**Introduction**

The Management Application is an application that allows rendering Blender, Dyad and FFMPEG files on the Azure Batch service provided by Microsoft. This is needed to process larger tasks since C# functions don’t allow such big rendering times. In this file all the functionalities will be tested by one or more tester. Every test done will be written down and shown in a table.

# Scope of Testing

During the test phase, we will focus on how the system interacts with Azure Batch.

We hope that, by virtue of these tests we can offer an improved platform where users can upload and render files and to guarantee that the functionalities of the application can work normally in a real environment.

# Features not to be tested

The features below will not be tested, because they are either not implemented or they are not included in the software requirements specification.

***Not included***

* Anything device-hardware related
* Anything device-software related
* Anything frontend related

# Test Methods

**Usertesting**

Usertesting is a process where real users will test the interface and functions of a website by performing specific tasks in realistic conditions. The purpose of this process is to evaluate the usability of our website to decide whether it’s ready to be launched for real users and to receive feedback to improve the product.

# Who has performed the tests?

|  |  |
| --- | --- |
| Test Method | Usertesting |
| Tester 1 | Gerben Put |
| Tester 2 | Joris Wessels |

# TestCases

|  |  |
| --- | --- |
| ID | Container Creation (Completed) |
| Priority | High |
| Description | A user must be able to create a container |
| Pre-Conditions | User is logged in and is connected to a tenant. |
| Step |  |
| 1 | User starts a new Render task |
| 2 | Input and Output Container will be created and will be named ( Input/Output + Username + TenantID) |

|  |  |
| --- | --- |
| ID | File Upload (Completed) |
| Priority | High |
| Description | A user must be able to upload files to Azure Batch |
| Pre-Conditions | User is logged in and is connected to a tenant. |
| Step |  |
| 1 | User starts a new Render task |
| 2 | Files will be unzipped |
| 3 | Files will be uploaded |

|  |  |
| --- | --- |
| ID | Pool Creation (Completed) |
| Priority | High |
| Description | A user must be able to create a Pool on Azure Batch |
| Pre-Conditions | User is logged in and is connected to a tenant. |
| Step |  |
| 1 | User starts Create Pool function and gives Debian or Windows option |
| 1 | User starts a new render task and gives Debian or Windows option |
| 2 | An Azure Batch pool will be created |

|  |  |
| --- | --- |
| ID | Job Creation (Failed) |
| Priority | High |
| Description | A user must be able to create a Job on Azure Batch |
| Pre-Conditions | User is logged in and is connected to a tenant. A Pool is created. |
| Step |  |
| 1 | User starts a new render task |
| 2 | A job will be created |

|  |  |
| --- | --- |
| ID | Task Creation (Failed) |
| Priority | High |
| Description | A user must be able to create a Task on Azure Batch |
| Pre-Conditions | User is logged in and is connected to a tenant.  A pool is created  A job is created |
| Step |  |
| 1 | User starts a new render task |
| 2 | Job is created |
| 3 | Task is created inside the Job |
| 4 | Task will run untill complete or failure. |

|  |  |
| --- | --- |
| ID | Monitoring Tasks (Completed) |
| Priority | High |
| Description | A user must be able to upload files to Azure Batch |
| Pre-Conditions | User is logged in and is connected to a tenant. |
| Step |  |
| 1 | User starts a new render task |
| 2 | Job is created |
| 3 | Task is created |
| 4 | Once all tasks completed and/or failed. It will throw runtime and a task completion/fail message. |

|  |  |
| --- | --- |
| ID | Auto Installing Software (Completed) |
| Priority | High |
| Description | A user must be able to upload files to Azure Batch |
| Pre-Conditions | A pool is created running on Windows or Debian |
| Step |  |
| 1 | If debian pool: FFMPEG and Blender will automatically be installed |
| 1 | If windows pool: Dyad is automatically installed |

|  |  |
| --- | --- |
| ID | Autoscaling Pools (nodes) (Failed) |
| Priority | High |
| Description | A user must be able to upload files to Azure Batch |
| Pre-Conditions | A pool is created |
| Step |  |
| 1 | User gives autoscale interval |
| 2 | Pool autoscales nodes depending on interval |

# Failed Tests + Fix

Job creation failed due to the job previously already existing. The job will now be created with a randomized number behind the name so this won’t occur anymore.

|  |  |
| --- | --- |
| ID | Job Creation (Completed) |
| Priority | High |
| Description | A user must be able to create a Job on Azure Batch |
| Pre-Conditions | User is logged in and is connected to a tenant. A Pool is created. |
| Step |  |
| 1 | User starts a new render task |
| 2 | A job will be created |

Task creation failed due to the task previously already existing. The task will now be created with a randomized number behind the name so this won’t occur anymore.

|  |  |
| --- | --- |
| ID | Task Creation (Completed) |
| Priority | High |
| Description | A user must be able to create a Task on Azure Batch |
| Pre-Conditions | User is logged in and is connected to a tenant.  A pool is created  A job is created |
| Step |  |
| 1 | User starts a new render task |
| 2 | Job is created |
| 3 | Task is created inside the Job |
| 4 | Task will run untill complete or failure. |

Tasks wouldn’t run on autoscaling pools due to a to short autoscaling time. This is avoided by not allowing autoscaling intervals under 10 minutes.

|  |  |
| --- | --- |
| ID | Autoscaling Pools (nodes) (Completed) |
| Priority | High |
| Description | A user must be able to upload files to Azure Batch |
| Pre-Conditions | A pool is created |
| Step |  |
| 1 | User gives autoscale interval |
| 2 | Pool autoscales nodes depending on interval |