Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**System Test Plan**

(TINF21C, SWE)

**Project:** Modelling Wizard Improvements

**Customer:** Markus Rentschler

Christian Holder

**Team:** Project Manager – Robin Ziegler ([inf21100@lehre.dhbw-stuttgart.de](mailto:inf21100@lehre.dhbw-stuttgart.de))

Developer – Nils Hoffmann ([inf21194@lehre.dhbw-stuttgart.de](mailto:inf21194@lehre.dhbw-stuttgart.de))

Test Manager – Michael Grote ([inf21111@lehre.dhbw-stuttgart.de](mailto:inf21111@lehre.dhbw-stuttgart.de))

System Architect – Fabian Kreuzer ([inf21106@lehre.dhbw-stuttgart.de](mailto:inf21106@lehre.dhbw-stuttgart.de))

Tech. Documentation – Dana Frey ([inf21099@lehre.dhbw-stuttgart.de](mailto:inf21099@lehre.dhbw-stuttgart.de))

Developer – Sophie Kirschner ([inf21083@lehre.dhbw-stuttgart.de](mailto:inf21083@lehre.dhbw-stuttgart.de))

Product Manager – Maximilian Trumpp ([inf21123@lehre.dhbw-stuttgart.de](mailto:inf21123@lehre.dhbw-stuttgart.de))

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 11.03.2023 | Michael Grote | created |
| 0.2 | 16.03.2023 | Michael Grote | Add features |
| 0.3 | 04.04.2023 | Michael Grote | Add testcases |
| 0.4 | 16.04.2023 | Michael Grote | Add testcases |
| 0.5 |  |  | Add file paths |

Table of Contents

[1. Scope 1](#_Toc132661392)

[2. Glossary 1](#_Toc132661393)

[3. Product Names and Attributes 1](#_Toc132661394)

[4. Features 1](#_Toc132661395)

[5. Test Preparation Strategy 2](#_Toc132661396)

[6. Test Execution Strategy 2](#_Toc132661397)

[7. Test Equipment 2](#_Toc132661398)

[8. Test Schedule and Budget 2](#_Toc132661399)

[9. Test Planning 3](#_Toc132661400)

[10. References 3](#_Toc132661401)

[11. Appendix: Testcases 4](#_Toc132661402)

[<TS-001> User interaction with Data 4](#_Toc132661403)

[<TC-001-001> (Create new device) 4](#_Toc132661404)

[<TC-001-002> (Edit attribute of device) 4](#_Toc132661405)

[<TC-001-003> (Change the theme) 5](#_Toc132661406)

[<TC-001-004> (Load a new valid library) 5](#_Toc132661407)

[<TS-002> File operations 6](#_Toc132661408)

[<TC-002-001> (Loading of a valid file with validation) 6](#_Toc132661409)

[<TC-002-002> (Export of a valid device in a file with validation) 6](#_Toc132661410)

[<TS-003> Error Handling 7](#_Toc132661411)

[<TC-003-001> (Loading of an invalid file with validation) 7](#_Toc132661412)

[<TC-003-002> (Export of an invalid device in a file with validation) 7](#_Toc132661413)

[<TC-003-003> (Load new invalid library) 8](#_Toc132661414)

[Figures 9](#_Toc132661415)

# Scope

This STP (System Test Plan) specifies the test strategy and test planning. It references test to be performed to verify the accordance of the demanded features given by the SRS (System Requirements Specification) to the implemented features. The document derived from the STP in the STR (System Test Report) where additionally the results are given.

# Glossary

**TC** Testcase

**TS** Test suite

**TD** Test data

**GUI** Graphical User Interface

# Product Names and Attributes

The following test objects must be verified:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.-Id.** | **Product Number** | **Product Name** | **Product Description** |
| 1 | Build v1.0 | Standalone Modelling Wizard | Windows Standalone Application with a GUI |

# Features

The following requirements must be verified, as long as they are not classified as “not to be tested”. This table shows the test coverage between functionality and test suite.

|  |  |  |  |
| --- | --- | --- | --- |
| **Req.-ID** | **Functionality** | **Priority** | **Test suite** |
| LF10: Import | Imports file by absolute path | A | TS-002 |
| LF20:  File validation | Detect wrongly formatted imported files and throw an error to the user | B | TS-002,  TS-003 |
| LF30:  Error handling | Handle errors (e.g. wrong user input) | A | T-S003 |
| LF40:  GUI | Display a GUI and accept user input. | A | TS-001 |
| LF50:  Display device in a readable way | Display the attributes of selected device. | A | TS-001 |
| LF60:  Edit device | Edit the attributes of a device. | A | TS-001 |
| LF70:  Create device | Create an own device by the imports from the libraries. | A | TS-001 |
| LF80:  Export device | Save created device into file. | A | TS-002 |

# Test Preparation Strategy

The creation of tests will be application case based. Two main application cases can be identified, the new GUI and file operations.

The GUI is the first main application case. The GUI provides a view of the loaded device with input fields in which the respective device data is displayed. These fields must be checked and features to edit and save device must be validated.

File operations represent the second main application case. Device files need to be loaded, validated and saved to ensure full functionality of the application for the user.

# Test Execution Strategy

Most functions of the application had been changed and should be tested. Because it is not possible to test all cases, we test the main functions. The test is divided into the following phases:

1. Graphical User Interface
2. File operations

Since the user interaction are needed for the application to work, these must be tested first. This includes the start of the program and the execution of the main features of the application in the GUI.

Then the file operations can be tested. The created devices should be exported, and existing devices should be imported.

# Test Equipment

The following equipment must be available for testing:

* A computer with Windows 10 or higher
* The standalone Device Modelling Wizard software

# Test Schedule and Budget

The testing of the application begins as soon as the application is completed. This makes it possible to make the necessary corrections quickly.

No budget is needed for the tests, as they are all performed by hand. Only the tester needs to get paid.

# Test Planning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testsuite** | **Test objective** | **Testplan Creator** | **Testplan Reviewer** | **Tester** |
| TS-001 | User interaction with Data | Michael Grote |  |  |
| TS-002 | File operations | Michael Grote |  |  |
| T-S003 | Error Handling | Michael Grote |  |  |

# References

The Software Requirements Specification (SRS) can be found in Github. (See <https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/wiki/Software-Requirements-Specification-%5BSRS%5D>)

# Appendix: Testcases

## User interaction with Data

### (Create new device)

|  |  |  |  |
| --- | --- | --- | --- |
| Testcase-ID: | | TC-001-001 | |
| Testcase-Name: | | Create new device | |
| Req.-ID: | | LF40, LF50, LF60, LF70 | |
| Description | | The test case verifies that the application can create a new device. | |
|  | |  | |
| Test Steps | | | |
| Step | Action | | Expected result |
| 1 | Open the Application Modelling Wizard. | | Application starts without problems. |
| 2 | Create a new device by clicking “New”. | | In “Generic Data” should be one device. |
| 3 | Change the mode from “Easy mode” to “Advanced mode”. Click “Options” in the menu and “Advanced Mode”. | | There should be displayed more configurations for the attributes and the label under “Option” should change to “Easy Mode”. Also, there should be a new button in “Generic Data” with the label “Add Role Class”. |
| 4 | Click on the “Add Role Class”-Button in “Generic Data”. Check one device and then click “Add”. | | A dialog should be opened and after clicking on “Add” the selected devices should be added into “Generic Data” and the dialog should be closed. |

### (Edit attribute of device)

|  |  |  |  |
| --- | --- | --- | --- |
| Testcase-ID: | | TC-001-002 | |
| Testcase-Name: | | Edit attribute of devices | |
| Req.-ID: | | LF40, LF50, LF60 | |
| Description | | The test case verifies that the application can create a new device. | |
|  | |  | |
| Test Steps | | | |
| Step | Action | | Expected result |
| 1 | Open the Application Modelling Wizard. | | Application starts without problems. |
| 2 | Create a new device by clicking “New”. | | In “Generic Data” should be one device. |
| 3 | Open the first device in Generic Data. Now change some data in the right grid. Then select another attribute in the grid. | | The user input should be accepted and displayed correctly. |

### (Change the theme)

|  |  |  |  |
| --- | --- | --- | --- |
| Testcase-ID: | | TC-001-003 | |
| Testcase-Name: | | Change the theme | |
| Req.-ID: | | LF40, LF50 | |
| Description | | The test case verifies that the application can change the colour theme. | |
|  | |  | |
| Test Steps | | | |
| Step | Action | | Expected result |
| 1 | Open the Application Modelling Wizard. | | Application starts without problems. |
| 2 | Click on the “Options” in the menu bar and click on “Darkmode” or “Lightmode” depends on your actual theme. | | The theme of the application should be changed. |
| 3 | Open the “Generic Data”, “Interfaces” and “Attachments” and compare the design with the expected design pattern. | | The colour theme should be used in every menu. In Darkmode the text colour should be white and the background light grey. In Lightmode the background should be white and the text colour should be black. |

### (Load a new valid library)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase-ID: | | | TC-001-004 | |
| Testcase-Name: | | | Load a new valid library | |
| Req.-ID: | | | LF40, LF70 | |
| Description | | | The test case verifies that the application can load a new aml-library. | |
|  | | |  | |
| Test Steps | | | | |
| Step | Action | | | Expected result |
| 1 | Open the Application Modelling Wizard. | | | Application starts without problems. |
| 2 | Click “Libraries” in the menu bar, then “Add library”. | | | The explorer should be opened. |
| 3 | Select the library file to import it. | | | The library should be loaded into the Application and displayed in the list under “Libraries” in the menu bar. |
|  | |  | |  |
| Test data: TD-001-004 | | | | |
| Dataset | | File | | Validation |
|  | |  | |  |

## File operations

### (Loading of a valid file with validation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase-ID: | | | TC-002-001 | |
| Testcase-Name: | | | Loading of a valid file with validation | |
| Req.-ID: | | | LF10, LF20, LF50 | |
| Description | | | The test case verifies that the application can load a valid file. | |
|  | | |  | |
| Test Steps | | | | |
| Step | Action | | | Expected result |
| 1 | Open the Application Modelling Wizard. | | | Application starts without problems. |
| 2 | Select a valid input file for the validation, by selecting “Open”, and then choose the file in explorer. | | | The validation is executed successfully, and the file is loaded completely and correctly without error message. |
| 3 | Check that the data has been interpreted correctly Compare the expected data with the data in “Generic Data” and their “Attributes”, also compare the data in “interfaces”. | | | All data should be displayed readable and correctly. |
|  | |  | |  |
| Test data: TD-002-001 | | | | |
| Dataset | | File | | Validation |
| 1 | |  | | valid |
| 2 | |  | | valid |

### (Export of a valid device in a file with validation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase-ID: | | | TC-002-002 | |
| Testcase-Name: | | | Export of a valid device in a file with validation | |
| Req.-ID: | | | LF10, LF20, LF50, LF60, LF80 | |
| Description | | | The test case verifies that the application can export a valid file. | |
|  | | |  | |
| Test Steps | | | | |
| Step | Action | | | Expected result |
| 1 | Open the Application Modelling Wizard. | | | Application starts without problems. |
| 2 | Select a valid input file for the validation, by selecting “File”, and then “Open” and choose the file in explorer. | | | The validation is executed successfully, and the conversion is completed correctly without error message. |
| 3 | Edit some attributes in “Generic Data”, be sure to add some new attributes and change some old ones. | | | Changes are displayed correctly and an icon popup to show unsaved changes. |
| 4 | Click on “File” and select “Save”, select location in the file explorer and save file. | | | Valid file can be saved without errors and filename is generated automatically by device name and manufacturer. |
| 5 | Open the new file in the Application and check if the changes were applied correctly and the file is still valid. | | | File opened without error and changes of the attributes are displayed correctly. |
|  | |  | |  |
| Test data: TD-002-002 | | | | |
| Dataset | | File | | Validation |
|  | |  | |  |

## Error Handling

### (Loading of an invalid file with validation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase-ID: | | | TC-003-001 | |
| Testcase-Name: | | | Loading of an invalid file with validation | |
| Req.-ID: | | | LF10, LF20, LF30 | |
| Description | | | The test case verifies that the application can validate a file and find invalid files by opening. | |
|  | | |  | |
| Test Steps | | | | |
| Step | Action | | | Expected result |
| 1 | Open the Application Modelling Wizard. | | | Application starts without problems. |
| 2 | Select an invalid input file for the validation, by selecting “Open”, and then choose the file in explorer. | | | The validation is executed successfully, and the file is not loaded. The User get a message about the error and the reason. |
|  | |  | |  |
| Test data: TD-003-001 | | | | |
| Dataset | | File | | Validation |
| 1 | |  | | invalid |
| 2 | |  | | invalid |

### (Export of an invalid device in a file with validation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase-ID: | | | TC-003-002 | |
| Testcase-Name: | | | Export of an invalid device in a file with validation | |
| Req.-ID: | | | LF10, LF20, LF30, LF60, LF80 | |
| Description | | | The test case verifies that the application can handle a wrong input from the user and that it informs the user right. | |
|  | | |  | |
| Test Steps | | | | |
| Step | Action | | | Expected result |
| 1 | Open the Application Modelling Wizard. | | | Application starts without problems. |
| 2 | Select a valid input file for the validation, by selecting “File”, and then “Open” and choose the file in explorer. | | | The validation is executed successfully, and the conversion is completed correctly without error message. |
| 3 | Delete the manufacturer of the first element in “Generic Data”. | | | The attribute name should be displayed in red. |
| 4 | Click on “File” and select “Save”, select location in the file explorer and save file. | | | The file cannot be saved, and the application should show an message with the reason. |
|  | |  | |  |
| Test data: TD-003-002 | | | | |
| Dataset | | File | | Validation |
|  | |  | |  |

### (Load new invalid library)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase-ID: | | | TC-003-003 | |
| Testcase-Name: | | | Load a new invalid library | |
| Req.-ID: | | | LF30, LF40 | |
| Description | | | The test case verifies that the application can find an invalid aml-library. | |
|  | | |  | |
| Test Steps | | | | |
| Step | Action | | | Expected result |
| 1 | Open the Application Modelling Wizard. | | | Application starts without problems. |
| 2 | Click “Libraries” in the menu bar, then “Add library” | | | The explorer should be opened. |
| 3 | Select the invalid library file to import it | | | The library should not be loaded into the Application. There should be message about the failed loading and the reason for it. |
|  | |  | |  |
| Test data: TD-003-003 | | | | |
| Dataset | | File | | Validation |
|  | |  | |  |

# Figures

[Figure 1 – MVC Architecture 8](#_Toc102130525)

[Figure 2 - Logic of the plugin 9](#_Toc102130526)

[Figure 3 - Programme Concept 10](#_Toc102130527)

[Figure 4 - Class design from the main application 10](#_Toc102130528)

[Figure 5 - Class design from the Modelling Wizard 11](#_Toc102130529)

[Figure 6 - MVC pattern 11](file:///C:/Users/Lukas_Ernst/Downloads/TINF18C_SAS_Team_3_v1.0.docx#_Toc102130531)