

Software description: Audio/video redactor with many file editing functions

Feasibility Study - Input

Budget: \$10 000

Scope and Audience:

1. Alternatives: [list of similar popular programs]
2. End Users: People of any editing experience
3. Experts: Software Engineers with experience in this field

Quantitative study - Potential Operational Issues:

1. Long response times
2. Unintuitive outputs / incorrect outputs based on perceived expected result
3. Corruption of files
4. Bad UI
5. Payment Issues

Qualitative study - Potential Technical Issues:

1. Security concerns (specifically regarding updates & online user made libraries)
2. Compatibility with files and user made libraries

Feasibility Study - Output

Specified Functions & Features (Budget: \$10 000):

1. Advanced file editing
2. Many formats support
3. Online asset library
4. A freemium model with basic features available for free and advanced for €10/month
5. Customer support system on the website, being able to help users with their problems when requested

Risk Management:

1. Regular Updates: Keep the software updated with the latest security patches and features.
2. Risk Assessment: Regularly assess and mitigate risks related to security, technology, and market changes.

Legality and Compliance:

1. Data Protection: Ensure compliance with data protection laws (like GDPR) for user data security.
2. IPR and Licensing: Secure all necessary intellectual property rights and licenses for the software and any third-party integrations

List of Keywords and Recommendations

Keywords	Recommendations
File Editing	The program must support many formats of audio and video files and allow complex editing of them, including file format conversion
Stability	An effort must be made to make the software stable and responsive (minimize crashes and response times)
Availability	The software must have a comprehensive user interface to make it possible for inexperienced people to work. The work of the program must also be intuitively readable. In case of problems, the website will contain customer support, which can be contacted to help solve issues
Flexibility	The addition of online browsing tool for user made libraries will allow users to modify and enhance their experience at their own will, and making a useful framework for developing add-ons will only strengthen this side of the project
Safety	With online features implemented, a maximal focus on protecting the users must be made. In case of online library, implementing a strong and robust malware detection system will be the bare minimum required

Decision: Continue the development

Requirement Analysis - Input

Threshold for operations not dependent on project size: Around 5 seconds, may vary

Session time for online features: Around 30 seconds, may vary

Online functions table

Function	Input Parameter	Operation type	Operation on database
Create Account	Email Password	Soft + Hard	Write
Log in	Email Password	Soft + Hard	Read, then Write
Upgrade to Premium	Bank Details	Soft + Hard	Write
Issue a Support Ticket	Initial Information	Soft + Hard	Write
Answer a Support Ticket	-	Soft + Hard	Read, then Write
Mark a Ticket as Solved	Ticket Number	Soft + Hard	Write

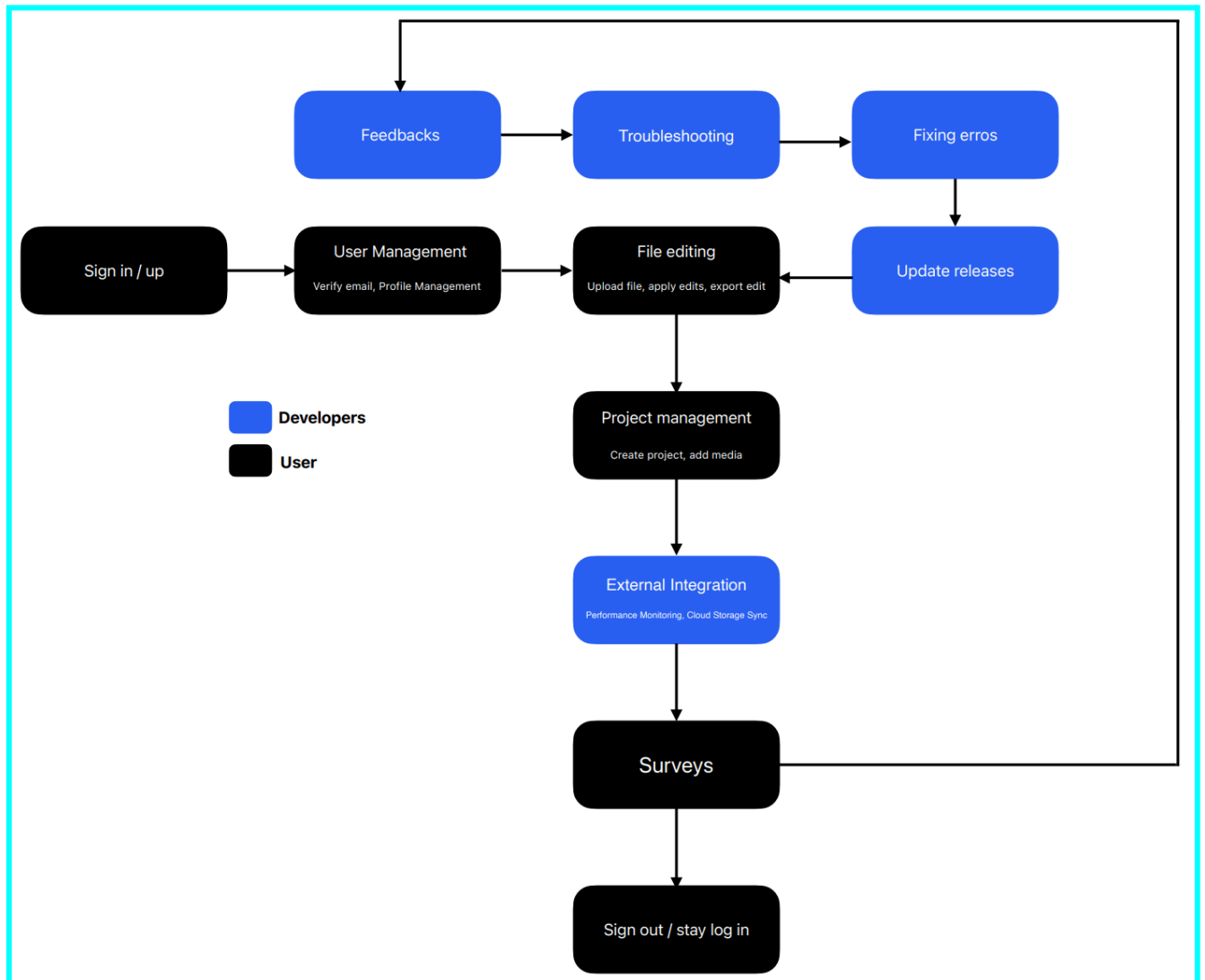
Requirement Analysis - Output

Table of Users, Activities and Measurements

Users	Activities available for users	Measurements (Frequency of being chosen / average response time)	Success (Threshold)	Success (Session Time) for online
Basic user	Create Account	100% / 5s	8s	30s
	Log In	100% / 2s	5s	20s
	Upgrade to Premium	8% / 4s	10s	30s
	Create Project	100% / 6s	8s	
	Load File	80% / 1.5s	3s	
	Add Effect	75% / depends on effect	depends	
	Cut	70% / 0.1s	0.5s	
	Delete File	90% / 0.4s	0.5s	
	Save Project	95% / 1s	5s	
	Export File	60% / depends on file size	depends	
	Issue Support Ticket	5% / 3s	5s	30s
Premium user	All Basic Activities			
	Stereo Sound	30% / 2s	5s	
	Advanced Effects	15% / depends on effect	depends	
Developer / Tester	Edit File Metadata	2% / 0.2s	1s	
	Select Exported File Size	5% / depends on file size	depends	
	Randomize Effects	10% / 1s	5s	
Customer Support	Answer a Support Ticket	1s	5s	60s
	Mark a Ticket as Solved	0.5s	1s	30s

Requirement Definition - Input

Workflow



Use Case Diagram

User: Register → Log In → Download Software

User (Premium): Register → Log In → Upgrade to Premium → Download Software

User (Needs Help): Log In → Request Online Support → Add Initial Information → Send

Project: Create → Open → [Interact] → Save = Set Attributes

Data Structure

Types of Users: Basic User, Premium User (Software - access to new features), Developer (Website and software - access to new features), Support (Website - access to new features, no features of basic user)

Metadata

User	Activities	Items or Services
Basic Premium Customer Support	Register (number of) Log In (number of) Download Software (number of) Upgrade to Premium (number of, current price) Issue Support Ticket (number of, reason) Answer Support Ticket (time spent, user feedback)	Software Tier Ticket Information Feedback

List of Classes and Objects

Class	Object Instance	Attributes / Metadata	Functions
user	person's account	account name account password user type	register account log in upgrade to premium request online support
project	a locally saved project	name contents creation time last edit time file path special file type	create open delete
support ticket	a support ticket	ID user attached initial information, given by user state of solving	add initial information send mark as solved

Requirement Definition - Output

Functional and Nonfunctional Requirements:

See Table of Users, Activities and Measurements

Software Specification:

1. Hardware Components

Operating System	Windows 7 or higher; Mac OS 10.8 or higher
Processor	800Mhz or faster
RAM	256MB or more
Video Card	64MB or more
Sound Card	64MB or more

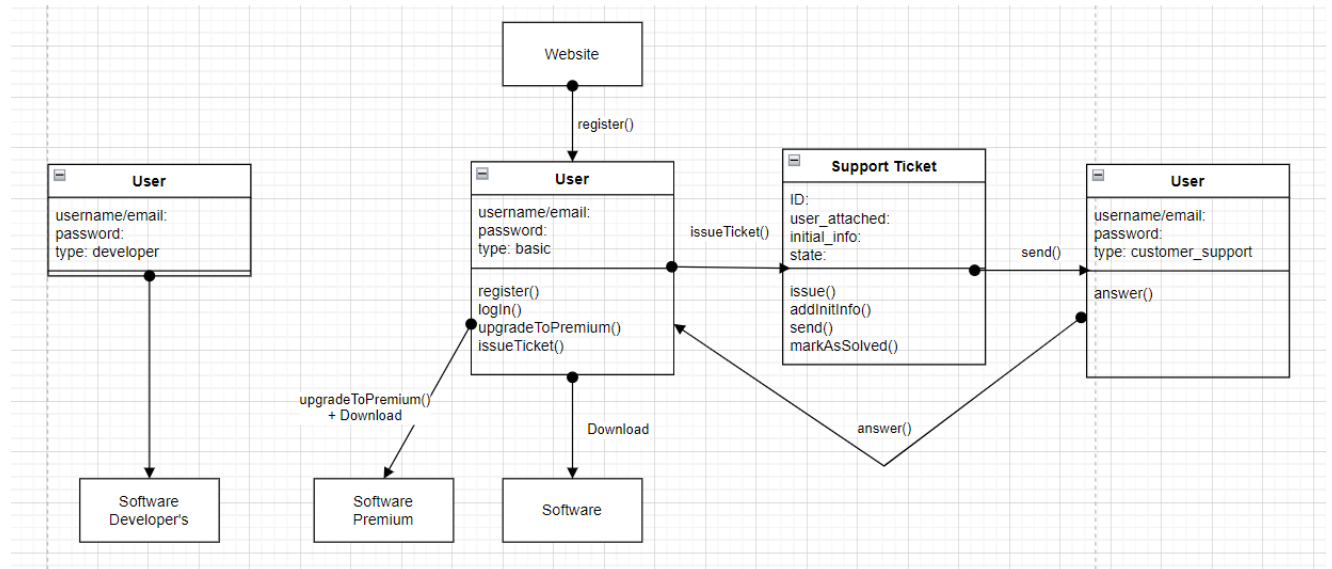
2. Software Components

Programming Language & Software	Python; Pycharm Professional Edition
Version Control System	Git
Additional Requirements	-

3. Quality Level

Thresholds and Session Times are defined in Requirement Analysis

Design



User Interface Design:

- **Layout:** For an editing software, prioritize easy access to common tools and features. Use tabs or menus for organization. A tabbed layout with sections for 'File', 'Edit', 'Effects', and 'Export'.
- **Interactivity:** Ensure interactive elements like buttons and sliders are responsive and provide feedback (like changing color when hovered over).
- **Accessibility:** Incorporate accessibility features such as keyboard shortcuts, screen reader compatibility, and adjustable text sizes.
- **User Experience (UX):** Conduct user testing to refine the UI, ensuring it meets the needs of both novice and experienced users.

Data Flow and Processing:

- **Data Input:** Implementing a drag-and-drop area for file uploads and a file browser for importing.
- **Editing Process:** A visual timeline where users can cut, drag, and drop different media segments.
- **Data Output:** Options to export videos in formats like MP4, AVI, or MOV, with customisable resolution and bitrate settings.

Component Design:

- **Editing Engine:** Capable of real-time rendering for effects like color correction, transitions, and text overlays.
- **Asset Library:** A searchable database of free and premium soundtracks, transitions, and visual effects.
- **Customer Support System:** A ticketing system for detailed support requests.

Testing and Quality Assurance:

- **Unit Testing:** Testing the file upload module to handle different formats and sizes.
- **Integration Testing:** Ensuring the export function works correctly after editing, applying effects, and using assets.
- **User Testing:** Setting up a beta test group of varied skill levels to use the software and provide feedback.
- **Performance Testing:** Simulating high-load scenarios to check if the software maintains performance without crashing.