

Titlul proiectului este „Editor Video” și este realizat de Agavriloaei Marina 1409A, Iva Antonin 1409B și Stroici Andrei 1409A.

Aplicația propusă este un editor video. În cadrul proiectului am ales să folosim mai multe biblioteci. Bibliotecile folosite până la momentul actual și versiunile lor (unde este cazul) sunt prezentate în lista următoare:

- PyQt6-Qt6 versiunea 6.10.0
- PyQt6\_sip versiunea 13.10.2
- PySide6 versiunea 6.10.0
- PySide6\_Addons versiunea 6.10.0
- PySide6\_Essentials versiunea 6.10.0
- shiboken6 versiunea 6.10.0
- ffmpeg

Pentru realizarea acesteia am ales să folosim paradigma orientată obiect. În Figură 1 Diagrama de clase este diagrama de clase propusă.

În partea stângă a diagramei se pot observa clasele VideoPreview, Toolbar, TimeLineAndTracks, MediaTabs, EnhancementsTabs și VideoEditorUI. Acestea sunt clasele care modelează interfața grafică și elementele din aceasta. În Figură 2 Schiță interfața grafică, Figură 3 Schiță interfața grafică și Figură 4 Schiță interfața grafică se ilustrează modul cum va fi structurată interfața grafică.

În această secțiune sunt descrise clasele din partea stângă a diagramei de clase (Figura 1), care modelează componentele interfeței grafice ale aplicației de editare video. Toate aceste clase moștenesc QWidget și reprezintă elemente UI personalizate, având roluri bine delimitate în cadrul aplicației.

Clasa VideoPreview (moștenește QWidget) este responsabilă pentru previzualizarea conținutului video pe parcursul editării. Aceasta oferă un set de funcționalități de control al redării, afișate în ordine de la stânga la dreapta:

- salt la începutul videoclipului
- redarea inversă a videoclipului
- navigarea la cadrul anterior
- redare/Pauză
- navigarea la cadrul următor
- creșterea/scăderea vitezei de redare
- salt la finalul videoclipului

Această componentă este utilizată atât pentru controlul redării, cât și pentru sincronizarea vizuală cu timeline-ul.

Clasa Toolbar, derivată din QWidget, reprezintă meniul principal poziționat în partea superioară a interfeței. Aceasta pune la dispoziția utilizatorului următoarele acțiuni:

- adăugarea unui fișier
- deschiderea unui director
- salvarea stării proiectului
- export în format video
- anularea ultimei acțiuni (Undo)
- refacerea unei acțiuni anulate (Redo)

Toolbar-ul servește drept punct central pentru operațiile frecvent utilizate în procesul de editare.

Clasa TimeLineAndTracks modelează zona timeline-ului și a canalelor de editare, fiind un widget personalizat bazat pe QWidget. Aceasta gestionează structura secvențelor video și permite utilizatorului să opereze asupra canalelor. Funcționalitățile principale sunt:

- adăugarea de noi canale
- alinierea canalelor
- tăierea unui canal
- ultimul buton este replasare în timeline a unui fișier pe alte coordonate

Această componentă reprezintă elementul central în gestionarea structurii timeline-ului proiectului.

Clasa MediaTabs moștenește clasa QWidget și are rolul de a afișa fișierele deschise de utilizator.

Clasa EnhancementsTabs, derivată din QWidget, gestionează panoul de filtre, ajustări și transformări aplicabile videoclipului. Aceasta oferă acces la filtre, transformări și efecte.

Clasa VideoEditorUI, care moștenește clasa QWidget, este clasa care conține fereastra principală a aplicației. În interiorul acesteia se găsesc elementele din interfață descrise mai sus. Aceasta reprezintă punctul de intrare în aplicație (locul unde se află main-ul). Pentru a putea aplica operațiile de editare asupra videoclipului, care este salvat în memorie, aceasta conține o referință spre VideoEditor.

Clasa VideoEditorUI, care moștenește QWidget, reprezintă fereastra principală a aplicației și punctul de intrare în interfață. Aceasta conține și orchestrează toate elementele de UI descrise anterior: Toolbar, VideoPreview, TimeLineAndTracks, MediaTabs și EnhancementsTabs.

De asemenea, clasa deține o referință către componenta logică VideoEditor, responsabilă cu operațiile de procesare și transformare a videoclipului încărcat în memorie. Prin intermediul acestei relații, interfața poate solicita aplicarea efectelor, modificărilor sau generarea rezultatelor de editare.

Clasa VideoEditor este clasa care se va ocupa de gestionarea operațiilor aplicate asupra videoclipului. Aceasta va conține referințe la clasele abstracte care descriu operațiile care pot fi aplicate.

Clasa ImagePlayer, derivată din QObject, are rolul de a converti o imagine într-un flux video, astfel încât aceasta să poată fi integrată și redată pe un timeline.

Clasa TimelineTrackWidget este un widget personalizat (scris cu PySide6) care implementează timeline-ul unui editor video. Este o componentă avansată care se ocupă cu:

- afișarea unei rigle temporale (secunde, minute)
- afișarea clipurilor (dreptunghiuri colorate)
- gestionarea playhead-ului (linia roșie de poziție)
- selecția clipurilor, drag-ul playhead-ului, coordonate timp – pixeli
- scalarea automată a timeline-ului
- emiterea semnalului seek\_request(int) când utilizatorul mută playhead-ul

Clasa VideoPreviewButtons este un widget care conține toate butoanele de control ale playerului video folosit în VideoPreview. Botanele disponibile sunt:

- to\_start\_button - merge la începutul videoclipului
- playback\_button - pornește redarea înapoi (reverse)

- prev\_frame\_button - sare la cadrul anterior
- play\_pause\_button - play / Pauză pentru videoclipul curent
- next\_frame\_button - sare la cadrul următor
- faster\_button - crește viteza de redare (fast-forward)
- to\_end\_button - merge la sfârșitul videoclipului

Clasa VideoTabContent este componenta completă care afișează conținutul unui tab din VideoPreview. Este partea care se ocupă de încărcarea, afișarea și controlează videouri, imagini și audio, oferind toate funcțiile de player.

Clasa ClickableListWidget este o extensie simplă a QListWidget care adaugă un comportament suplimentar la click: dacă utilizatorul face click într-o zonă goală (fără item), lista își șterge selecția și focusul. Practic, permite deselectarea itemilor printr-un click pe fundalul listei, lucru care nu se întâmplă implicit în QListWidget.

Clasa History este clasa care menține lista de stări a videoclipului prin intermediul căreia se realizează operațiile de undo și redo.

BaseProcessor este o clasă abstractă care oferă mecanismul comun de procesare media folosind FFmpeg. Metoda protejată \_apply\_ffmpeg aplică un filtru video sau audio asupra fișierului asociat obiectului QTimeLine, generând automat un fișier temporar cu rezultatul procesării.

Clasa gestionează:

- citirea și validarea fișierului de intrare (input\_file)
- păstrarea fișierului original pentru a preveni suprascrierea accidentală
- construirea și executarea comenzii FFmpeg (-vf pentru video, -af pentru audio)
- ștergerea fișierelor intermediare
- actualizarea proprietății input\_file din QTimeLine cu fișierul procesat

Această clasă centralizează logica de aplicare a filtrelor, permițând ca utilizatorii specializați să specifice doar filtrul dorit, fără a gestiona direct detaliile tehnice ale interacțiunii cu FFmpeg.

În cele ce urmează vom prezenta clasele care modelează operațiile disponibile.

O altă clasă abstractă este Composition. Aceasta este o clasă care este o abstracțiune pentru operațiile de compoziție care vor fi disponibile. Aceasta conține o singură metodă abstractă, care va fi implementată de subclase. Această metodă primește ca argument un videoclip și întoarce videoclipul rezultat în urma aplicării operației de compoziție. Clasele care o moștenesc sunt:

- BlendVideos - care implementează o operație în care videoclipurile sunt combinate folosind un efect vizual de tip blend( amestecare), plasând videoclipurile pe layer-e diferite. Constructorul primește ca argument videoclipul peste care se aplică compoziția, un parametru alfa (transparența) și modul în care se face amestecarea.
- Chorus - simulează mai multe „voci” ale aceluiași audio, ușor decalate în timp și în frecvență, rezultând un sunet mai „larg”, mai „plin”, specific efectelor muzicale chorus. Constructorul primește ca argumente întârzierea exprimată în milisecunde, amplitudinea modulației întârzierii și efectul aplicat pe întârziere.
- Delay – are un efect de întârziere a semnalului audio din videoclip, cu un timp specificat în milisecunde și un efect. Cele 2 fac parte din constructorul cu argumente.
- Echo - generează nu doar o singură întârziere, ci o serie de ecouri succesive, fiecare mai slab decât precedentul, controlate prin decay. Constructorul primește ca argument întârzierea, decăderea și nivelul.

- Overlay - suprapune al doilea videoclip peste primul, la o poziție specificată (x, y) și cu o transparență opțională (alpha). Constructorul primește ca argument videoclipul care se va suprapune, alpha – parametrul de opacitate și poziția.

În categoria filtrelor avem o clasă abstractă Filters care moștenește BaseProcessor și oferă o abstracțiune pentru clasele care realizează efectiv filtrarea claselor. Clasa Filters are o metodă abstractă care aplică filtrul peste videoclip, care are ca parametru un videoclip. Clasele care moștenesc această abstracțiune sunt următoarele:

- BlurFilter – care realizează efectul de estompare. Aceasta are ca parametru pentru constructor raza filtrului de estompare.
- EdgeDetect – aceasta este un filtru care realizează detecția marginilor. Constructorul acestei clase primește ca argument metoda de detecție și un prag necesar detecției.
- KernelFiltering – este un filtru care îmi determină cum se modifică un anumit pixel. Constructorul acceptă ca argument o matrice de valori care modelează aplicarea filtrului de convoluție și o valoare logică care indică dacă se face o normalizare sau nu asupra valorilor obținute.
- NoiseReduction – acesta este un filtru pentru reducerea zgomotului din imagine. Constructorul primește ca argumente tăria filtrului și metoda folosită pentru reducerea zgomotului.
- Tempo – acesta este un filtru audio care are rolul de a modifica viteza unei piste audio. Constructorul acestei clase primește ca argument factorul care indică modificarea de viteză.
- Volume – acest filtru amplifică volumul sunetului după parametrul care este dat în constructor, semnificând câștigul în decibeli.

Pentru a putea aplica operația de desenare de text peste videoclip am creat clasa abstractă TextOperation care moștenește BaseProcessor. Această clasă abstractă adaugă o metodă applyText care primește ca argument videoclipul peste care se va aplica operația de desenare de text și întoarce videoclipul obținut după aplicarea acestei operații. TextOperation este implementată de:

- DrawText – această clasă desenează un text peste videoclip. Constructorul primește ca argument textul care va fi scris, poziția unde va fi plasat textul, fontul textului, dimensiunea și culoarea specificată sub formă de șir de caractere, folosind numele culorilor din limba engleză.

Clasa TimelineOperation este o clasă care definește în mod abstract o operație pe timeline-ul videoclipului. Această clasă are o metodă abstractă applyOperation care primește ca argument un videoclip și rezultatul returnat este videoclipul obținut în urma aplicării operației. Aceasta e implementată de:

- ConcatVideo – care realizează concatenarea videoclipului. Constructorul primește ca argument videoclipul peste care se aplică operația.
- CutVideo - realizează tăierea unei porțiuni din videoclip. Constructorul primește ca argumente momentul în care începe tăietura și momentul în care se termină.

Clasa Timing moștenește BaseProcessor și este un tip general de procesor temporal care definește operații ce manipulează o linie temporală. Aceasta are o metodă abstractă applyOperation care primește ca argument un videoclip și întoarce videoclipul obținut după aplicarea operației. Clasele care o implementează sunt:

- FadeInOut - este o clasă care realizează o apariție sau dispariție treptată. Constructorul acestei clase primește ca argumente durata pentru apariție și tipul.

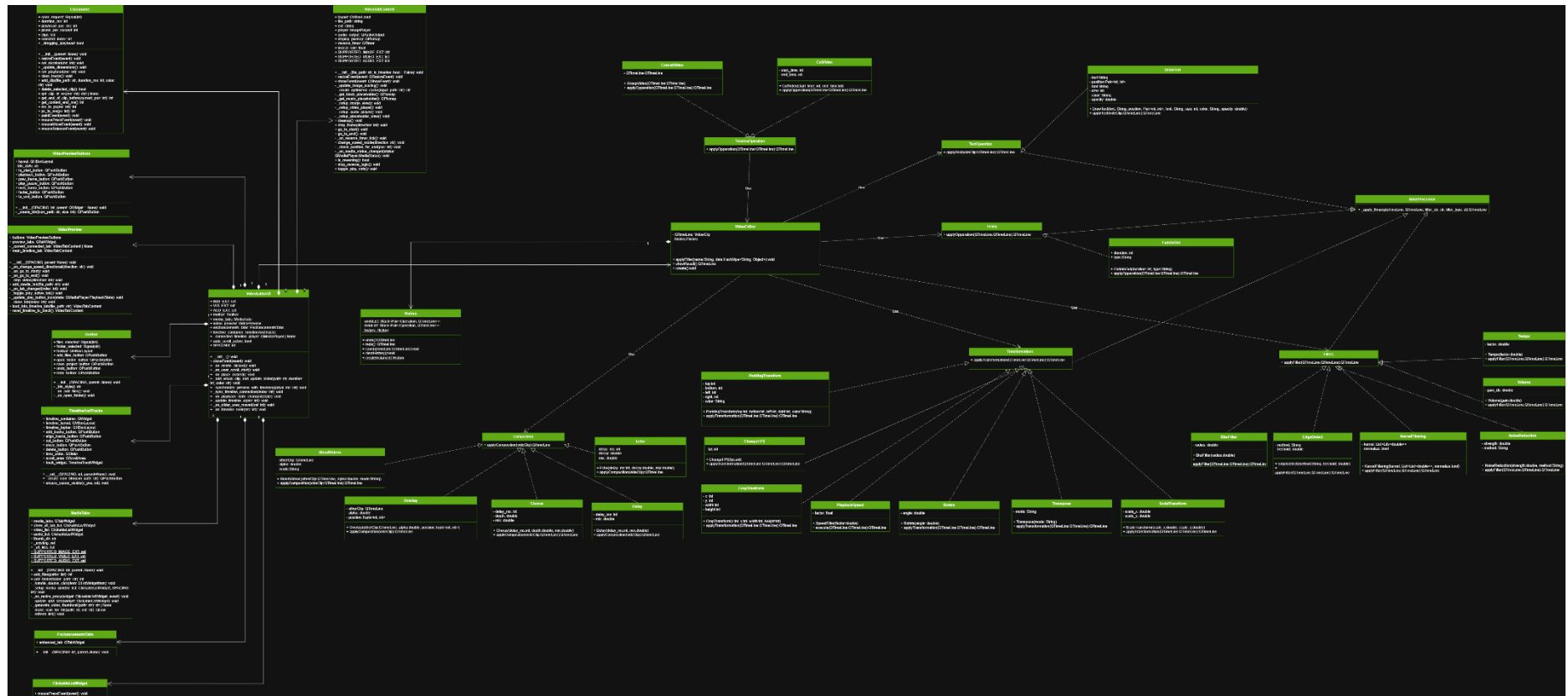
Ultima clasă care moștenește BaseProcessor este Tranfomrmations. Aceasta este o clasă abstractă care modelează transformările care se pot aplica asupra videoclipului. Transfromations introduce o metodă abstractă care aplică transformarea asupra videoclipului. Această metodă primește ca argument videoclipul și va întoarce videoclipul după aplicarea transformării. Tranfomations e implementată de:

- ChangeFPS - această clasă modifică numărul de cadre pe secundă ale videoclipului. Constructorul primește ca argument noul număr de cadre.
- CropTranform – această clasă realizează o decupare în imagine. Constructorul clasei primește coordonatele punctului din stânga jos și lățimea, respectiv înălțimea dreptunghiului care realizează decuparea.
- PaddingTransform – este o transformare care adaugă margini de o culoare la videoclip. Constructorul primește ca argumente marginile de sus, jos, stânga și dreapta și culoarea padding-ului.
- PlaybackSpeed - este o transformare care modifică viteza de redare a videoclipului. Constructorul primește ca argument factorul de modificare a vitezei de redare.
- Rotate – este o transformare care rotește videoclipul cu un unghi specificat în parametrul constructorului.
- ScaleTransform – este o transformare care realizează o scalare după axa x sau axa y în funcție de valorile setate prin parametrii constructorului: factorul de scalare pe x și factorul de scalare pe y.
- Transpose - este o clasă de transformare video care permite: rotirea clipului (90° în sensul acelor de ceas, invers, cu flip etc.) sau oglindirea clipului pe orizontală sau verticală. Constructorul primește ca argument modul de transpunere.

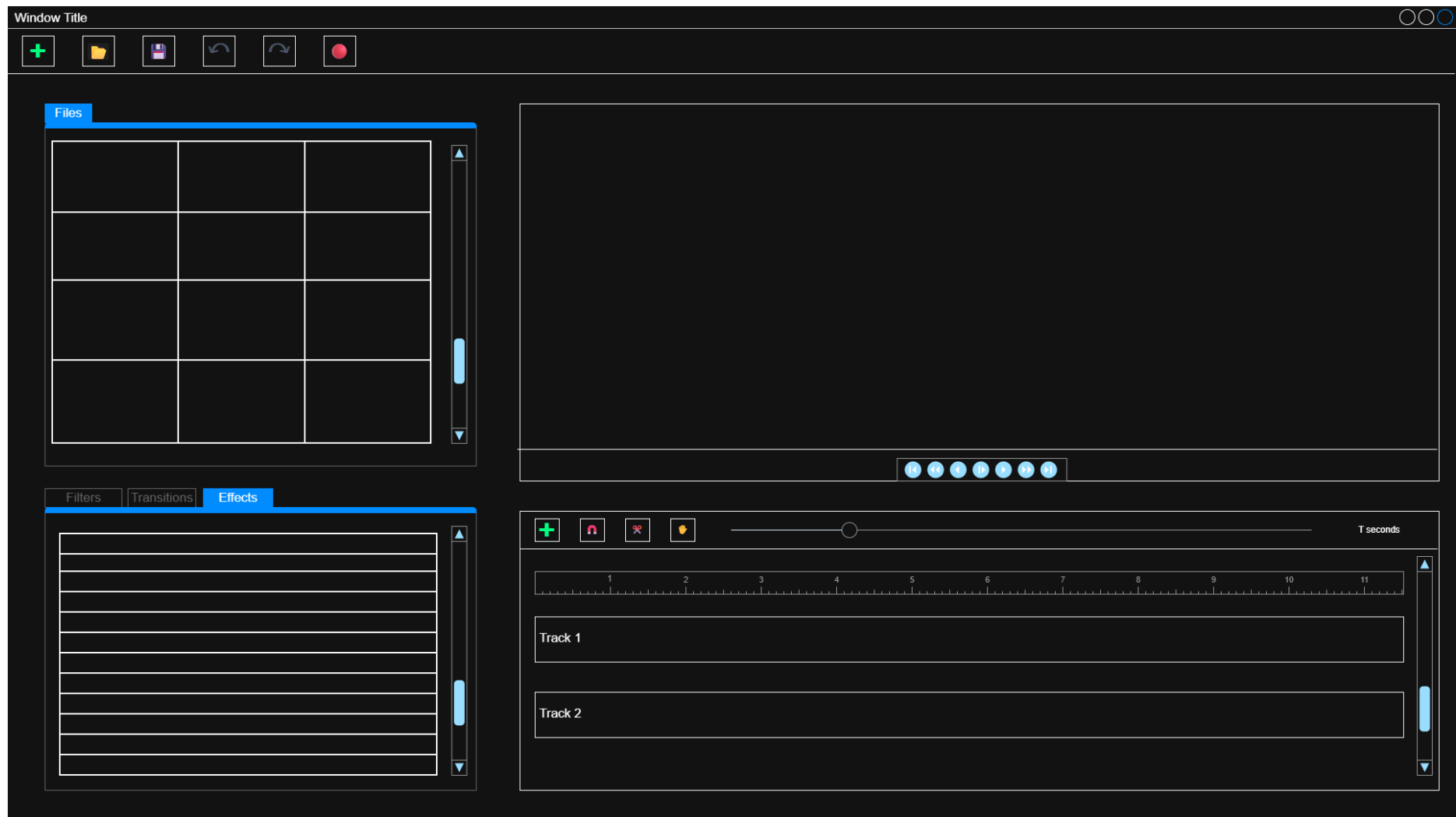
La momentul încărcării documentului de față sunt implementate următoarele funcționalități:

- Toate operațiile care se vor putea aplica pe videoclip, dar fără a fi legate de interfața grafică
- Încărcare de fișiere
- Ștergere fișiere
- Vizualizare fișiere cu dublu click.

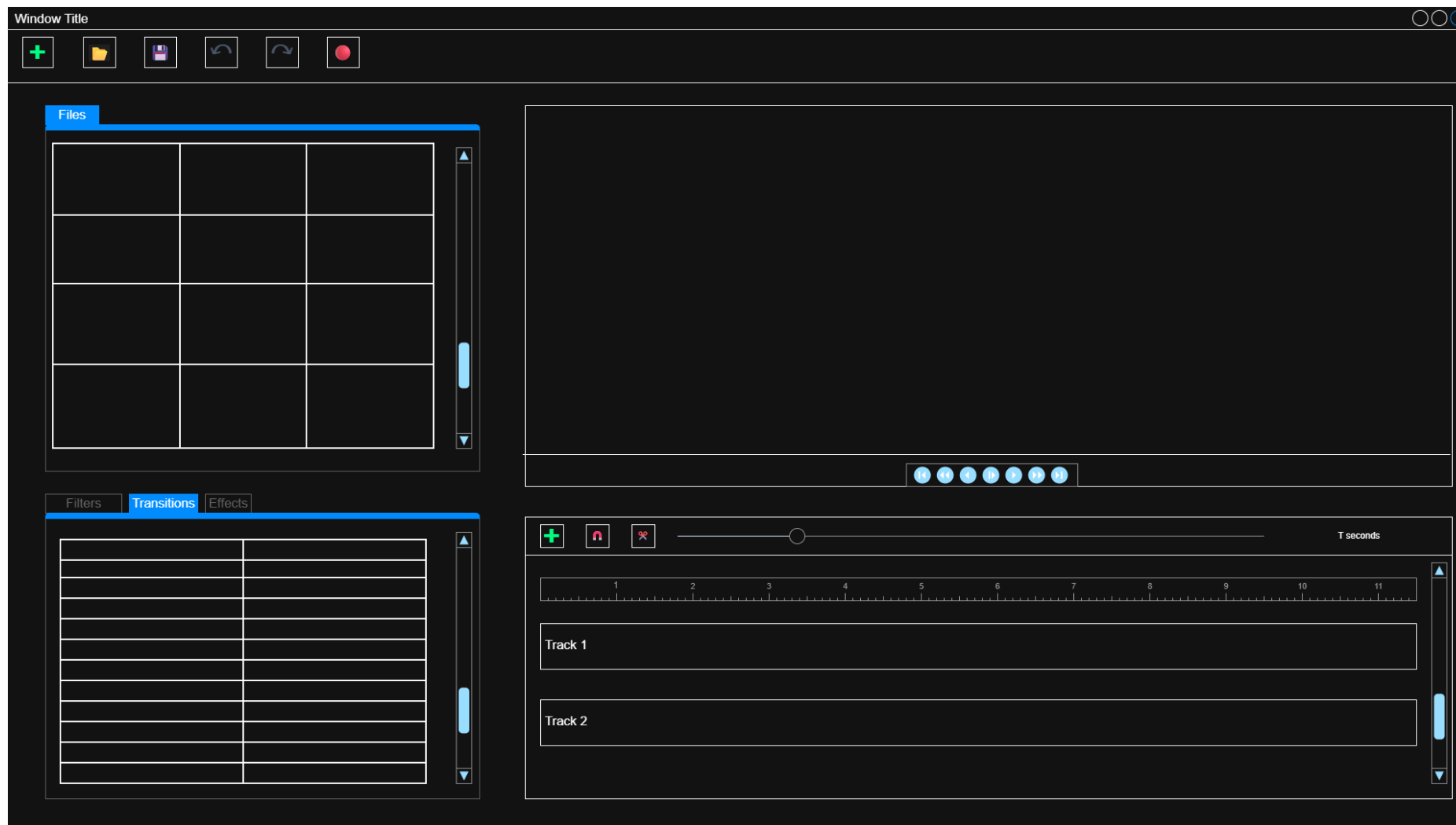
Pentru testarea aplicației am oferit la început bibliotecile necesare. Verificarea funcționalităților implementate presupune pornirea interfeței și în cadrul acesteia se pot verifica încărcarea fișierelor, ștergerea lor și vizualizarea fișierelor folosind dublu click. Pentru testarea operațiilor de editare video se poate folosi codul aflat în pachetul Python tests.



Figură 1 Diagrama de clase

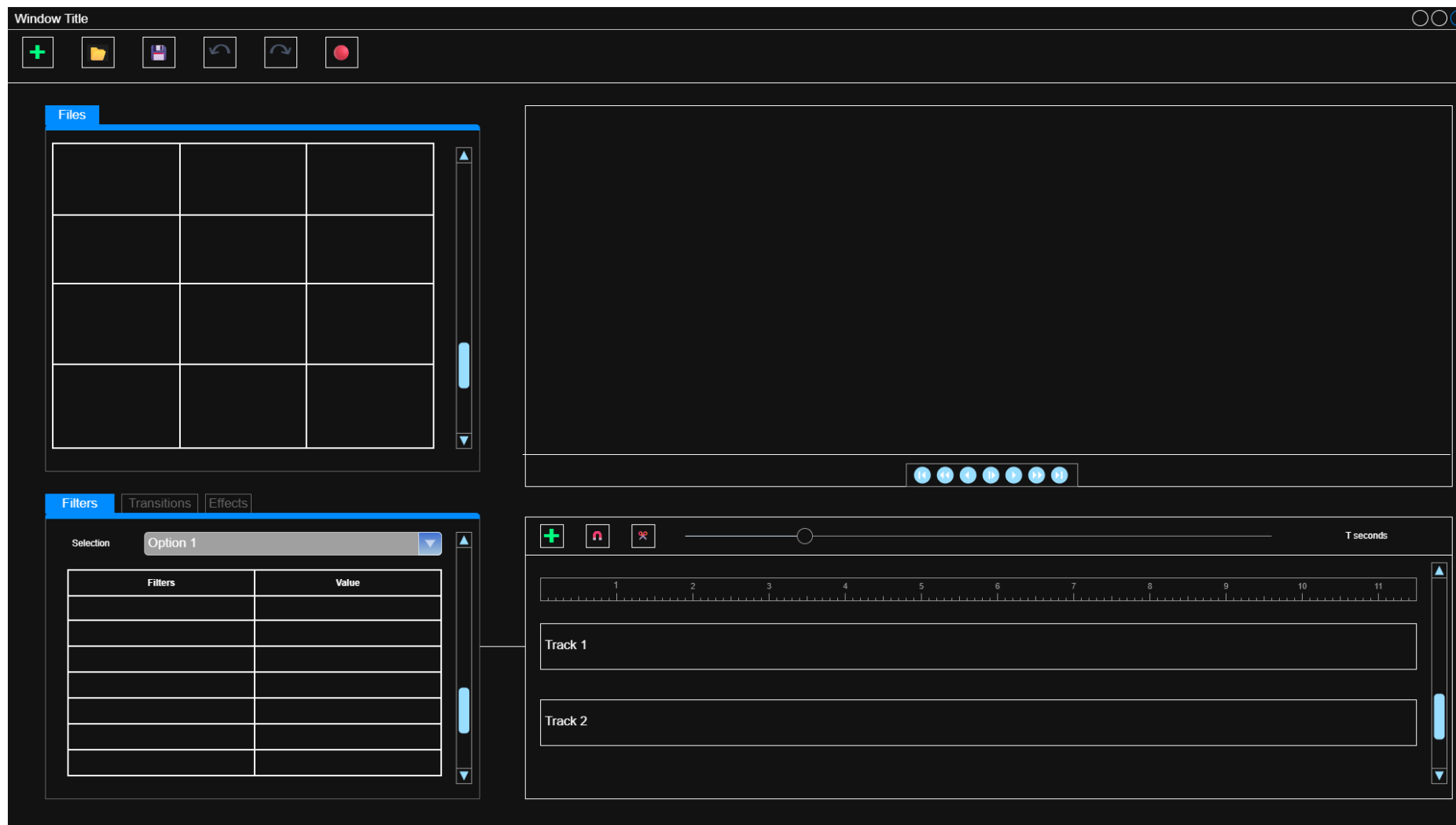


Figură 2 Schiță interfața grafică



Figură 3 Schiță interfața grafică





Figură 4 Schiță interfața grafică