



Software Requirements Specification for Video Merger App

Prepared by: [Bacsain, Jave A.

Parro, Carl Gerald J.

Prestado, Marc Justin N.]

Date: October 6, 2025

Version: 1.0

1. Introduction

The Video Merger App is a desktop tool that will be created to make video editing and uploading as simple and efficient as possible. It will help users combine multiple video clips into a single file and upload the finished video directly to YouTube, no need for complicated editing software or manual uploads. The app will be especially useful for streamers, educators, and content creators who want to save time by automating repetitive steps like merging, naming, and uploading clips. This document outlines what the system will do, how it will work, and what requirements must be met to build it successfully.

1.1 Purpose

This Software Requirements Specification (SRS) describes the goals and requirements for the Video Merger App, version 1.0. The purpose of this document is to guide the development team in creating a simple and reliable video merging and uploading application. It also serves as a reference for all stakeholders from developers to testers to ensure everyone has a clear understanding of what the system is expected to achieve.

1.2 Scope

The Video Merger App is a desktop application that will enable users to:

- Select and arrange multiple video clips.
- Merge those clips into a single video file in a chosen format and resolution.
- Automatically upload the merged video to their YouTube account, with options to add titles, descriptions, and tags.

For Users (General Content Creators):

They can easily combine their video segments (e.g., recorded gameplay, lectures, vlogs) into one file without using advanced editing tools.

For YouTube Uploaders:

They can directly publish their merged videos to YouTube through the app's built-in integration, saving time otherwise spent switching between editing and browser interfaces.

The system will not include advanced editing features (e.g., trimming, color correction, effects, or multi-track timelines).

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification



- UI: User Interface
- API: Application Programming Interface
- YouTube API: Google's interface for uploading and managing videos programmatically
- FFmpeg: A multimedia framework used for processing and merging video files

1.4 References

- ISO/IEC/IEEE 29148:2018 — Systems and software engineering — Life cycle processes — Requirements engineering
- YouTube Data API v3 Documentation (Google Developers)
- FFmpeg Official Documentation

1.5 Overview

This document is divided into three sections:

- Section 1: Introduction to the app and purpose of the document
- Section 2: Overall description including product perspective, user types, and constraints
- Section 3: Specific functional and non-functional requirements

2. Overall Description

This section describes the general factors that affect the product and its requirements

2.1 Product Perspective

The Video Merger App is a standalone desktop application that will be built using Python and FFmpeg for video processing. It will also integrate with the YouTube Data API to authenticate users and upload their videos directly from the app.

The app will feature a graphical user interface (GUI) that will allow drag-and-drop file selection, merging order adjustment, progress tracking, and upload management.

2.2 Product Functions

Key features will include:

- Video Selection and Arrangement: Users can import multiple clips and reorder them as needed.
- Video Merging: The app merges all selected clips into a single output file using FFmpeg.
- YouTube Uploading: Users can log in to their YouTube account and upload the final video directly.
- Metadata Editing: Users can input video titles, descriptions, tags, and privacy settings before uploading.
- Progress Tracking: The app displays merging and upload progress to keep the user informed.
- Error Handling: Provides alerts for failed merges, invalid files, or upload errors.



2.3 User Characteristics

- General User / Content Creator: A basic to moderately tech-savvy individual familiar with desktop applications and video content creation. Their main goal is to streamline the video preparation and uploading process.
- YouTuber / Streamer: Regularly merges and uploads recordings, often handling multiple short clips per session. Prioritizes efficiency and reliability.
- Educator / Lecturer: Creates lecture recordings or tutorial videos and seeks an easy solution to merge and upload them without technical complexity.

2.4 Constraints

- The application must run on Windows 10 or later.
- The app must integrate with FFmpeg for all video merging operations.
- Upload functionality must comply with YouTube Data API v3 authentication and upload policies.
- All user authentication tokens must be stored securely and encrypted locally.
- The system must be developed using Python with the Flet framework for the graphical user interface.

2.5 Assumptions and Dependencies

- Users must have a stable internet connection to upload videos.
- Users must have a YouTube account with permission to upload.
- FFmpeg and YouTube API services must be available and functioning.

3. Specific Requirements

This section contains the detailed requirements necessary to build the system.

3.1 Functional Requirements

FR-001: User Authentication

- The system shall allow users to sign in using their YouTube account through the YouTube Data API (OAuth 2.0).
- After login, the system shall save a secure session token for future uploads.

FR-002: Video Selection and Arrangement

- The user shall be able to import multiple video files from local storage.
- The user shall be able to reorder the video clips before merging.



FR-003: Video Merging

- The system shall merge all selected video clips into one continuous video file using FFmpeg.
- The user shall be able to choose the output format (e.g., MP4) and resolution.

FR-004: YouTube Upload

- The system shall upload the merged video directly to YouTube using the user's account.
- The user shall be able to enter a title, description, tags, and privacy setting (Public, Unlisted, or Private).

FR-005: Progress Tracking and Notifications

- The system shall display real-time progress for both merging and uploading.
- The system shall notify the user when merging or uploading is completed or fails.

FR-006: Error Handling and Logging

- The system shall handle errors such as missing files, upload failures, or invalid credentials gracefully.
- The system shall log all errors for troubleshooting.

3.2 Non-Functional Requirements

This details how the system should perform its functions.

NFR-001: Performance

- Merging performance shall depend on system resources, but the app must handle up to 10 clips within a reasonable time (<5 minutes for standard HD).
- Upload feedback must appear within 5 seconds of starting the upload.

NFR-002: Usability

- The interface must be simple and intuitive, allowing a new user to merge and upload a video without training.

NFR-003: Security

- User authentication data (tokens) must be encrypted and stored locally.
- No passwords will be stored by the app.



NFR-004: Reliability

- The app should recover gracefully from upload interruptions (e.g., network loss).
- All merging and upload logs should be stored locally for future review.

NFR-005: Maintainability

- The system should be modular and allow easy updates for new features or API changes.