Software Requirements Specification (Wiki Export)

Version 0.12

Version History

Version	Date	Author	Comment
0.1	17.10.2022	Paul Brenner	Created
0.2	24.10.2022	Paul Brenner	GUI Previews
0.3	30.10.2022	Paul Brenner	Implemented Feedback (Issue#13)
0.4	30.10.2022	Paul Brenner	Table of contents
0.5	08.11.2022	Paul Brenner	Changes 1.4 Features and Layout
0.6	09.11.2022	Paul Brenner	AASM-UC.04, AASM-UC.05 – Modification Involved Roles
0.7	09.11.2022	Paul Brenner	Correction AASM-UC.07, Update Team-Number
0.8	17.03.2023	Paul Brenner	Add user privilege Matrix
0.9	04.04.2023	Paul Brenner	Creation of Markdown-Wiki
0.10	17.04.2023	Paul Brenner	Remove Business Processes, general improvements
0.11	17.04.2023	Paul Brenner	Fixed Requirements, create Use Case Add/Edit Asset
0.12	11.05.2023	Paul Brenner	Add Use-Cases

Table of contents

1 Concept and general conditions

This section illustrates the goal and product environment and presents the characteristics of the web application.

1.1 Goal

The goal of this project is to develop a web application that acts as a management system for the “Asset Administration Shell” (AAS). This specific web application shall have an identity and access management as well as a user administration with persistent data storage in MongoDB. The user administration enables a role distribution of the users in the user groups “Admin”, “Advanced” and “Basic”, whereby the role distribution is carried out manually via the Admin. Each role is equipped with different access rights and read permissions (“Advanced” gets full read access to all AAS and their submodels and “Basic” gets read access only to the basic submodels to all AAS), with the admin also having functions for managing AAS content and user management. This uses the specification of the concept as a REST API in openapi.

1.2 Product Environment

The AAS is a concept of the Industrie 4.0 platform for the standardized implementation of “Industrie 4.0 components”, consisting of the digital twin in the form of the AAS and the associated physical object (the asset). This makes it possible in industry to provide digital twins that can be shared and combined across manufacturers and accessed via standardized interfaces.

1.3 System Environment

1.3.1 General

AAS Management is a web application and therefore only requires an Internet connection and an installed web browser on the computer work.

1.3.2 Product Data - /AASM-D10/ MongoDB

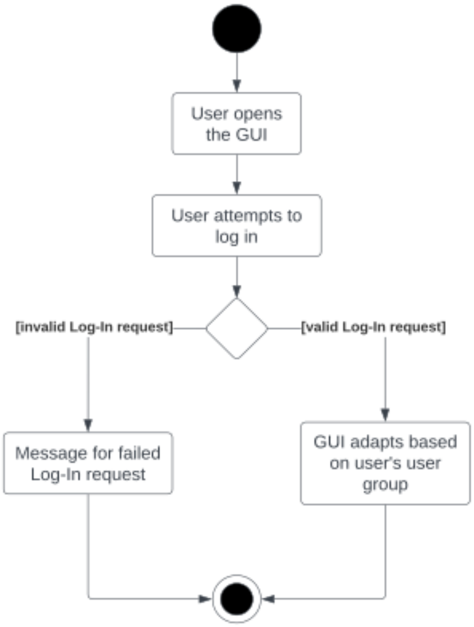
The web application should enable persistent data storage with MongoDB as the database management system used.

MongoDB is hosted on a Synology NAS using Ubuntu 22.4 and MongoDB Version 4

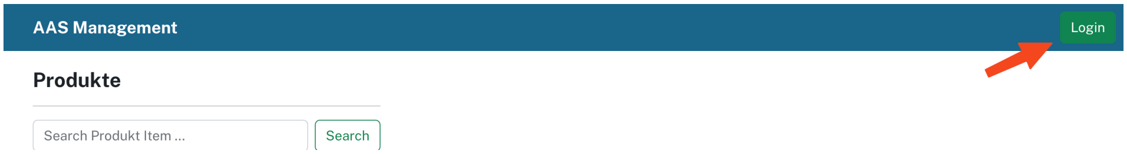
2 Use Cases

2.1 UC01 Login

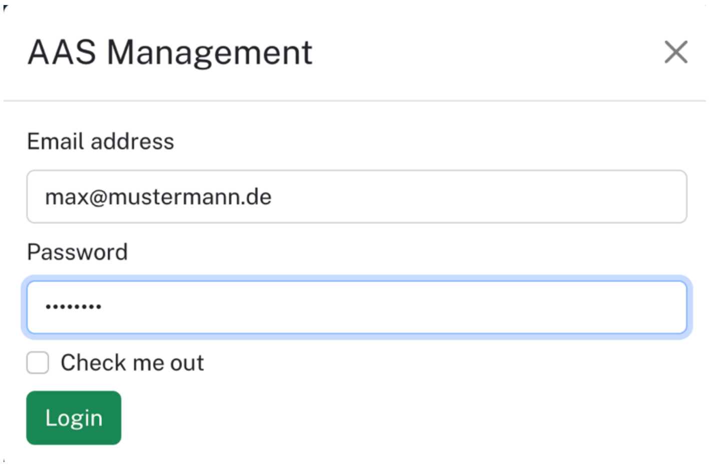
Nr. / ID	<AASM-UC.01>
Use Cases Objective	User wants to log into account.
System Boundary	
Precondition	The user's combination of e-mail address and password are valid and stored in the database, and the database system is in operation.
Postcondition on success	The web application shall not be closed until the user is logged in to his account and the user interface is customized according to the user group.
Involved Roles	Any user, GUI and database interface
Triggering Event	User opens the GUI and wants to log into user's account.



Login



Login Button



AAS Management

Email address

max@mustermann.de

Password

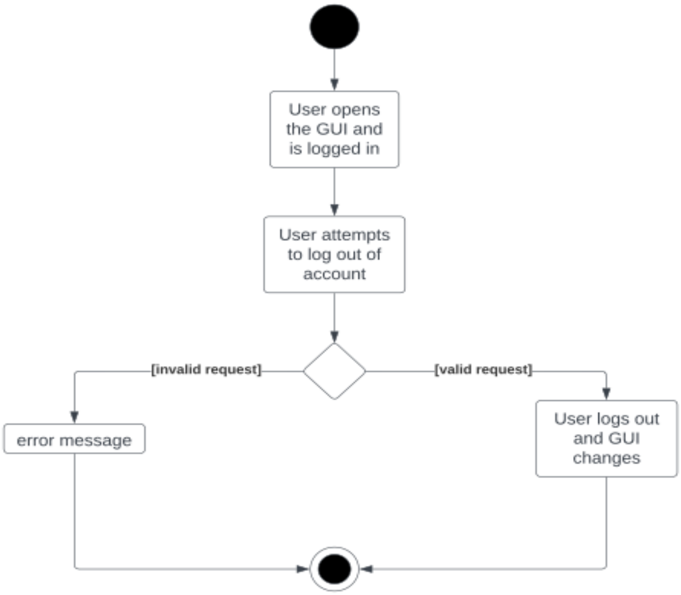
••••••••

□ Check me out

Input Dialoge for User Credentials

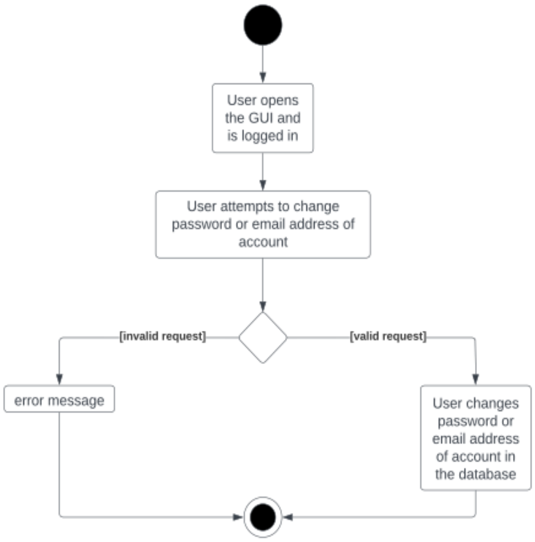
2.2 UC02 Logout

Nr. / ID	<AASM-UC.02>
Use Cases Objective	User wants to log out of his account.
System Boundary	
Precondition	User is already logged in.
Postcondition on success	The web application shall not be closed until the user is logged out.
Involved Roles	Any user and GUI
Triggering Event	User wants to log out of account.

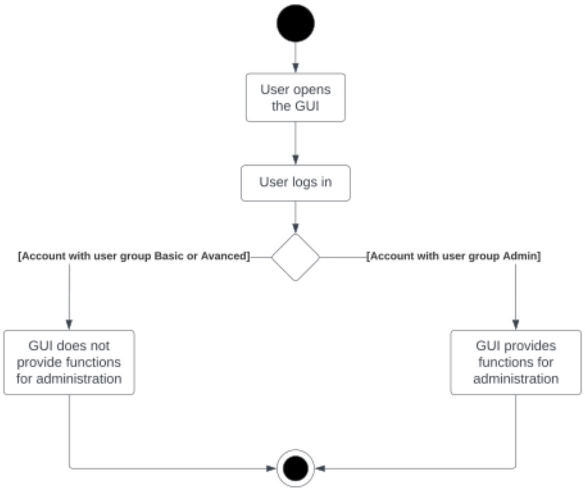

User opens the GUI and is logged in
User attempts to log out of account
[invalid request]
[valid request]
error message
User logs out and GUI changes

2.3 UC03 Managing own Account

Nr. / ID	<AASM-UC.03>
Use Cases Objective	The user wants to use the self-services for e-mail and password change to modify these components of the account.
System Boundary	
Precondition	The user is already logged in, has the current password of the account and the database system is running.
Postcondition on success	The web application may not be closed until these changes have also been changed in the database system.
Involved Roles	Any user, GUI and database interface
Triggering Event	User wants to change password and/or e-mail address.


User opens the GUI and is logged in
User attempts to change password or email address of account
[invalid request]
[valid request]
error message
User changes password or email address of account in the database

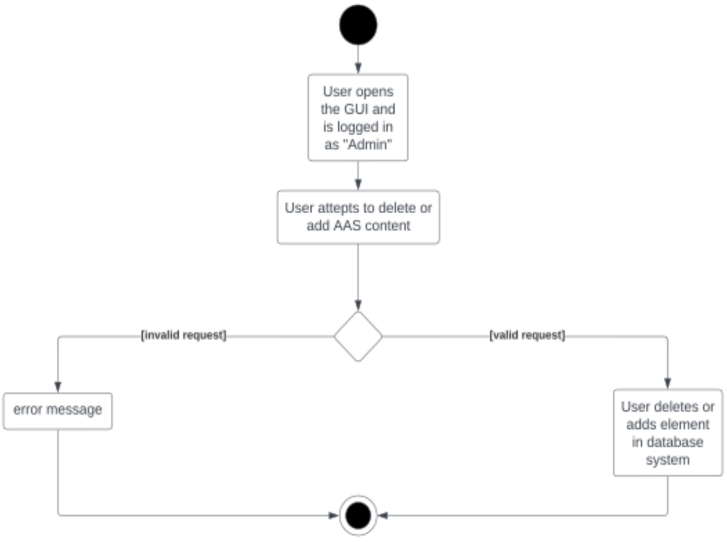
Managing Own Account


User opens the GUI
User logs in
[Account with user group Basic or Avanced]
[Account with user group Admin]
GUI does not provide functions for administration
GUI provides functions for administration

User Assignment

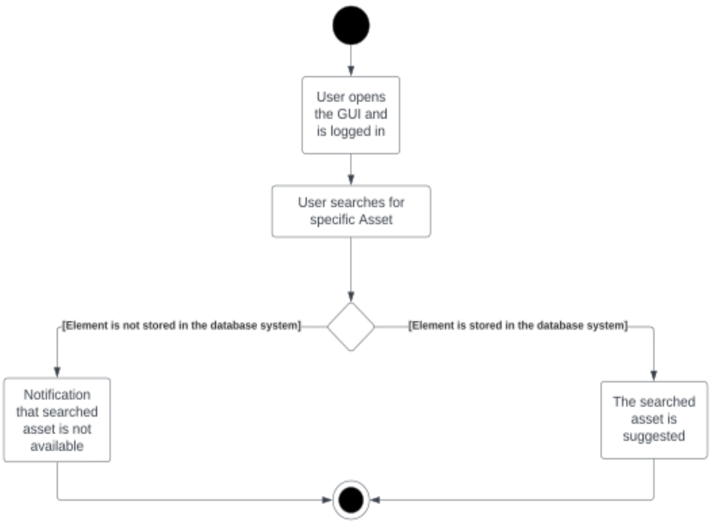
2.4 UC04 Administrate AAS content

Nr. / ID	<AASM-UC.04>
Use Cases Objective	The user wants to add or delete AAS content.
System Boundary	
Precondition	The user's user group is Admin, and the database system is operation.
Postcondition on success	The web application may not be closed until these changes have also been changed in the database system.
Involved Roles	User with role "Admin", GUI and database interface
Triggering Event	User wants to add AAS content by uploading a .aasx file ("Add Asset") or delete AAS content.


User opens the GUI and is logged in as "Admin"
User attepts to delete or add AAS content
[invalid request]
[valid request]
error message
User deletes or adds element in database system

2.5 UC05 Administrate Accounts

Nr. / ID	<AASM-UC.05>
Use Cases Objective	The user wants to add or delete accounts. The user also wants to divide the accounts into user groups (Basic, Advanced and Admin).
System Boundary	
Precondition	The user's user group is Admin, and the database system is operation.
Postcondition on success	The web application may not be closed until these changes have also been changed in the database system.
Involved Roles	User with role "Admin", GUI and database interface
Triggering Event	User wants to add or delete accounts or divide the accounts into user groups.


User opens the GUI and is logged in
User searches for specific Asset
[Element is not stored in the database system]
[Element is stored in the database system]
Notification that searched asset is not available
The searched asset is suggested

2.6 UC06 Find asset

Nr. / ID	<AASM-UC.06>
Use Cases Objective	The user searches for an asset to find it in the GUI or receives a notification that the asset is not present.
System Boundary	
Precondition	Assets are successfully loaded
Postcondition on success	
Involved Roles	Any user, GUI and database interface
Triggering Event	User enters search term into the search field

2.7 UC07 Display asset

Nr. / ID	<AASM-UC.07>
Use Cases Objective	The user finds the asset and clicks on it for display purposes and further details.
System Boundary	
Precondition	For a role-based access the current user role need to be validated. To view restricted data (based on user role) user may need to be logged in.
Postcondition on success	
Involved Roles	Any user, GUI and database interface
Triggering Event	User selects an Asset (by click)

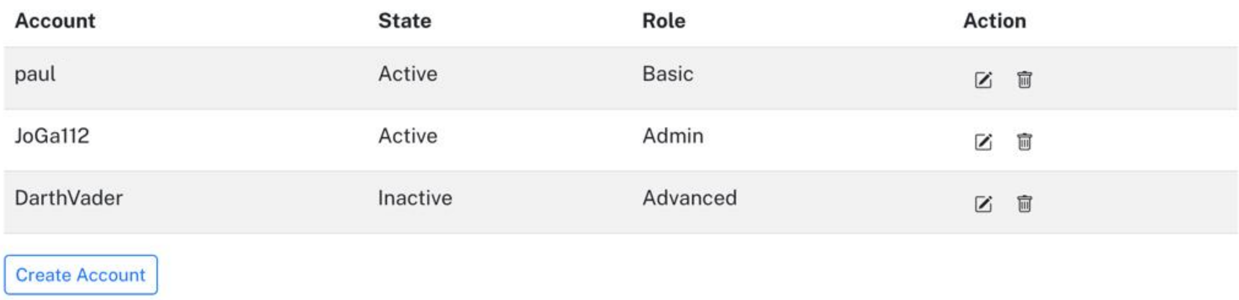
3 Functional Requirements

3.1 Frontend

3.1.1 /AASM-REQ1/ Default landing page - GUI Components

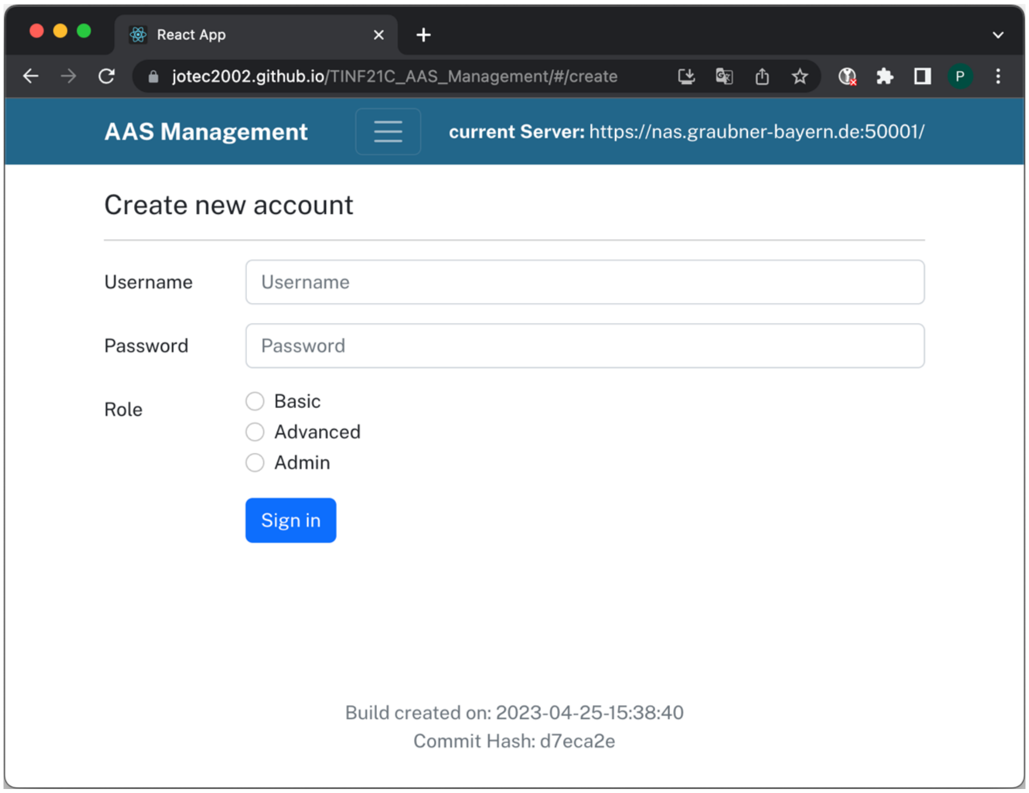
The GUI has a home page where the user can browse the package explorer with AAS content and log in to an account. When the user logs in, he gets access to a customized package explorer with an extension of more functions and dynamically customized content depending on the user's user group.

3.1.2 /AASM-REQ2/ Identity & Access Management



Account	State	Role	Action
paul	Active	Basic	✎ 🗑
JoGa112	Active	Admin	✎ 🗑
DarthVader	Inactive	Advanced	✎ 🗑

User Management - Settings Admin v0.2



current Server: https://nas.graubner-bayern.de:50001/

Create new account

Username: Username

Password: Password

Role
○ Basic
○ Advanced
○ Admin

Build created on: 2023-04-25-15:38:40
Commit Hash: d7eca2e

User Management - Add user v0.2

The web application shall have centralized management of identities and access permissions, with role-based access rights. The web application shall have functions for authentication and authorization of users and a self-service for users such as email and password change.

3.1.3 /AASM-REQ3/ AAS content data management

The user (more precisely the user with the role of Admin) has the possibility to upload and delete AAS content. The content to be uploaded is in .aasx file format.

3.1.4 /AASM-REQ4/ Search functionalities

The user should be able to effectively find desired AAS content using search functions. The user should be able to effectively find the desired content of the AAS using search functions.

3.1.5 /AASM-REQ5/ Rest-API Support

The software shall support the V3 Rest-API spezified in https://v3.admin-shell-io.com/swagger/index.html standard for communication between the client and database system.

3.1.6 /AASM-REQ6/ Error Display

The software should check whether an error has occurred. If an error occurs, the user should receive an easily readable and understandable error message.

3.2 Backend

3.2.1 /AASM-REQ7/ MongoDB Integration

The web application should enable persistent data storage with MongoDB as the database management system used. MongoDB is hosted on a Synology NAS using Ubuntu 22.4 and MongoDB Version 4

4 Non-functional Requirements

4.1.1 /AASM-NF10/ Usability

Usability is to be ensured by a simple layout and by a human-centered interface. It should be clear and easy to use so that the user can achieve the objectives efficiently and satisfactorily.

4.1.2 /AASM-NF20/ Reliability

Reliability is to be ensured by a responsive GUI with high error tolerance and a database with accurate and consistent data.

4.1.3 /AASM-NF30/ Performance

The software should orientate on the recommended page load time by Google (2 seconds). Yet this time can vary, based on the internet connection and file size (e.g. images), therefore a loading time shorter than 7 seconds fulfils the requirements.

4.1.4 /AASM-NF40/ License

The software is published with a MIT License on GitHub.

5 User privilege Matrix

	Basic	Advanced	Admin
Login	X	X	X
Managing own account	X	X	X
User management			X
View Asset title	X	X	X
View Asset details – Basic information:	X	X	X
View Asset details – Advanced information		X	X
Download Asset files		X	X
Administrate AAS Content			X
PUT/POST/DELETE API-Calls			X

5.1 Security Roles

Security role informations are stored in the “myAASwithGlobalSecurityMetaModel”.

5.1.1 Role 0

Deny Access to Security Asset Submodels for all users except the admin role.

5.1.2 Role 1

Gives the Advanced user the ability to use the complete API with read calls.

5.1.3 Role 2

Gives the Admin user the ability to use the complete API with read, create, update and delete calls. Therefore it is possible to fulfil above requirements (e.g. Adding AAS-Models)

5.1.4 Role 3

Gives the basic user the ability to access the submodels with the semantic id “/aas/identification” or the semantic id “/aas/nameplate”

Document information

Document author	Paul Brenner
Created on	17.10.2022
Export on	11.05.2023

Duale Hochschule Baden-Württemberg