

**VES Project**

Requirements analysis and specifications document

Filippo Solimando

**Table of Contents**

1. Introduction

1.1. Project Scope

1.2. Target Clients

1.3. Functionalities

1.4. Goals

1.5. Limitations

2. Functionalities details

2.1. Actors

2.2. Functionalities description

2.2.1. Session Initialization

2.2.2. Video File Upload

2.2.3. Subtitles File Upload

2.2.4. Resize Inizialization

~~2.2.5. Video Information Query~~

2.2.6. Session Status Query

2.2.7. Burn Video

2.2.8. Burn Stop

2.2.9. Download Burned Video

3. Scenarios

3.1. Resource Definition

3.1.1. File Upload Insufficient Server Resources

3.1.2. Burn Insufficient Server Resources

3.1.3. File Upload After Burn

3.1.4. Resize Definition After Burn

3.2. Download Video

3.2.1. Download Video Before Burn

3.2.2. Download Video After Burn

3.2.3. Download Video After Error

4. System requirements

4.1. Constraints

4.1.1. Design constraints

4.1.2. Performance constraints

4.1.3. Database constraints

4.1.4. Hardware constraints

4.1.5. Software constraints

4.1.6. Compliance to standards

4.2. Software properties

4.2.1. Reliability

4.2.2. Availability

4.2.3. Security

4.2.4. Maintainability

4.2.5. Portability

4.3. Technical requirements

4.3.1. Hardware

4.3.1.1. Memory

4.3.1.2. Network interface

4.3.1.3. Processor

4.3.2. Software

4.3.2.1. Database management system

4.3.2.2. Network protocols

4.3.2.3. Internal tools

4.3.2.4. Java virtual machine

**Introduction**

**1.1. Project Scope**

VES (Video Editing Service) is a web application to provide basic video manipulation features. Videos uploaded in VES can be resized or completed with subtitles. VES offers its functionality by a specific REST api set.

**1.2. Target** Clients

There is no specific on client type. Any entity can access to VES if connected to the network.

**1.3. Functionalities**

The system will implement these functionalities:

- Video resize

- Subtitle integration into the video frames

**1.4. Goals**

VES is aiming to accomplish:

- To offer functionalities by RESTful api and communicate over http/https protocol

- Asynchronous video processing

- To be distributed into a Dockers Container Image

**1.5. Limitations**

The software will suffer from the following limitations:

- The system must be reliable

- The system has limited local resources

- No specific security and privacy requirements about video files will be implement

**Functionalities details**

**2.1. Actors**

Any entity able to communicate to VES by given REST APIs. We will refer to it as Client

**2.2. Functionalities description**

**2.2.1. Session Initialization**

In order to access to any further functionalities, the Client must request a new working session. The informations returned by this request will be used to identify and execute any other action.

**2.2.2. Video File Upload**

Upload the video file the Client wants to manipulate. Only one file can be processed in a working session.

**2.2.3. Subtitles File Upload**

Upload files containing video subtitles. \*.SRT, \*.SUB and \*.SBV file types are supported by VES. Subtitles are not mandatory.

**2.2.4. Resize Inizialization**

Specify the final video size. Video resize request is not mandatory.

**~~2.2.5. Video Information Query~~**

~~Return uploaded video properties, like duration, size, etc.~~

**2.2.6. Session Status Query**

Return current session status, file actually uploaded into the session, percentage of completiotion if the video process has been started.

**2.2.7. Burn Video**

Execute the video processing. In order to start the action, a video file must be uploaded and at last a subtitles file must be uploaded or a resize inizialization must be specified. Subtitles and resize can be executed simultaneously

**2.2.8. Burn Stop**

Interrupt the video processing if already started.

**2.2.9. Download Burned Video**

Download the processed video file as soon as the elaboration is finished.

**Scenarios**

Scenarios are written in the gherkin formal syntax.

**3.1. Resource Definition**

**3.1.1. File Upload Insufficient Server Resources**

Given I am a client

and there is not enought space on the server

when I upload a file

then the system must return error

and cancel the session

**3.1.2. Burn Insufficient Server Resources**

Given I am a client

and I upload a video file

and I upload a subtitle file or declared a resize

and there is not enought space on the server

and I execute Burn action

then the system will stop the elaboration

and put the session in an error mode

**3.1.3. File Upload After Burn**

Given I am a client

and I executed Burn action

and Burn process started without error

and I upload a file

then the system return invalid action error

**3.1.4. Resize Definition After Burn**

Given I am a client

and I executed Burn action

and Burn process started without error

and I declare a resize modification

then the system return invalid action error

**3.2. Download Video**

**3.2.1. Download Video Before Burn**

Given I am a client

and I executed Burn action

and Burn process started without error

and Burn has not finished yet

and I request to download processed video

then the system return work in progress error

**3.2.2. Download Video After Burn**

Given I am a client

and I executed Burn action

and Burn process started without error

and Burn completed the video elaboration

and I request to download processed video

then the system return the video processed

and cancel the session

**3.2.3. Download Video After Error**

Given I am a client

and I executed Burn action

and Burn process started without error

and Burn did not complete the process cause an error

and I request to download processed video

then the system return the error during Burn process

and cancel the session

**System details**

**4.1. Constraints**

**4.1.1. Design constraints**

The system must be implemented in Java using the J2EE platform and JAX-RS technology.

**4.1.2. Performance constraints**

The system response time must be acceptable. It must be able to handle many concurrent requests: the order of magnitude of the userbase is unknown.

**4.1.3. Database constraints**

Not specified.

**4.1.4. Hardware constraints**

Not specified.

**4.1.5. Software constraints**

Not specified.

**4.1.6. Compliance to standards**

Whenever possible, the code and organization of the system should obey to existing standards. This allows for an easier maintainability and quicker accessibility for new programmers.

**4.2. Software properties**

The produced software must offer a standard set of properties. Here follows a description of how this system complies to those properties.

**4.2.1. Reliability**

The system must assure that every data entered by client is persisted until the client decides to cancel the session or the processed video is successfully downloaded.

**4.2.2. Availability**

Not specified.

**4.2.3. Security**

Not specified.

**4.2.4. Maintainability**

Standard compliance will aim future developers during software maintenance.

**4.2.5. Portability**

By using Java technologies, our product can be installed virtually anywhere, provided that the target operating system has a JVM (version >= 1.8) installed.

**4.3. Technical requirements**

**4.3.1. Hardware**

The computer which the system will be installed on needs to have the following requirements.

**4.3.1.1. Memory**

Depending on the amount of video process session running in the same moment, VES could need a huge amount of free storage space, we reccomand at last 100GB each service installed.

**4.3.1.2. Network interface**

The application needs to instantiate a web server, so a network interface is required.

**4.3.1.3. Processor**

The system runs one thread every video process, a multi-core processor will impact a lot on the overall performance.

**4.3.2. Software**

The system needs several software modules in order to work properly.

**4.3.2.1. Database management system**

MongoDB (version >= 3.0.3)

**4.3.2.2. Network protocols**

The application need to work on the following network protocols:

● HTTP, on TCP, port 80 suggested

● HTTPS, on TCP, port 443 suggested

**4.3.2.3. Internal tools**

ffmpeg library must be installed

**4.3.2.3. Java virtual machine**

Any instance of JVM, from version 1.8 onward.