System Test Plan

TINF22F, Software-Engineering |

Practical Project 2023/2024

<u>Project:</u> AAS Webclient

<u>Customer:</u> Markus Rentschler, Ivan Bogicevic

Rotebühlstraße 133, 70197 Stuttgart

<u>Project Information:</u>

Project Leader: Ilir Emini (inf22198@lehre.dhbw-stuttgart.de)

Product Manager: David Bauer (inf22167@lehre.dhbw-stuttgart.de)

System Architect: Lara Lorke (inf22132@lehre.dhbw-stuttgart.de)

System Architect: Armin Taktar (inf22148@lehre.dhbw-stuttgart.de)

Documentation: Ümmühan Ay (inf22124@lehre.dhbw-stuttgart.de)

Developer: Rafael Sancho Pernas (inf22078@lehre.dhbw-stuttgart.de)

Test Manager: Kyle Zieher (inf22210@lehre.dhbw-stuttgart.de)

<u>Version History:</u>

Version	Date	Author	Comments
0.1	29.04.2024	Kyle Zieher	Initial draft with table of contents and basic structure
0.2	05.05.2024	Kyle Zieher	Added detailed test cases and strategies
1.0	14.05.2024	Kyle Zieher	Finalized the documentation

Content

1. Introduction

The AAS-Webclient is desigend to be a comprehensive solution for managing and interacting with Asset Administration Shells (AAS) in a network environment. The purpose of this System Test Plan (STP) is to outline the approach, coverage, methods and schedules that will be used in the system testing of the AAS-Webclient. This document will serve as a blueprint to ensure that the application meets all its technical and functional requirements as specified in the project documentation.

1.1. Purpose

The purpose of this document is to summarize the result of the system testing phase of the AAS-Webclient project. It provides an overview of the testing activities, test results and the quality of the system being tested.

1.2. Scope

This STP covers the following aspects:

Functional testing to verify each function of the software as the SRS.

Usability testing to ensure the interface is intuitive and conforms to specified user requirements.

1.3. Product names and attributes

The following test object will be examined:

Ref.-Id	Product Number	Product Name	Product Description
1	Build v1.0 (commit d5094d5)	AAS-Webclient	User friendly web-tool for displaying AAS-Servers and its contents while providing filter and search functionalities

2. Features

The following requirements must be verified through the testing process. The table below shows the mapping between the requirements from the SRS and the according Test suites or Test cases.

Requirement ID	Functionallity	Priority	Test case ID
REQ10: Server Data Display	Displays the right data from the AAASX-Server in a meaningful way	A	001-TC-ASSET-SERVER
REQ20: Error Handling	Handle and report errors like server not available, server not found, etc.	B	002-TC-ASSET-ERROR
REQ30: Support of Server browsing and direct URL-call	Download information for a selected asset	A	003-TC-ASSET-BROWSING
REQ40: User-/Expert-Mode	Switch between normal information or detailed information of a selected asset	A	004-TC-ASSET-MODE
REQ50: Hierarchical Structures enabling Bills of Material	Shows sub models for each asset	A	005-TC-ASSET-BOM
REQ60: Backend server connection	Runs the application without a server connection	A	006-TC-ASSET-BACKEND

3. Overall strategy and approach

3.1. Testing strategy

The test strategy for the AAS-Webclient involves a combination of Requirements-Based Testing and Exploratory testing to cover all features and functionalities of the software. The strategy is outlined with the intent to detect defects in the testing cycle, ensuring good software quality and meet the set requirements.

3.2. System testing entrance criteria

In order to start system testing, the software has to be signed off as ready by the development team.

3.3. Test preparation

Two main application cases can be identified: server management, and asset exploration.

Server management is about being able to add AAS-Servers to the application which are then used to fetch data. Those servers should also be able to be switched out as needed. The system should be able to detect false server-URLs and throw an error, whenever a server cannot be reached. It should be ensured, that all assets within the server are actually loaded.

Asset exploration involves all steps that involve browsing through the assets that appear as soon as the server has connected successfully. The user should then be able to scroll through the list of assets and see their names as well as a preview image. If needed, the user should also be able to search for assets by their name, filter for specific manufacturers and sort the assets by year. The user should be able to select a certain asset by clicking on it and then see a detailed view with all information that is available about it. This should also include all the sub-models.

3.4. Testing Types

3.4.1. Usability testing

User interface, cosmetic presentation and content will be tested for accuracy and general usability. The goal of usability testing is to ensure consistent and appropriate access and navigation to the different functionalities of the application.

3.4.2. Functional testing

The goal of functional testing is to ensure that all requirements and functions that were specified in the SRS are working accordingly to the expectations. This refers to functional requirements, business goals or conditions as well as other requests that were coming into scope throughout the course of this project (like resolutions of issues, change requests or feedback)

3.5. Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

3.5.1. Suspension Criteria

Testing will be suspended if an incident is found that does not allow further testing of the application. If testing is stopped and changes are made to the frontend or backend, they will be paused until the test manager decides that the resumption requirements are met.

3.5.2. Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been fixed and retested successfully. In this case the whole system test plan is re-executed to ensure that the bug-fixing did not cause any other problems.

4. Test equipment

The following prerequisites are required for testing:

- A computer with a browser supporting the HTML5 standards
- A working internet connection

5. Test data

All the data for the application is provided by AAS-Servers through REST- calls. This will be also the fact for the test conduction. The only input that is to be made by the user is therefore the json file to start the application if the server is currently not running.

6. Test specifications

6.1. Naming Conventions

Testcases and test suites will be named after the following structure:

Test case = **SEQNR** -TC-**TCFUNC**
SEQNR = sequential numbering of testcase (001)
TCFUNC = Abbreviation for the related requirement

6.2. Test cases <TC-ASSET>

6.2.1. Test case <001-TC-ASSET-SERVER>

TEST CASE	
ID:	001-TC-ASSET-SERVER
NAME:	Server Data Display
REQ.-ID:	REQ10
DESCRIPTION:	The test case verifies the correct functionality of the connection between AASX-Server and the AAS-Webclient. It display an error message in case the given URL is incorrect. The test set up consists of a computer, with an active internet connection and a working browser.

TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Click on the button "Server Menu" in the upper right corner	Dropdown server menu opens, providing an input field for a server URL as well as some predefined AAS-server URLs
2	Click on one of the predefined Server URLs	The selected URL is displayed as the current server and its assets are loaded on the left side of the screen (names and preview images)
PASSED		

6.2.2. Test case <002-TC-ASSET-ERROR>

TEST CASE	
ID:	002-TC-ASSET-ERROR
NAME:	Error Handling
REQ.-ID:	REQ20
DESCRIPTION:	The test case verifies code functionality and quality along with better user experience. It verifies the display of an error message in case the server is not available. The test set up consists of a computer, with an active internet connection and a working browser.
TEST STEPS	

STEP	ACTION	EXPECTED RESULT
1	Click on the button "Server Menu" in the upper right corner	Dropdown server menu opens, providing an input field for a server URL as well as some predefined AAS-server URLs
2	Click on one of the predefined Server URLs	The selected URL is not reachable and the server data can't be accessed. It displays the error: "Server not available".
PASSED		

6.2.3. Test case <003-TC-ASSET-BROWSING>

TEST CASE	
ID:	003-TC-ASSET-BROWSING
NAME:	Support of Server browsing and direct URL-call
REQ.-ID:	REQ30
DESCRIPTION:	The test case verifies the successful download of all the asset information of the selected asset. The test set up consists of a computer, with an active internet connection and a working browser.

TEST STEPS		
STEP	ACTION	EXPECTED RESULT
1	Select "Expert Mode".	Toggled switch "Expert Mode".
2	Select an asset listed from the left side.	The selected asset is shown with detailed information.
3	Click on the button "JSON download".	The asset with all its information should be downloaded. Visible on the top right side of your browser.
PASSED		

6.2.4. Test case <004-TC-ASSET-MODE>

TEST CASE	
ID:	004-TC-ASSET-MODE
NAME:	User-/Expert-Mode
REQ.-ID:	REQ40
DESCRIPTION:	The test case verifies the functionality of the "User Mode" and "Expert Mode" switch on the AAS-Webclient Navbar. It should validate the difference of shown information for each mode. The test set up consists of a computer, with an active internet connection and a working browser.

TEST STEPS		
STEP	ACTION	EXPECTED RESULT
1	"User Mode" is set as default. Click on an asset.	Normal information of the selected asset is shown.
2	Select "Expert Mode" and click on the same asset or any other asset again.	Detailed information is shown of the selected asset.
PASSED		

6.2.5. Test case <005-TC-ASSET-BOM>

TEST CASE	
ID:	005-TC-ASSET-BOM
NAME:	Hierarchical Structures enabling Bills of Material
REQ.-ID:	REQ50
DESCRIPTION:	The test case verifies the correct visualization of the sub models of a selected asset. The test set up consists of a computer, with an active internet connection and a working browser.

TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Select "Expert Mode" and select an asset.	Switch changes to "Expert Mode" and the asset shows detailed information.
2	Click on the "Assetsubmodels" button.	The dedicated sub models of the selected asset should be shown in numerical order with a link.

PASSED

6.2.6. Test case <006-TC-ASSET-BACKEND>

TEST CASE	
ID:	006-TC-ASSET-BACKEND
NAME:	Backend server connection
REQ.-ID:	REQ60
DESCRIPTION:	The test case verifies the use of the application without a server connection to the AASX-Server. The test set up consists of a computer, with an active internet connection and a working browser.

TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Click on the button "Server Menu" in the upper right corner	Dropdown server menu opens, providing an input field for a server URL as well as some predefined AAS-server URLs
2	Click on the last URL in the list where it says "localhost"	The selected URL is displayed as the current server and its assets are loaded on the left side of the screen (names and preview images). Its imitating the real Server just for demonstration purposes because the AASX-Server doesn't run successfully.

PASSED