|  |  |
| --- | --- |
|  |  |

**Testreport**

**Authors**Rimac Valdez Quispe  
Robert Kettler  
Daniel Weidle

# Table of contents

Table of contents 2

Abbildungsverzeichnis 3

Tabellenverzeichnis 3

Abkürzungsverzeichnis 4

Version History 5

Preface 6

1 Introduction 7

1.1 Ziel der Testphasendokumentation 7

1.1 Testphasen 7

2 Testfallgenerierung 8

2.1 Testbasis 8

3 Test summary 10

3.1 Testing Scope 10

3.2 Unit/Module Test 11

3.3 Usability Test 11

3.4 Performance Test 11

3.5 User Acceptance Test 12

4 Test results 13

4.1 Unit/Module Test 13

4.2 System Test 13

4.3 Usability Test 13

4.4 Performance Test 13

4.5 User Acceptance Test 15

5 Metrics 16

5.1 Bug report 16

5.2 List of open bugs 17

# Abbildungsverzeichnis

Abbildung 1: Verbinden eines Textteils mit einer Formatvorlage 9

# Tabellenverzeichnis

Tabelle 1: Unterschiede zwischen Abbildungen und Tabellen 10

# Abkürzungsverzeichnis

# Version History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 0.1 | 13.06.2016 | Initial |
| 0.2 | 17.06.2016 | Added open bugs list, bugs in total |
|  |  |  |

# Preface

Dieses Dokument dient zur Dokumentation der in den Testphasen durchgeführten Aktivitäten. Die Testphasen orientieren sich an den von der Projektleitung festgelegten Release Zeiträumen. Des Weiteren dient dieses Dokument der Nachvollziehbarkeit, der Testfallgeneration.

# Introduction

Dieses Kapitel gibt einen Überblick, über das Ziel dieses Dokumentes und ordnet den Release Versionen den in den Testphasen durchgeführten Tests zu.

## Ziel der Testphasendokumentation

Das Ziel diese Dokumentation ist, die durchgeführten Tests innerhalb des Testprozesses zu dokumentieren.

## Testphasen

Die Testphasen orientieren sich an den von der Projektplanung vorgegebenen Release Terminen. Es handelt sich dabei um drei Phasen. Jede dieser Phasen endet mit dem Release einer Version. Die nachfolgende Tabelle zeigt eine Zuordnung der Phasen zu den korrespondierenden Versionen, sowie den durchgeführten Tests.

Tabelle 1: Phasenzeiträume und Release Versionen

|  |  |  |  |
| --- | --- | --- | --- |
| **Beginn** | **Ende** | **Version** | **Durchgeführte Tests** |
| |  |  | | --- | --- | | 09.05.2016 | 06/Jun/16 | | 06.06.2016 | 0.1 | ? |
| 06.06.2016 | 13.06.2016 | 0.2 | ? |
| 13.06.2016 | 24.06.2016 | 0.3 | ? |

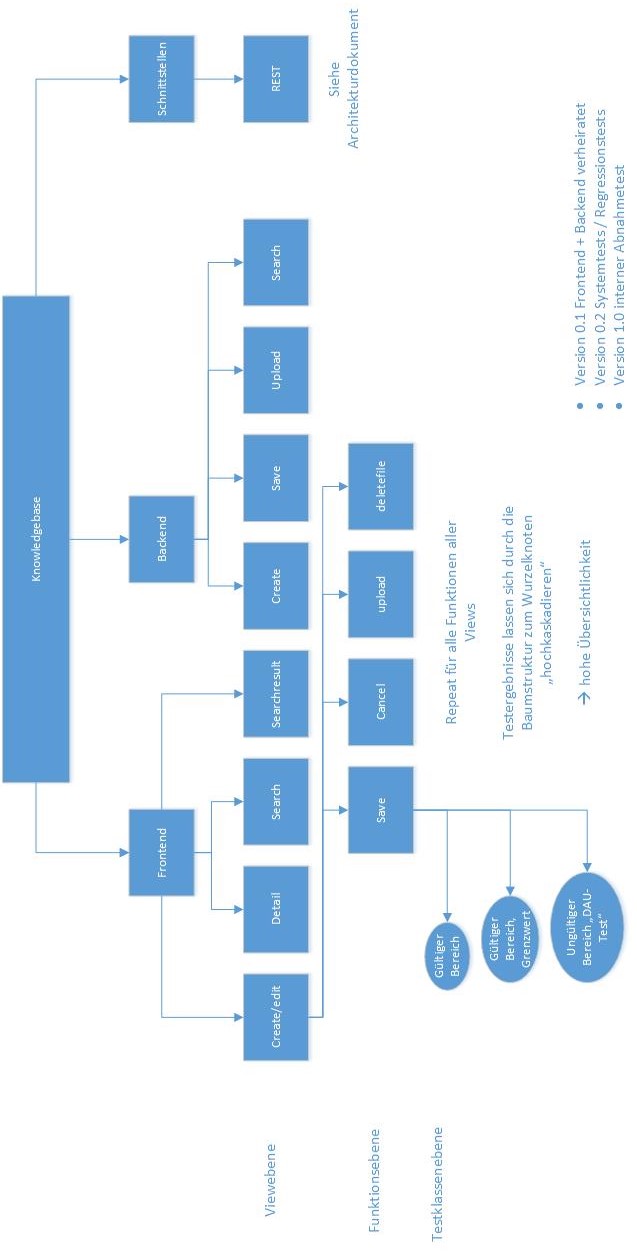
# Testfallgenerierung

Die in diesem Dokument aufgeführten Testfälle leiten sich aus dem, vom Qualitätsmanagment, erstellten Testplan ab. In Abschnitt 2.1 wird auf den Aufbau des Testplans eingegangen, um dem Leser ein Verständnis für die Testfallgenerierung zu vermitteln.

## Testbasis

Das in Abbildung 1 dargestellte Organigramm zeigt den Aufbau des Testplans. Auf Basis dieses Planes lassen sich drei Testbereiche identifizieren. Es handelt sich dabei um das Frontend, das Backend, sowie die Restschnittstellen zwischen dem Backendserver und der Searchengine. Neben der logischen Aufteilung spezifiziert der Testplan drei aufeinander aufbauende Ebenen. Die Viewebene entspricht den vom Benutzer aufrufbaren Seiten. Jede einzelne dieser Seiten bietet dem Benutzer eine bestimmte Anzahl an Funktionalitäten. Diese werden in der Funktionsebene dargestellt. Anschließend werden für sämtliche Funktionen Testfälle mithilfe von Äquivalenzklassenbildung generiert. Definiert sind die Testklassen, „Gültiger Bereich“, „Gültiger Bereich – Grenzwert“, sowie „Ungültiger Bereich.“. Auf Basis dieses Plans werden von den Testern Testfälle generiert. Eine Auflistung sämtlicher durchgeführter Tests findet sich in Kapitel 3.

Abbildung 1: Organigramm des Testplans



# Test summary

All tests are performed on the recommended hardware specifications described below. The results are based on local communication – network traffic is excluded.

Table 1: Recommended system requirements

|  |  |
| --- | --- |
|  | **Recommended requirements** |
| **Operating system** | Windows 7, Linux or MacOS |
| **CPU** | Core 2 Duo at 2,6 GHz with VT-X |
| **Memory** | 2 GB RAM |
| **Hard drive** | At least <<XX>> GB free hard disk space |
| **Network** | 1 Gbit/s |
| **Internet** | Metis requires internet access to download the Docker images. After the build is done, internet connection isn’t required anymore. |

## Testing Scope

The following list shows the tested and not tested aspects of Metis software.

In scope

Functional Testing for the following modules are in scope of testing

* Database Connector
* File System Connector
* Search Engine Connector

Integration Testing are performed to proof the interaction of the application with the services

* Open search server
* Nginx webserver
* Mongodb
* ClientUI

Usability tests….<<RIMAC>>

Regression Testing

Out of scope

* Security and Penetration tests was not done for this application

## Unit/Module Test

These tests aim on the key functionalities of the product. According the “Qualitätshandbuch” they cover functional and nonfunctional Requirements. Like described in the Architecture Documentation, the software is split in modules. According to this, the tests are also split in these modules. Table 2 gives a distribution overview of the tests to the modules.

Each Testcase was built out of the underlying requirements specification. This approach ensures, that …<< sinnvoller satz mit äquivalenzklassenbildung hier>> .

Table 2: Distribution of unittests on modules

|  |  |
| --- | --- |
| **Module** | **# Tests** |
| Database Connector | 4 |
| File System Connector | 23 |
| Search Engine Connector | 14 |
| Service API | 27 |

Unit and Module Tests were performed automatically and continuously during the development of the software on every build by the build server. The test environment mirrors the production environment, which was specified in the requirements. In discharge of this, the integration of services like the open search server was tested early in the development.

## System Test

<<RIMACS SELENIUM>>

## Usability Test

<<RIMAC/MARCEL>>

## Performance Test

The performance of <<PRODUKTNAME>> was tested on every release with the Apache Benchmark tool ApacheBench. According the specification in the requirements document Metis is able to handle up to 1000 concurrent requests without losing significant performance - this is 50x more than specified.

## User Acceptance Test

<<????????>>

# Test results

## Unit/Module Test

Tester: R. Kettler; D. Weidle  
Release Version: 1.0

All test’s are executed using the Mocha testrunner for node.js. The following testreport is generated by Mocha’s spec reporter. The code tested is found in the git repository version tagged „1.0“. The following list gives an overview over all executed tests, passing and failing.

GET

✓ request with q=test (1150ms)

✓ request with q=asdfg

✓ request with q=

With Article:

/api/articles?q=

✓ request with q=test (48ms)

✓ request with q=

✓ request with unknown q (140ms)

✓ search in article (49ms)

✓ search in title (44ms)

✓ search author

✓ search author maxlength

✓ search author with spaces

✓ search author with keyword (66ms)

✓ search author with spaces with keyword

✓ search in file (74ms)

✓ search filename

✓ search filename with extension

/api/articles?ids=

✓ request with ids=ArticleIds[0]

✓ request with ids=[ArticleIds] (78ms)

✓ request with ids=

✓ request with ids=1,2,3

✓ request with valid invalid ids mixed (43ms)

/api/articles/:ArticleId

✓ request the articles just after save (80ms)

/api/articles/:ArticleId?old

✓ request the article old after save

POST

/api/articles/

✓ request with all valid information

With Article:

/api/articles/:articleId/documents

✓ upload json testfile

/api/articles/:articleId/documents

1) upload testfile 10 times to test replacement

PUT

With Article:

/api/articles/:ArticleId

✓ put with valid data

✓ put without data

✓ put with empty json

✓ put with string as data

✓ put with invalid json as data

email

✓ invalid

author

✓ to long

title

✓ to long

DELETE

With Article:

/api/articles

✓ delete existing article without files

/api/articles

✓ delete existing article with files

/api/articles

✓ delete not existing article

/:articleId/documents/:filename

✓ delete file from existing article

✓ delete article.html

✓ delete server.js

/:articleId/documents/:filename

✓ delete file from not existing article

Database

✓ findAllPermArticleIds

✓ deleteTemporaryArticlesOlderThan

✓ should fail for values equal 0

✓ should fail for values less 0

Filesystem

✓ deleteEmptyArticles (2002ms)

✓ deleteTemporaryArticles

file\_system\_connector

test extract html title content

✓ should pass

✓ should pass with empty title

✓ should pass with empty title immediately closed tag

✓ should pass with nested tags

✓ should pass with malformed tags

✓ should pass with malformed tags

✓ should pass without tags

✓ should pass two opening one closing tag

✓ should pass new line

test extract html body content

✓ should pass

✓ should pass empty body

✓ should pass regular text

✓ should pass missing closing tag

✓ should pass missing opening tag

✓ should pass missing tags

✓ should pass with escaped tags

✓ should pass with newline

test wrap content in html

✓ should pass

✓ should pass both params empty

✓ should pass both params whitespace

✓ should pass both params whitespace

To 1): The test „upload testfile 10 times to test replacement“ tries to upload a file 10 times concurrently. We rated this bug as low, becouse this scenario cannot occure as long as out browser client is used.

## System Test

28

<< RIMAC FRONTEND TEST>>

## Usability Test

<<RIMAC/MARCEL>>

## Performance Test

Performance Tests are performed on every release since Version 0.2 to make sure the major functionalities are working fine in defined conditions. Detailed reports can be found on appendix.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#ID** | **Description** | **Tester** | **Version** | **Pass/Fail** |
| P-01 | Handle 50 concurrent search- and 3 create requests and respond them within 2 seconds. | R. Kettler | 0.2 | Pass |
| P-02 | Handle 100 concurrent search- and 3 create requests and respond them within 2 seconds. | R. Kettler | 0.2 | Pass |
| P-03 | Handle 200 search- and 3 create requests and respond them within 2 seconds. | R. Kettler | 0.2 | Pass |
| P-04 | Handle 500 concurrent search- and 3 create requests and respond them within 2 seconds. | R. Kettler | 0.2 | Pass |
| P-05 | Handle 1000 concurrent request and respond them within 2 seconds. | R. Kettler | 0.2 | Pass |
| P-06 | Handle 50 concurrent search- and 3 create requests and respond them within 2 seconds. | R. Kettler | 1.0 | Pass |
| P-07 | Handle 100 concurrent search- and 3 create requests and respond them within 2 seconds. | R. Kettler | 1.0 | Pass |
| P-08 | Handle 200 concurrent request and respond them within 2 seconds. | R. Kettler | 1.0 | Pass |
| P-09 | Handle 500 concurrent search- and 3 create requests and respond them within 2 seconds. | R. Kettler | 1.0 | Pass |
| P-10 | Handle 1000 concurrent search- and 3 create requests and respond them within 2 seconds. | R. Kettler | 1.0 | Pass |

## User Acceptance Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #ID | Description | Tester | Version | Pass/Fail |
| UA-01 |  |  |  |  |
| UA-02 |  |  |  |  |

# Metrics

## Bug report

|  |  |  |  |
| --- | --- | --- | --- |
| **Test cases planned** | **Test cases executed** | **Passed** | **Failed** |
|  |  |  | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Low** | **Middle** | **High** | **Highest** |
| Closed | 1 | 43 | 2 | 1 |
| Open | 1 | 3 | 0 | 0 |

## List of open bugs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[KNOW-218]**[**Öffnen eines Artikels, der bereits gelöscht wurde, leitet auf eine leere Artikelseite ohne Fehlermeldung**](https://danielweidle.de/jira/browse/KNOW-218)Erstellt: 14/Jun/16  Aktualisiert: 14/Jun/16 | | | | |
| **Status:** | | Backlog | | |
| **Projekt:** | | [Knownana](https://danielweidle.de/jira/secure/BrowseProject.jspa?id=10000) | | |
| **Komponente(n):** | | [Frontend](https://danielweidle.de/jira/issues/?jql=project%3D10000%20AND%20%22component%22%3D10000%20ORDER%20BY%20priority%20ASC) | | |
| **betrifft Version(en):** | | 0.2 | | |
| **Lösungsversion(en):** | | 1.0 | | |
| **Typ:** | Bug | | **Priorität:** | Medium |
| **Autor:** | [Rimac Valdez](https://danielweidle.de/jira/secure/ViewProfile.jspa?name=rvaldez) | | **Bearbeiter:** | [Christoph Brutscher](https://danielweidle.de/jira/secure/ViewProfile.jspa?name=cbrutscher) |
| **Lösung:** | Nicht erledigt | | **Stimmen:** | 0 |
| **Beschreibung** | Der Benutzer erhält kein direktes Feedback, warum der Artikel leer ist.  Toast mit "No article found" oder ähnliches wäre gut. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[KNOW-217]**[**Der Inhalt von secured pdf dateien kann nicht geparst werden.**](https://danielweidle.de/jira/browse/KNOW-217)Erstellt: 14/Jun/16  Aktualisiert: 16/Jun/16 | | | | |
| **Status:** | | Backlog | | |
| **Projekt:** | | [Knownana](https://danielweidle.de/jira/secure/BrowseProject.jspa?id=10000) | | |
| **Komponente(n):** | | [Backend](https://danielweidle.de/jira/issues/?jql=project%3D10000%20AND%20%22component%22%3D10001%20ORDER%20BY%20priority%20ASC) | | |
| **betrifft Version(en):** | | 0.2 | | |
| **Lösungsversion(en):** | | 1.0 | | |
| **Typ:** | Bug | | **Priorität:** | Medium |
| **Autor:** | [Rimac Valdez](https://danielweidle.de/jira/secure/ViewProfile.jspa?name=rvaldez) | | **Bearbeiter:** | [Alex Schramm](https://danielweidle.de/jira/secure/ViewProfile.jspa?name=aschramm) |
| **Lösung:** | Nicht erledigt | | **Stimmen:** | 0 |
| **Beschreibung:** | Der Inhalt von secured pdf dokumenten kann nicht geparst werden. Daher kann der Inhalt der Dateien nicht über die Suche gefunden werden. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **[KNOW-210]**[**Die Oberfläche funktioniert nicht, wenn keine Internetverbindung vorhanden ist.**](https://danielweidle.de/jira/browse/KNOW-210)Erstellt: 14/Jun/16  Aktualisiert: 14/Jun/16 | | | |
| **Status:** | Backlog | | |
| **Projekt:** | [Knownana](https://danielweidle.de/jira/secure/BrowseProject.jspa?id=10000) | | |
| **Komponente(n):** | [Frontend](https://danielweidle.de/jira/issues/?jql=project%3D10000%20AND%20%22component%22%3D10000%20ORDER%20BY%20priority%20ASC) | | |
| **betrifft Version(en):** | 0.2 | | |
| **Lösungsversion(en):** | 1.0 | | |
| **Typ:** | Bug | **Priorität:** | Medium |
| **Autor:** | [Rimac Valdez](https://danielweidle.de/jira/secure/ViewProfile.jspa?name=rvaldez) | **Bearbeiter:** | [Christoph Brutscher](https://danielweidle.de/jira/secure/ViewProfile.jspa?name=cbrutscher) |
| **Lösung:** | Nicht erledigt | **Stimmen:** | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **[KNOW-211]**[**Darstellung Files**](https://danielweidle.de/jira/browse/KNOW-211) Erstellt: 14/Jun/16  Aktualisiert: 14/Jun/16 | | | |
| **Status:** | Selected for Development | | |
| **Projekt:** | [Knownana](https://danielweidle.de/jira/secure/BrowseProject.jspa?id=10000) | | |
| **Komponente(n):** | [Frontend](https://danielweidle.de/jira/issues/?jql=project%3D10000%20AND%20%22component%22%3D10000%20ORDER%20BY%20priority%20ASC) | | |
| **betrifft Version(en):** | 0.2 | | |
| **Lösungsversion(en):** | 1.0 | | |
| **Typ:** | Bug | **Priorität:** | Low |
| **Autor:** | [Daniel Weidle](https://danielweidle.de/jira/secure/ViewProfile.jspa?name=dweidle) | **Bearbeiter:** | [Yannic Soethoff](https://danielweidle.de/jira/secure/ViewProfile.jspa?name=ysoethoff) |
| **Lösung:** | Nicht erledigt | **Stimmen:** | 0 |
| **Beschreibung:** | Ist einem Artikel eine Datei angehängt, wird in der Vorschau ein Komma an den Dateinamen des hochgeladenen Files angehängt. | | |