**System Test Specification (STS)**

1. **Introduction**

a) Background

The System described in System Requirement Specification needs tests to prove its stability and correct performance. The results will be the subject of software testing documentation and base for further development and use of external projects.

b) Purpose

The System test Specification for the project will verify that the functionality of the Report Management System (RMS) meets the requirements as documented in the System Requirements Specification. It provides a detailed description of each test. Each requirement from the STS includes a test name, purpose, description of the initial state, test procedure and overview of expected results.

Tests are to be executed in order to demonstrate the correct behaviour of the system. Each test traces to the components of the Formal Design that it demonstrates. Analysis of the tracing then allows a demonstration that all aspects of the Formal Design have been covered by these tests.

1. **Testing approach**

Tested aspects:

* Graphical user interface (GUI)
* Database interact
* Server tests

Representative (GUI) tests shall release by human reception in browser.

Database and server tests used for the project shall be created by means of Eiffel Studio embedded tool with respect to the requirements specification. Tests shall be stored in the separated cluster in order to prevent their delivery with the application code. Part of them (especially checking database) should make by SQLite console.

1. **Tests cases**
2. **Graphical user interface**

|  |  |
| --- | --- |
| Test name | Section traverse correctness |
| Purpose | All links on homepage must implement correct web page transition. |
| Initial State | Homepage of RMS website opened. |
| Test Procedure | Visit all sections and check their correctness and their content. Content means fields that page allow you to fill and send on server. |
| Expected results | All links are correct and sections deliver expected information fields. |

|  |  |
| --- | --- |
| Test name | Comfortable filling of fields |
| Purpose | While user write his text fields should perform predictable reflection as auto increase text area when borders are reached, go to next field after TAB button. |
| Initial State | Every section page opened. |
| Test Procedure | Print massive text (more than 40 symbols) in every field and change by different ways selected field. |
| Expected results | Incorrect and strange page behavior not reached. |

|  |  |
| --- | --- |
| Test name | Sending complete section information on server |
| Purpose | After section fields are completed user allow to send results on server. |
| Initial State | Any section web page is opened. All necessary fields are filled in. |
| Test Procedure | Try to cheat site and send fields information without its completeness as example - to send answer on server with one or more uncompleted field. Fill field and then delete all result then try to send answer. |
| Expected results | Page not allow to send answer on server without required fields are completed fully on moment of sending. |

|  |  |
| --- | --- |
| Test name | Filled information lost protect |
| Purpose | When user in report complete process he able to accidentally close this page and lost information. |
| Initial State | Section page is opened. Something fields are completed. |
| Test Procedure | Try to close page when something fields on your page are filled. |
| Expected results | Page must invoke floating window and ask your attempting to leave this page and provide information that unsent information will lost. |

|  |  |
| --- | --- |
| Test name | Traverse by incorrect link |
| Purpose | If user try to access unreached on our service page (or page which he has not access) he must reach error page. |
| Initial State | Home page is opened. |
| Test Procedure | Try to add or change current URL to have access admin area or not existed page. |
| Expected results | User must meet error page |

|  |  |
| --- | --- |
| Test name | Correct database web performance |
| Purpose | On administrator page must locate correct information from database. |
| Initial State | Server running. |
| Test Procedure | Visit administrator page and compare information received from server with information keeping in database tables |
| Expected results | Information on web page must duplicate information in database. |

**b) Database interaction**

|  |  |
| --- | --- |
| Test name | Database connection test |
| Purpose | Connect to database can be not creatable. |
| Initial State | Server running. |
| Test Procedure | Realise basic operations in database (Create, Read, Update, Delete) with testing data. |
| Expected results | Database tables change their state as expected. |

|  |  |
| --- | --- |
| Test name | Rewrite database collision information |
| Purpose | If section field information from current user submit is actually in database (database table collision reached) in that case server should save exactly last submit information. |
| Initial State | Database has not empty field with something key (id) KEY. |
| Test Procedure | Make server send different information with same KEY in Database. |
| Expected results | Field with key KEY must contain the last sent information. |

|  |  |
| --- | --- |
| Test name | Database protect |
| Purpose | Attackers would like to destroy or modify service database. |
| Initial State | Server running. |
| Test Procedure | By SQL injections or other ways try to get access to database or modify existing information. |
| Expected results | Database accept only described in SRS operations from predicted web pages. |

|  |  |
| --- | --- |
| Test name | Privacy test |
| Purpose | User can not allow access to information of other users. |
| Initial State | Database has information of 2 or more users. |
| Test Procedure | By any way try to get information as an identified user from database that not allowed to this user. |
| Expected results | Can’t reach illegal information. |

**c) Server tests**

|  |  |
| --- | --- |
| Test name | Submission parsing |
| Purpose | Information reached by server must correctly be sent in database. |
| Initial State | Server reached request. |
| Test Procedure | After adding submit information into database table check new rows and compare them with request fields. |
| Expected results | Submitted information equals information which currently stored in database. |

|  |  |
| --- | --- |
| Test name | Correct information received |
| Purpose | Server get exactly equally information as user sent. |
| Initial State | Web page section has complete fields. |
| Test Procedure | Send information from web page on server. Take information on server and compare its accuracy. |
| Expected results | Received information is fully equals to sent information. |

|  |  |
| --- | --- |
| Test name | Correct page perfomance return |
| Purpose | Server keep templates of web pages and must send them to user in designed condition. |
| Initial State | Server running. |
| Test Procedure | Check web page source received from server. As the way is to save web page on computer and look files structure. |
| Expected results | Page respond by browser must contain html file, style files and script files. |