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

**User Stories Document**

Knowledge-base System "Metis"

**Author**Marcel Hoffmann

# Table of contents

Table of contents 2

History 4

Preamble 5

1 functional User stories 6

2 Non-functional user stories 8

# History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 1.0 | 17.06.2016 | Initial document creation |
| 1.1 | 17.06.2016 | Review |

# Preamble

This document contains the user stories of Metis, a knowledge base system. These user stories have been identified with the aid of interviews and workshops with a representative of NTT Data Germany. Hence, the content of this document forms the basis of the requirements elicitation. It is used to infer use cases from which the requirements are formed from.

# Functional User Stories

As a user I must be able to create new articles in order to provide my know-how to other users.

As a user I must be able to read articles in order to acquire know-how for a certain issue.

As a user I must be able to edit articles in order to have the possibility to correct faults in articles.

As a user I must be able to upload files and documents in order to attach these to corresponding articles and thereby increase the shareability and understandability.

As a user I must be able to download documents attached to an article to save it on my local machine and being able to edit these.

As a user I must be able to search for articles in order to find relevant articles for my issue, based on the search term.

As a user I must be provided with the following meta information for each article:

* Name of the author
* Date and time of the last change
* Name of the last person who changed the article

As a user I must be able to format articles during creation and editing in order to increase understandability and readability.

* Font size, headline, bolt, italic, underlined
* Bullet list
* Tables

As a user I must be able to add pictures to articles in order delineate information for complex matters more accurately.

As a user I must be able to delete articles in order to remove obsolete information.

As a user I must be able to delete documents attached to an article in order to decrease information redundancy and avoid overhead.

As a user I would like to be able to use formatting templates during creation of an article to establish standardization easier.

As a user I would like to be able to use the system on my mobile device in order to be more flexible and facilitate the use of the product.

As a user I can contact the author or the last editor via email in order to indicate faults in his article.

As a user I can mark an article as helpful in order to emphasize this article for other users.

As a user I would like to access my last visited articles in order to save time and increase the business value of this system.

As a user I can access the previous version of an article in order to trace changes.

# Non-functional User Stories

As a user I would like to search articles in order to find relevant know-how quickly. Therefore, the system must be able to provide the search results within one second (server sided) searching time in order to not decelerate my work flow.

* search performance

As a user I would like to find articles in order to access information quickly. Therefore, the system should only provide relevant search results, based on the inserted search term in order to support my work flows efficiently.

* search reliability

As a user I would like to gain an additional benefit from this software in order to accelerate and simplify my work flows and to reach higher efficiency. Therefore, the system must be usable, that means easy to use and intuitive (training period < 15 minutes) in order to integrate the system quick and easy into my work flows.

* usability training period

As a user I would like to share and access know-how across departments in order to acquire internal know-how easier and more independent. Therefore, the system must hold the performance criteria described in paragraph search performance, even if there are up to 200 reading and 25 writing request taking place at the same time.

* scalability

As an administrator I would like to be able to replace the current search engine in order to establish new technologies and to simplify the maintainability of the system. Therefore the search engine must have loose coupling on the complete system in order to provide easy changeability.

* exchangeability