**3. Specific Requirements**

**3.1. Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Requirements | General information | Priority |
| **Provide automatic document validation** | **Description**  For providing a relevant result, we have to check input files for structural and syntax errors.  **Input**  XML-files provided by the user.  **Display**  Message window with validation result.  **System** **Processing**  Prepared XSD-schema in SOAP WSDL  **System** **Output**  Log message. | Critical |
| **Visualizing input (tree mode)** | **Description**  Visualizing input document in tree view using Angular JS framework.  **Input**  XML-files provided by user.  **Display**  RDF graph. (wrong) just a tree from the input  **System** **Processing**  Script for RDF graph construction.  **System** **Output**  Log message. | High |
| **Map each document topology onto each other and produce an integration file** | **Description**  Using SPARQL queries, ontologies and text mining perform document cross matching.  **Input**  XML-files, prepared beforehand ontologies.  **Display**  Operation status message and download url.  **System** **Processing**  SPARQL, regular expressions (text mining tools) with Java and JENA triple store on back end.  **System** **Output**  File, log message. | Very High |
| **Ability to test the output against the Gold Standard** | **Description**  Manual development of a Gold Input and Output files and testing tool for comparing teacher value with achieved one. Precision recall, matching quality.  **Input**  XML-files, processed output, Golden Standard.  **Display**  Mismatching node graphs if any.  **System** **Processing**  Strict matching and cross-checking of the resulting document with the manually developed.  **System** **Output**  Log messages. | Medium |
| **Ability to run SPARQL queries on top of the resulting file** | **Description**  There should be a web-based instrument to query a resulting file. The result of the query is highlighted on the tree.  **Input**  Sparql query  **Display**  The result of the query on the already visualized tree  **System** **Processing**  Executing a query using a document as a data-source  **System** **Output**  Log messages. | High |
| **Match quality visualization** | **Description**  There should be an option of visualizing the difference between the resulting file and the “gold” standard file.  **Input**  Either files (different level of maturity) or user action in session.  **Display**  Mismatching node graphs if any.  **System** **Processing**  Strict matching.  **System** **Output**  Log messages. | Medium |

**3.2. Non-functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Description** | **Priority** |
| **Provide security and fault tolerance** | **Description**  Penetration prevention, spoofing/sniffing protection, data privacy and protection.  **System** **Processing**  Implementation on server side, utilizing NGINX web-server and Tomcat servlet container. | Very high |
| **User-friendly interface** | **Description**  The system will be able to be visualized in different devices such as smart-phones, tablets and computers | Low |
| **Semantic fuzzy matching** | **Description**  To perform sufficient and determined level of matching with high level of quality enable “soft” or non-strict matching. | Low |
| **Time tolerant processing** | **Description**  Time optimization of performance. | Medium |
| **Scalability** | **Description**  The complexity of matching algorithm should not be exponential in order for the rising amount of incoming data to perform rapid and time-efficient processing. Should be tested using different file sizes and multiple runs (load test) | Medium |

**3.3. Use-case specifications**

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Upload and validate document | | |
| Description | A user should be able to upload a document and get a message of validation result | | |
| Preconditions | The user is logged in the system and is located in the main interface | | |
| Postcondition | A document is loaded into current session, and is checked for consistency and structure. | | |
| Main course |  | **Actor input** | **System** |
| 1 | The user press the “Upload file” button |  |
| 2 |  | The system shows a file explorer to the user |
| 3 | The user chooses a file |  |
| 4 |  | The system validates the document |
| 5 |  | The document is saved |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | **Document visualization** | | |
| Description | Visualize a document in a tree view. The user should be able to navigate the tree with the ability to show and hide nodes | | |
| Preconditions | The user is logged in the system and is located in the main interface.  The user has loaded a file, and a file proceeded through verification | | |
| Postcondition | An interactive tree appears on the main page | | |
| Main course |  | **Actor input** | **System** |
| 1 | The user press the button “Visualize” |  |
| 2 |  | Visualizes a tree-view for a document |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | **Map each document topology onto each other and produce an integration file** | | |
|  |  | | |
| Description |  | | |
| Preconditions | The user is logged in the system and is located in the main interface having already attached two files, that were already verified | | |
| Postcondition | Result file appears in current user session. Available for download by user. | | |
| Main course |  | **Actor input** | **System** |
| 1 | The user presses the button to generate a result file |  |
| 2 |  | The system generates a file |
| 4 |  | The system visualizes the file in tree view |
| 5 |  | The document is saved in session |
| Title | **Ability to test the output against the Gold Standard** | | |
|  |  | | |
| Description | A user should be able to compare a system-generated result file and a handmade one. | | |
| Preconditions | The user is logged in the system and is located in the main interface.  A resulting file is already in active session | | |
| Postcondition | A comparison is provided to user in percentage | | |
| Main course |  | **Actor input** | **System** |
| 1 | The user press the “Upload “Gold” standard” button |  |
| 2 |  | The system shows a file explorer to the user |
| 3 | The user chooses a file |  |
| 4 |  | The system validates the document |
| 5 |  | The system provides analysis and visualizes the results |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | **Ability to run SPARQL queries on top of the resulting file** | | |
|  |  | | |
| Description | A user should be able to call a window for SPARQL input in order to get values from resulting document. | | |
| Preconditions | The user is logged in the system and is located in the main interface  The result document is in active session | | |
| Postcondition | A result document is highlighted according to query result | | |
| Main course |  | **Actor input** | **System** |
| 1 | The user presses “RunSPARQL” button |  |
| 2 | The user puts in query text and runs it | The system check user input for syntax errors |
| 3 |  | The system performs query operation |
| 4 |  | The system adds visualization for the results on existing tree |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | **Match quality visualization** | | |
|  |  | | |
| Description | A user should be able to visualize the discrepancies between the generated and pattern files | | |
| Preconditions | The user is logged in the system and is located in the main interface  Pattern file is attached in session. Result file is stored in session | | |
| Postcondition | A table of disparities is shown to user | | |
| Main course |  | **Actor input** | **System** |
| 1 | The user press the “Visualize difference” button |  |
| 2 |  | The system provides comparing between the resulting file and the pattern |
| 3 |  | The system show the output in table format |