# System Requirements Specification

TINF21C, Software Engineering I

Practical project 2022/23

## *Project*

AAS-Web client

## *Customer*

Markus Rentschler, Christian Holder

Rothebühlplatz 41, 70178 Stuttgart

## *Supplier*

Project Leader: Samara Dominik (inf21001@lehre.dhbw-stuttgart.de)

Product Manager: Martin Rittmann (inf21157@lehre.dhbw-stuttgart.de)

System Architect: Marcel Hintze (inf21056@lehre.dhbw-stuttgart.de)

Test-Manager: Anja Niedermeier (inf21097@lehre.dhbw-stuttgart.de)

Developer: Severin Helms (inf21047@lehre.dhbw-stuttgart.de)

Technical documentor: Tom Engelmann (inf21010@lehre.dhbw-stuttgart.de)

| *Version* | *Date* | *Author* | *Comment* |
|---|---|---|---|
| 0.1 | 01.11.2022 | S. Helms | Created Document, added structure |
| 0.2 | 04.11.2022 | S. Helms | Product requirements added |
| 0.3 | 06.11.2022 | S. Helms | Product requirements added |
| 0.4 | 03.05.2023 | S. Helms | Added final requirements |
| 0.5 | 3.05.2023 | S. Helms | Added Analysis |

Purpose ... 3
Product Environment ... 3
Use Cases ... 4
<UC.001>: Add AAS-Server ... 4
<UC.002>: Browse through digital twins ... 5
<UC.003>: Display digital twin ... 6
<UC.004>: Search for digital twin ... 7
<UC.005>: Filter for manufacturer ... 8
<UC.006>: Sort by year ... 9
Product Requirements ... 10
/REQ10/ Import Server ... 10
/REQ20/ Server validation ... 10
/REQ30/ Error handling ... 10
/REQ40/ Display content in a clear way ... 10
/REQ50/ Display digital twin ... 10
/REQ60/ Sort by year ... 10
/REQ70/ Filter for manufacturer ... 10
/REQ80/ Search for digital twin ... 10
Non-functional Requirements ... 11
/NF10/ GUI ... 11
/NF20/ Browser ... 11
/NF30/ Efficiency ... 11
/NF40/ Usability ... 11
Data Model ... 12
/LF10/ Data ... 12
Analysis of Existing Solutions ... 12
Admin-shell-io.com ... 12
ZVEI Online Demonstrator ... 13
demo-digital-twin.r-stahl.com/ ... 14

## Purpose

The ultimate end goal of this software shall be that the web-based-interface written in java-script displays the products in a user-friendly and structured way. It should be possible to search for products and filter them. It should also be possible to add new servers.

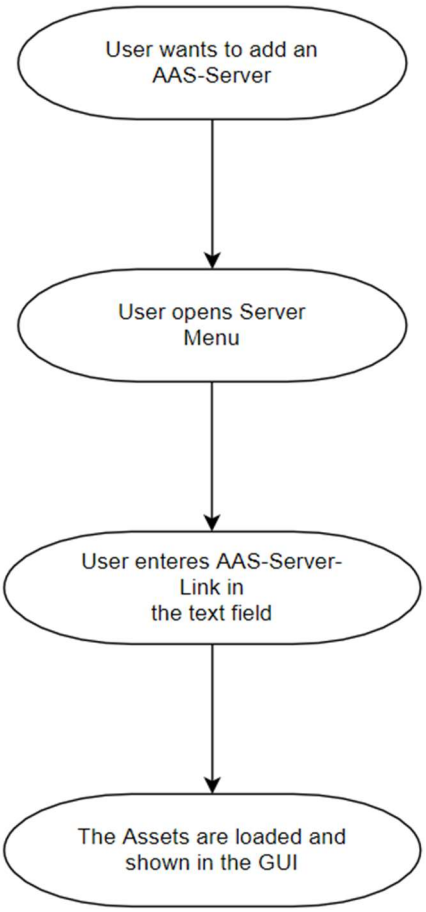
## Product Environment

The Asset Administration Shell (AAS) is the Standard for modelling digital twins in Industry 4.0. Digital twins are models of so-called assets. Those can be tools, systems and products, etc. The twins shall be as like the existing asset as possible, even if it hasn’t even been built before. It should simulate a real asset in every (technical) detail. You can virtually build machines with digital twins or test them in virtual environments before building the machine to test its functionality, usability, etc.

# Use Cases

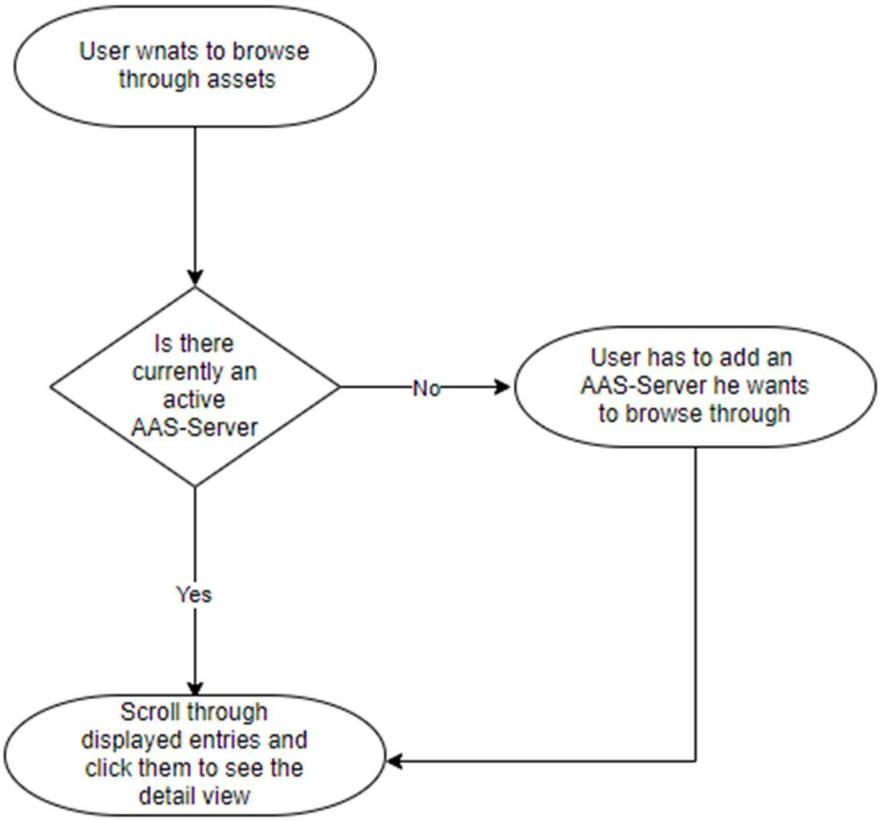
## <UC.001>: Add AAS-Server

| | |
|---|---|
| *Use Cases Objective:* | *User wants to add an AAS-Server by inserting the server's URL manually into the webpage.* |
| *System Boundary:* | *The application itself.* |
| *Precondition:* | *The user needs to have the URL to the AAS-Server. The webpage must be open and connected to the Internet. There must not be a currently active AAS-Server.* |
| *Postcondition on success:* | *The server's content is displayed on the webpage.* |
| *Involved Users:* | *Every end-user of the application.* |
| *Triggering Event:* | *When the user presses "Server Menu" on the webpage, the user enters the URL and presses "Add Server".* |



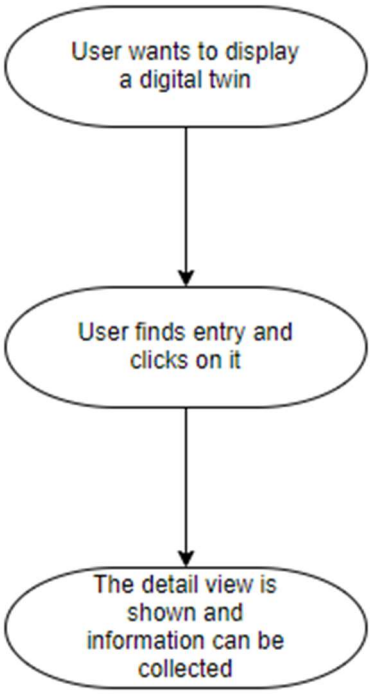
## <UC.002>: Browse through digital twins

| | |
|---|---|
| *Use Cases Objective:* | *The digital twins are displayed alphabetically sorted on the webpage. The user can scroll through them.* |
| *System Boundary:* | *The application itself* |
| *Precondition:* | *The user added a working AAS-Server.* |
| *Postcondition on success:* | *The user can browse through all content of the server and find digital twins coincidentally.* |
| *Involved Users:* | *Every end-user of the application.* |
| *Triggering Event:* | *When the user wants to look through all content of the server without the need to find a specific digital twin.* |



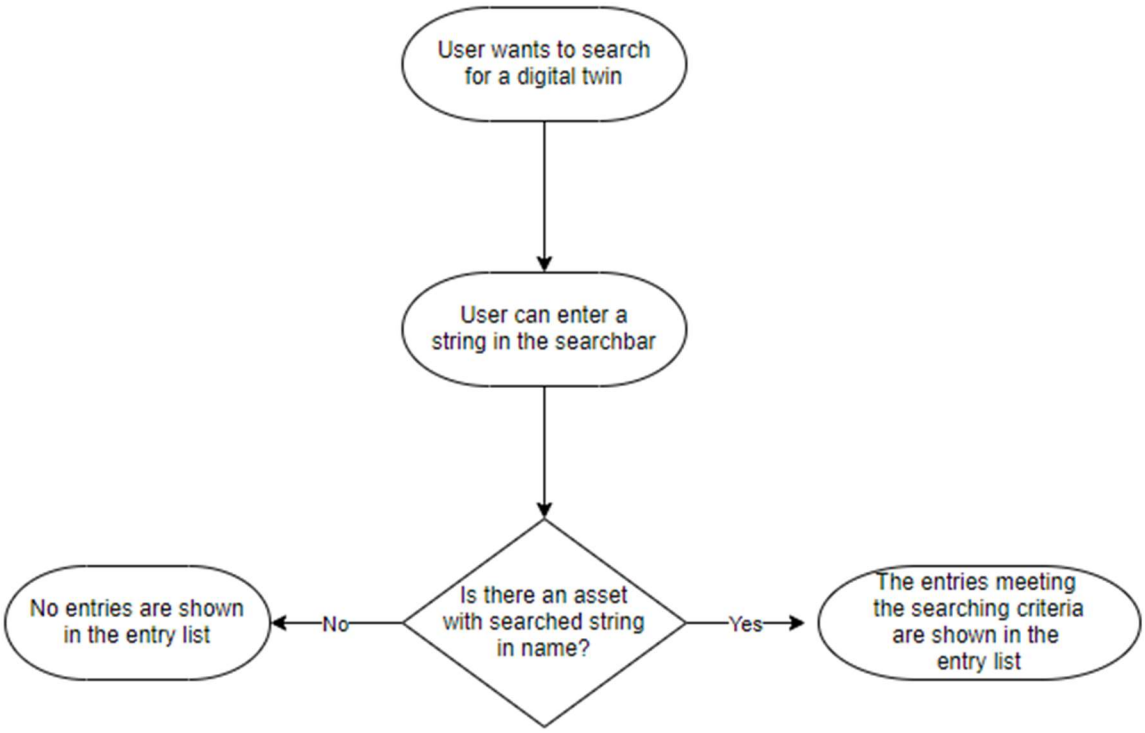
## <UC.003>: Display digital twin

| | |
|---|---|
| *Use Cases Objective:* | *User can see the information of a specific digital twin* |
| *System Boundary:* | *The application itself* |
| *Precondition:* | *The user added a working AAS-Server.* |
| *Postcondition on success:* | *The user can see the information regarding a specific digital twin* |
| *Involved Users:* | *Every end-user of the application.* |
| *Triggering Event:* | *The user clicks on an entry of a digital twin on the webpage* |



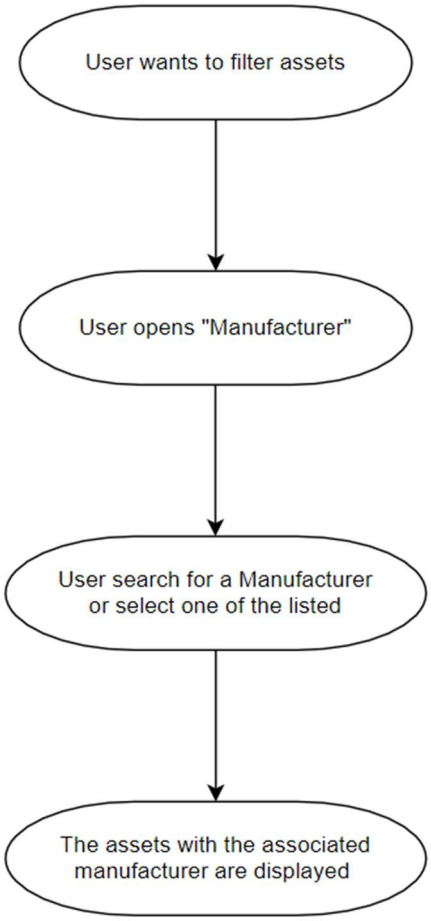
## <UC.004>: Search for digital twin

| | |
|---|---|
| *Use Cases Objective:* | *Search for a specific digital twin by serial number or by free text search.* |
| *System Boundary:* | *The application itself* |
| *Precondition:* | *The user added a working AAS-Server.* |
| *Postcondition on success:* | *A reduced number of digital twins is displayed on the webpage in the same way it was displayed before the search.* |
| *Involved Users:* | *Every end-user of the application.* |
| *Triggering Event:* | *The user enters a text in the search bar and presses "search" or Enter and wants to find a specific digital twin. For each letter, a clickable list of corresponding assets is displayed.* |



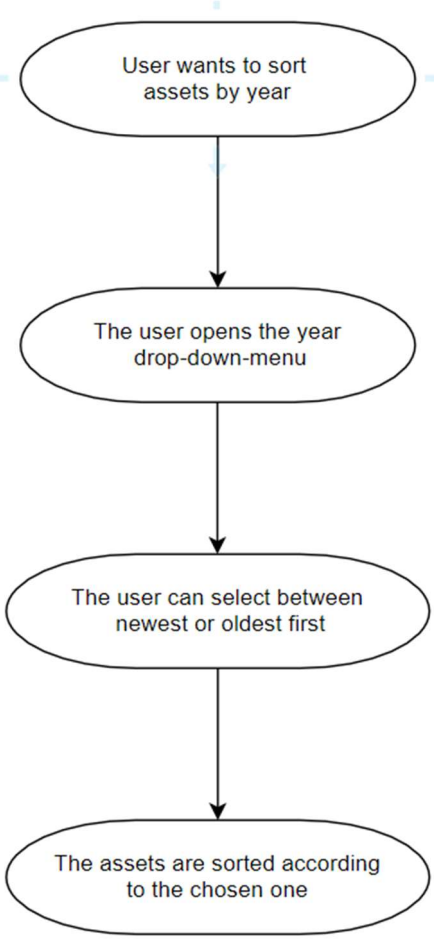
## <UC.005>: Filter for manufacturer

| | |
|---|---|
| *Use Cases Objective:* | *Filter all content on the webpage to only show assets of one manufacturer name.* |
| *System Boundary:* | *The application itself.* |
| *Precondition:* | *The user added a working AAS-Server.* |
| *Postcondition on success:* | *The entries are specified and displayed on the webpage.* |
| *Involved Users:* | *Every end-user of the application.* |
| *Triggering Event:* | *The user enters wished manufacturer in designated area.* |



## <UC.006>: Sort by year

| | |
|---|---|
| *Use Cases Objective:* | *Sort all content on the webpage to show the newest/oldest assets at the top of the list based on manufacturing year* |
| *System Boundary:* | *The application itself.* |
| *Precondition:* | *The user added a working AAS-Server.* |
| *Postcondition on success:* | *The entries are sorted and displayed on the webpage and entries without manufacturing year are shown at the end of the list.* |
| *Involved Users:* | *Every end-user of the application.* |
| *Triggering Event:* | The user opens the drop-down menu for the year and can choose between the newest and the oldest year. |



# Product Requirements

The following functions should be implemented in the applications.

## /REQ10/ Import Server

The webpage should be able to import a server by its URL and display its content.

| Input Description | Expected Output |
|---|---|
| Server URL | JSON containing the AAS data |

## /REQ20/ Server validation

The system shall be able to detect false server-URLs when adding a new server and throw an error to the user.

| Input Description | Expected Output |
|---|---|
| Wrong Server URL | Error message |

## /REQ30/ Error handling

The system shall be able to handle errors (no entries found, unexpected errors, false server-URL, …) and throw an error to the user.

| Input Description | Expected Output |
|---|---|
| Unexpected Input | Error message |

## /REQ40/ Display content in a clear way

The entries including the digital twins are shown in a clear and readable way as a list one entry under the other.

| Input Description | Expected Output |
|---|---|
| Server URL | List of AAS items |

## /REQ50/ Display digital twin

The user should be able to see more information about the digital twin.

| Input Description | Expected Output |
|---|---|
| Click on an asset | listed information |

## /REQ60/ Sort by year

The user should be able to sort the displayed assets by manufacturer year.

| Input Description | Expected Output |
|---|---|
| Selected sorting direction (oldest/newest first) | Sorted list of AAS items |

## /REQ70/ Filter for manufacturer

The user should be able to filter the assets by manufacturer.

| Input Description | Expected Output |
|---|---|
| Manufacturer name | List of AAS items only by the specific manufacturer. |

## /REQ80/ Search for digital twin

The user can search for digital twins by entering the name of the asset.

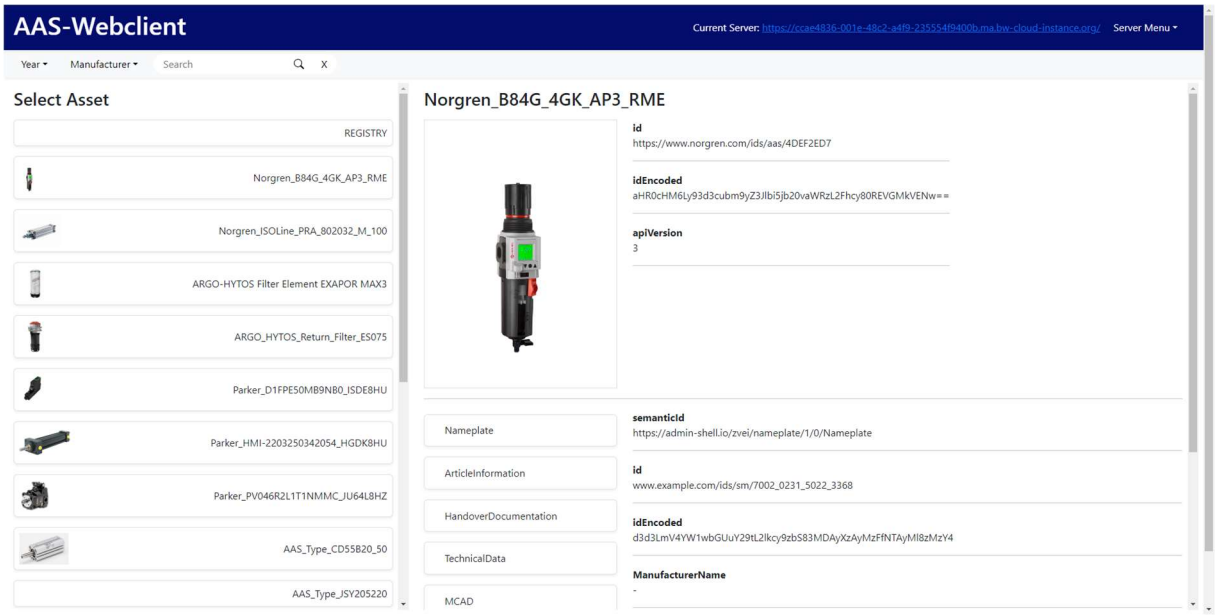
| Input Description | Expected Output |
|---|---|
| Name of asset | List of AAS items with the specific name |

# Non-functional Requirements

This section describes the already known non-functional requirements for the product.

## /NF10/ GUI

The webpage shall display a graphical user interface (GUI) to the user. This GUI must display every function provided to the user in a simple and intuitive way. It will be the only way to interact with the application.




## /NF20/ Browser

The webpage shall work in every Browser supporting the HTML 5 standard.

## /NF30/ Efficiency

The webpage shall add servers and apply filters in the fastest way possible as well as the user being able to find desired results with the lowest possible amount of steps.

## /NF40/ Usability

A user searching a specific product shall find that product as fast as possible, intuitively know how the webpage works and there shall be enough online documentation for new users. No further training or experience is required.

# Data Model

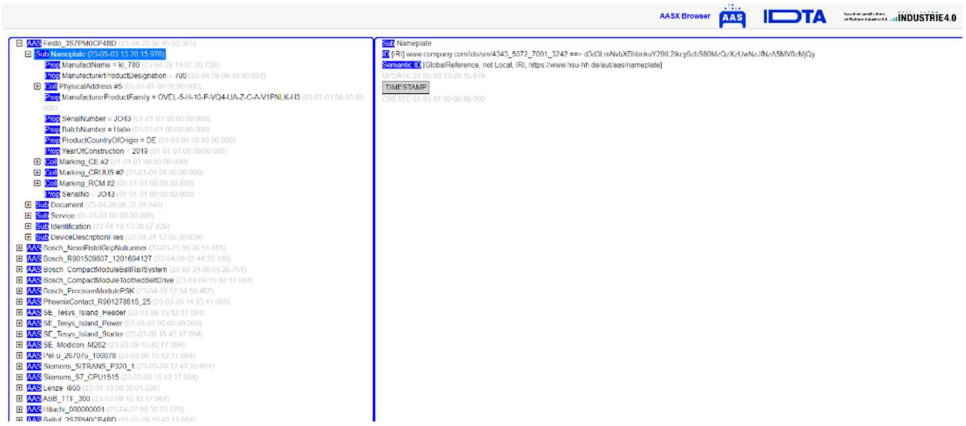
This section describes the data, that should be displayed on the webpage.

## /LF10/ Data

The data displayed in the webpage is delivered through an AAS-Server with REST-Calls. There is no functionality to export data or import other data than an AAS-Server by its URL.

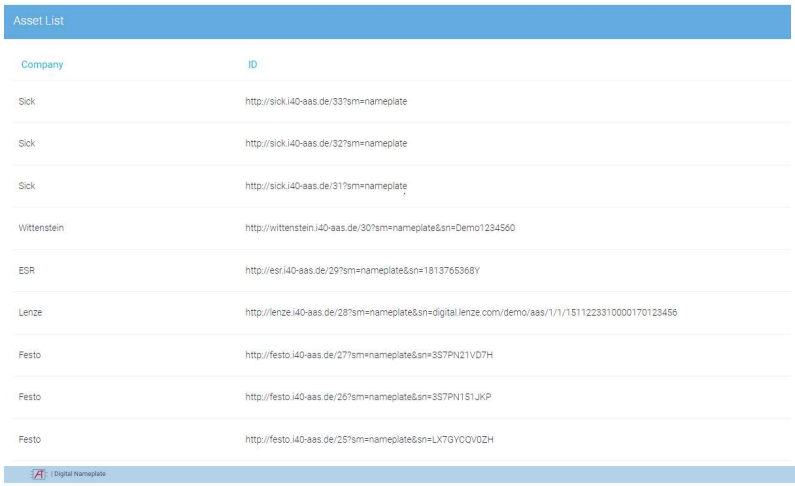
# Analysis of Existing Solutions

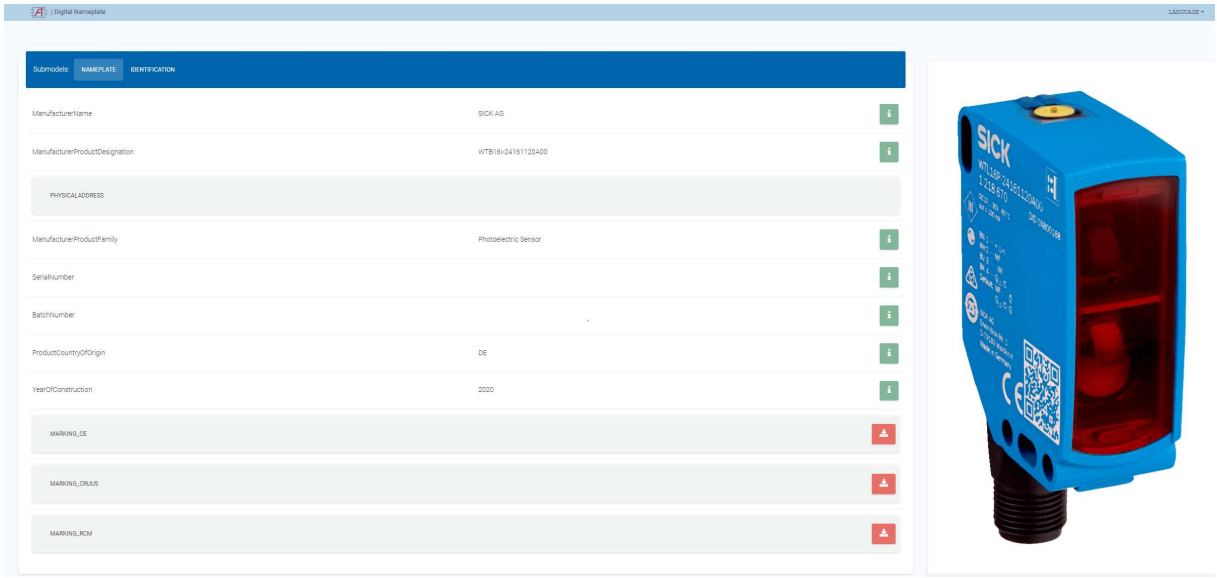
## Admin-shell-io.com




| Strengths | Weaknesses |
|---|---|
| The list shows the assets clearly | It quickly becomes confusing due to the many dropdown menus on the website |
| | It is difficult to look for the content and find the right one |

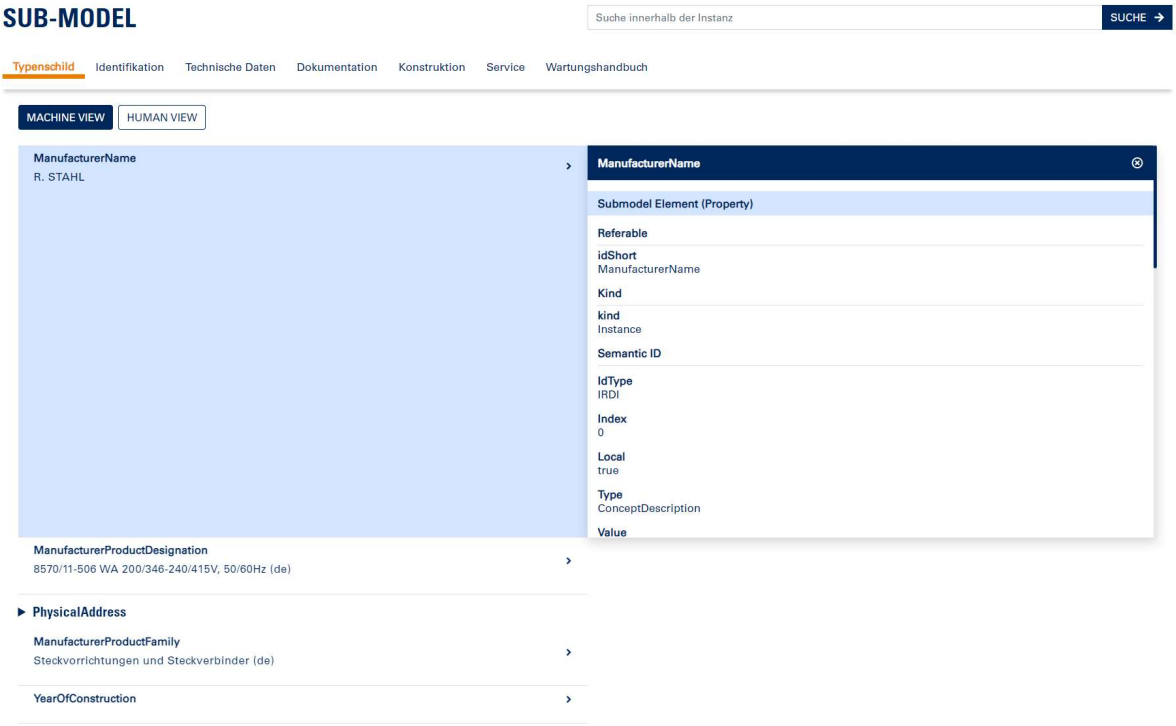
## ZVEI Online Demonstrator




| Strengths | Weaknesses |
|---|---|
| The list shows the assets clearly | There are no names in the asset selection |
| The submodels are clear and structured | |

[demo-digital-twin.r-stahl.com/](demo-digital-twin.r-stahl.com/)




| Strengths | Weaknesses |
|---|---|
| - | The display of an asset is extremely confusing, nested and unclear because of the many dropdowns and many fields. |