**Project Research Document**

**RenCloud**

**X00202628 Vlad Pocris**

**Section 1**

My aim is to build a light-weight video editing software for windows users that would allow them to manipulate/create/produce content/media in a very easy and intuitive way. It would focus on a very strong UX/UI design/functionality that would make possible many features known in video editing, such as: crop; trim; merge; delete; concatenation and more. You could also double these functionalities if we count the features included, since the application will also make possible for user to manipulate audio which is a different thing therefore it is not only about video but audio at the same time.  
Who are the users?  
It could be anyone, maybe you need to make a compilation of good moments of your life using some recorded videos from many weekends, maybe you just want to crop out some seconds of the video, maybe you want to delete the audio from the video, or just to delete a specific portion of the audio, downgrade the video quality for efficient use of storage, in plain words unlimited use-cases.  
It could include from families to content creators, since it would be very intuitive and easy to work with while being light-weight which means minimum resource usage which as a result will make possible to use on any windows-OS device that holds weak hardware while most importantly holding the functionalities of some of the most known video editing applications.

If we were to talk in great details about the functionality this would be described in my UCLA for certain use-cases, there I will explain how data is manipulated and how media files are used in detail with additional algorithms explained.

The approach for this project is to provide lots of functionality with less user interactions, by that I mean using the way modern video editing software are created I can achieve 50% of the features just by making a dynamic and easy to understand way of handling them:  
**(a)Segmentation –** say we upload a video file, it would represent a segment on the timeline that could be split into many segments e.g. a video of 1 minute split at 20 seconds would result in 2 segments on the timeline, one would be 20 seconds and the second one would be 40 seconds.

**(b)Deletion –** continuing from last example, if we were to delete the segment 1 (20 seconds) then on the timeline would be only 1 segment which would be the segment 2 (40 seconds) this feature also requires the **selection** of segments feature.

**(c)Moving the segment –** alternative continuation of first example, say we want the video to play the segment 2(40seconds) from start and then the segment 1(20 seconds) at the end, the user would be able to drag the segment to the necessary position which will then change/swap their positions accordingly also considering there are images (small image previews from video clip) by that I want to say it would be complex to build it, since there are many use-cases such as moving with neighbour segment (easy to achieve), moving with segments far e.g. if we have 3 segments **a**,**b** and **c** then if **c** is dragged and passes the middle point of the segment **a** the swap should occur which must recalculate the position of the whole timeline, also different logic applies for swapping with a segment from left and from right.

**Section 2**

(1) CapCut. ([link](https://www.capcut.com/tools/online-video-editor))

(2) Filmora. ([link](https://filmora.wondershare.net/tools/short-video-maker.html?gad_source=1&gclid=Cj0KCQiAwOe8BhCCARIsAGKeD56yLamydk3TIZ3ikzDbCxLDhaR2dyTT4em4R041M9rMDCYsGCrOsOgaAmkYEALw_wcB))

(3) Clipchamp. ([link](https://clipchamp.com/en/))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Application | Similarities | User Interface & Experience | Advanced editing features | Pricing Model | AI Capabilities |
| CapCut | Cloud-based video editing, AI-assisted tools, easy drag-and-drop functionality, however cloud-based is very limited. | Simple and beginner-friendly, mobile-first design. | Basic transitions, filters, and effects; lacks advanced layering. | Free with limited features, Pro version for premium tools aka. More effects & more transitions. | AI-driven background removal, auto-captioning, and face tracking. |
| Filmora | Multi-platform video editing, pre-built templates, AI-powered editing. | Intuitive timeline, appealing UI, suitable for beginners and intermediates. | Advanced effects, motion tracking, keyframing. | One-time purchase & subscription based. | AI noise reduction, smart cutout, speech-to-text. |
| Clipchamp | Web-based video editing, template-based workflow, AI-enhanced features. | Web app with drag-and-drop simplicity, integrated with Microsoft ecosystem. (Most of windows have this application preinstalled on their OS such as Windows 11) | Basic trimming, transitions, audio editing. | Free version with watermark, paid tiers for full access. | AI-powered text-to-speech, auto-captions, branding tools. |

**Section 3**

The framework for the project would be WFA – Windows forms application with C# .Net 4.8 Framework with platform target for any CPU for Windows OS machines.

After carefully researching the editing industry most of the applications if not every applications use [ffmpeg](https://www.ffmpeg.org/about.html) library(multimedia framework) for video/audio manipulations, therefore I would be using this library in order to render to the final product.

Another library would be the [VLC](https://www.videolan.org/vlc/libvlc.html) library as a media player, initially windows media player has been one of the options, however it wouldn’t be suitable for this matter since it doesn’t hold the capabilities of updating it’s preview constantly which would cause the media to freeze until the video is paused and started again. Why would this be important for a video editing software? When the user searches the moment that he needs to manipulate it would scroll through the timeline until it gets there, and the preview should be updated each millisecond so it provides an accurate time-line position where the pointer would be scrolled.

At this stage these two libraries provide with all I need keeping it light weight also.

The app doesn’t need to use any cloud-based features to do the expected job from a video editing software, the cloud features would be just some add-ons e.g. save project on the cloud based on your login, render the project on a cloud server and then just retrieve it. While this is nice, the modern application doesn’t provide something like this, only CapCut does, but it is more a business-case rather than actual functionality, I will explain: Say I share my project online, then it becomes a property of CapCut which will use it to share with users, so users could just change very minimum stuff in it, like the clips used only and nothing more, which is not a video editing software but just a cool thing.

**Section 4**

There are some risks in relation with the framework chosen because of its limitation, usually it is based on width limitation of components, after some research the limitation would be 30,000 pixels max-width which will place a limitation of circa only 10 minutes length of videos that can be used (off course if you upload 2 clips one with 8 minutes and you get only 30 seconds from it you will still be able to add up to 9 minutes and 30 seconds of video/audio). I do exactly know a workaround for this limitation however I will focus on providing functionality to the application rather than focusing on that alone since the workaround would be complex.

One more risk is the time available to provide enough functionality to the software. These applications often are complex to build but very useful for many business-cases. The existing applications I did describe in **Section 2** usually have a vast number of developers while providing new features/functionalities once 2 to 3 months which can give an Idea how complex the project would be. However, my clear vision on what do I have to do will allow me to do the most important features faster.

The libraries that I intend to use doesn’t provide key functionality rather than displaying the changes made to clip and rendering the changes, most of the logic is pure back-end without use of any SDK’s which also will provide even more complexity to the project, by that I mean, I won’t use any prebuilt components/algorithms nor for back-end or front-end.

The project itself is well organized which is deemed to success with right effort and time provided, it was researched well.

Since I have to finish the page I could mention about the cross platform issue, but not quite much since it is C# with .Net framework, only thing that would have to be changes is the libs for other OS systems and the file manipulation, but file manipulations would be considered as minimal since variables would be used so I won’t label it as big risk but still a very minimal risk. When it comes to android devices or IOS it won’t be possible to use the same solution, still the functions can be used and the only change needed here would be the frontend.

And last risk would be running the application, it would be labelled as an unknown source, that would warn the antivirus, still you can run the application by ignoring it but usually depending on the machine windows defender configuration it could remove the .exe (application) straight away.