

# Appendix 1 Technical requirements

## 1 Project goals

The Customer desires to acquire and the Developer will develop & deliver the web-based service for automated online video rendering of book presentation trailers. The project goal is to create such website for private users and a web application service for corporate customers.

Potential service users are the individuals that work in book publishing industry, publishing houses that need to promote the books, online and offline bookstores that wish to create simple advertisement for the books they sell.

The service will provide following functionality:

- Perform user registration.
- Provide selection of preliminary created scenarios of book presentations with possibilities to customize the content of the presentation (texts: content, font, color, etc., images).
- Define the parameters (text, options, upload images where appropriate) and create the customized video based on scenario.
- Watch the small preview of the customized scenario in 3D (using WebGL) (free).
- Order (purchase) the full-sized HD quality video created from the customized scenario.
- After the video is created, it can be downloaded from the website through a direct link. The video file availability time should depend on user account characteristics.
- The website will store customized scenario for further use.

The service design and implementation must consider that the functionality described above (except for registration and payment) is accessible through API to enable integration into other websites.

## 2 Main tasks

In order to achieve the project goals the Developer will do the following tasks:

- Design the code-first entity model for the project.
- Design and create the Microsoft SQL Server database that will conform to the entity model.
- Design and develop ASP.Net application implementing the functionality of the service, which will provide the JSON (and optionally XML) interface for data manipulation (API).
- Create the functionality to display the WebGL file parameterized with user input (texts, images, etc.)
- Create the procedure for rendering of book trailer video from scenario file(s) using virtual machines from Microsoft Azure cloud; implement the scripts to execute rendering and video encoding task.
- Bind the payment solution provided by the Customer to the web site.
- Implement the website that operates through the API with the service and provides the functionality of the service to private users.

The Developer should implement the tasks according to the requirements and guidelines specified in this document.

## 3 Functional requirements

### 3.1 User interface requirements

The website that implements private user access to the book trailer creation functionality will contain following sections:

1. Landing page with general service information
2. User registration form that will provide fields to enter username, password, password confirmation, e-mail address
3. User login form where user specifies username and password to identify and authorize
4. User “office” – set of pages with user-specific information:
  - a. Authorization information pages view/change password and/or e-mail
  - b. List of book trailers (projects) with both unfinished projects and the projects with purchased/rendered trailers. The projects with purchased videos are read-only with options to re-render the video or create a new project copying all current settings into it. Read-only attribute is necessary to ensure that the user is able to create the video in different resolution or render it once more if the video file is removed from storage.
  - c. Detailed view of the project with options to continue editing (if the project is incomplete), with the option to create a new project using currently selected as a template, and the link to download the videos for the project (if they exist).
  - d. Billing information: the list of payments and invoices.
5. Page with list of available trailer scenarios – each list item contains basic information about the scenario (description, timing, screenshot from a sample video, price).
6. Page with detailed information about scenario containing preview, available list of trailer parameters, such as image files for the book (book cover, book spine, book forth coverage, two pages featured inside the book – Left and Right), image file for the company logo, book title, author name, background image or video, soundtrack, followed by list of trailer clips (parts) and their parameters.
7. Page for user customization of the book trailer – creation of a book trailer for particular book. The page contains all the fields from scenario page with possibility to fill them with user-specific information.
  - a. Page with 3D (webGL) preview
  - b. Page with order (purchase) information for a customized book trailer with option to select the resolution and proceed to payment page.
8. Payment page with binding to PayPal payment gateway (the Customer will provide the credentials information and will perform communications with the PayPal support if necessary).
9. Page with notification that the payment was accepted/declined. If the payment is accepted, the website will also display information about the rendering task that is launched with the link to the page where the user will receive the downloadable video.

The user interface should be plain, intuitive, with minimal decorative attributes. The Developer will provide the general website layout for the Customer to approve in 10 days after the development starts. The Developers will provide sample landing page and 2-3 sample pages that provide website look and feel for review and approval.

Webpages layout and/or logic may be changed during the development with Customer approval.

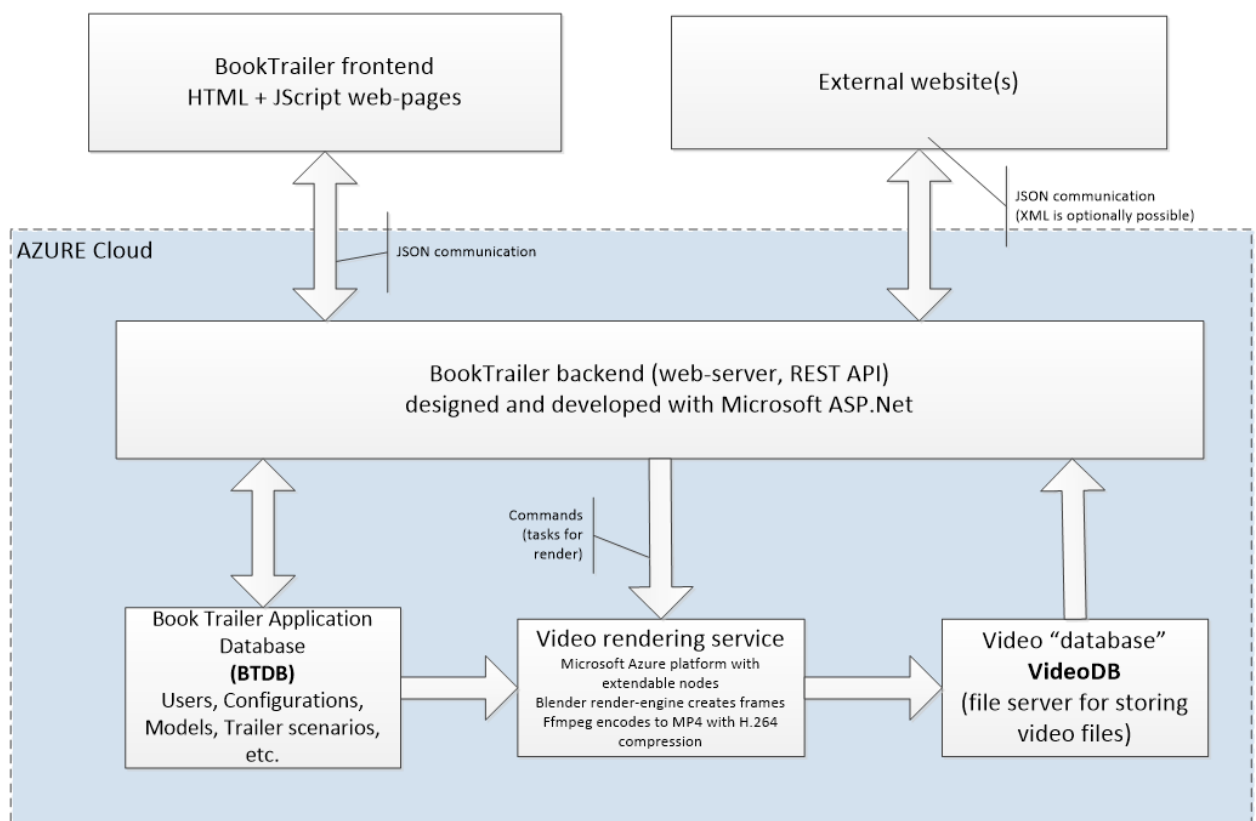
The site has to support multilingual pages and creation of book trailers in multiple languages.

## 3.2 Technological requirements

### 3.2.1 Architecture and platform

The Developers will implement the entire solution as a 3-tier application. This will help to achieve the goal to be able to use the core of the web site to deliver the services/functionality to other websites.

1. The frontend layer (the book trailer website) implements the UI for the clients. It communicates with the backend through and ONLY through API, provided by the ASP.Net application.
2. The API layer, which implements the business logic and is implemented using Microsoft ASP.Net. It publishes REST API for frontend applications for ALL functions.
3. The “database” and “render farm” layer consist of three components:
  - a. Book Trailer web site database (**BTDB**) for storing site-specific information like user (client) information, trailer scenarios and templates, configurations of client tasks, payment information, etc.
  - b. Render farm, which is essentially Microsoft Azure batch (combination of nodes with tasks running on them) with Blender renderer launched there with parameters taken from task and database (BTDB). We will ensure to configure the nodes (computers) pool to achieve optimal performance with automatic extension when necessary and releasing the resources when they are no longer needed.
  - c. Source image/videos/audio and result video files database (**VideoDB**), that stores uploaded files by client and the video files and provides them to the client.



### 3.2.2 Database architecture

**Microsoft SQL server** will be the platform to host the application SQL-based relational database.

**Microsoft Entity Framework Code First** technology will be used to create/modify database structure and relations between database objects in order to speed-up and simplify database development phase. The Entity Framework Code First approach allows the development team to focus on other aspects of web development such as enhanced user experience design, usability and video preview generation engine.

Web application database stores the following information:

- User registration and contact information
- User billing information
- User book trailers information (projects)
- User customized sequences
- Predefined book trailer information (project templates)
- Predefined book trailer previews (pre-rendered trailers)
- Predefined animation effects
- Soundtracks (premium soundtracks, user soundtracks)
- Backgrounds (premium backgrounds, user soundtracks)

In order to avoid fast database growth caused by binary files storing (images, videos, sound files) these files will be stored in Microsoft Azure storage (VideoDB) as separate files with identifiers, names and types being stored in application database.

This approach ensures that the database size growth will not require upgrading of the database hosting. It is more efficient and cheap to reserve extra space in generic storage in Azure cloud.

### 3.2.3 Database content

Developer will define the list of entities in database in order to implement the functionality of the project. The application database will presumably contain following entities:

- Trailer template – description of the trailer with list of sequences it consists of, list of parameters' descriptions and default (sample) values, preview video reference. Contains list of sequence templates
- Sequence template – description of a trailer part with set of its parameters
- Trailer instance – the trailer created by the user from a template. Stores the actual values of parameters for rendering.
- Sequence instance – part of trailer instance, created from sequence template, contains actual parameters used for rendering.
- Video file - contains information about the rendering task and clip. There will be (potentially) several video file records corresponding to single trailer instance, as the video may exist in various resolutions and rendering may occur several times.
- Set of tables for user identity management
- Font – set of font parameters. May be bound to sequence or trailer templates and instances.
- Storage link – auxiliary table that contains identification information (i.e. file name) and description of a binary resource stored outside database.
- Invoice and payments information – this will be designed after the payment model is finally determined by the Customer.

Each entity will have a set of common fields, used for database management and maintenance. This includes, but not limited to ID column, Creation date, and Modification date. These and other auxiliary fields are used for user activity tracking and data consistency.

Developer will define the exact list of fields in all entities during the development. Presumably, the entities will contain the following fields:

User information:

- User name
- Password hash
- Password "salt"

- Name/First name of the contact
- E-mail address
- Company Name (if any)
- Full address of the company + Country
- Phone
- VAT number (if any)
- “Parent” or “group” user account reference
- Reference to user quotas record
- List of trailers requests

User quotas (contains information allowing to check if this particular user can perform the rendering – will contain financial and/or administrative limitations); actual fields list will extend during development when payment models will be finalized.

- Number of trailers allowed
- Amount (limit) of money to spend (may contain advance payment information)
- Date – end of “free rendering” period
- Other fields.

Task (trailer) requests information

- Reference to trailer template
- Date/time of render request
- Date/time of render completion
- Cost billed
- Payment information

Trailer template:

- Name (title) to identify among other templates
- 3D file (blender) that will be used for rendering
- Video file for preview
- Duration
- Description
- Base price
- List of parameters (with values that correspond to the preview video file).
  - Title
  - Book cover image file
  - Book spine image file
  - Book 4th cover image file
  - Some pages of the book (depending of the project with open book)
  - Book spine shape (square or rounded)
  - Book cover quality (hard or soft)
  - Company logo image file
  - Soundtrack file
  - Trailer background file (main background: image or video file).

Trailer instance:

- Reference to trailer template used to create the instance
- Reference to the user account to specify the user that owns the instance/project
- Parameters:

- Title
- Book cover image file
- Book spine image file
- Book 4th cover image file
- Some pages of the book (depending of the project with open book)
- Book spine shape (square or rounded)
- Book cover quality (hard or soft)
- Company logo image file
- Soundtrack file
- Trailer background file (main background: image or video file).

Sequence template:

- Reference to trailer template
- Order of sequence in trailer
- Duration
- Parameters:
  - Text that will be displayed (text position information is taken from Blender 3D scenario source file)
  - Reference to font information (font type, size, style, etc.)
  - Image (optional, depends on sequence scenario)
  - Sequence background file (image or video) – if it differs from the trailer background.

Sequence instance:

- Reference to trailer
- Order of sequence in trailer
- Parameters:
  - Text that will be displayed
  - Reference to font information (font type, size, style, etc.)
  - Image (optional, depends on sequence scenario)
  - Sequence background file (image or video) – if it differs from the trailer background.

Premium image/video

- Price
- Category (background image, texture, etc.)
- Reference to image/video file

Premium music

- Price
- Category (music thematic)
- Reference to music file

Font description

- Font name
- Size (in points)
- Color (RGB)
- Italics flag
- Bold flag

### 3.2.4 Image, video and sound restrictions

The images and videos may be either loaded by the user or chosen from predefined premium image and video collections. If the user loads the image, then application will expect that the image match the necessary aspect ratio: i.e. the application may expect the image to be square-shaped or the sides should match 4:3 ratio. The user can load image with different aspect ratio, but application will have to warn that the image will be stretched/shrunk during rendering, which will lead to distortions when rendering video.

The video and audio files that the user loads as background or soundtrack may have duration that differs from the trailer duration. If the trailer duration is less than the video or sound duration, application will truncate the user file to the necessary duration. If the user file duration is less than the trailer duration, application will loop the playback of the video or sound file when rendering the trailer.

The application will restrict the size of loaded files of all types. Developers will define the restrictions and store them to the application configuration file or into the database. Application will apply different restrictions for different file types.

Other restrictions may apply to the files uploaded for video rendering service by the users. We will publish these restrictions on the website and will check if the uploaded files comply with requirements.

### 3.2.5 Multilanguage support

The website will support multiple languages. The support for these languages will include functionality of the website (tracking of current user locale and user language selection) and the database storage with multilingual support.

Current user language will be stored in cookie file – to restore user language choice when he revisits the website. The website URL addresses will have the spelling with language identification within:

- "videotrailer.tv/en/home/index"
- "videotrailer.tv/fr/home/index"
- "videotrailer.tv/ru/home/index"

Database will include the dictionary (table) of supported languages and the table with translations of all text fields and other text-containing objects on the website.

Also the TrailerTemplate and TrailerInstance entities will have the language attribute – to enable language-dependent trailers.

The developers will implement the English version and the Customer should provide translations for other languages.

## 3.3 Payment model requirements

The Customer will design the payment model according to business requirements no later than in 1 month after the development starts.

Payments will proceed through the PayPal payment system. The Customer will provide all necessary information to implement the payment page, deliver all credentials and do the necessary communication with PayPal support if necessary.

The Developer will add entities to store user payments and invoices to the application database.

## 3.4 3D preview

User will be able to view the 3D animated model of the trailer with customizations. The client computer will be the rendering in real-time using WebGL control. If the user browser does not support WebGL the

control will not be available. If the client computer does not have sufficient processing power, the frame rate of visualization will not be real-time.

The application may adjust the resolution and/or color depth of images used for rendering to optimize performance of 3D visualization.

### 3.5 Video rendering requirements

Azure cloud computers will perform video rendering tasks. Developer will create the image of the render computer (choose OS and prepare set of applications that are needed for rendering and encoding).

When user requests (orders, purchases) the video, the application will perform following tasks:

- Initialize the instance of render computer in the cloud.
- Upload the image of operating system and applications.
- Upload the source files (3D scenario file, images, video and sound files) to the computer and optionally change their names so that the 3D scenario file references the custom files.
- Launch the render application to form frames (images)
- Launch the encoding application to create video stream (MPEG, H.264 encoding)
- Store the result file to storage and create a record in database that will bind the video to the user trailer.
- Shutdown the render computer when it finishes the task (to minimize the cost of render computing).

The Developers will use Blender open-source application for frame rendering.

The Developer should use ffmpeg utility for video encoding.

3D scenario file is preliminary created in Blender. The frame rate of the target video is specified in a 3D scenario file. This frame rate will be used to create the video. If necessary, the target frame rate of the video may differ from target frame rate, but this requires that the artist (designer) adjust the 3D scenario model to avoid the render speed distortions.

## 4 Delivery

The Developers will deliver the project as a full-functional website. It is necessary to deploy the website, database, file storage and render “farm” on Microsoft Azure cloud hosting.

Developer will deliver the results of the work as a Windows application that can run on Windows computer with Microsoft.Net 4.5 installed.

The Customer creates the Azure account and passes the credentials to the installer application. The application performs the following tasks:

- Set up Azure environment automatically; Azure environment configuration will include:
  - App Service plan
    - App service - web site
  - SQL server (logical server)
    - SQL database
  - Storage account (create if it does not exist)
  - Batch account - linked to storage account
    - Application packages
      - AzCopy - for azure storage account manipulations
      - Blender - for trailer rendering to PNG image files



- FFMpeg - for trailer encoding to MP4 files
  - Render - set of command files for using in Batch tasks
  - Pool - set of computers for rendering
    - Computer configuration (OS, computer type, processor, disk and memory)
    - Auto scale configuration (script for shutting down unused computers from render pool)
- Deploy the Book Trailer application
  - Create MS SQL database, user and setup the database structure
  - Seed the database (fill the default values, dictionaries, etc.)
  - Copy files for sample trailers to the storage and create records in database for trailers and their parameters.
  - Deploy and configure the website.

The Developer will test the installer application against the Developer's Azure account.