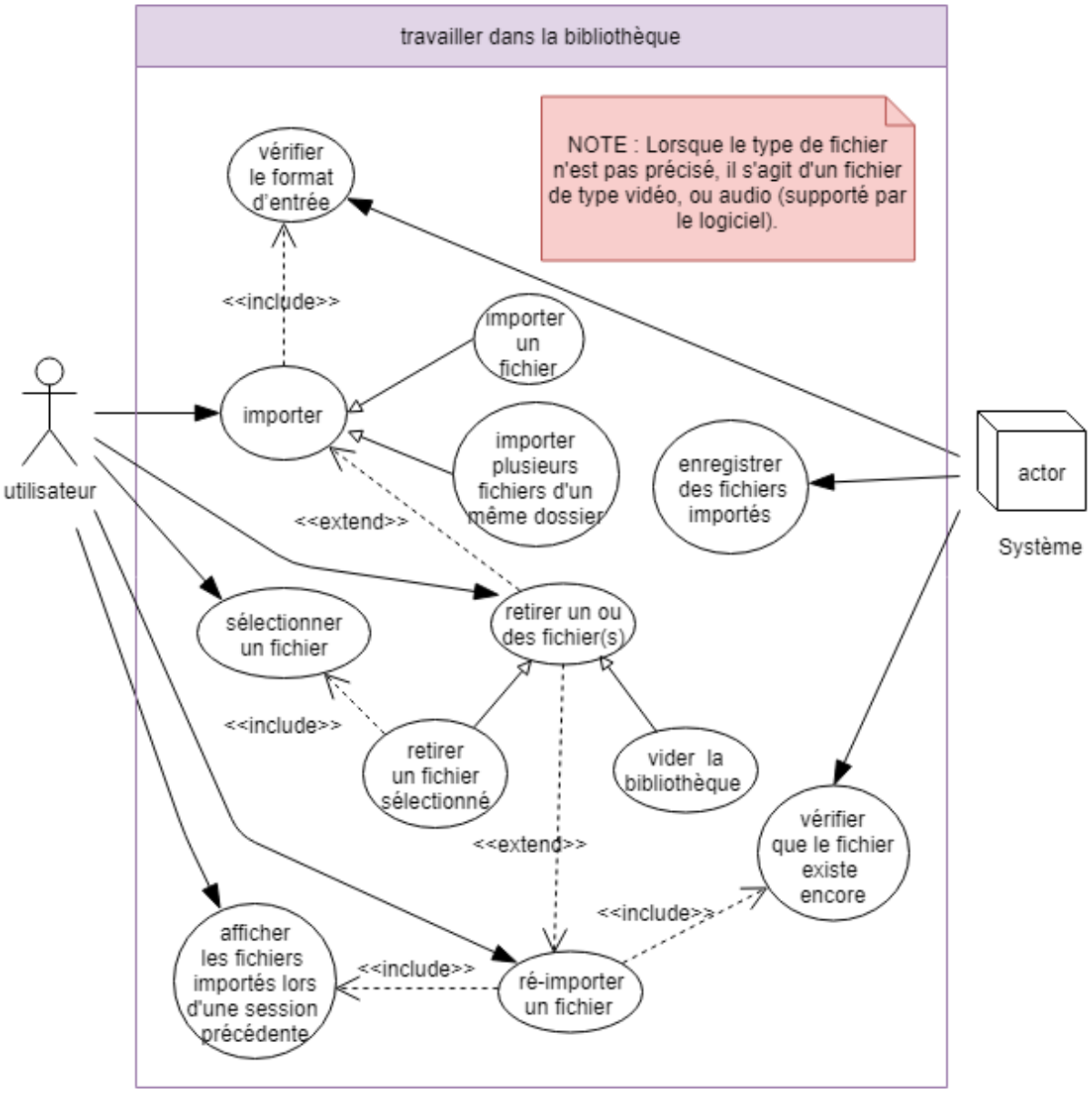
3.2) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LE MODE CONVERSION



3.3.1) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER SUR UN FICHIER



3.3.2) DIAGRAMME DE CAS D'UTILISATION TRAVAILLER DANS LA BIBLIOTHEQUE

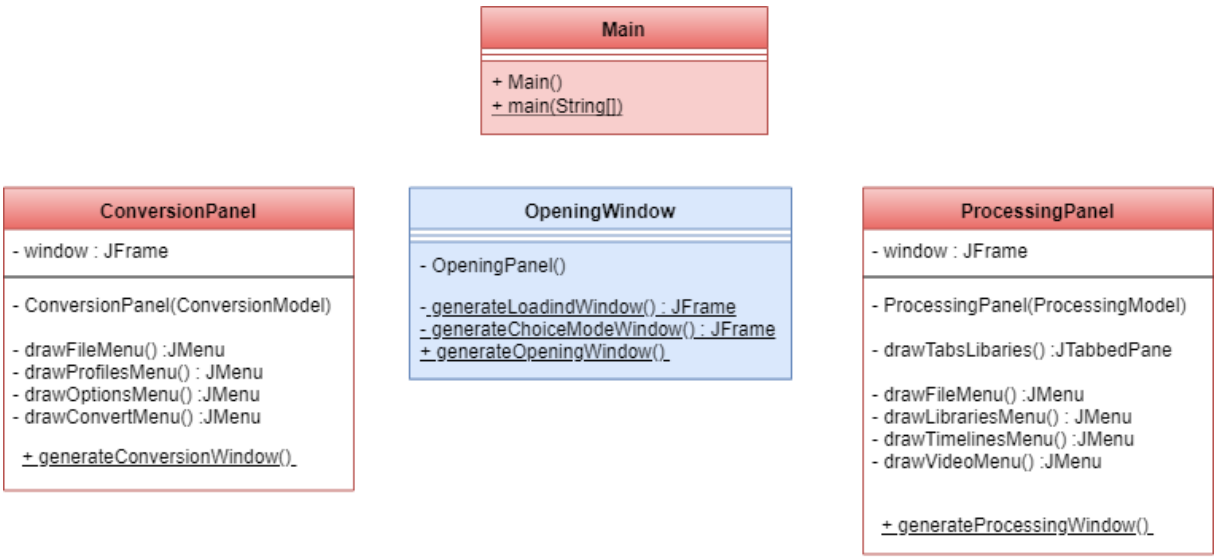


4) PROGRAMMATION : DIAGRAMMES DE CLASSES ET FONCTIONNEMENT

4.1) PROGRAMMES FEDERATEURS

Nous appelons programmes fédérateurs, les programmes qui « fédère » l'ensemble des programmes du logiciel. Ils construisent l'interface à partir de l'ensemble des programmes implémentés dans le logiciel, et ils réalisent le lien entre les objets.

4.1.1) DIAGRAMME DE CLASSES



4.1.2) DESCRIPTION DU FONCTIONNEMENT

Tout d'abord nous avons la classe main qui se contente de contenir une seule ligne dans sa méthode main. Cette ligne la voici :« **OpeningWindow.generateOpeningWindow() ;** ». Cette ligne génère le lancement de tout le logiciel.

La méthode **generateOpeningWindow()** appelée, appelle à son tour la méthode interne (= déclarée en private) **generateLoadingWindow() : JFrame**, qui génère une fenêtre de présentation du logiciel. La méthode a retourné l'objet JFrame qui génère la fenêtre de présentation, et au bout d'un certain temps la méthode **generateOpeningWindow()** appelle **JFrame.dispose()** pour fermer la fenêtre de présentation.

Puis ensuite **generateOpeningWindow()** appelle la méthode interne **generateChoiceModeWindow() : JFrame**, qui génère la fenêtre de choix du mode. Une fois le mode choisi, et selon celui qui est choisi, c'est soit **ConversionPanel.generateConversionWindow()** qui est appelée soit **ProcessingPanel.generateProcessingWindow**() qui est appelée après que la fenêtre du choix du mode soit fermée. La fenêtre du mode choisi est alors générée.

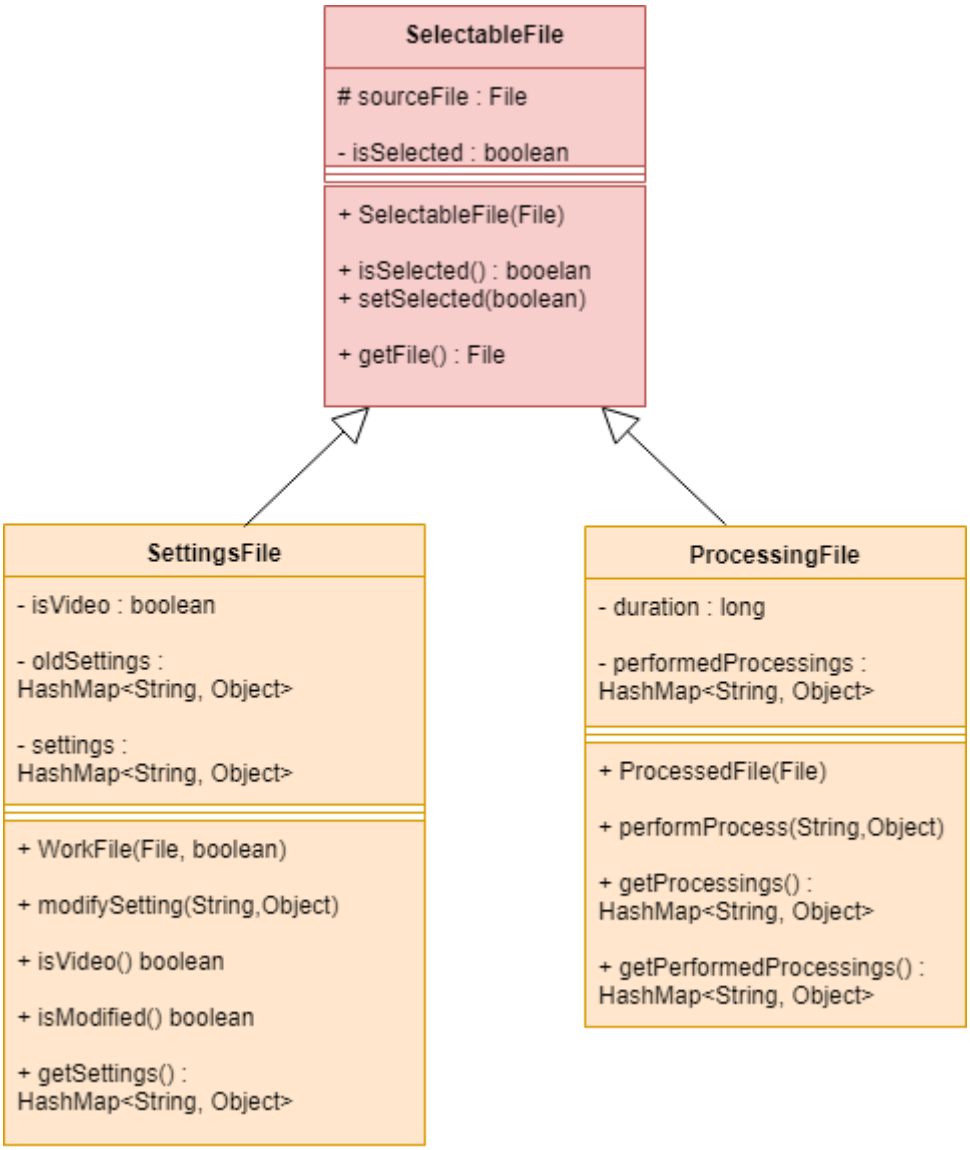
On peut remarquer que les constructeurs des classes ProcessingPanel et ConversionPanel sont privés, il est impossible d'instancier de l'extérieur des classes un Objet à partir de ces classes. Ces classes sont fédératrices elles instancient les modèles, les vues, les contrôleurs, les menus. Elles lient

les vues (Observers) au modèle (Observable) et les contrôleurs aux vues. Elles construisent les menus dans des méthodes internes et attribut des actions aux menus par le biais d'auditeurs anonymes etc.

4.2) PACKAGE files

On peut remarquer l'emploi d'un **PATRON DE CONCEPTION STRATEGIE**. Ce package files contient les objets qui représenteront nos fichiers dans le logiciel.

4.2.1) DIAGRAMME DE CLASSES



4.2.2) DESCRIPTION DU FONCTIONNEMENT

SelectedFile est la superclasse du package. L'idée est **qu'on a besoin de savoir lorsqu'un fichier est sélectionné** dans le logiciel, le booléen **isSelected** sert à cela.

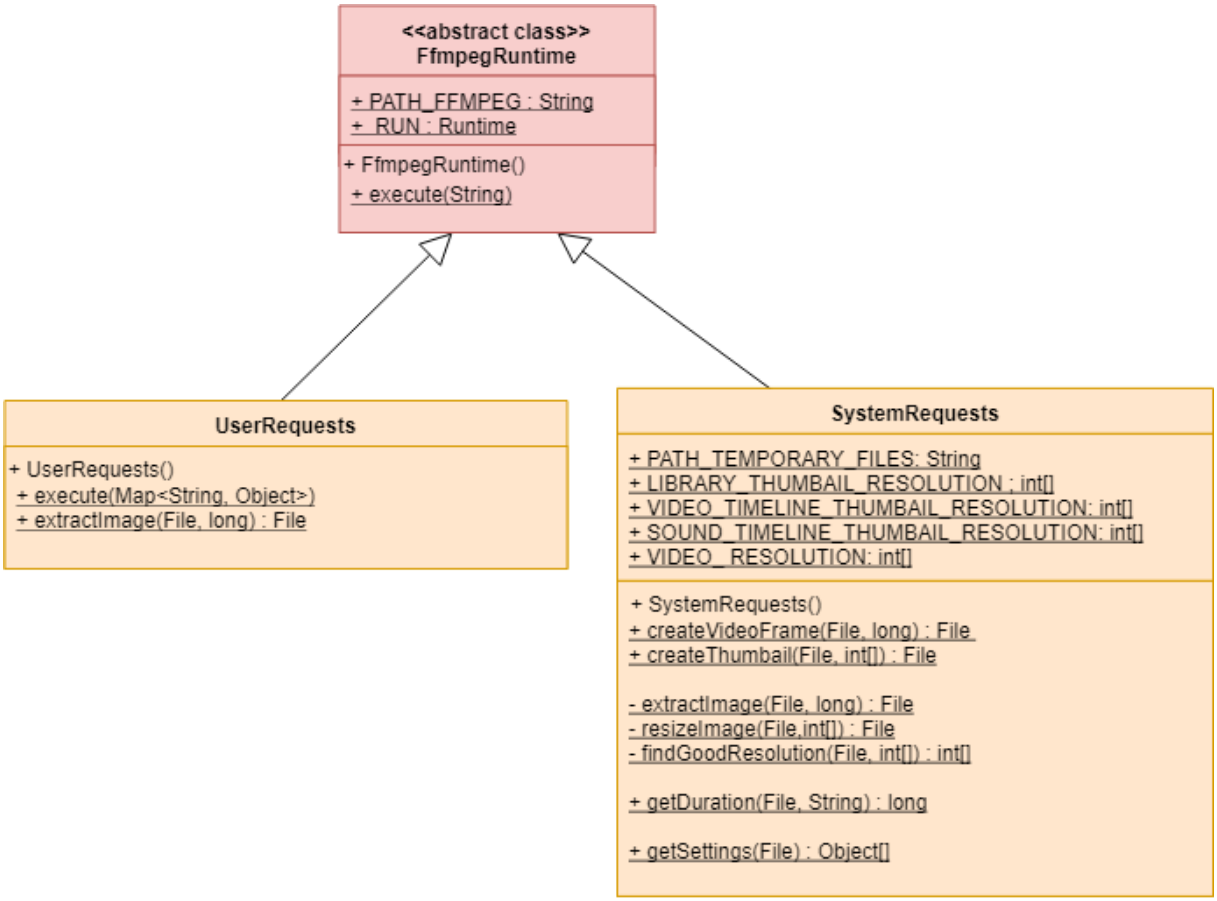
ProcessingFile est une sous-classe, qui sert dans la fenêtre de traitement. L'attribut **duration informe de la durée en secondes** du fichier (ce qui explique le choix du type **long** pour les entiers avec beaucoup de chiffres), et l'attribut performedProcessings sert à retenir **les traitements effectués** sur le fichier.

SettingsFile est une sous-classe, qui sert dans la fenêtre de conversion. Les attributs oldSettings et settings servent respectivement à retenir **les paramètres anciens de la vidéo avant modification, et les paramètres actuels de la vidéo**.

4.3) PACKAGE ffmpeg_tools

Les classes de ce package ont pour but réaliser l'interfaçage de FFmpeg dans notre logiciel. Il y a ici l'utilisation d'un **PATRON DE CONCEPTION STRATEGIE**.

4.3.1) DIAGRAMME DE CLASSES



4.3.2) DESCRIPTION DU FONCTIONNEMENT

FfmpegRuntime est la superclasse, et implémente une méthode execute(String) qui reçoit une requête FFmpeg en paramètre et l'exécute avec un **objet Runtime et la méthode Runtime.exec(String).** L'objet Runtime utilisé est déclaré en constante de classe publique avec le nom **RUN**. Cette classe est déclarée en abstract car on ne veut pas d'instances de cette classe. La constante de classe publique **PATH_FFMPEG** est un String qui **contient le chemin relatif vers l'emplacement du répertoire bin de FFmpeg** dans le répertoire ffmpeg du logiciel.

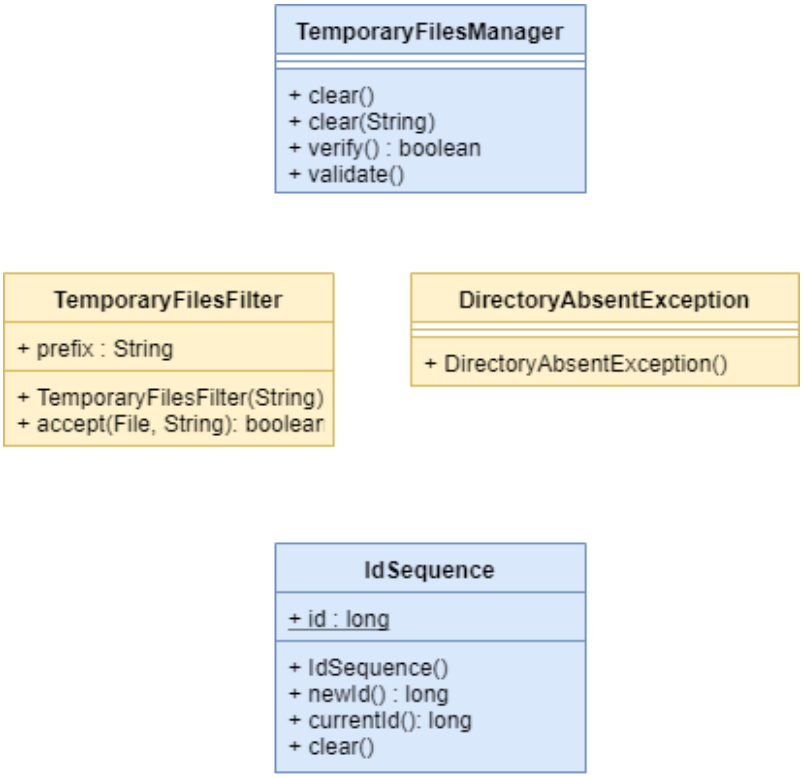
La classe UserRequests est une sous-classe, qui **exécute les requêtes FFmpeg correspondant aux opérations réalisées par l'utilisateur sur des fichiers**. Elle est appelée dans les méthodes ProcessingModel.export() et dans ConversionModel.convert().

La classe SystemRequests est une sous-classe, qui **exécute les requêtes FFmpeg correspondant aux opérations du système**, du genre extraire la première image d'une vidéo, puis la redimensionner pour créer une miniature de cette vidéo dans une bibliothèque ; ou alors pour connaitre la durée d'une vidéo, ou les informations générales d'une vidéo. **On utilise entre autres un objet Process appliqué sur un objet Runtime pour récupérer les réponses de FFmpeg à ce genre de requêtes**. La constante de classe publique **PATH_TEMPORARY_FILES** est un string qui **contient le chemin relatif vers le répertoire où on stocke les fichiers temporaires**, par exemple les miniatures pour les bibliothèques, ou encore les images extraites pour afficher la vidéo.

4.4) PACKAGE tools

Ce package contient les classes nécessaires pour la gestion des fichiers temporaires.

4.4.1) DIAGRAMME DE CLASSES



4.4.2) DESCRIPTION DU FONCTIONNEMENT

TemporaryFilesManager permet de vérifier et de recréer le répertoire des fichiers temporaires si nécessaire. Il permet également de **vider** le répertoire ou de **supprimer seulement certains fichiers** temporaires.

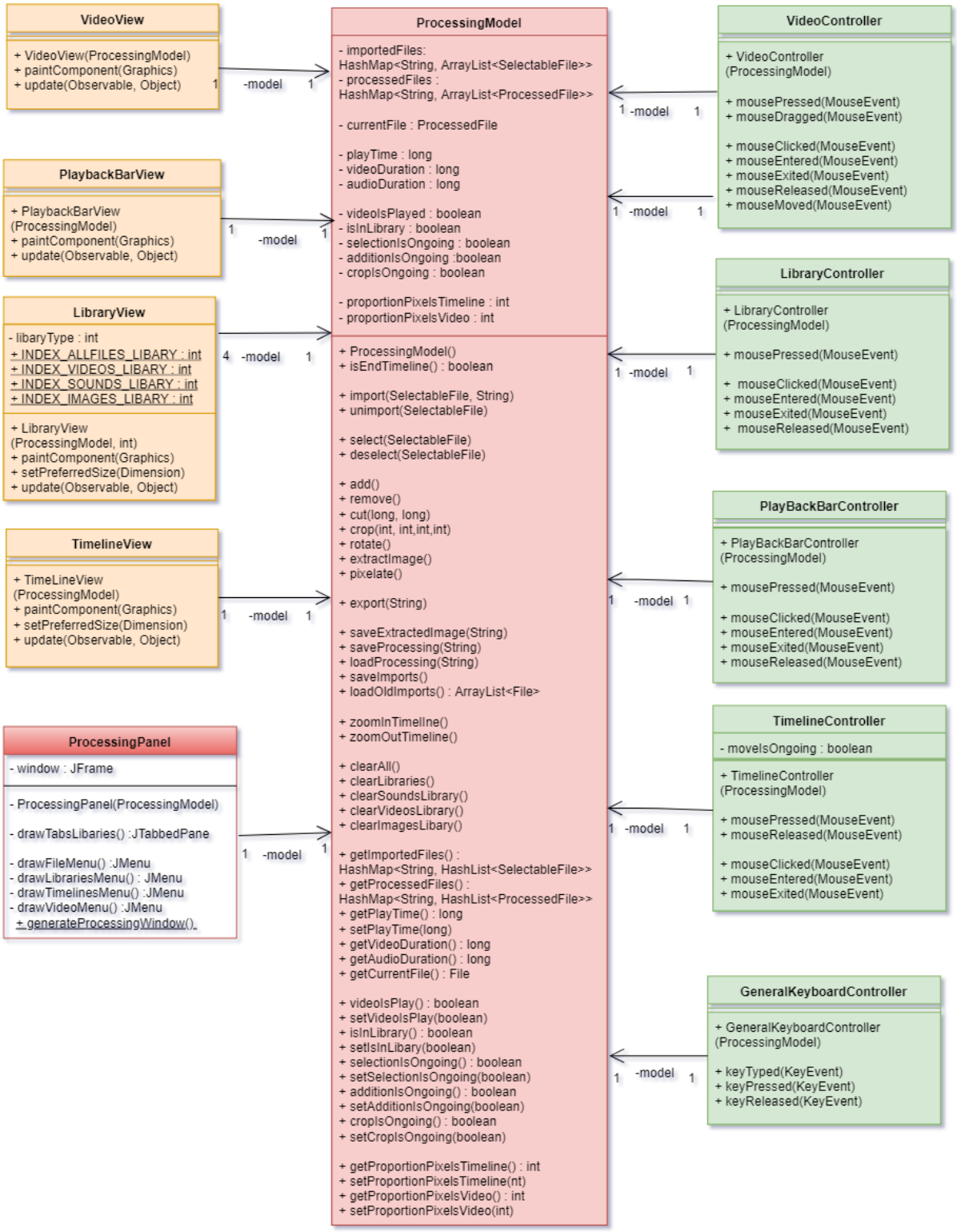
DirectoryAbsentException fait planter le logiciel s'il le répertoire des fichiers temporaires n'existe plus et que le système n'arrive pas créer un nouveau répertoire.

TemporaryFilesFilter implémente l'interface FilenameFilter qui permet de **trier des fichiers selon nos critères**. Cette classe sert à filtrer les fichiers existants trouvés par la méthode File.listFile(FilenameFilter).

Enfin la classe **IdSequence** permet de **générer des numéros en séries**, afin qu'on ne manque pas d'id pour créer des fichiers, et surtout de **s'assurer d'utiliser des noms de fichier pas déjà utilisé.** On utilise un **long** pour l'id, au cas où on crée beaucoup de fichiers temporaires au cours d'une même session de travail.

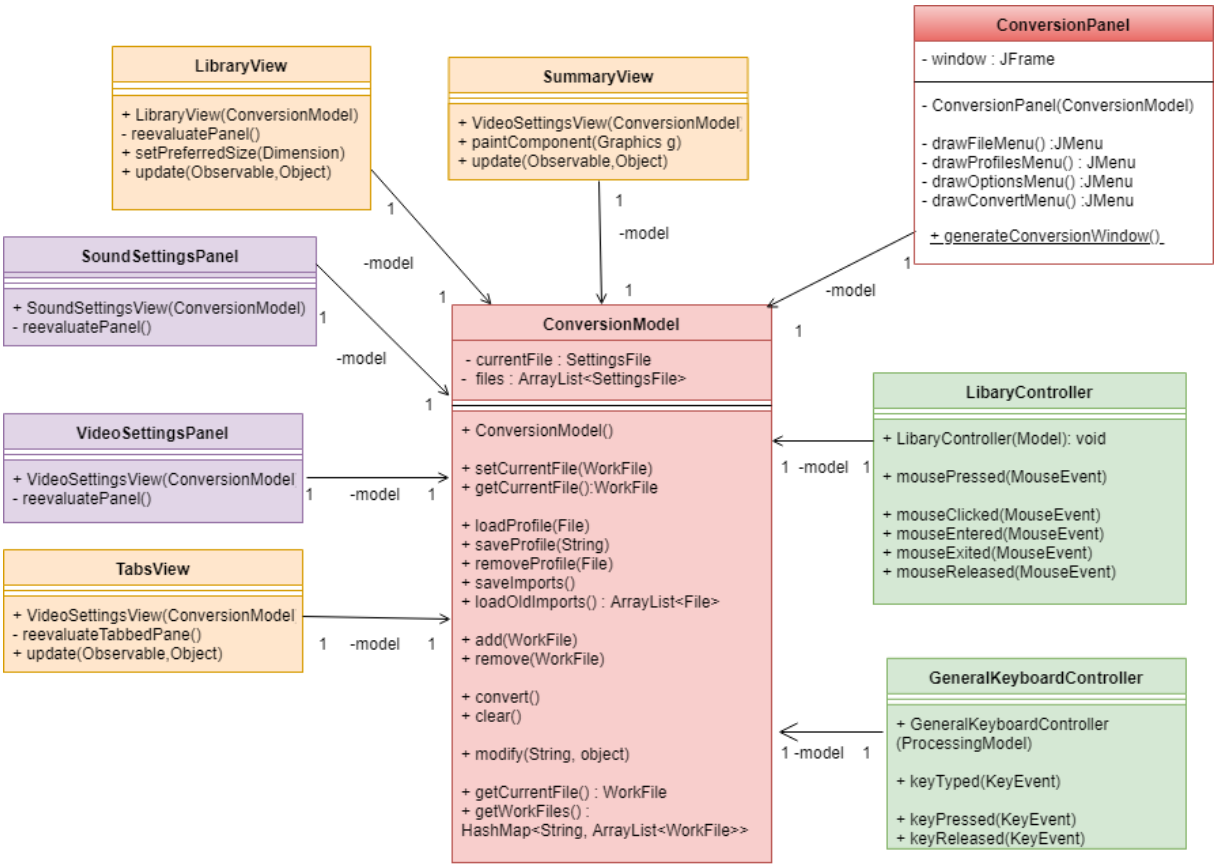
4.5) PACKAGE gui_processing

Ce package mets un en place un **PATRON DE CONCEPTION MVC** pour la fenêtre de traitement.



4.6) PACKAGE gui_conversion

Ce package mets un en place un **PATRON DE CONCEPTION MVC** pour la fenêtre de conversion.



5) CONCLUSION

Pour conclure cette étude préalable a permis d'entrer dans le **concret** de l'étude du logiciel. Désormais **nous nous rapprochons un peu plus de la solution**. Pour être franc, **quelques diagrammes de séquences n'auraient pas été de trop,** mais notre équipe a été absorbée par la réalisation des diagrammes de classes.

Les **risques** qui planent au-dessus de notre projet sont :

(1) Le risque que le logiciel soit **ralenti légèrement par l'interfaçage entre le logiciel et ffmpeg**. Risque qui ne se serait pas présenté si on avait choisi de développer dans le langage des bibliothèques de FFmpeg.
(2) Le risque que nous **soyons obligés d'utiliser des Threads pour faire tourner en simultané certaines tâches**, et donc que nous soyons obligés de modifier de manière conséquente nos diagrammes de classes.
(3) Le risque que JAVA soit **incapable d'actualiser convenablement les frames de la vidéo** (de manière à fournir une lecture fluide de la vidéo).