

18.12.2025 – HS 2025/2026

BSL Support Dashboard

Studienarbeit

Projektteam: Jeriel Frei
Simeon Rhyner

Projektbetreuer: Joel Schwab

Praxispartner: Berufsschule Lenzburg

Inhaltsverzeichnis

I Abstract	1
II Management Summary	3
Ausgangslage und Zielsetzung	4
Realisierung und Technologie	4
Ergebnis und Ausblick	8
III Danksagung	9
IV Product Documentation	11
1 Requirements	12
1.1 Actors	12
1.1.1 Supportmitarbeiter	12
1.1.2 Applikationsverwalter	12
1.1.3 Interne Dienste	12
1.1.4 Externe Dienste	12
1.2 Use Case	13
1.2.1 UC01 - Modulares Dashboard verwalten	14
1.2.2 UC02 - Passwort zurücksetzen	15
1.2.3 UC03 - PRTG Monitoring	16
1.2.4 UC04 - Powerautomate Flow Error	17
1.2.5 UC05 - Administration	18
1.3 Non-Functional Requirements	19
1.3.1 Erklärung	19
1.3.2 Functionality	19
1.3.3 Usability	20
1.3.4 Reliability	20
1.3.5 Supportability	21
2 Architecture	22
2.1 Systemarchitektur	22
2.2 Sicherheitskonzept	24
2.2.1 Login	24
2.2.2 Absicherung der API	25
2.3 Technologie	26
2.3.1 React	26
2.3.2 .Net	26
2.3.3 PostgreSQL	26
2.3.4 Docker	26
2.3.5 Swagger	26

2.3.6	GitLab	27
2.4	CI/CD	28
2.4.1	Gitlab	28
2.4.2	Docker	28
2.4.3	Pipeline	29
3	Umsetzung	31
3.1	Login und Dashboard	31
3.1.1	Kurzbeschreibung	31
3.1.2	Aktivitätsdiagramm	32
3.1.3	Mockups	33
3.1.4	Feedback aus dem Supportteam	33
3.1.5	Umsetzung	34
3.1.6	Tests	35
3.2	AD Modul Beschreibung	36
3.2.1	Kurzbeschreibung	36
3.2.2	Aktivitätsdiagramm	37
3.2.3	Mockups	38
3.2.4	Feedback aus dem Supportteam	40
3.2.5	Umsetzung	41
3.2.6	Schnittstellen	44
3.2.7	Spezielles	45
3.2.8	Tests	45
3.2.9	Erledigte Functional Requirements	45
3.3	Modul PRTG Server Monitor	46
3.3.1	Kurzbeschreibung	46
3.3.2	Aktivitätsdiagramm	46
3.3.3	Mockups	47
3.3.4	Umsetzung	48
3.3.5	Schnittstellen	50
3.3.6	Tests	50
3.3.7	Erledigte Functional Requirements	50
3.4	Power Automate Runs Fehler	51
3.4.1	Kurzbeschreibung	51
3.4.2	Aktivitätsdiagramm	52
3.4.3	Mockups	53
3.4.4	Feedback aus dem Supportteam	53
3.4.5	Umsetzung	54
3.4.6	Schnittstellen	56
3.4.7	Tests	57
3.4.8	Erledigte Functional Requirements	57
3.5	Einstellungen	58
3.5.1	Kurzbeschreibung	58
3.5.2	Aktivitätsdiagramm	58
3.5.3	Mockups	59
3.5.4	Umsetzung	60

3.5.5 Schnittstellen	61
3.5.6 Tests	61
3.5.7 Erledigte Functional Requirements	61
4 Quality Measures	62
4.1 Working Environment	62
4.1.1 Dokumentation	62
4.1.2 Code	62
4.1.3 Definition of Done	62
4.1.4 Version Control	62
4.1.5 Review Process	64
4.1.6 Automatic Quality Control	64
4.2 Test Concept	65
4.2.1 Übersicht	65
4.3 Test Protocols	66
4.3.1 Legende Testbewertung	66
4.3.2 Acceptance Tests Non-Functional Requirements	66
IV Project Documentation	68
5 Initial Project Proposal	69
5.1 Kunde und Ansprechpartner	69
5.1.1 Praxispartner	69
5.2 Thema	70
5.3 Kurzbeschreibung und erwartete Resultate	71
5.4 Randbedingungen und Infrastruktur	73
5.4.1 Infrastruktur:	73
5.5 Sonstige Hinweise	74
6 Project Plan	75
6.1 Project Management	75
6.2 Long Term Planning	76
6.2.1 Milestones	76
6.3 Risk Management	78
6.3.1 Risikomatrix 01.10.2025	81
6.3.2 Risikomatrix 17.10.2025	81
6.3.3 Risikomatrix 30.11.2025	82
7 Persönlicher Bericht	83
7.1 Jeriel Frei	83
7.2 Simeon Rhyner	83
8 Schlussfolgerungen und Ausblick	84
8.1 Schlussfolgerungen	84
8.2 Ausblick	85
9 Time Tracking Report	86
9.1 Zeit pro Person	86
9.2 Zeit pro Kategorie	86
10 Glossary und Abkürzungen	87

11 Abbildungsverzeichnis	88
12 Tabellenverzeichnis	90
V Anhang	91
Hilfsmittelverzeichniss	92

Part I

Abstract

Das IT-Supportteam der Berufsschule Lenzburg stand vor ineffizienten Support-Prozessen: Passwort-Resets erforderten mehrere manuelle Schritte, die Serverüberwachung war unübersichtlich, und Power Automate Flow-Fehler mussten einzeln überprüft werden.

Diese Studienarbeit entwickelte ein modulares Dashboard-System zur Zentralisierung und Verbesserung dieser wiederkehrenden Aufgaben. Die Lösung basiert auf einem React-Frontend, einem .NET 8.0-Backend und einer PostgreSQL-Datenbank, deployed als Docker-Container.

Drei Kernmodule wurden implementiert:

- AD Passwort Reset ermöglicht Benutzersuche via E-Mail, Benutzername oder Anzeigenname und generiert automatisch sichere Passwörter.
- PRTG Server Monitor aggregiert Serverstatus mit Filterfunktionen.
- Power Automate Monitoring zeigt alle fehlgeschlagenen Flow-Runs an.

Sensible Funktionen, wie das AD-Modul, werden durch ein separates Admin-Authentifizierungssystem geschützt. Dies gewährleistet, dass nur autorisierte Support-Mitarbeiter kritische Aktionen durchführen können.

Das Dashboard ist vollständig personalisierbar mit drei Layout-Optionen und persistenter Benutzerkonfiguration.

Usability-Tests mit dem BSL IT-Leiter verliefen positiv. Die Passwort-Reset-Zeit wurde von mehreren Minuten auf unter 30 Sekunden reduziert. Alle Functional Requirements, die als Zwingend indentifiziert wurden, wurden erfüllt.

Die modulare Architektur ermöglicht einfache Erweiterungen.

Das System wird produktiv von der BSL übernommen und demonstriert erfolgreich die Verbesserung repetitiver IT-Support-Aufgaben durch moderne Webtechnologien.

Part II

Management Summary

Ausgangslage und Zielsetzung

Das IT-Supportteam der Berufsschule Lenzburg stand vor zeitaufwendigen, manuellen Prozessen: Passwort-Resets erforderten mehrere Schritte über einen Jumphost, die PRTG-Serverüberwachung war unübersichtlich verteilt, und Power Automate Flow mussten einzeln auf Fehler überprüft werden.

Projektziel: Entwicklung eines zentralen Dashboard-Systems zur Verbesserung dieser wiederkehrenden Support-Aufgaben.

Realisierung und Technologie

Das Projekt wurde in zwei Phasen durchgeführt.

Phase 1 umfasste eine Technologie-Evaluation: React für das Frontend, .NET für das Backend und PostgreSQL als Datenbank. Die Komponenten wurden mittels Docker containerisiert und über eine GitLab CI/CD-Pipeline deployed.

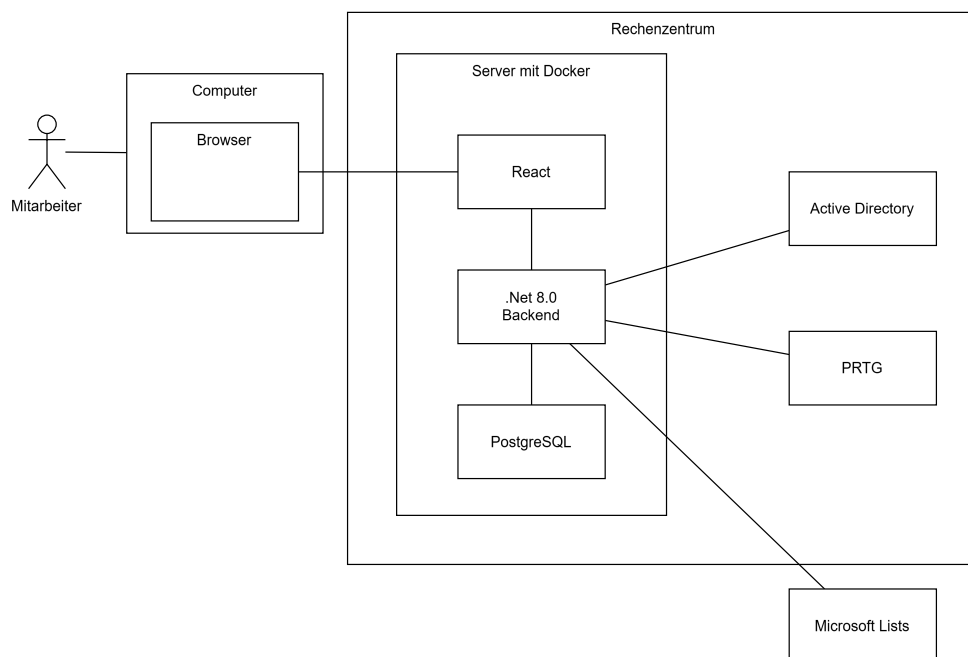


Figure 1: Systemarchitektur des BSL Support Dashboards

Phase 2 beinhaltete die Entwicklung von drei Kernmodulen:

AD Passwort Reset ermöglicht das schnelle Zurücksetzen von Benutzerpasswörtern durch Eingabe der E-Mail-Adresse, Benutzername oder Anzeigename. Das System generiert automatisch ein sicheres Passwort gemäss BSL-Richtlinien. Eine separate Admin-Authentifizierung schützt diese sensible Funktion.

The screenshot shows a web interface titled "Active Directory Passwort Reset". It features three input fields: "john.doe" in the first, "Display Name" in the second, and "Email" in the third. Below these fields are two buttons: "Suchen" (blue) and "Zurücksetzen" (grey). Underneath the buttons, the text "Suchergebnisse (1)" is displayed. A light grey box contains the search results for the user "Doe John". The results include: Display Name: Doe John, Username: john.doe, Email: john.doe@stud.bslenzburg.ch, Distinguished Name: CN=Doe John,OU=Benutzer,OU=stud,DC=BSL,DC=lan, Description: Testbenutzer Berufslernender Grundbildung & BM, and Account Status: Aktiv (indicated by a green checkmark). At the bottom of the results box is a blue button with a key icon and the text "Passwort zurücksetzen".

Active Directory Passwort Reset

john.doe Display Name

Email

Suchen Zurücksetzen

Suchergebnisse (1)

Display Name: Doe John
Username: john.doe
Email: john.doe@stud.bslenzburg.ch
Distinguished Name: CN=Doe John,OU=Benutzer,OU=stud,DC=BSL,DC=lan
Description: Testbenutzer Berufslernender Grundbildung & BM
Account Status: ✔ Aktiv

Passwort zurücksetzen

Figure 2: AD Passwort Reset Modul

PRTG Server Monitor bietet eine übersichtliche Darstellung aller Server und Sensoren mit Filterfunktionen nach Status, was eine schnelle Problembehebung ermöglicht.

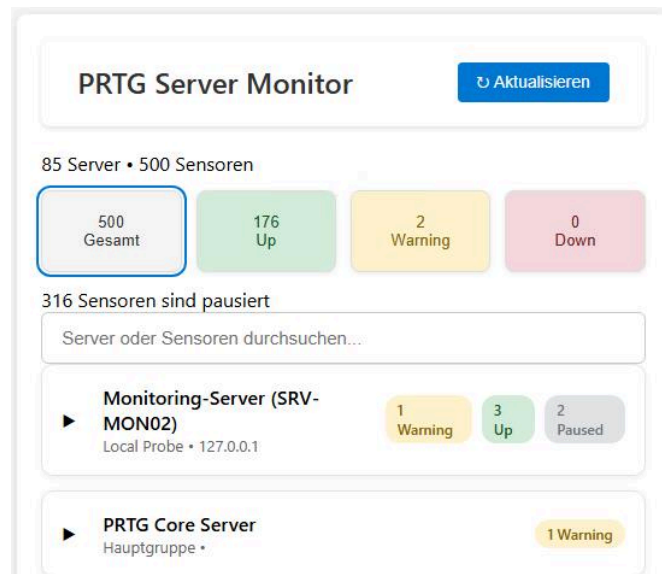


Figure 3: PRTG Server Monitor

Power Automate Monitoring zentralisiert die Überwachung aller Flow-Runs und hebt fehlgeschlagene Ausführungen hervor, die dann direkt als erledigt markiert werden können.

Power Automate Runs Fehler				
			☒ Abgeschlossene anzeigen	Aktualisieren
Run ID	Flow Name	Startzeit	Status	Aktionen
2	AntragExkursion ExternerUnterricht	17.09.2025, 08:16	Ausstehend	Run anzeigen Item öffnen Als abgeschlossen markieren
3	AntragExkursion ExternerUnterricht	24.09.2025, 15:11	Abgeschlossen	Run anzeigen Item öffnen ✓ Abgeschlossen
4	AntragExkursion ExternerUnterricht	26.09.2025, 11:21	Abgeschlossen	Run anzeigen Item öffnen ✓ Abgeschlossen
5	'INPRO-Beschaffung-Antrag Kostenvoranschlag-Antrag	02.09.2025, 11:07	Abgeschlossen	Run anzeigen Item öffnen ✓ Abgeschlossen

Figure 4: Power Automate Monitoring

Das Dashboard ist vollständig personalisierbar: Drei Layout-Optionen stehen zur Verfügung, Module können per Drag-and-Drop angeordnet werden, und alle Einstellungen werden persistent pro Benutzer gespeichert.

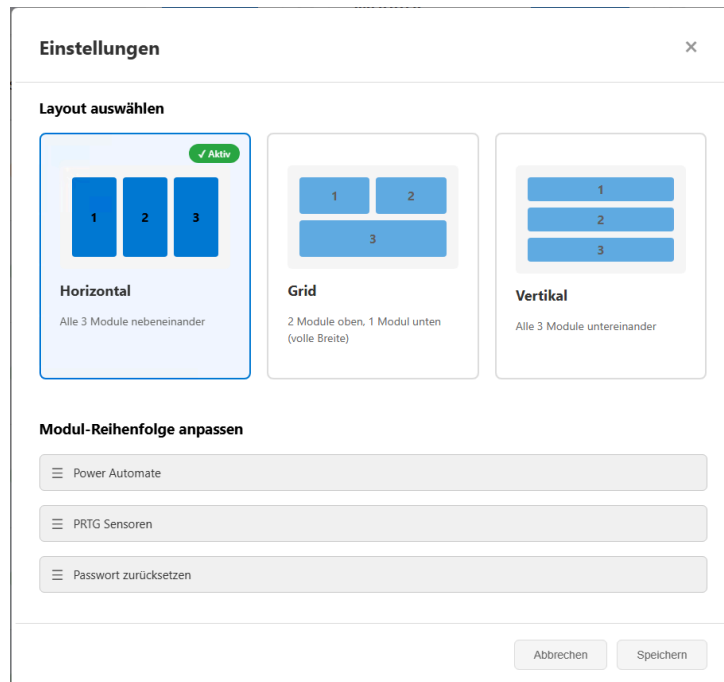


Figure 5: Einstellungen

Ergebnis und Ausblick

Das entwickelte System erfüllt alle definierten Functional Requirements, die als zwingend eingestuft wurden, und wurde erfolgreich in die Infrastruktur integriert. Usability-Tests mit dem IT-Leiter verliefen durchweg positiv.

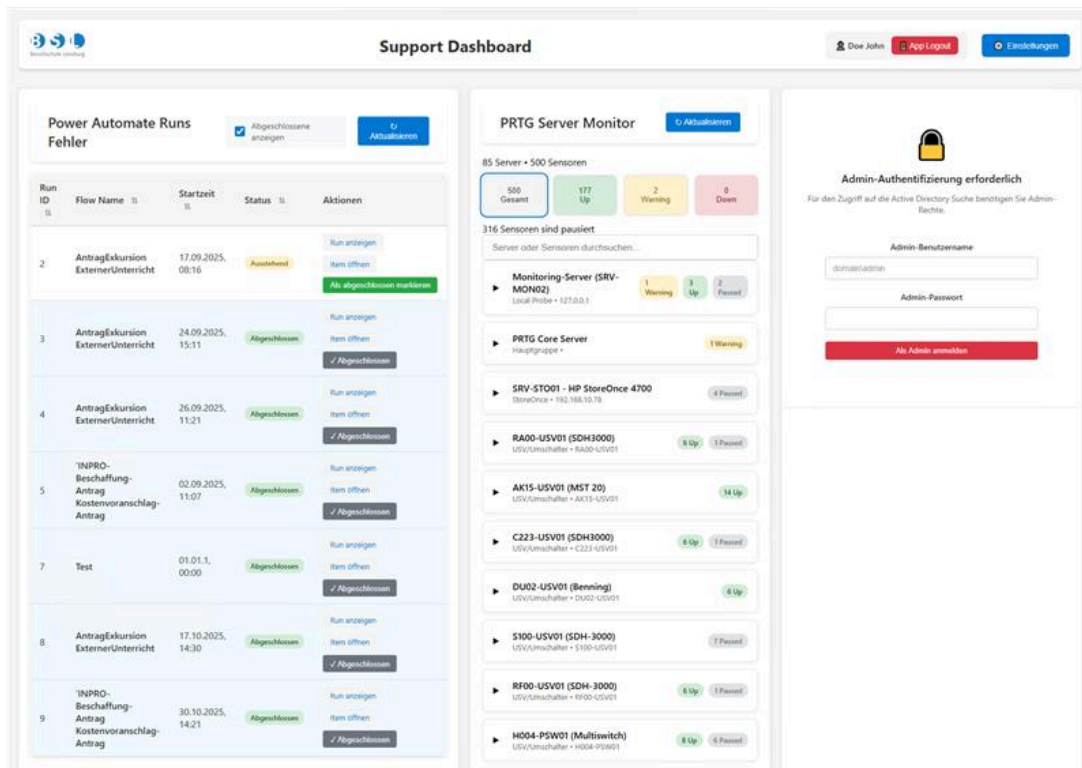


Figure 6: BSL Support Dashboard Gesamtansicht

Messbare Verbesserungen:

- Passwort-Reset: von mehreren Minuten auf unter 30 Sekunden
- Serverüberwachung: 85 Server und 500 Sensoren zentral überwacht
- Flow-Monitoring: Alle Fehler auf einen Blick statt manuellem Durchsuchen

Die modulare Architektur ermöglicht eine einfache Erweiterung um zusätzliche Funktionen wie Ticket-System-Integration oder Lizenzmanagement. Das BSL Support Dashboard demonstriert eindrucksvoll, wie moderne Webtechnologien genutzt werden können, um repetitive Support-Aufgaben zu verbessern und die Effizienz eines Support-Teams signifikant zu steigern.

Part III

Danksagung

Wir möchten uns herzlich bei allen Personen bedanken, die zum Gelingen dieser Studienarbeit beigetragen haben.

Unser besonderer Dank gilt Marco Gaiffi und dem IT-Team der Berufsschule Lenzburg für die ausgezeichnete Zusammenarbeit, die Bereitstellung der Infrastruktur und die wertvollen Praxiseinblicke. Die konstruktiven Feedbacks während der Usability-Tests haben wesentlich zur Qualität des BSL Support Dashboards beigetragen.

Herzlich danken möchten wir unserem Projektbetreuer Joel Schwab für die kompetente fachliche Begleitung, die hilfreichen Anregungen und die Unterstützung bei der Umsetzung des Projekts.

Ein besonderer Dank geht an Nico Tuscano und Tobias Locher für das sorgfältige Korrekturlesen dieser Dokumentation und die wertvollen Hinweise zur Verbesserung der sprachlichen Qualität.

Rapperswil, Dezember 2025
Jeriel Frei und Simeon Rhyner

Part IV

Product Documentation

1 Requirements

1.1 Actors

Für das BSL Support Dashboard gibt es verschiedene Actors.

1.1.1 Supportmitarbeiter

Der Supportmitarbeiter ist eine Person, die das BSL Support Dashboard verwendet.

1.1.2 Applikationsverwalter

Der Applikationsverwalter ist eine Person, die für die Verwaltung des BSL Support Dashboards verantwortlich ist.

1.1.3 Interne Dienste

Interne Dienste sind Dienste, die von der Applikation verwendet werden, um bestimmte Funktionen bereitzustellen.

1.1.4 Externe Dienste

Externe Dienste sind Dienste, die von der Applikation verwendet werden, um Funktionen bereitzustellen, welche nicht intern implementiert sind.

1.2 Use Case

Da wir im Rahmen unserer Studienarbeit die zwingend relevanten Functional Requirements umsetzen werden, erfolgt keine Priorisierung der Use Cases selbst, sondern nur der Functional Requirements. Stattdessen werden innerhalb der Use Cases die Functional Requirements priorisiert. Diese sind entsprechend ihrer Relevanz und Notwendigkeit für die minimale Erfüllung des jeweiligen Use Cases beschrieben. In der zugehörigen Tabelle ist ersichtlich, welche Functional Requirements für die minimale Umsetzung erforderlich (zwingend) sind und welche darüber hinausgehende Erweiterungen darstellen. So stellen wir sicher, dass jeder Use Case vollständig umgesetzt wird, während gleichzeitig die wichtigsten Funktionen zuerst realisiert werden, um eine minimale, funktionsfähige Lösung sicherzustellen.

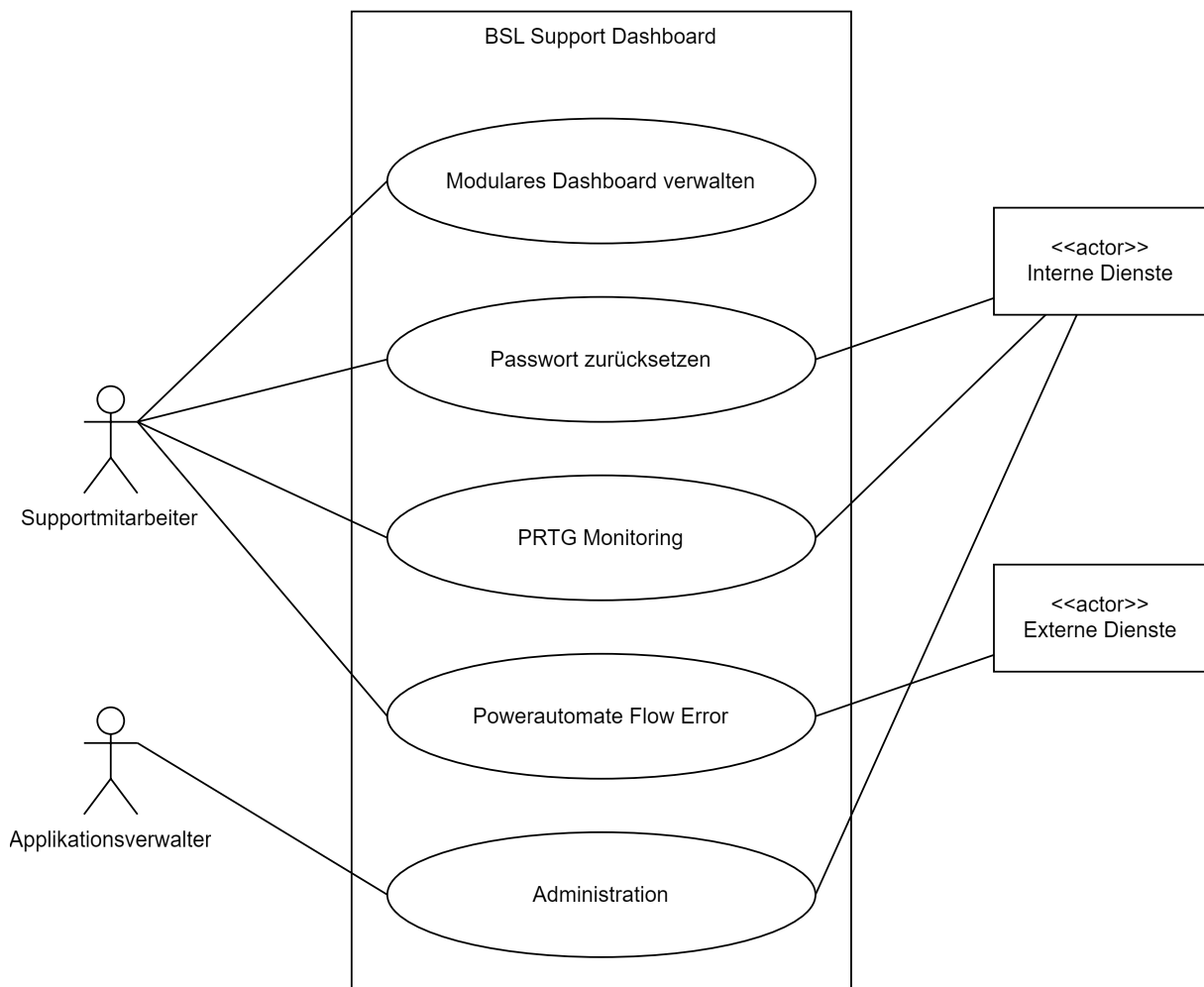


Figure 7: Use Case Diagram

1.2.1 UC01 - Modulares Dashboard verwalten**Involved Actors**

- Supportmitarbeiter

Beschreibung

Der Supportmitarbeiter kann sein Dashboard durch Anpassen von Grösse, Sichtbarkeit und Anordnung der Module personalisieren.

Abgeleitete Functional Requirements

ID	Name	Beschreibung	Zwingend	Prio
FR011	Anordnung ändern	Die Anordnung der Module auf dem Dashboard kann vom Nutzer geändert werden.	Ja	1
FR012	Konfiguration speichern	Das System speichert die Konfiguration pro Nutzerprofil.	Ja	2
FR013	Sichtbarkeit steuern	Die Sichtbarkeit der Module auf dem Dashboard kann vom Nutzer gesteuert werden.	Nein	3
FR014	Grösse anpassen	Die Grösse der Module auf dem Dashboard kann vom Nutzer angepasst werden.	Nein	4

Table 1: Liste aller Functional Requirements, die aus dem Use Case UC01 abgeleitet wurden

1.2.2 UC02 - Passwort zurücksetzen**Involved Actors**

- Supportmitarbeiter

Beschreibung

Der Supportmitarbeiter kann für Lernende oder Mitarbeiter ihr AD Account Passwort zurücksetzen.

Abgeleitete Functional Requirements

ID	Name	Beschreibung	Zwingend	Prio
FR021	Passwort zurücksetzen	Das System muss es ermöglichen, dass der Nutzer ein Passwort mithilfe der Account-E-Mail das Passwort zurücksetzen kann. Das neue Passwort wird dem Lernenden oder Mitarbeiter im System angezeigt.	Ja	1

Table 2: Liste aller Functional Requirements, die aus dem Use Case UC02 abgeleitet wurden

1.2.3 UC03 - PRTG Monitoring

Involved Actors

- Supportmitarbeiter

Beschreibung

Das System zeigt dem Supportmitarbeiter die aktuellen PRTG Monitoring Informationen an.

Abgeleitete Functional Requirements

ID	Name	Beschreibung	Zwingend	Prio
FR031	PRTG Filterung	Das System muss es ermöglichen, dass die PRTG Ergebnisse gefiltert werden können.	Ja	2
FR032	PRTG Suche	Das System muss es ermöglichen, dass der Nutzer die PRTG Ergebnisse durchsuchen kann.	Nein	3
FR033	PRTG Pin	Das System muss es ermöglichen, dass der Nutzer die PRTG Ergebnisse anheften kann.	Nein	4
FR034	PRTG Serverauswahl	Das System muss es ermöglichen, dass der Nutzer die PRTG Ergebnisse auf bestimmte Server beschränken kann	Nein	5

Table 3: Liste aller Functional Requirements, die aus dem Use Case UC03 abgeleitet wurden

1.2.4 UC04 - Powerautomate Flow Error**Involved Actors**

- Supportmitarbeiter

Beschreibung

Das System zeigt dem Supportmitarbeiter die fehlerhaften Powerautomate Flows an und ermöglicht es ihm, diese abzuschliessen.

Abgeleitete Functional Requirements

ID	Name	Beschreibung	Zwingend	Prio
FR041	Powerautomate Filter	Das System muss einen Standard Filter bereitstellen, welcher die erledigten Powerautomate Flows rausfiltert.	Nein	1

Table 4: Liste aller Functional Requirements, die aus dem Use Case UC04 abgeleitet wurden

1.2.5 UC05 - Administration**Involved Actors**

- Applikationsverwalter

Beschreibung

Der Applikationsverwalter kann administrative Änderungen im BSL Support Dashboard vornehmen.

Abgeleitete Functional Requirements

ID	Name	Beschreibung	Zwingend	Prio
FR051	Serviceaccount Umstellung	Das System muss es ermöglichen, dass der Applikationsverwalter die Serviceaccounts für die Applikation ändern kann.	Nein	5
FR052	Accessmanagement	Das System muss es ermöglichen, dass der Applikationsverwalter das Accessmanagement für die Applikation verwalten kann.	Nein	5

Table 5: Liste aller Functional Requirements, die aus dem Use Case UC05 abgeleitet wurden

1.3 Non-Functional Requirements

In diesem Kapitel sind die Non-Functional Requirements nach FURPS dokumentiert.

1.3.1 Erklärung

FRUPS unterteilt die Qualitätsanforderungen an Software in folgende Gruppen:

- Functionality - NFR0XX
- Usability - NFR1XX
- Reliability - NFR2XX
- Performance - NFR3XX
- Supportability - NFR4XX

Je tiefer die Zahl bei der Priorisierung ist, desto wichtiger ist das Requirement. Für den MVP nicht relevante Requirements werden mit keiner Priorität versehen, da diese erst zu einem späteren Zeitpunkt geplant werden würden.

1.3.2 Functionality

ID	Name	Akzeptanzkriterium	Verifizierung	Priorität
NFR001	Nur berechnigte Benutzer dürfen die jeweils vorgesehenen Module sehen und nutzen.	Ein Benutzer ohne entsprechende Berechtigungen kann keine Module aufrufen, für die ihm die Rechte fehlen	Durchführung von Zugriffstest mit unterschiedlichen Benutzerrollen	1
NFR002	Alle Benutzereingaben müssen validiert werden, um fehlerhafte oder sicherheitskritische Daten zu verhindern.	Ungültige Eingaben werden erkannt und entsprechende Fehlermeldungen angezeigt	Durchführung von Tests mit verschiedenen ungültigen Eingaben	2

Table 6: Functionality Non-Functional Requirements

1.3.3 Usability

ID	Name	Akzeptanzkriterium	Verifizierung	Priorität
NFR101	Fehlermeldungen müssen klar und verständlich formuliert sein	Alle Fehlermeldung enthalten eine verständliche Beschreibung des Problem, ohne technische Details preiszugeben	Usability-Test mit Testnutzer	1
NFR102	Fehlermeldungen müssen in einem einheitlichen Design präsentiert werden, unabhängig vom Modul	Alle Fehlermeldungen verwenden dieselbe Formatierung (Farbe, Icons, Layout)	Visuelle Inspektion in des UI	2

Table 7: Usability Non-Functional Requirements

1.3.4 Reliability

ID	Name	Akzeptanzkriterium	Verifizierung	Priorität
NFR201	Alle Fehler müssen in einem zentralen Log gespeichert werden	Jeder im System auftretende Fehler wird automatisch im zentralen Fehler-Log erfasst	Überprüfung der Log-Dateien nach provozierten Fehlern, Review der Logging-Konfiguration	1
NFR202	Alle Passwort-Resets müssen in einem separaten Log protokolliert werden	Jeder durchgeführte Passwort-Reset wird mit Zeitstempel und Benutzererkennung im Passwort-Reset-Log dokumentiert	Funktionstest durch mehrfaches Passwort-Reset, anschliessende Log-Analyse	1
NFR203	Logs im Modul Powerautomate Flow Errors dürfen keine sensiblen personenbezogenen Daten enthalten	Kein Log-Eintrag enthält sensible personenbezogene Daten	Log-Analyse	3

Table 8: Reliability Non-Functional Requirements

1.3.5 Supportability

ID	Name	Akzeptanzkriterium	Verifizierung	Priorität
NFR401	Die Anwendung muss modular aufgebaut sein, um die einfache Integration neuer Module zu ermöglichen	Neue Module können durch Konfiguration oder klar definierte Schnittstellen eingebunden werden, ohne Anpassung am Kernsystem	Code-Review der Architektur und Schnittstellenbeschreibung	2

Table 9: Supportability Non-Functional Requirements

2 Architecture

In dem folgenden Kapitel wird die Architektur der Applikation beschrieben.

2.1 Systemarchitektur

Die Systemarchitektur des BSL Support Dashboards folgt dem klassischen 3-Tier-Muster und ist in folgender Grafik dargestellt.

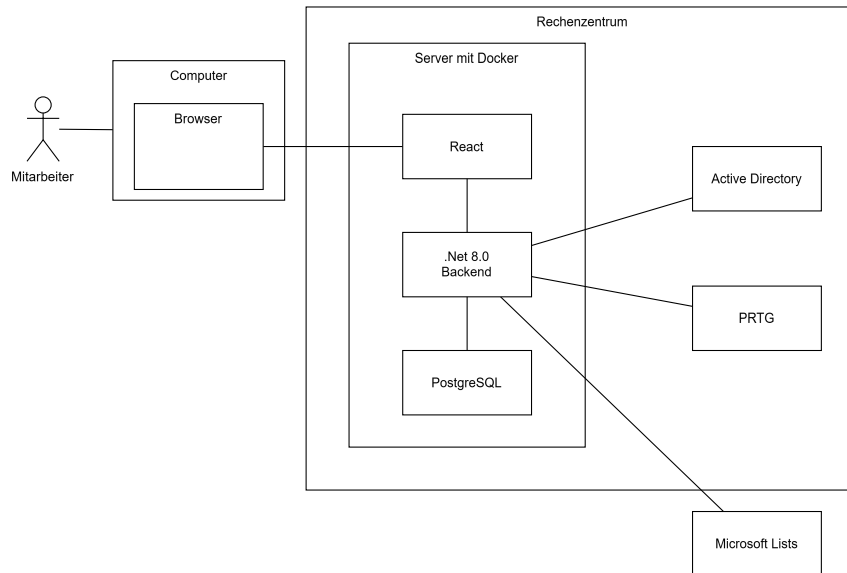


Figure 8: Systemarchitektur des BSL Support Dashboards

Die Architektur besteht aus folgenden Hauptkomponenten:

Präsentationsschicht: Das React-Frontend ist als Docker-Container auf dem Server deployed und wird von dort an den Browser des Mitarbeiters ausgeliefert. Die Single-Page-Applikation kommuniziert über REST-API-Aufrufe mit dem Backend.

Logikschicht: Das .NET 8.0 Backend läuft ebenfalls als Docker-Container auf dem Server. Es verarbeitet die gesamte Geschäftslogik, orchestriert die Kommunikation mit den externen Systemen, validiert Anfragen, authentifiziert Benutzer mittels JWT-Tokens und stellt eine sichere Schnittstelle zu sensiblen Ressourcen bereit.

Datenschicht: Die PostgreSQL-Datenbank ist als dritter Docker-Container auf dem Server deployed und speichert persistente Daten wie Dashboard-Konfigurationen.

Docker-Deployment: Alle drei Komponenten (React-Frontend, .NET Backend, PostgreSQL-Datenbank) laufen als separate Docker-Container auf dem Server und werden über Docker Compose orchestriert. Diese Container-Architektur ermöglicht eine einfache Skalierung, Wartung und Portabilität der Anwendung.

Externe Integrationen: Das Backend bindet drei externe Systeme an:

- **Active Directory:** Authentifizierung der Mitarbeiter und Passwort-Reset-Funktionalität für das AD-Modul
- **PRTG:** Abruf von Server- und Sensor-Status über die PRTG-API für das Monitoring-Modul
- **Microsoft Lists:** Überwachung von Power Automate Flow-Runs via Microsoft Graph API für das Flow-Error-Modul

2.2 Sicherheitskonzept

Der folgende Abschnitt beschreibt umfassend, wie das Sicherheitskonzept in der Applikation konzipiert und technisch umgesetzt wurde. Dabei werden verschiedene sicherheitsrelevante Aspekte behandelt, die für den Schutz der Anwendung und der Benutzerdaten essenziell sind. Die Hauptaspekte umfassen die mehrstufige Authentifizierung in der Applikation sowie die durchgängige Absicherung der API-Schnittstellen.

2.2.1 Login

In unserer Applikation wurde ein differenziertes Login-Konzept mit zwei verschiedenen Authentifizierungsmechanismen implementiert, um unterschiedliche Sicherheitsanforderungen optimal abzudecken.

2.2.1.1 Primärer Login (Applikationszugriff)

Der erste Login-Mechanismus stellt die zentrale Zugriffskontrolle der Applikation dar und wird unmittelbar nach dem Start der Applikation aufgerufen. Dieses Login dient als Hauptauthentifizierung für die gesamte Anwendung.

Die Zugriffsberechtigung wird über die Integration mit Active Directory-Gruppen gesteuert. Dadurch kann zentral und granular kontrolliert werden, welche Benutzer oder Benutzergruppen Zugriff auf das Tool erhalten. Diese Anbindung an bestehende AD-Strukturen ermöglicht eine konsistente Rechteverwaltung im Einklang mit der organisationsweiten IT-Sicherheitsstrategie.

Ein weiterer Vorteil dieser Authentifizierung besteht darin, dass alle vom Benutzer vorgenommenen Einstellungen und Konfigurationen bei der Speicherung automatisch dem jeweiligen Benutzer zugeordnet werden können.

2.2.1.2 Sekundärer Login (AD-Passwort zurücksetzen)

Der zweite Login-Mechanismus wurde speziell für das hochsensible Modul "AD-Passwort zurücksetzen" entwickelt und implementiert eine zusätzliche Sicherheitsebene.

Ohne erfolgreiche Authentifizierung über diesen zweiten Login wird das Modul vollständig verborgen und dem Benutzer lediglich das Login-Fenster angezeigt. Diese Designentscheidung verhindert, dass unbefugte Benutzer überhaupt Kenntnis von der Existenz dieser privilegierten Funktionalität erlangen.

Die Notwendigkeit dieses separaten Login-Mechanismus ergibt sich aus den spezifischen Sicherheitsanforderungen der Berufsschule Lenzburg: Das Zurücksetzen von Active Directory-Passwörtern ist dort ausschliesslich mit einem dedizierten Service-Account mit erweiterten Berechtigungen möglich. Durch die Trennung der beiden Login-Prozesse wird sichergestellt, dass normale Benutzer mit ihren Standard-Credentials keinen Zugriff auf diese kritische Funktionalität erhalten, selbst wenn sie grundsätzlich zur Nutzung der Applikation berechtigt sind.

2.2.1.3 Authentifizierungsprozess und Token-Management

Für beide Login-Mechanismen folgt der Authentifizierungsprozess einem einheitlichen und bewährten Sicherheitsprotokoll.

Benutzer müssen jeweils ihren AD-Benutzernamen sowie das zugehörige AD-Passwort eingeben. Diese Credentials werden über eine verschlüsselte Verbindung an das Backend übermittelt und dort gegen das Active Directory validiert. Die Authentifizierung erfolgt somit stets gegen die zentrale Benutzerverwaltung, wodurch eine einheitliche Passwort-Policy gewährleistet wird.

Bei erfolgreicher Validierung der Credentials generiert das Backend einen JSON Web Token (JWT) für den authentifizierten Benutzer. Dieser Token enthält relevante Benutzerinformationen und Berechtigungen in verschlüsselter Form und wird an den Client zurückgegeben. Der JWT-Token dient als Nachweis der erfolgreichen Authentifizierung für alle nachfolgenden Interaktionen mit der Applikation und der API.

2.2.2 Absicherung der API

Die API-Absicherung bildet einen zentralen Pfeiler des Sicherheitskonzepts und stellt sicher, dass ausschliesslich authentifizierte und autorisierte Benutzer auf die Backend-Funktionalitäten zugreifen können.

Um einen umfassenden Schutz zu gewährleisten, implementiert die API eine strikte Token-basierte Authentifizierung: Bei jedem API-Aufruf muss der Client einen gültigen JWT-Token im Authorization-Header der HTTP-Anfrage mitschicken. Das Backend validiert diesen Token vor der Ausführung jeglicher Geschäftslogik.

Anfragen ohne Token oder mit einem ungültigen, abgelaufenen oder manipulierten Token werden automatisch mit einem HTTP 401 (Unauthorized) Status-Code abgelehnt und nicht weiter verarbeitet. Dadurch wird verhindert, dass unauthentifizierte Zugriffe auf sensible Daten oder Funktionen erfolgen können.

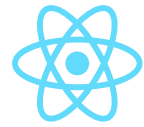
2.3 Technologie

Im folgenden Abschnitt werden die Entscheidungen für unsere verwendeten Technologien begründet.

2.3.1 React

Für den Frontend-Bereich unserer Applikation haben wir uns auf React geeinigt. React setzte sich gegenüber anderen Frameworks durch, da es uns durch seine einfache und flexible Struktur besonders zugesagt hat.

Ein weiterer Vorteil von React liegt in der komponentenbasierten Architektur. Dadurch können wir einmal erstellte Komponenten mehrfach wiederverwenden und so die Entwicklung effizienter und übersichtlicher gestalten.



2.3.2 .Net

Für das Backend unserer Applikation haben wir .Net gewählt. Der Vorteil von .Net in diesem Projekt liegt darin, dass beide Teammitglieder bereits erste Erfahrungen mit .Net machen konnten. Zudem hat .Net eine einfache Anbindung an AD, was für unser Projekt sehr wichtig ist.



2.3.3 PostgreSQL

Zur Speicherung der Daten in unserem Projekt haben wir uns für eine PostgreSQL-Datenbank entschieden. Diese Datenbank ist uns aus dem Modul „Datenbanken“ bekannt und wir haben bisher nur positive Erfahrungen damit gemacht. Zudem sind alle Teammitglieder mit dem zugehörigen Verwaltungstool pgAdmin vertraut. Ebenso ist PostgreSQL eine Open Source Lösung, somit kann die Datenbank komplett auf unsere Bedürfnisse angepasst werden.



2.3.4 Docker

Um das Frontend, das Backend und die Datenbank auf dem Server bereitzustellen, haben wir uns für den Einsatz von Docker entschieden. Docker ist eine weit verbreitete Plattform zur Containerisierung von Anwendungen. Im Gegensatz zur klassischen Virtualisierung virtualisiert Docker keine ganzen Betriebssysteme, sondern isoliert Anwendungen leichtgewichtig in Containern. Dies ermöglicht es uns, die Ressourcen pro Service gezielt zu steuern und die Auslastung des Servers effizient zu optimieren.



2.3.5 Swagger

Für die Dokumentation unserer API haben wir Swagger gewählt. Swagger ist in der Industrie weit verbreitet, um APIs zu dokumentieren.



2.3.6 GitLab

Für die Versionsverwaltung und Zusammenarbeit im Team verwenden wir GitLab. GitLab bietet uns nicht nur ein zentrales Repository, sondern unterstützt uns auch aktiv im Entwicklungsprozess durch integrierte Funktionen wie Merge Requests oder Code Reviews. Zusätzlich nutzen wir die CI/CD-Pipeline von GitLab, um unsere Anwendung und die Dokumentation automatisiert zu builden, zu testen und zu deployen. Dadurch können wir Fehler frühzeitig erkennen und sicherstellen, dass neue Änderungen zuverlässig und konsistent ausgeliefert werden.



2.4 CI/CD

Unser Projekt liegt als Monorepo auf dem Gitlab (unter [Gitlab](#)).

Die Gitlab Pipeline verarbeitet den Sourcecode zu Docker Images (und Dokumentation) und legt sie in unserem Projekt Registry ab. Von dort können sie dann in einem weiteren manuellen Schritt auf unserer Server Instanz per Docker-Compose deployt werden.

2.4.1 Gitlab

Feature-Branches werden in den Main-Branch gemerget, wobei die MR-Pipeline fehlerfrei durchlaufen muss. Zudem muss per fast-forward auf Main gemerget werden. Die Commits des Feature-Branches werden vor dem Merge gesquasht, damit die Commithistory übersichtlich bleibt.

2.4.2 Docker

Die Frontend und Backend Container werden in der Pipeline mittels Dind aus Dockerfiles gebildet. Docker-Compose wird auf beiden Environments unserer Entwicklung gebraucht (local(dev), server(prod)). Die benötigten Secrets kommen aus lokalen Umgebungsvariablen (zB. per .env).

Unsere eigenen Images werden auf der Registry des Projekts gespeichert und geladen.

Folgende Container kommen zum Einsatz:

- **frontend**: Enthält die React-Anwendung für das Support Dashboard.
- **backend**: Enthält die .Net-Anwendung für das Backend des Support Dashboards.
- **postgres**: Enthält die PostgreSQL-Datenbank für das Support Dashboard und wird aus dem Internet bezogen.

Diese Images befinden sich alle im selben Netzwerk und können über ihre jeweiligen Servicenamen miteinander kommunizieren. Der Postgres Container verwendet ein lokales Volume.

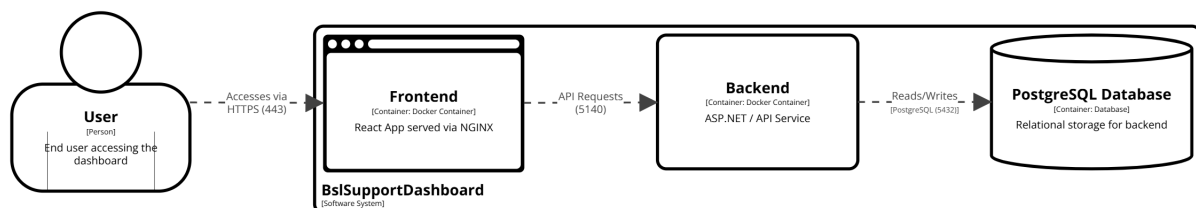


Figure 9: C4 Diagramm der Container

2.4.3 Pipeline

Die Pipeline ist in mehrere Teile unterteilt:

- Im ersten Teil, der aus der Typst Dokumentation ein PDF generiert. Dieses PDF wird fix auf den Feature-Branch committet. Das stellt sicher, dass keine Typst-Infrastruktur benötigt wird um ein Setup durchzuführen (welche erst im generierten PDF beschrieben ist). Nach dem Commit wird die Pipeline mit einem Fehler gestoppt, aber der Commit triggert eine neue Pipeline. Das wird nur ausgeführt, wenn tatsächlich Veränderungen im documentation-Ordner sind.
- Im zweiten Teil wird unsere Applikation getestet, gebuildet und released. Das wird nur ausgeführt, wenn Veränderungen im project-Ordner sind. Die release-Stage wird nur auf der Main-Branch-Pipeline ausgeführt. Alle Schritte müssen "passen", um zu mergen.
- Als letztes gibt es noch einen default-pass Job, der immer auf MR-Pipelines succeedet. Ohne diesen könnte es passieren, dass bei einem MR kein Job läuft und daher nicht gemerget werden kann.

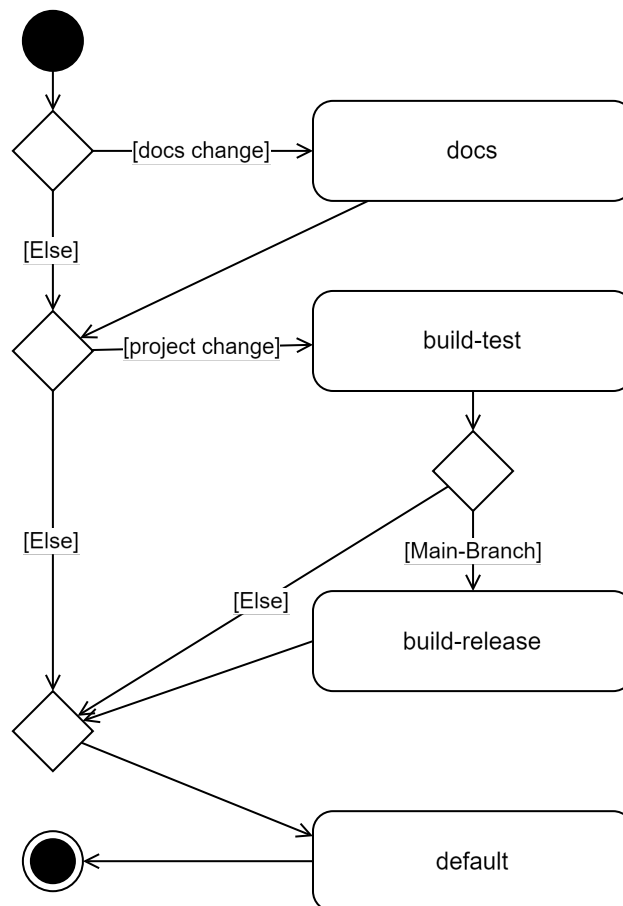


Figure 10: Stages der Pipeline

Erwähnenswert ist auch, dass die “docs”-Stage nach dem Pushen des neuen PDFs der Dokumentation die Pipeline effektiv abbricht. Aufgrund des neuen Commits wird eine neue Pipeline gestartet, die die Jobs der “docs”-Stage bypassst.

Dank der Containerisierung kann dieses Deployment für eine zukünftige Skalierung sehr flexibel angepasst werden. Man könnte zum Beispiel mehrere Container parallel laufen lassen oder die Datenbank separat auf einen dedizierten Server deployen.

3 Umsetzung

In den folgenden Kapiteln werden die einzelnen Module der Applikation beschrieben.

3.1 Login und Dashboard

In diesem Kapitel wird das erste Login und das Dashboard beschrieben.

3.1.1 Kurzbeschreibung

Durch das Dashboard können wir die verschiedenen Module, die in den folgenden Kapiteln beschreiben werden, an einem zentralen Ort darstellen. Zudem kann das Dashboard durch die Einstellungen, welche im Kapitel “Einstellungen” beschreiben werden, individuell angepasst werden. Um die Einstellungen einem User zuzuordnen und den Zugriff auf das Tool einzuschränken, wurde ein Login implementiert. Das Login ist im Kapitel “Sicherheitskonzept” genauer beschreiben.

3.1.2 Aktivitätsdiagramm

Die folgende Grafik zeigt einen typischen Ablauf der Benutzung des Logins und des Dashboards:

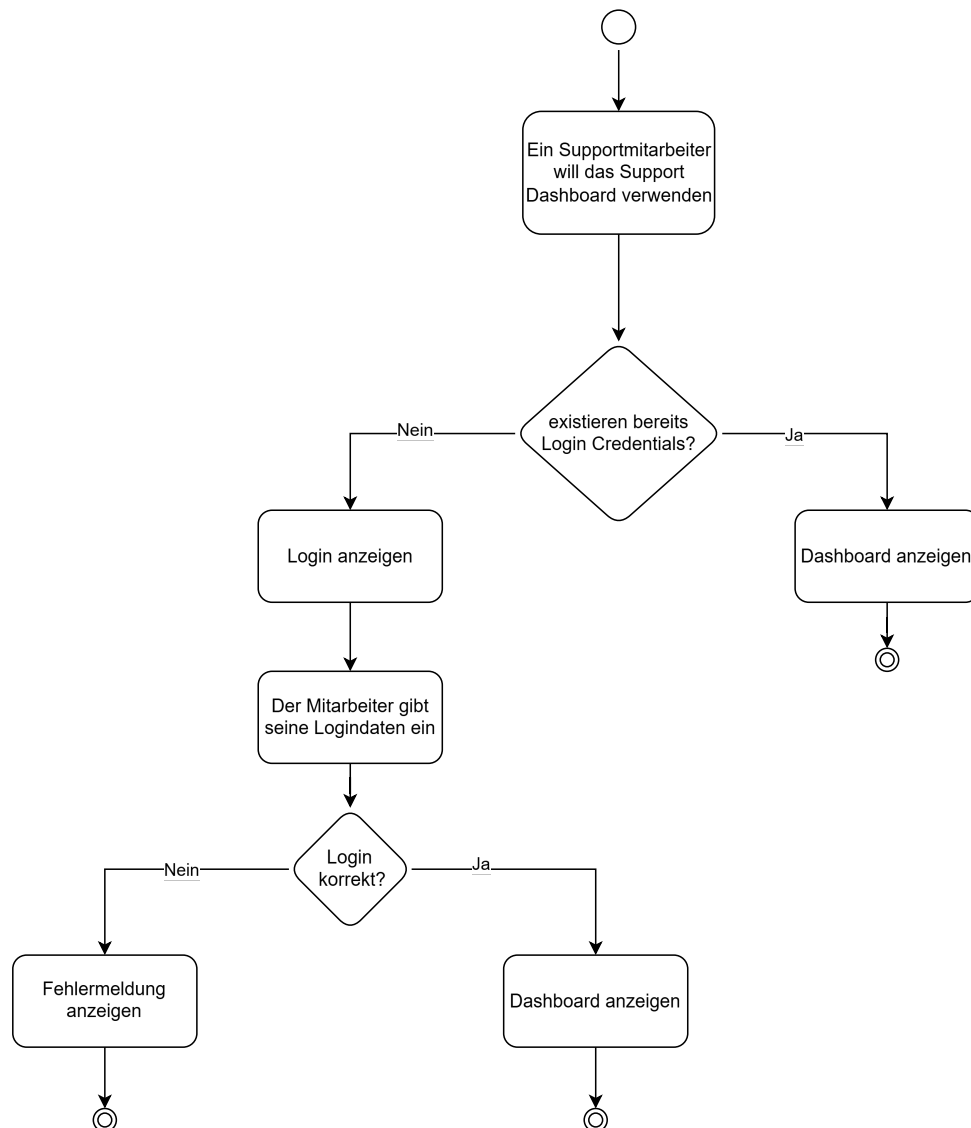


Figure 11: Aktivitätsdiagramm des Logins und des Dashboards

3.1.3 Mockups

Als Startpunkt, wurden von diesem Modul Mockups gezeichnet und einem Mitarbeiter aus dem Support Team der Berufsschule Lenzburg gezeigt.

Auf dem Dashboard können auf die erstellten Module angezeigt werden.

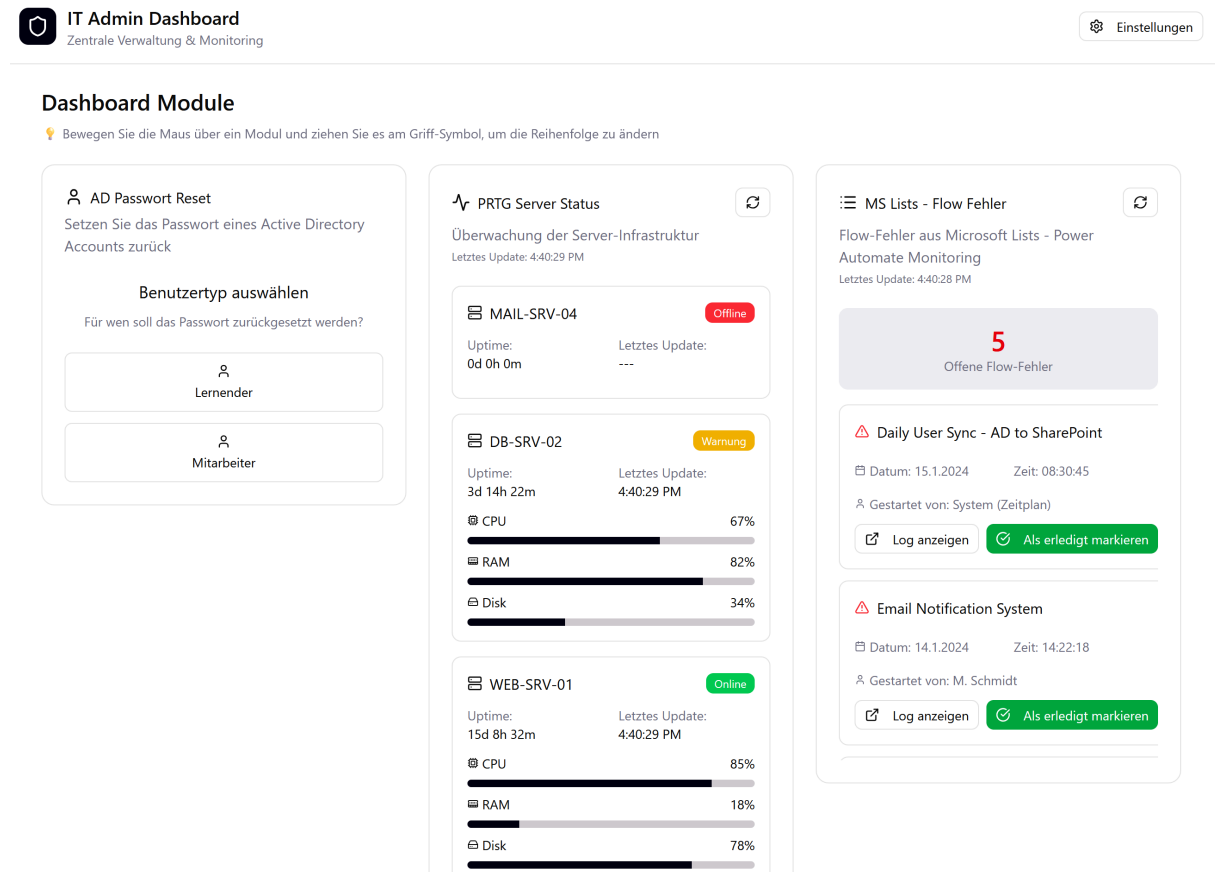


Figure 12: Mockup: Dashboard

3.1.4 Feedback aus dem Supportteam

Die gezeichneten Mockups wurden einem Mitarbeiter des Supportteams gezeigt. Das Feedback wurde dann in die finale Version, im Programm eingearbeitet.

Folgende Punkte wurden als Feedback mitgeteilt:

- Cooles Layout
- Übersichtliche Titel, nicht immer prägnant

3.1.5 Umsetzung

Mit dem Feedback aus dem Supportteam und den Mockups wurde die finale Version, welche auch implementiert wurde, entwickelt.

Das Login der Applikation sieht wie folgt aus:

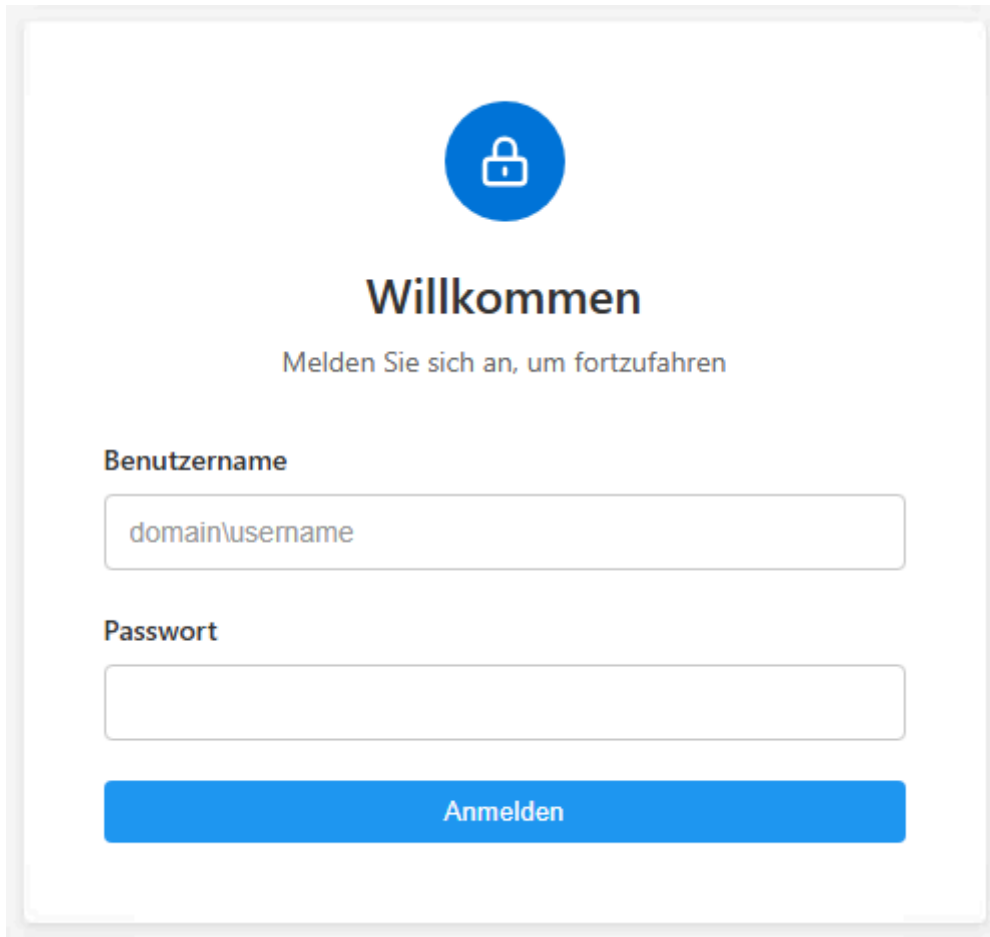
The image shows a login form for an application. At the top center is a blue circular icon containing a white padlock. Below this icon, the word "Willkommen" is written in a large, bold, black font. Underneath "Willkommen" is the text "Melden Sie sich an, um fortzufahren" in a smaller, regular black font. The form contains two input fields: the first is labeled "Benutzername" and contains the text "domain\username"; the second is labeled "Passwort" and is empty. Below these fields is a blue button with the text "Anmelden" in white.

Figure 13: Login der Applikation

Das Dashboard der Applikation sieht wie folgt aus:

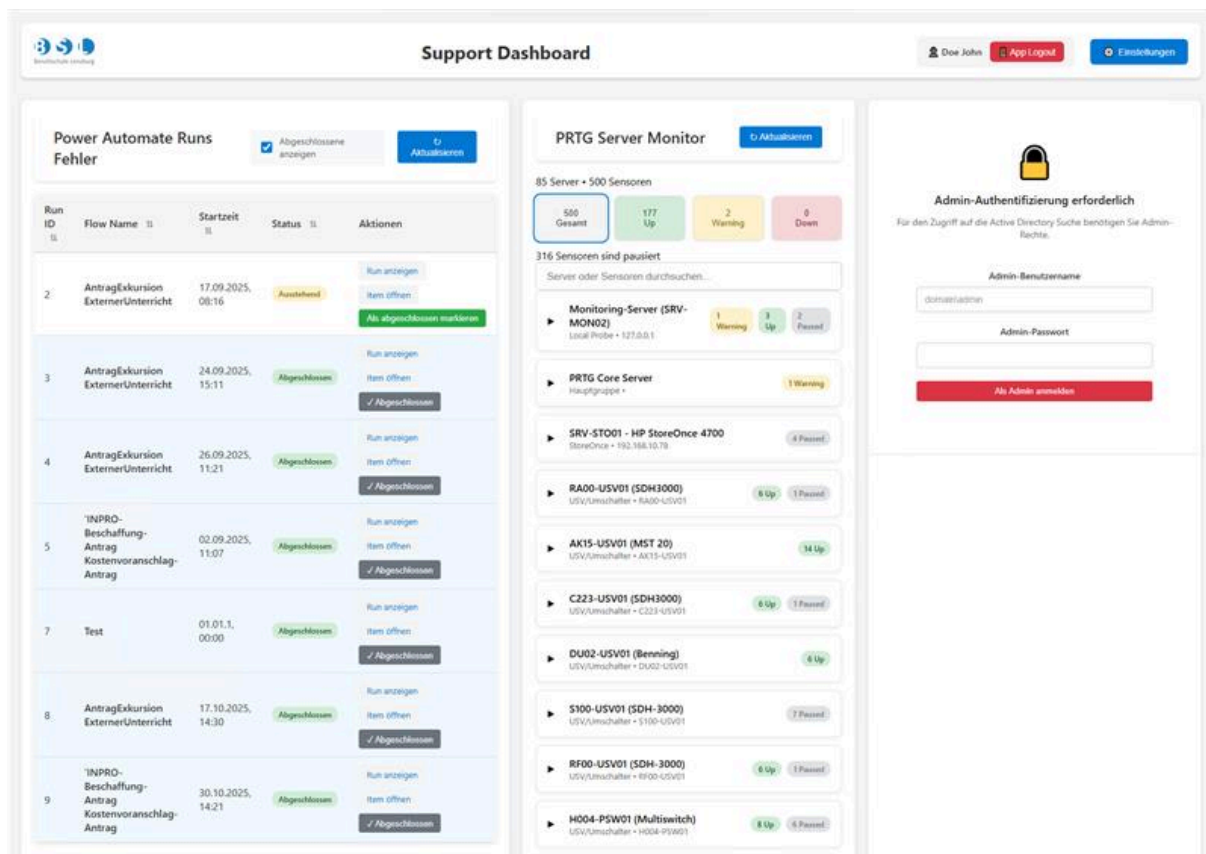


Figure 14: Dashboard der Applikation

3.1.6 Tests

Nach der Entwicklung des Modules wurde ein Usability Test mit dem Leiter der IT der Berufsschule Lenzburg durchgeführt. Dabei testete er das Modul ohne eine Einweisung, folgender Punkt ist dabei aufgefallen:

- Die Titel der Module sahen nicht gleich aus

Dieser Punkt wurde dann noch in der finalen Version implementiert.

3.2 AD Modul Beschreibung

In diesem Kapitel wird das Modul AD Passwort Reset beschrieben.

3.2.1 Kurzbeschreibung

Durch diese Modul soll der Prozess um Passwort zurücksetzen für den Support der Berufsschule Lenzburg deutlich vereinfacht werden. Aktuell muss man dafür auf einen Jumphost connecten und dann über die Active Directory Suche den Benutzer suchen. In einem weiteren Schritt muss man sich ein Passwort ausdenken und dem Benutzer dieses Passwort als sein neues Passwort setzen. Als letztes muss man das ausgedachte Passwort dem Benutzer übergeben.

Mithilfe von diesem Modul soll der oben beschriebene Prozess soweit vereinfacht werden, dass man die Benutzer über die Email-Adresse, den Benutzernamen oder den Anzeigenamen suchen kann. Danach kann mit einem einfachen Button Klick ein neu generiertes Passwort gesetzt und dem Benutzer angezeigt werden.

3.2.2 Aktivitätsdiagramm

Die folgende Grafik zeigt einen typischen Ablauf der Benutzung des Modules AD Passwort Reset:

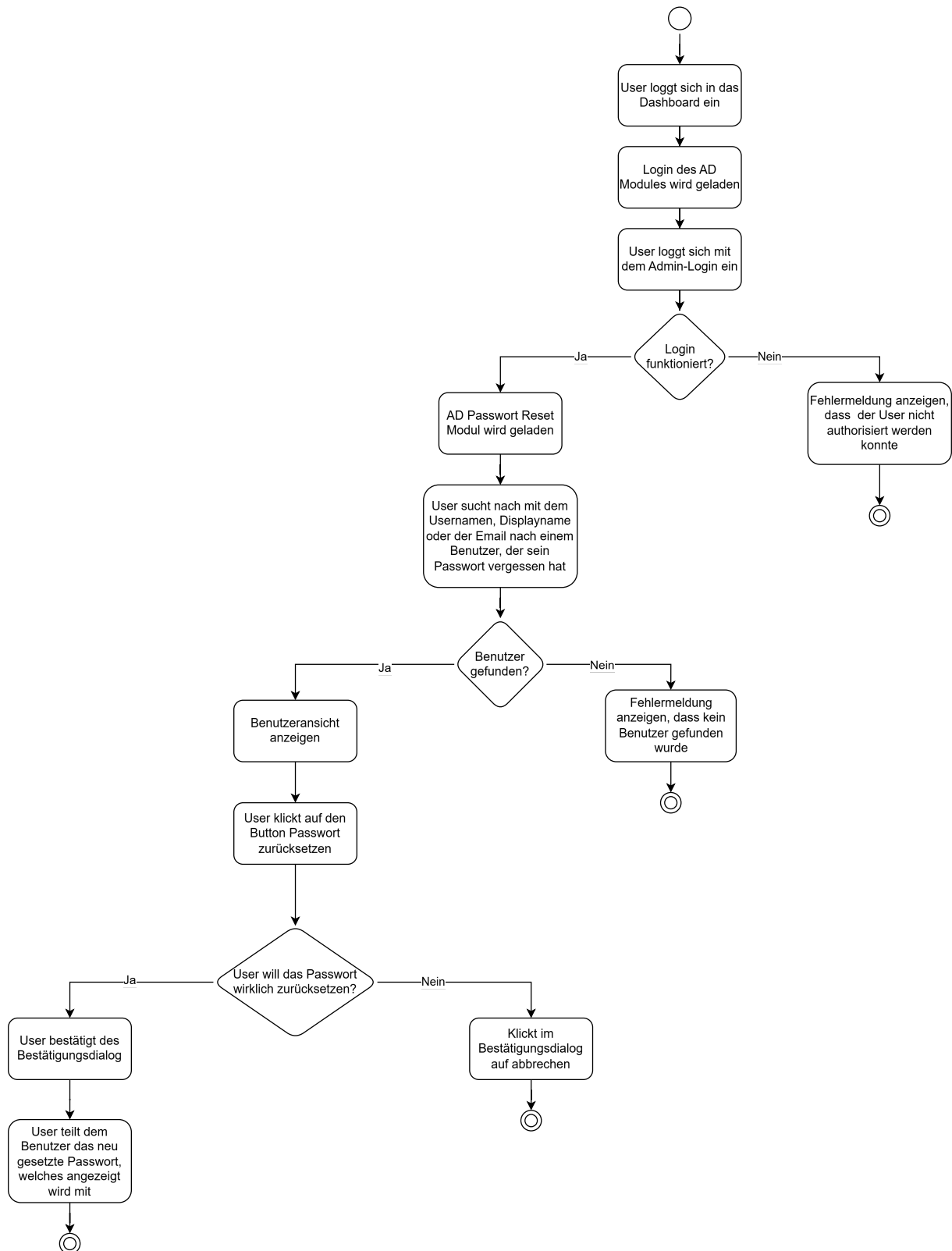


Figure 15: Aktivitätsdiagramm des Modules AD Passwort Reset

3.2.3 Mockups

Als Startpunkt, wurden von diesem Modul Mockups gezeichnet und einem Mitarbeiter aus dem Support Team der Berufsschule Lenzburg gezeigt.

Da das Modul für den Support von Lernenden sowie den Mitarbeitern verantwortlich ist, muss man als erstes die Funktion des Benutzers ausgewählt werden.

The mockup shows a screen titled 'AD Passwort Reset' with a subtitle 'Setzen Sie das Passwort eines Active Directory Accounts zurück'. Below this is a section 'Benutzertyp auswählen' with the question 'Für wen soll das Passwort zurückgesetzt werden?'. There are two buttons: 'Lernender' and 'Mitarbeiter', each with a person icon above the text.

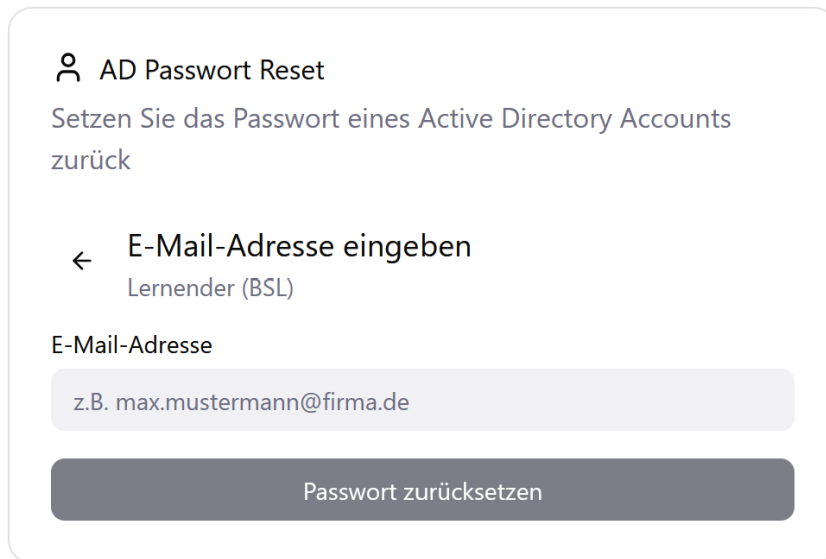
Figure 16: Mockup: Auswahl der Funktion des Benutzers

Der Support der Berufsschule Lenzburg ist für die Berufsschule, sowie auch für das Weiterbildungszentrum verantwortlich. Daher war die Überlegung, dass man als erstes auswählen kann, aus welchem Bereich der Benutzer kommt.

The mockup shows a screen titled 'AD Passwort Reset' with a subtitle 'Setzen Sie das Passwort eines Active Directory Accounts zurück'. Below this is a section 'Standort auswählen' with a left arrow icon and the question 'An welchem Standort ist der Lernende?'. There are two buttons: 'BSL Berufliche Schule' and 'WBZ Weiterbildungszentrum', each with the location name above the text.

Figure 17: Mockup: Auswahl des Bereiches vom Benutzer

Um das Passwort des Benutzers zurückzusetzen, muss der Benutzer eindeutig identifiziert werden können. Daher muss man als nächstes die Email-Adresse des Benutzers, welcher ein neues Passwort erhalten soll eingeben.



The mockup shows a user interface for resetting an AD password. It starts with a user icon and the title 'AD Passwort Reset'. Below this is the instruction 'Setzen Sie das Passwort eines Active Directory Accounts zurück'. A back arrow is followed by the heading 'E-Mail-Adresse eingeben' and the user role 'Lernender (BSL)'. The label 'E-Mail-Adresse' is positioned above a text input field containing the example email 'z.B. max.mustermann@firma.de'. At the bottom is a dark button labeled 'Passwort zurücksetzen'.

👤 AD Passwort Reset

Setzen Sie das Passwort eines Active Directory Accounts zurück

← E-Mail-Adresse eingeben
Lernender (BSL)

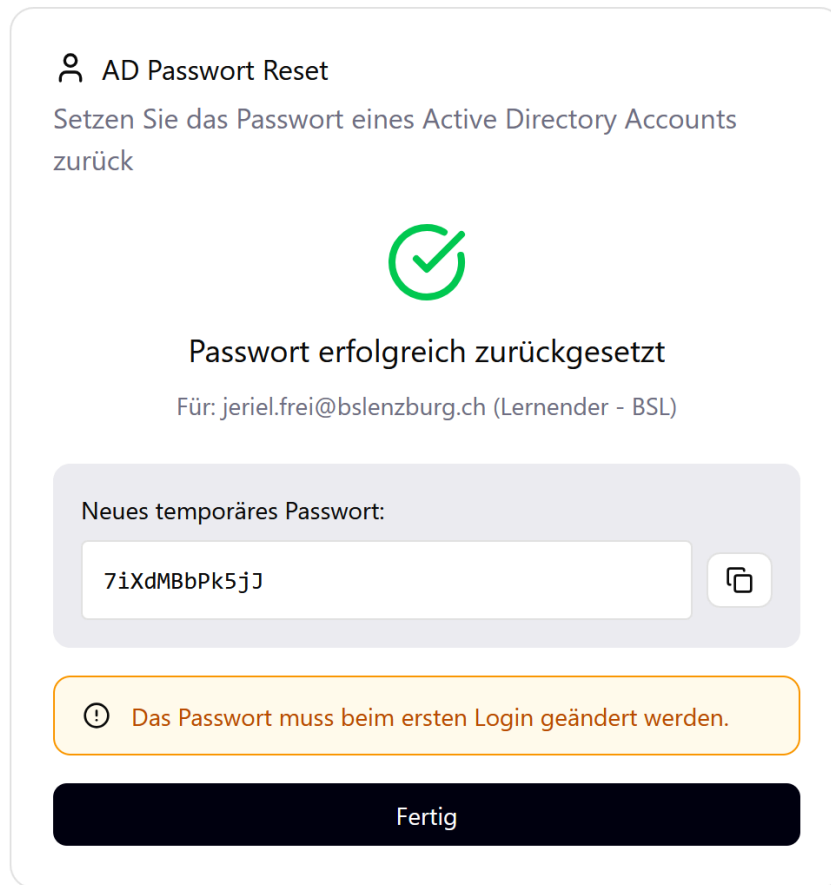
E-Mail-Adresse

z.B. max.mustermann@firma.de

Passwort zurücksetzen

Figure 18: Mockup: Eingabe der Email-Adresse

Nachdem das Passwort erfolgreich zurückgesetzt wurde, wird das Passwort dem Benutzer angezeigt. Zusätzlich sollte das Flag, dass das Passwort beim nächsten Login geändert werden muss, gesetzt werden.



AD Passwort Reset

Setzen Sie das Passwort eines Active Directory Accounts zurück

Passwort erfolgreich zurückgesetzt

Für: jeriel.frei@bslenzburg.ch (Lernender - BSL)

Neues temporäres Passwort:

7iXdMBbPk5jJ

Das Passwort muss beim ersten Login geändert werden.

Fertig

Figure 19: Mockup: Anzeige des neuen Passworts

3.2.4 Feedback aus dem Supportteam

Die gezeichneten Mockups wurden einem Mitarbeiter des Supportteams gezeigt. Das Feedback wurde dann in die finale Version, im Programm eingearbeitet.

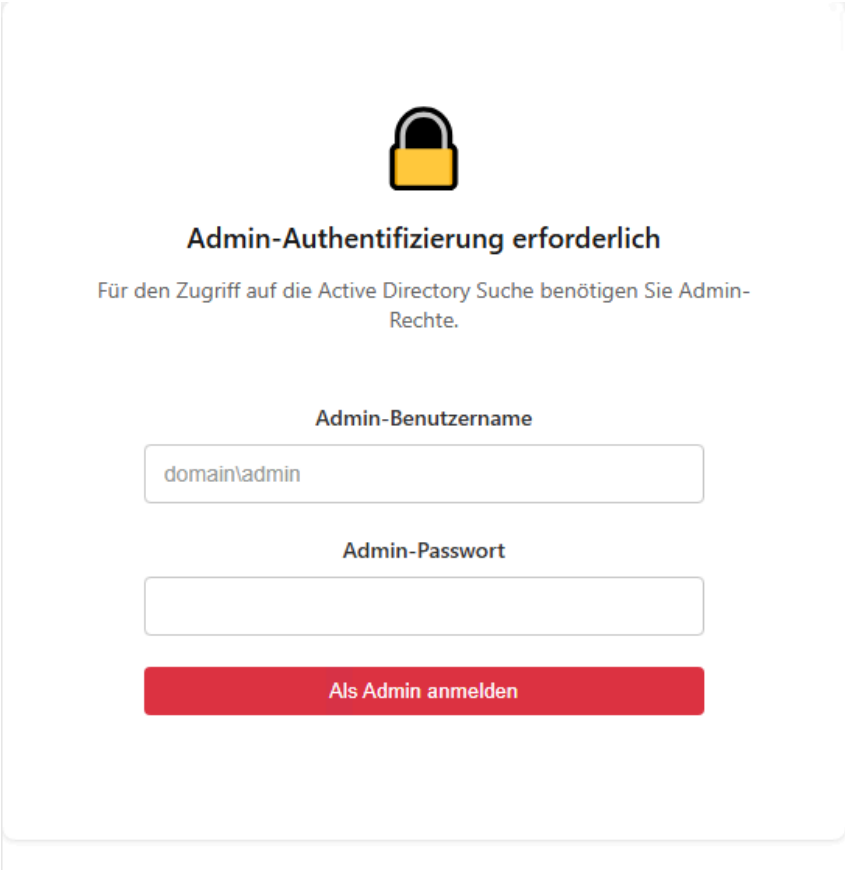
Folgende Punkte wurden als Feedback mitgeteilt:

- Ein Benutzer sollte über den Benutzernamen oder die Email-Adresse gesucht werden können, das hat den Hintergrund, dass sonst Benutzer mit dem gleichen Namen nicht richtig gefunden werden, da sie dann eine 1 in der Email-Adresse haben.
- Wenn es mehrere Benutzer mit dem gleichen Namen gibt, soll nicht nur der Erste angezeigt werden, sondern es sollen alle Benutzer mit diesem Namen angezeigt werden.
- Es sollen noch die Informationen zur Klasse, welche in der Beschreibung des Benutzers hinterlegt sind, angezeigt werden. Damit man den User besser verifizieren kann.

3.2.5 Umsetzung

Mit dem Feedback aus dem Supportteam und den Mockups wurde die finale Version, welche auch implementiert wurde, entwickelt.

Da nur spezielle Benutzer im Umfeld der BSL die Rechte besitzen um AD Passwörter zurückzusetzen, wurde ein zweites Login angelgt. Bei diesem Login muss der Benutzer, der sich einloggen will, eine spezielle Benutzergruppe haben, diese Gruppe kann im .env File angepasst werden.





Admin-Authentifizierung erforderlich

Für den Zugriff auf die Active Directory Suche benötigen Sie Admin-Rechte.

Admin-Benutzername

domain\admin

Admin-Passwort

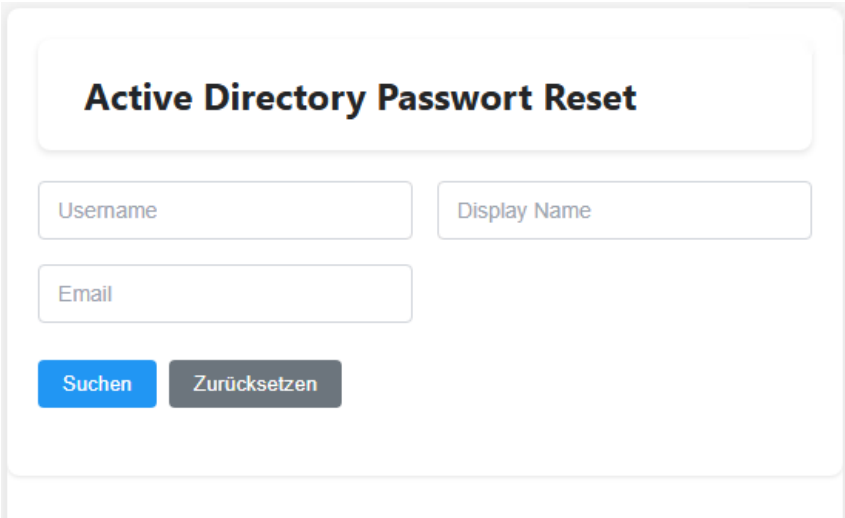
Als Admin anmelden

Figure 20: Login Seite des AD Modules

Sobald man sich erfolgreich eingeloggt hat, kann man nach einem Benutzer über den Benutzernamen, die Email-Adresse oder dem Namen des Benutzers suchen. Folgendes Mapping wird für die Suche verwendet:

Suchfeld	AD Feld
Benutzername	sAMAccountName
Email-Adresse	mail
Name	displayName

Table 10: Mapping zwischen den Suchfeldern und den AD Feldern



The screenshot shows a web interface titled "Active Directory Passwort Reset". Below the title, there are three input fields: "Username", "Display Name", and "Email". At the bottom of the form, there are two buttons: a blue "Suchen" button and a grey "Zurücksetzen" button.

Figure 21: Benutzersuche im AD Modul

Nachdem auf den Button Suche geklickt wurde, werden alle Benutzer die den Suchkriterien entsprechen, angezeigt. Im folgenden Bild sieht man welche Werte alles angezeigt werden.

The screenshot shows a web interface titled "Active Directory Passwort Reset". It features a search section with input fields for "john.doe" (labeled "Display Name") and "Email". Below these are two buttons: "Suchen" (blue) and "Zurücksetzen" (grey). The results section, titled "Suchergebnisse (1)", displays user details for "Doe John":
- Display Name: Doe John
- Username: john.doe
- Email: john.doe@stud.bslenzburg.ch
- Distinguished Name: CN=Doe John,OU=Benutzer,OU=stud,DC=BSL,DC=lan
- Description: Testbenutzer Berufslernender Grundbildung & BM
- Account Status: ✔ Aktiv
At the bottom of the results box is a blue button with a key icon labeled "Passwort zurücksetzen".

Figure 22: Suchergebnisse im AD Modul

Bei jedem Suchergebnis wird ein Button Passwort zurücksetzen angezeigt. Wenn man diesen Button klickt, taucht ein kleiner Dialog auf, welche nachfragt, ob man das Passwort wirklich zurücksetzen will, da es beim Testing mehrfach vorkamm, dass ein Passwort versehentlich zurückgesetzt wurde.

The screenshot shows a confirmation dialog titled "Passwort zurücksetzen". The text inside asks: "Möchten Sie das Passwort für john.doe wirklich zurücksetzen?". At the bottom, there are two buttons: "Abbrechen" (grey) and "Bestätigen" (red).

Figure 23: Bestätigungsdialog zum Zurücksetzen des Passworts

Sobald dieser Dialog bestätigt wurde, wird das Passwort des Benutzers zurückgesetzt und dem Benutzer angezeigt. Das neue Passwort entspricht den Passwort Richtlinien der Berufsschule Lenzburg. Diese werden in einem späteren Kapitel Spezielles_AD-Passwort_zurücksetzen genauer erläutert.

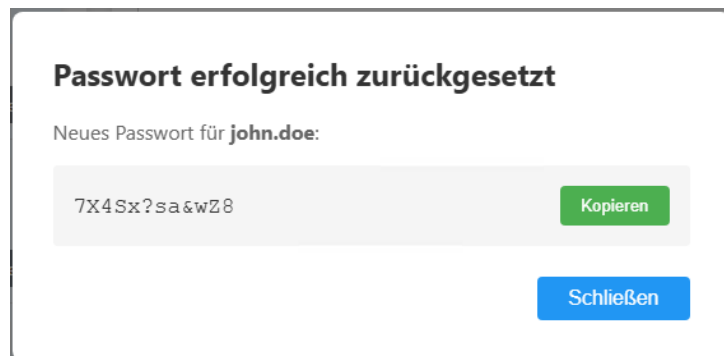


Figure 24: Anzeige des neuen Passworts

In den Mockups wurde das Feld "Passwort muss beim nächsten Login geändert werden", angewählt, dies wurde in der Finalen Version nicht umgesetzt, da es in der Vergangenheit bereits versucht wurde, und diese Einstellung mit einem On Prem AD nicht funktioniert.

3.2.6 Schnittstellen

In diesem Modul werden die folgenden REST Endpoints zwischen dem Frontend und dem Backend verwendet.

ActiveDirectory		^
GET	/api/ActiveDirectory/search	∨
POST	/api/ActiveDirectory/unlock	∨
POST	/api/ActiveDirectory/reset-password	∨

Figure 25: Backend Endpoints des AD Modules

3.2.7 Spezielles

In diesem Kapitel werden Besonderheiten des Modules AD Passwort zurücksetzen beschrieben.

Passwortrichtlinien

Das neu erstellte Passwort erfüllt die Passwortrichtlinien der Berufsschule Lenzburg, welche die folgenden sind:

- Länge des Passwortes: min 10 Zeichen
- Gross/Klein-Buchstaben
- Mindestens eine Zahl und Sonderzeichen

Die erstellten Passwörter von unserem Modul erfüllen diese Angaben durch folgende Einstellungen:

- Passwortlänge zwischen 12 und 15 Zeichen
- Mindestens einen Grossbuchstaben und Sonderzeichen, sowie mindestens eine Zahl

Folgende Zeichen werden aus den Passwörtern ausgeschlossen, da diese schnell zu einer Verwechslung führen können: "i, l, o, I, L, O, 0, 1"

3.2.8 Tests

Nach der Entwicklung des Modules wurde ein Usability Test mit dem Leiter der IT der Berufsschule Lenzburg durchgeführt. Dabei testete er das Modul ohne eine Einweisung, folgende Punkte sind dabei aufgefallen:

- Die Fehlermeldung, wenn kein Benutzer gefunden wurde, sind nicht verständlich, da es sich um eine Stackfehlermeldung handelt.
- Es wäre schön wenn man auch die Email Nicknamen anzeigen könnte.
Beispiel Max.Muster@bslenzburg.ch kann noch den Nicknamen M.Muster@bslenzburg.ch haben.

Diese Punkte wurden als Feedback entgegen genommen und in die Finale Version implementiert.

3.2.9 Erledigte Functional Requirements

Mit diesem Modul wurden die folgenden FRs aus dem Kapitel UC02 - Passwort zurücksetzen erledigt:

- FR021

3.3 Modul PRTG Server Monitor

In diesem Kapitel wird das Modul PRTG Server Monitor beschrieben.

3.3.1 Kurzbeschreibung

Das Modul PRTG Server Monitor soll dem Support der Berufsschule Lenzburg helfen, die Server und Netzwerkgeräte zu überwachen. Es soll eine Übersicht über den aktuellen Zustand der Geräte geben und einfache Filtermöglichkeiten beinhalten, um Probleme schnell zu erkennen.

3.3.2 Aktivitätsdiagramm

Die folgende Grafik zeigt einen typischen Ablauf der Benutzung des Moduls PRTG Server Monitor:

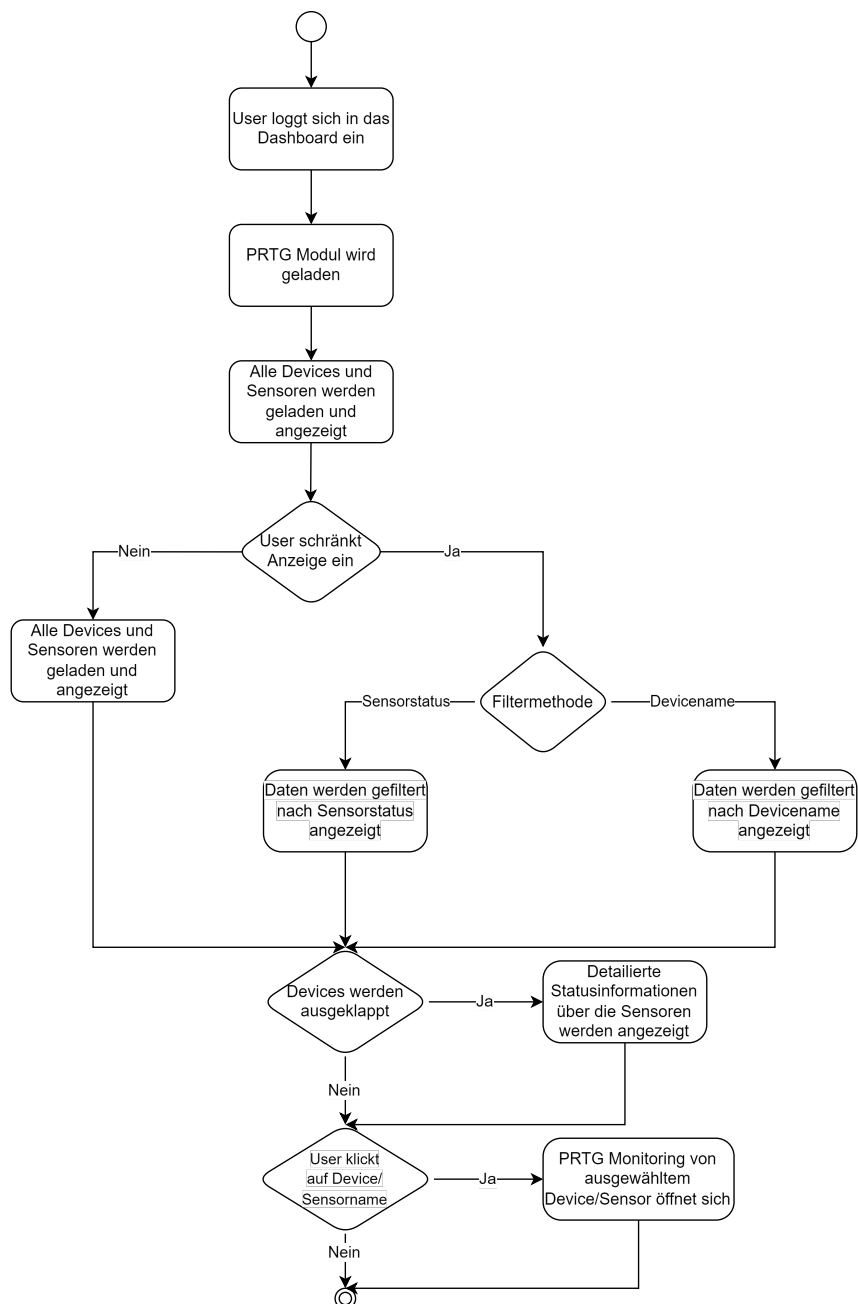


Figure 26: Aktivitätsdiagramm: PRTG Server Monitor Modul

3.3.3 Mockups

Als Startpunkt, wurden von diesem Modul Mockups gezeichnet und einem Mitarbeiter aus dem Support-Team der Berufsschule Lenzburg gezeigt.

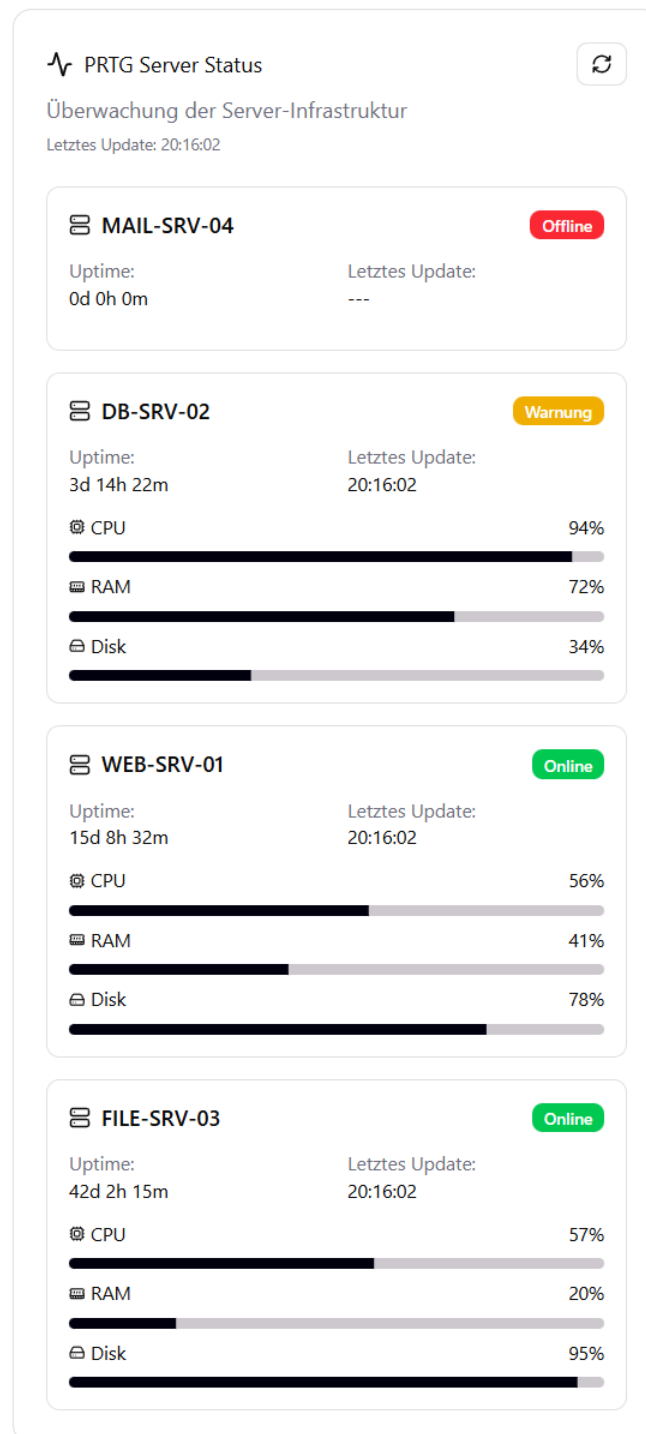


Figure 27: Mockup: PRTG Server Monitor Modul in Figma

Die Rückmeldung war sehr positiv und es wurden keine Anpassungen am Design vorgeschlagen.

3.3.4 Umsetzung

Durch das positive Feedback der Mockups, konnte mit der Umsetzung des Moduls begonnen werden.

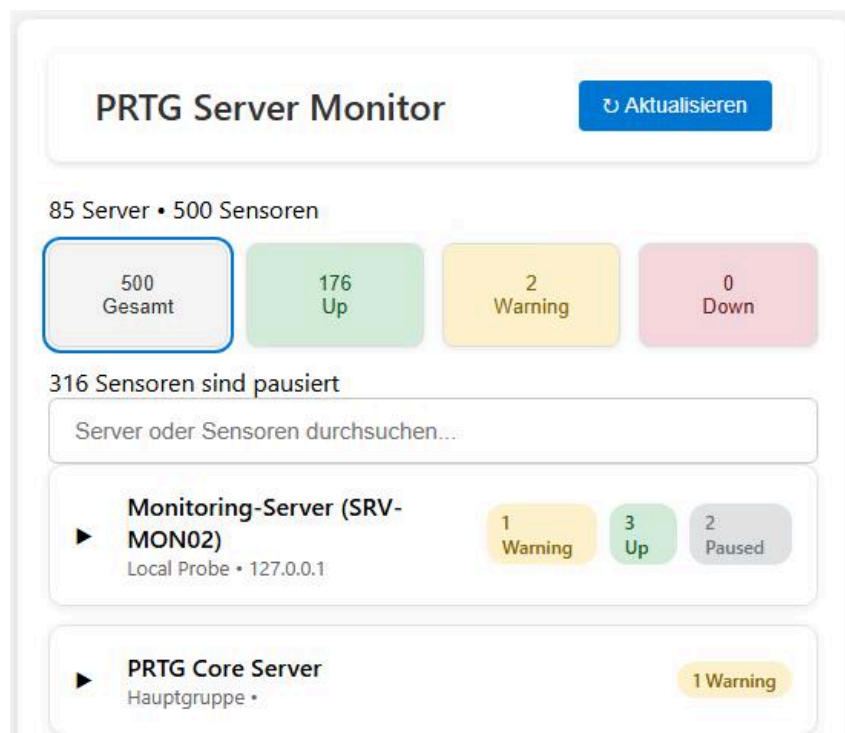


Figure 28: PRTG Server Monitor Modul

Alle Devices von PRTG werden in einer Tabelle dargestellt. Für jedes Device werden die zugehörigen Sensoren mit ihrem Status angezeigt.

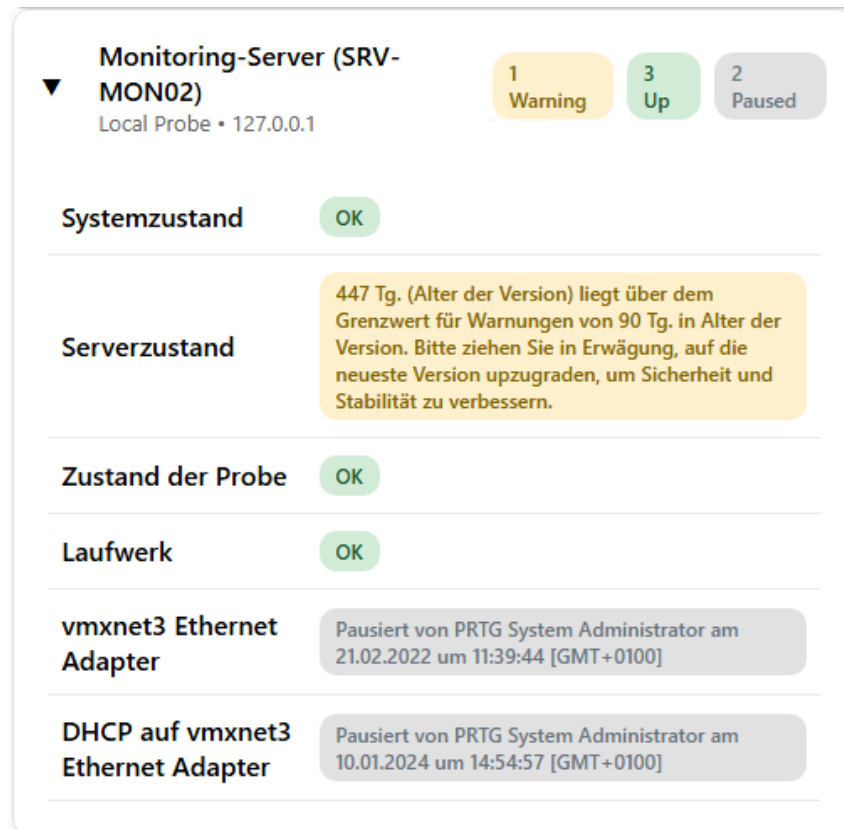


Figure 29: PRTG Server Monitor Modul mit erweitertem Device

Es gibt die Möglichkeit die Tabelle zu filtern und nach Servern zu suchen. Die Sensoren können nach ihrem Status mittels Buttons (Alle, OK, Warnung, Fehler) gefiltert werden.

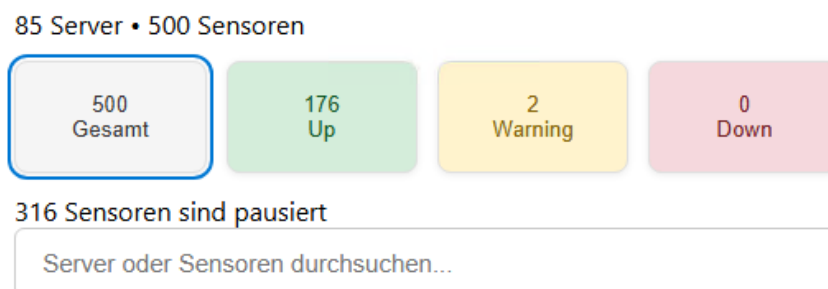


Figure 30: PRTG Filtermöglichkeiten

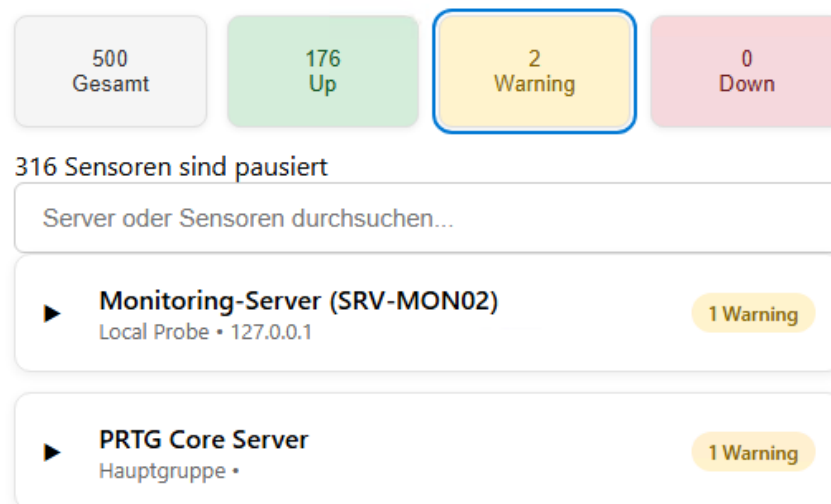


Figure 31: PRTG Filter auf Sensoren mit Warnung

3.3.5 Schnittstellen

In diesem Modul werden die folgenden REST Endpoints zwischen dem Frontend und dem Backend verwendet.

Prtg		^
GET	/api/Prtg/devices	▼
GET	/api/Prtg/devices/{id}	▼
GET	/api/Prtg/sensors	▼
GET	/api/Prtg/sensors/{id}	▼
GET	/api/Prtg/devices/{id}/sensors	▼

Figure 32: Backend Endpoints des PRTG Server Monitor Modules

3.3.6 Tests

Nach der Entwicklung des Moduls wurde ein Usabilitytest mit dem Leiter der IT der Berufsschule Lenzburg durchgeführt. Dabei testete er das Modul ohne eine Einweisung, folgender Punkt ist dabei aufgefallen:

- Es soll möglich sein, direkt auf das PRTG Dashboard zu gelangen mit eingebauten Links.

Dieser Punkt wurde als Feedback entgegengenommen und in der finalen Version implementiert.

3.3.7 Erledigte Functional Requirements

Mit diesem Modul wurden die folgenden FRs aus dem Kapitel UC03 - PRTG Monitoring erledigt:

- FR031: PRTG Filterung
- FR032: PRTG Suche
- FR034: PRTG Serverauswahl

3.4 Power Automate Runs Fehler

In diesem Kapitel wird das Modul Power Automate Runs Fehler beschrieben.

3.4.1 Kurzbeschreibung

Durch dieses Modul soll der Prozess um Fehler aus einem Powerautomate Run für den Support der Berufsschule Lenzburg besser erkennbar sein, somit kann auch schneller darauf reagiert werden. Aktuell musste man dafür sich mit einem separaten Benutzer auf die Microsoft Powerplattform einloggen und dann jeden einzelnen Flow separat nach fehlerhaften Flows durchsuchen.

Mithilfe von diesem Modul soll der oben beschriebenen Prozess deutlich vereinfacht werden, das man auf einen Blick alle fehlgeschlagenen Runs sieht. Ebenso kann man einen Flow als erledigt markieren, wenn sich der Support diesen kontrolliert und abgearbeitet hat.

3.4.2 Aktivitätsdiagramm

Die folgende Grafik zeigt einen typischen Ablauf der Benutzung des Module Power Automate Runs Fehler:

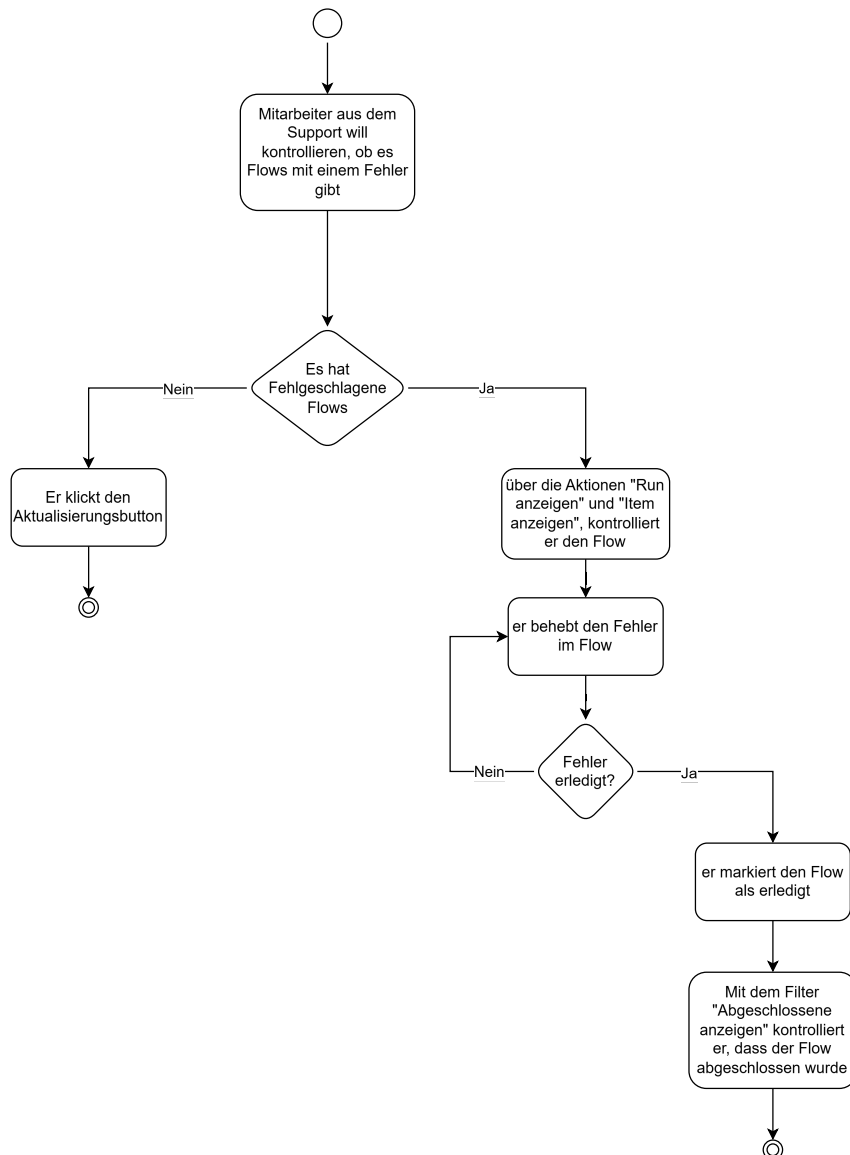


Figure 33: Aktivitätsdiagramm des Modules AD Passwort Reset

3.4.3 Mockups

Als Startpunkt für die Entwicklung, wurden Mockups in Figma, von diesem Module gezeichnet und einem Mitarbeiter aus dem Supportteam der Berufsschule Lenzburg gezeigt.

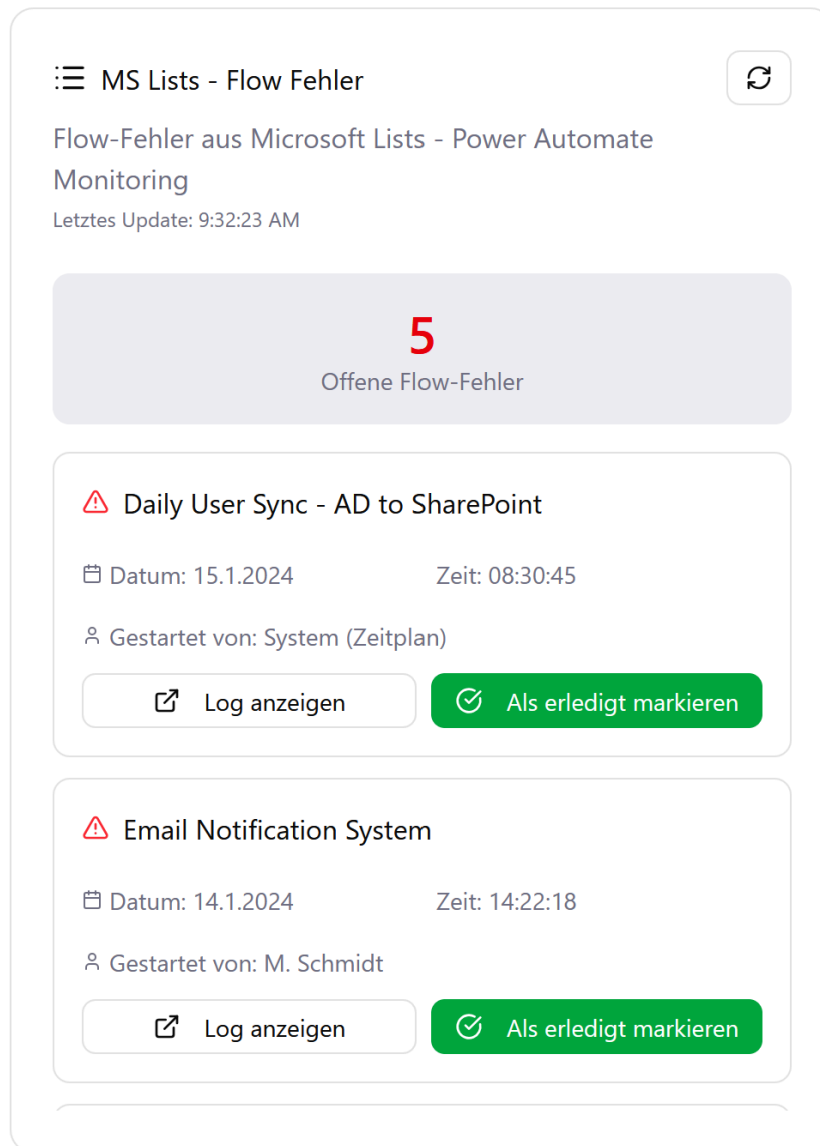


Figure 34: Mockup: Powerautomate Flow Runs Fehler Modul aus Figma

3.4.4 Feedback aus dem Supportteam

Das Feedback des Mitarbeiters war sehr positiv und es mussten nur kleinere Änderungen von Texten vorgenommen werden.

3.4.5 Umsetzung

Mit dem Feedback aus dem Supportteam und den Mockups wurde die finale Version, welche auch implementiert wurde, entwickelt.

In der Finalen Umsetzung wurde eine Liste mit der Run ID, Flow Name, Startzeit des Flows, Status und mehreren Aktionen entwickelt. Die Aktionen sind, Run anzeigen, Item öffnen und die Fehlermeldung als abgeschlossen zu markieren.

Die Startzeit in der Liste ist die Startzeit, wann der Flow gestartet wurde. Diese ist wichtig um Fehler zu finden, da ein Flow nach 30 Tagen automatisch abbricht.

Power Automate Runs Fehler				
			☐ Abgeschlossene anzeigen	<button>Aktualisieren</button>
1 von 7 Einträgen (ohne Abgeschlossene)				
Run ID	Flow Name	Startzeit	Status	Aktionen
2	AntragExkursion ExternerUnterricht	17.09.2025, 08:16	Ausstehend	<button>Run anzeigen</button> <button>Item öffnen</button> <button>Als abgeschlossen markieren</button>

Figure 35: Modul ohne aktiven Filter

Im Header des Modules gibt es zwei Funktionen, einen Filter, der auch die abgeschlossenen Runs anzeigen lässt und ein Button um die Liste manuell zu aktualisieren. Die Liste wird jede Minute automatisch aktualisiert, um immer aktuelle Daten zu haben.

Power Automate Runs Fehler				
		☒ Abgeschlossene anzeigen		Aktualisieren
Run ID	Flow Name	Startzeit	Status	Aktionen
2	AntragExkursion ExternerUnterricht	17.09.2025, 08:16	Ausstehend	Run anzeigen Item öffnen Als abgeschlossen markieren
3	AntragExkursion ExternerUnterricht	24.09.2025, 15:11	Abgeschlossen	Run anzeigen Item öffnen ✓ Abgeschlossen
4	AntragExkursion ExternerUnterricht	26.09.2025, 11:21	Abgeschlossen	Run anzeigen Item öffnen ✓ Abgeschlossen
5	'INPRO-Beschaffung-Antrag Kostenvoranschlag-Antrag	02.09.2025, 11:07	Abgeschlossen	Run anzeigen Item öffnen ✓ Abgeschlossen

Figure 36: Modul mit Einträgen aktivem Filter

Die Aktion Run anzeigen ist ein Link zum Powerautomte Run, in welchem dieser Fehler aufgetreten ist, damit kann man identifizieren, ob der Fehler ein Problem im Flow oder in den Daten ist.

Mithilfe der Aktion Item anzeigen, kann man direkt auf das Item springen, welches erstellt wurde als der Fehler aufgetreten ist.

Wenn der Support eine Fehlermeldung bearbeitet hat, kann man die Fehlermeldung als abgeschlossen markieren, dass sich diese Fehlermeldung nicht nochmals eine andere Person anschauen muss.

Power Automate Runs Fehler				
		☒ Abgeschlossene anzeigen	Aktualisieren	
Run ID	Flow Name	Startzeit	Status	Aktionen
2	AntragExkursion ExternerUnterricht	17.09.2025, 08:16	Abgeschlossen	Run anzeigen Item öffnen <input checked="" type="button" value="✓ Abgeschlossen"/>
3	AntragExkursion ExternerUnterricht	24.09.2025, 15:11	Abgeschlossen	Run anzeigen Item öffnen <input checked="" type="button" value="✓ Abgeschlossen"/>
4	AntragExkursion ExternerUnterricht	26.09.2025, 11:21	Abgeschlossen	Run anzeigen Item öffnen <input checked="" type="button" value="✓ Abgeschlossen"/>
5	'INPRO-Beschaffung-Antrag Kostenvoranschlag-Antrag	02.09.2025, 11:07	Abgeschlossen	Run anzeigen Item öffnen <input checked="" type="button" value="✓ Abgeschlossen"/>

Figure 37: Modul nach dem ausführen der Aktion "als abgeschlossen markieren"

3.4.6 Schnittstellen

In diesem Modul werden die folgenden REST Endpoints zwischen dem Frontend und dem Backend verwendet.

PowerAutomate	
GET	/api/PowerAutomate/runs
POST	/api/PowerAutomate/complete/{id}

Figure 38: Backend Endpoints des Powerautomate Flow Runs Fehler Modules

3.4.7 Tests

Nach der Entwicklung des Modules wurde ein Usability Test mit dem Leiter der IT der Berufsschule Lenzburg durchgeführt. Dabei testete er das Modul ohne eine Einweisung, folgender Punkt ist dabei aufgefallen:

- Ihm hat der Aktualisierungsbutton gefallen, da man so manuell die Liste aktualisieren kann.

Dieser Punkt wurde entgegengenommen und so in der Finalen Version belassen.

3.4.8 Erledigte Functional Requirements

Mit diesem Modul wurden die folgenden FRs aus dem Kapitel UC04 - Powerautomate Flow Error erledigt:

- FR041

3.5 Einstellungen

In diesem Kapitel werden die Einstellungen des modularen Dashboards beschrieben.

3.5.1 Kurzbeschreibung

Da das Dashboard von mehreren Benutzern verwendet wird, welche zum Teil unterschiedliche Anforderungen haben, daher ist es wichtig, dass jeder Benutzer seine eigenen Einstellungen vornehmen kann.

3.5.2 Aktivitätsdiagramm

Die folgende Grafik zeigt einen typischen Ablauf der Benutzung der Einstellungen:

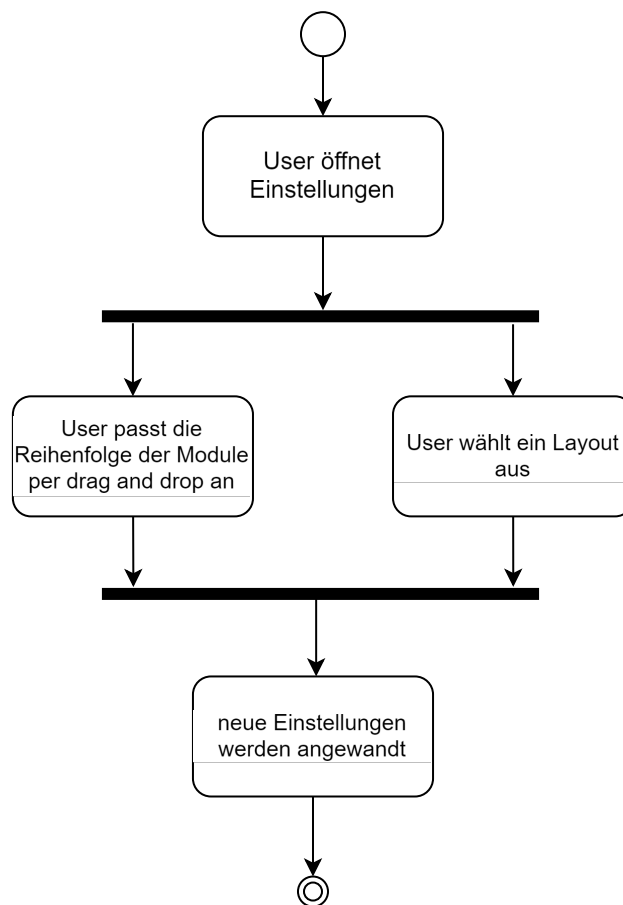


Figure 39: Ablaufdiagramm: Einstellungen

3.5.3 Mockups

Als Startpunkt wurde von den Einstellungen Mockups gezeichnet und einem Mitarbeiter aus dem Supportteam der Berufsschule Lenzburg gezeigt.

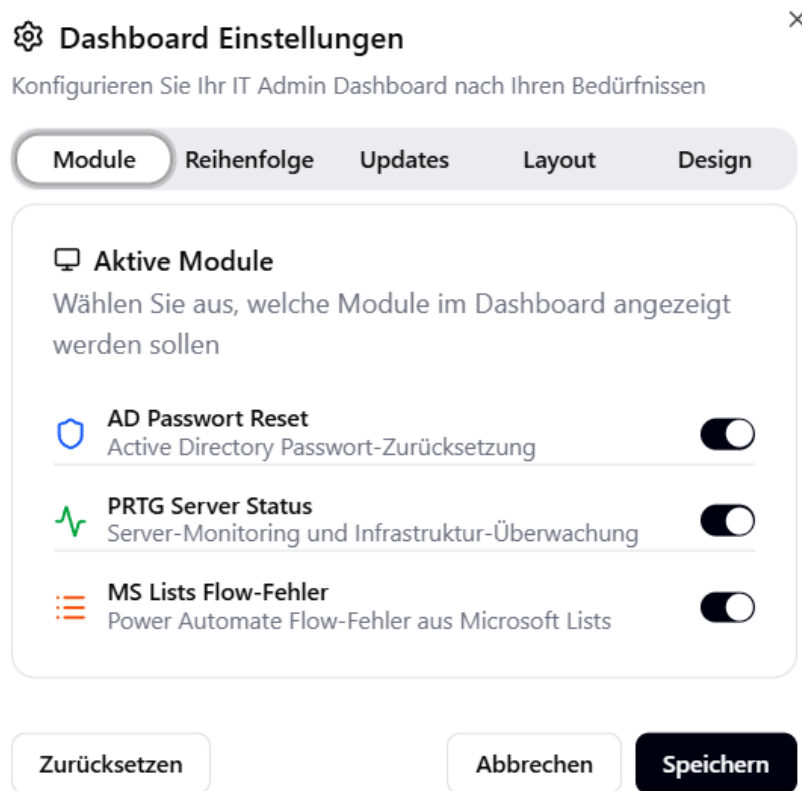


Figure 40: Mockup: Einstellungen im Figma

Das gezeigte Figma Mockup wurde vom Mitarbeiter des Supportteams der Berufsschule Lenzburg begutachtet und für gut empfunden.

3.5.4 Umsetzung

Die Einstellungen aus dem Mockup gestaltete sich allerdings schwieriger als gedacht, da ein komplett modulares Dashboard mehr Aufwand in der Umsetzung benötigt. Deshalb haben wir uns entschieden, die Einstellungen auf die wichtigsten Funktionen zu beschränken. Diese umfassen die Anordnung und Reihenfolge der Module.

Für die Anordnung der Module haben wir 3 verschiedene Layouts zur Auswahl gestellt und die Reihenfolge der Module kann per Drag & Drop angepasst werden.

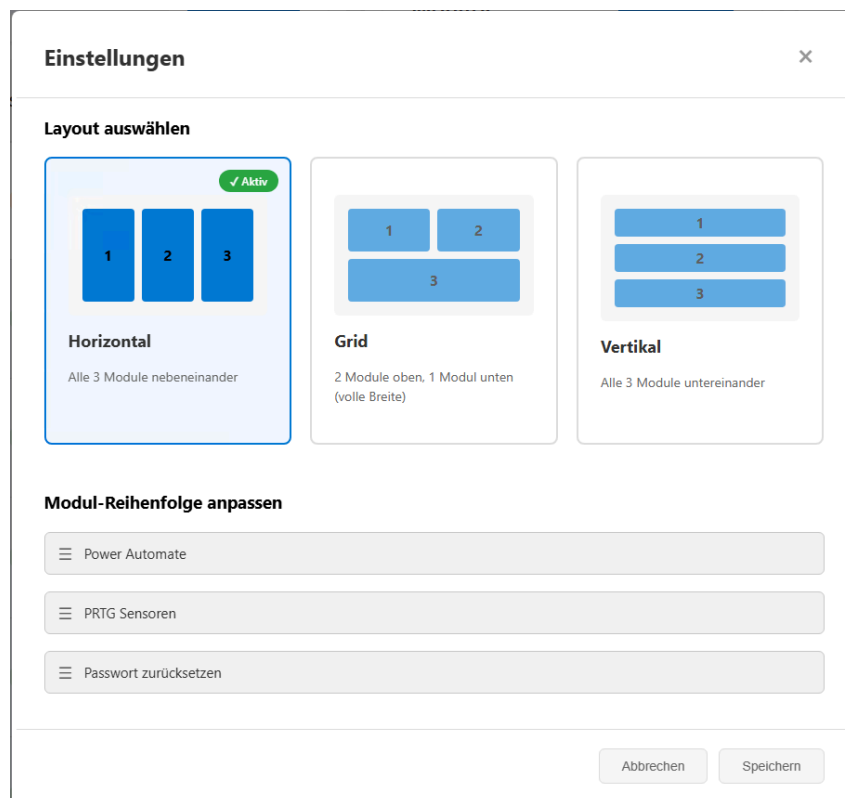


Figure 41: Screenshot der Einstellungen

Damit die Einstellungen auch persistent gespeichert werden, haben wir die Einstellungen in der Datenbank des Backends abgelegt und beim Laden des Dashboards wieder ausgelesen. Die Settings werden bei jedem Benutzer individuell gespeichert, sodass jeder Benutzer seine eigenen Einstellungen vornehmen kann.

DashboardSettings
primaryKey
userName
Layout
order

Figure 42: Postgres Tabelle der Einstellungen

3.5.5 Schnittstellen

In diesem Modul werden die folgenden REST Endpoints zwischen dem Frontend und dem Backend verwendet.

DashboardSettings	
GET	/api/DashboardSettings/get
POST	/api/DashboardSettings/save

Figure 43: Backend Endpoints der Einstellungen

3.5.6 Tests

Nach der Entwicklung des Modules wurde ein Usability Test mit dem Leiter der IT der Berufsschule Lenzburg durchgeführt. Dabei testete er das Modul ohne eine Einweisung und konnte die Einstellungen problemlos vornehmen. Anschliessend wurde das Feedback eingeholt, welches positiv ausfiel.

3.5.7 Erledigte Functional Requirements

Mit diesem Modul wurden die folgenden FRs aus dem Kapitel UC01 - Modulares Dashboard verwalten erledigt:

- FR014: Konfiguration speichern
- FR013: Anordnung ändern

4 Quality Measures

4.1 Working Environment

Dieser Abschnitt beschreibt die Quality Measures und Projekt-Guidelines, die für Konsistenz im Team und hohe Qualität in allen Projektbereichen sorgen.

4.1.1 Dokumentation

- Die Dokumentsprache ist Deutsch. Fachbegriffe oder übliche Überschriften können in Englisch geschrieben werden.
- Die Dokumentation wird in Typst verfasst.
- Es wird für jedes Hauptkapitel (z.B. 5) ein Ordner erstellt und für jedes Unterkapitel (z.B. 5.1) ein eigenes Typst-File erstellt.
- Namenskonvention für Ordner und Files in Typst: Kapitelnummer und Kapitelname in PascalCase (z.B. 04_QualityMeasures/01_WorkingEnvironment.typ)
- Um die Zusammenarbeit und das Mergen in Git zu vereinfachen, sollte jeder Satz in einer neuen Zeile begonnen werden.

4.1.2 Code

- Um eine gute Code-Qualität und eine einheitliche Code-Formatierung zu gewährleisten sowie Fehler frühzeitig zu erkennen, werden Linters eingesetzt.
- Der Code wird, wo passend, auf Englisch kommentiert.
- Klassen und Variablennamen sind aussagekräftig, einheitlich benannt und folgen den sprachüblichen Conventions (z.B. PascalCase für Klassen, camelCase für Variablen und Methoden).

4.1.3 Definition of Done

Ein Task wird nur dann abgeschlossen, wenn folgende Kriterien erfüllt sind:

- Die oben genannten Code-Guidelines sind umgesetzt.
- Die Pipeline ist erfolgreich durchgelaufen.
- Der Featurebranch wurde in den Mainbranch integriert.
- Alle im Asana-Ticket definierten Akzeptanzkriterien sind erfüllt.
- Zeit wurde im Asana eingetragen.

4.1.4 Version Control

Für die Version Control des Source Codes und der Dokumentation wird Gitlab verwendet und die Commits und Branches werden in Englisch verfasst. Genaue Guidelines werden in den folgenden Kapiteln beschrieben.

4.1.4.1 Branching

Für einzelne Tasks wird jeweils ein eigener Branch erstellt. Hierfür wird sich an die folgenden Naming Conventions gehalten:

Aufbau eines Branchname

Der Branchname wird in kebab-case geschrieben und hat folgenden Aufbau:

<Prefix>/<Task_id>-<kurze_Beschreibung>

z.B. "docs/29-test-concept"

Passender Prefix:

Prefix	Anwendung
feature/	Für die Entwicklung neuer Features
bugfix/	Um Bugs im Code zu beheben, oft in Verbindung mit einem Issue erstellt
hotfix/	Um kritische Bugs in der Produktion zu beheben
release/	Zur Vorbereitung einer neuen Version, typischerweise für letzte Anpassungen und Überprüfungen
docs/	Zum Schreiben, Ändern oder Korrigieren der Dokumentation
ci/	Für Änderungen an der Continuous Integration/Continuous Deployment (CI/CD)-Pipeline

Table 11: Branch Prefixes

4.1.4.2 Commits

Es gelten folgende Guidelines für alle Commit Messages:

Head = Maximale Länge von 50 Zeichen (Grobe Beschreibung)

Body = Maximale Länge von 72 Zeichen pro Zeile, kann mehrzeilig sein

Aufbau einer Commit-Message

Zeile 1: <Type>(<Task_id>): <Subject> (z.B. "feat(17): Add Admin-View")

Zeile 2: Leerzeile

Zeile 3: Body des Commits mit einer genaueren Beschreibung (optional)

Passender Type:

Type	Anwendung
build	Änderungen im Zusammenhang mit dem Build-Prozess (z.B. Hinzufügen von npm-Abhängigkeiten oder externen Bibliotheken)
feat	Ein neues Feature
fix	Ein Bugfix
docs	Änderungen an der Dokumentation
ci	Änderungen an der Continuous Integration/Continuous Deployment (CI/CD)-Pipeline
refactor	Code, der weder einen Bug fixt, noch ein Feature hinzufügt (z.B. semantische Änderungen wie das Umbenennen einer Variable oder Funktion)
perf	Code, der die Performance verbessert
test	Hinzufügen neuer Tests oder Änderungen an bestehenden Tests

Table 12: Commit Types

4.1.4.3 Merge Request

Nach Fertigstellung einer Änderung wird ein Merge Request erstellt. Dieser muss von der anderen Person reviewed und approved werden.

Der Name des Merge Requests entspricht dem Namen des zugehörigen Branches.

4.1.5 Review Process

Nachdem ein Merge Request erstellt wurde, muss die Story in Asana in die Spalte "IN REVIEW" gezogen und mittels Rules dem anderen assigned werden. Wenn das Review positiv verläuft, wird es wieder dem Entwickler zugewiesen und in die Spalte "MERGEN" gezogen. Bei einem negativen Review wird die Story in die Spalte "IN PROGRESS" anstelle von "MERGE" gezogen.

4.1.6 Automatic Quality Control

Für die automatisierte Qualitätskontrolle wird in der Pipeline der Code von ESLint und Prettier für das Frontend und dotnet-format für das Backend bei jedem Merge Request überprüft. Bei Fehlern schlägt die Pipeline fehl und der Merge Request kann nicht gemerged werden, bis die Fehler behoben sind.

4.2 Test Concept

4.2.1 Übersicht

Das Testkonzept für das BSL Support Dashboard umfasst sowohl funktionale als auch nicht-funktionale Anforderungen. Dieses Kapitel fokussiert sich auf die systematische Überprüfung der Non-Functional Requirements (NFRs) gemäss FURPS-Kategorisierung. Die Funktionalen Requirements wurden bereits in den einzelnen Modulen getestet und dokumentiert. Die Teststrategie orientiert sich an den definierten Verifizierungsmethoden aus den NFR-Tabellen.

4.2.1.1 Acceptance Tests

Für die im Projekt definierten Non-Functional Requirements (NFRs) wird keine klassische automatisierte Teststrategie verwendet. Stattdessen erfolgt die Verifikation der Anforderungen durch gezielte manuelle Prüfungen oder technische Analysen. Die jeweiligen Prüfmethoden wurden pro NFR im Kapitel Non-Functional Requirements bereits beschrieben und werden hier noch etwas ausgeführt.

4.3 Test Protocols

Nach durchgeführten Tests werden hier jeweils die Test-Protokolle ergänzt.

4.3.1 Legende Testbewertung

Feld	Bedeutung
OK	Test erfolgreich durchgeführt, Ergebnis wie erwartet
PARTIALLY OK	Test durchgeführt, Ergebnis teilweise wie erwartet
NOK	Test durchgeführt, aber mit fehlerhaftem Ergebnis
NOT TESTED	Test wurde noch nicht ausgeführt, da Feature noch nicht implementiert

Table 13: Legende Testbewertung

4.3.2 Acceptance Tests Non-Functional Requirements

4.3.2.1 Testprotokoll - 12.12.2025

Functionality

Testfall-ID	Ergebnis	Bewertung
<u>NFR001</u>	Ein AD User welcher nicht in der Richtigen AD Gruppe ist, kann nicht auf die entsprechenden Module zugreifen.	OK
<u>NFR002</u>	Es existieren keine Benutzereingaben, ausser Suchfelder, diese werden nicht validiert.	OK

Table 14: Testprotokoll Part Functionality - 12.12.2025

Usability

Testfall-ID	Ergebnis	Bewertung
<u>NFR101</u>	Alle Fehlermeldungen sind verständlich und hilfreich.	OK
<u>NFR102</u>	Fehlermeldungen werden in einem einheitlichen Format dargestellt.	OK

Table 15: Testprotokoll Part Usability - 12.12.2025

Reliability

Testfall-ID	Ergebnis	Bewertung
NFR201	Alle Fehlermeldungen werden in einem zentralen Log gespeichert.	OK
NFR202	Alle Passwörter Reset Anfragen werden im gleichen Log wie Fehlermeldungen gespeichert.	PARTIALLY OK
NFR203	In den Logs werden keine sensiblen Daten gespeichert.	OK

Table 16: Testprotokoll Part Reliability - 12.12.2025

Supportability

Testfall-ID	Ergebnis	Bewertung
NFR401	Neue Module können einfach integriert werden.	OK

Table 17: Testprotokoll Part Supportability - 12.12.2025

Part IV

Project Documentation

5 Initial Project Proposal

5.1 Kunde und Ansprechpartner

5.1.1 Praxispartner

Name des Praxispartners:	Berufsschule Lenzburg
Adresse des Praxispartners	Berufsschule Lenzburg Neuhofstrasse 36 5600 Lenzburg
Tätigkeitsgebiet des Praxispartners	Berufsschule
Name des Ansprechpartners	Marco Gaiffi
Telefonnummer des Ansprechpartners	+41 62 885 39 60
Email des Ansprechpartners	marco.gaiffi@bslenzburg.ch

5.2 Thema

Entwicklung eines Dashboard-Systems für Info-Screens und Support-Workstations

5.3 Kurzbeschreibung und erwartete Resultate

Im Rahmen dieses Projektes soll eine Anwendung entwickelt werden, diese Anwendung hat ihren Haupteinsatzbereich im IT-Supportteam der Berufsschule. Mit der Software sollen mehrere UseCases abgedeckt werden:

1. Es sollen die Stati der wichtigsten Server auf einen Blick ersichtlich sein.
2. Das Supportteam soll bei den häufigsten Supportanfragen unterstützt werden um diese zu vereinfachen und schnelle zu gestalten.
 - Ein Beispiel dafür wäre, dass ein Lernender sein Passwort vergessen hat, dann soll das Support Team nur die Email-Adresse des Lernenden eingeben müssen, und dann wird diesem AD-Benutzer ein neues Passwort gesetzt und zurückgegeben.
3. Im Backend soll das Modulare Design konfiguriert werden, dass heisst jede Person kann sich das Dashboard für sich selber zusammenstellen.
4. Spezielles Augenmerk soll auf folgende Punkte gelegt werden:
 - Das Tool soll die gängigen Sicherheitsstandards verwenden.
 - Das Design muss intuitiv und einfach zu bedienen sein.
5. Die verschiedenen angezeigten Module sollen in einer Cardview angezeigt werden.
 - Beispiel für eine solche Cardview ist unter folgendem Link einzusehen:
<https://jahresbericht.bslenzburg.ch/>
6. Im ersten Teil der Arbeit soll ein geeigneter Techstack ausgewählt werden.

Phase 1 Technologie Entscheidung und Prototyp:

In der ersten Phase soll eine Evaluation des Technologie Stack ausgearbeitet und sich für eine geeignete Lösung entschieden werden. Danach soll mit diesem Tech Stack ein Prototyp ausgearbeitet werden.

Phase 2 Anwendung entwickeln:

In der zweiten Phase sollen die Module, welche weiter unten beschrieben werden, umgesetzt werden. Dabei soll auf dem Prototyp aus Phase 1 aufgebaut werden.

Erwartete Resultate:

Qualitativ:

- Benutzerfreundliche, moderne Oberfläche
- Trennung zwischen Anzeige- und Verwaltungsbereich
- Modulares Design für einfache Erweiterbarkeit
- Dokumentation für Bedienung und Wartung
- Maintenance Aufwand für den Betrieb der Lösung soll in der Evaluation für den Tech Stack Entscheid einfließen.

Module

Passwort zurücksetzen:

In diesem Modul soll die Möglichkeit bestehen eine Email-Adresse von einem Lernenden am Schalter einzugeben. Danach wird das Passwort im AD direkt geändert und das neue random generierte Passwort wird zurückgegeben. Zusätzlich wird im AD das Flag gesetzt das der Lernenden das Passwort beim nächsten Login wechseln muss, aktiviert.

Serverstatus anzeigen:

In diesem Modul werden die Status, von User definierten Servern angezeigt. Das Ziel dieser Card ist es nicht ein eigenen Monitoring Service zu bauen, sondern eine bestehendes Tool über eine sicheren

Verbindung zu visualisieren. Der User kann aus einer Liste von bestehenden Servern auswählen welche er angezeigt haben möchte. Zusätzlich kann ein User auch neue Server hinzufügen, diese werden dann auch in der Liste angezeigt, damit auch andere User diesen einfach hinzufügen können.

Microsoft Power Automate Status:

Wir nutzen Power Automate, um verschiedene Prozesse zu automatisieren. Aktuell müssen die Ausführungen (Runs) jedes einzelnen Flows einzeln aufgerufen werden, um deren Ergebnisse zu überprüfen. In diesem Modul sollen die Runs zentral zusammengefasst werden, um einen schnellen Überblick über den Status aller Flows zu ermöglichen – insbesondere, um Fehler und fehlgeschlagene Ausführungen auf einen Blick zu erkennen.

Weitere Funktionalitäten

Login:

Damit Einstellungen gespeichert werden können und Änderungen nachvollzogen werden können, muss man sich in das Tool einloggen, dies soll mit SSO umgesetzt werden.

5.4 Randbedingungen und Infrastruktur

5.4.1 Infrastruktur:

Hardware: Standard-Windows-PCs

Betriebssystem: Windows 11

Programmiersprache: Wird in der ersten Phase des Projektes ermittelt.

Datenbank: Wird in der ersten Phase des Projektes ermittelt.

Netzwerk: Zugriff auf internes Netzwerk notwendig

Tools: Wird in der ersten Phase des Projektes ermittelt.

Besondere Infrastruktur:

Es wird ein Webserver notwendig, dieser kann intern bereitgestellt werden.

5.5 Sonstige Hinweise

Die entwickelte Software bleibt Eigentum der Berufsschule Lenzburg. Es gelten keine besonderen Vertraulichkeitsvereinbarungen, sofern keine sensiblen Daten integriert werden. Eine spätere Weiterentwicklung oder produktive Nutzung der Anwendung ist erwünscht.

6 Project Plan

6.1 Project Management

Zum Projektbeginn war die Projektmethode Scrum vorgesehen, da wir mit dieser in SE-Project gute Erfahrungen gemacht haben. Doch in den ersten Wochen der Prototypentwicklung zeigte sich, dass der damit verbundene organisatorische Aufwand für ein Projektteam bestehend aus zwei Personen nicht verhältnismässig war.

Aus diesem Grund wurde das Vorgehen angepasst zu einem leichtgewichtigen, welches eine pragmatische Zusammenarbeit ermöglicht. Wir trafen uns mindestens zweimal pro Woche zu strukturierten Abstimmungsrunden, in denen wir den Projektfortschritt besprachen, Prioritäten definierten und die nächsten Umsetzungsschritte planten. Diese flexible Arbeitsweise ermöglichte es uns, schnell auf neue Erkenntnisse zu reagieren und den Fokus auf die Entwicklung zu legen.

Obwohl wir auf ein etabliertes Vorgehen wie SCRUM verzichtet haben, orientierten wir uns weiterhin am ursprünglichen Projektplan und schlossen die definierten Meilensteine termingerecht ab.

Abschliessend ist festzuhalten, dass für zwei Personen eine reduzierte, flexible Arbeitsweise eine effiziente Alternative zu etablierten Methoden wie Scrum sein kann. Für Projekte mit einer grösseren Teamgrösse wäre der Einsatz einer strukturierten agilen Methode wie Scrum weiterhin sinnvoll, da in diesem Projekt der Nutzen die organisatorischen Aufwände überwiegen würde.

6.2 Long Term Planning

Für die Langzeitplanung verwenden wir RUP. Die Ziele der einzelnen Phasen sind als Epics in Asana abgelegt.

6.2.1 Milestones

Folgende Milestones wurden in Asana erstellt:

Milestone	Zieldatum	Beschreibung
Project Setup	19.09.2025	Das Projekt mit den entsprechenden Tools wurde angelegt und die ersten Aufgaben wurden erstellt.
Technologieentscheid & Architektur	03.10.2025	Die Technologien und die Architektur wurden festgelegt und aufgezeichnet.
Prototyp	10.10.2025	Der Prototyp wurde fertiggestellt und die Schnittstellen wurden getestet.
Modul 1 - Power Automate Monitoring	24.10.2025	Das Modul zur Überwachung von Power Automate ist implementiert und funktionsfähig.
Modul 2 - Serverstatus	07.11.2025	Das Modul zur Überwachung verschiedener Serverstatus ist implementiert und funktionsfähig.
Modul 3 - Passwort-Reset	14.11.2025	Das Modul zur Passwortzurücksetzung ist implementiert und funktionsfähig.
Modulares Dashboard	21.11.2025	Das modulare Dashboard ist implementiert und funktionsfähig.
Loginmechanismus	28.11.2025	Der Loginmechanismus ist implementiert und funktionsfähig.

Table 18: Milestones

Milestone	Zieldatum	Beschreibung
Testing	05.12.2025	Das Testing ist abgeschlossen und alle Tests sind erfolgreich.
Dokumentation	12.12.2025	Die Dokumentation ist abgeschlossen und alle relevanten Informationen sind enthalten.
Abgabe	19.12.2025	Das Projekt mit allen erforderlichen Dokumenten wurde erfolgreich abgegeben.

Table 19: Milestones

6.3 Risk Management

Dieses Kapitel beschreibt potenzielle Risiken im Projekt, bewertet sie mithilfe einer Risikomatrix und stellt Massnahmen zur Risikominimierung vor. So können Probleme frühzeitig erkannt und effektiv bewältigt werden.

Die Positionierung der Risiken in der Risikomatrix basiert auf einer sorgfältigen Einschätzung der Wahrscheinlichkeit und des Schweregrads jedes Risikos. Diese Bewertung berücksichtigt sowohl die beschriebenen möglichen Auswirkungen als auch die bereits definierten Massnahmen zur Risikominimierung. Dadurch spiegelt die Matrix nicht nur das reine Gefahrenpotenzial wider, sondern auch die erwartete Wirksamkeit der getroffenen Vorkehrungen.

Die Risikoanalyse ist ein kontinuierlicher Prozess und wird nicht einmalig durchgeführt.

Risiko-Listen

Nr.	Risiko	Beschreibung & mögliche Auswirkungen	Massnahmen zur Risikominimierung
Technisch			
RT01	Zugriff auf AD nicht möglich	Sollte der Zugriff auf das Active Directory nicht möglich sein, kann dies die Entwicklung und das Testen der Authentifizierungs- und Autorisierungsfunktionen erheblich beeinträchtigen. Dies könnte zu Verzögerungen im Projektzeitplan führen, da diese Funktionen zentral für die Anwendung sind.	Wir bauen einen Prototypen der die Schnittstellen, welche wir in den Modulen benötigen, mit einem Lesezugriff testet. Somit können wir frühzeitig erkennen, ob der Zugriff auf das AD funktioniert und wie die Datenstruktur aussieht.
RT02	PRTG API	Die PRTG API funktioniert nicht, oder anders als gedacht. Dies könnte die Entwicklung und das Testen der Überwachungsfunktionen behindern.	Wir testen die API mit einem Prototypen, um sicherzustellen, dass sie wie erwartet funktioniert.
RT03	SSO	Wir haben keine Erfahrung mit SSO. Dies könnte die Entwicklung behindern.	Bei grösseren Verzögerungen verwenden ein anderes Login-Verfahren, z.B. OAuth2, um die Authentifizierung zu gewährleisten.

Table 20: Risiko-Liste Teil Technisch

Nr.	Risiko	Beschreibung & mögliche Auswirkungen	Massnahmen zur Risikominimierung
Organisatorisch			
RO01	Server	Mangelnde Serverkapazitäten oder Ausfälle könnten die Entwicklung und das Testen der Anwendung behindern. Dies könnte zu Verzögerungen im Projektzeitplan führen.	Wir verwenden eine VM der OST für den Prototypen, damit wir bei Bedarf das Active Directory jederzeit testen können.

Table 21: Risiko-Liste Teil Organisatorisch

Legende

Farbe	Bedeutung
	Sehr Hoch
	Hoch
	Mittel
	Niedrig
	Beseitigt

Table 22: Legende

6.3.1 Risikomatrix 01.10.2025

Wahrscheinlichkeit	Schweregrad			
	Vernachlässigbar	Geringfügig	Kritisch	Katastrophal
Sicher				
Wahrscheinlich				
Möglich		RO01, RT03	RT01	
Unwahrscheinlich		RT02		
Selten				
Keine				

Table 23: Risikomatrix

6.3.2 Risikomatrix 17.10.2025

Wahrscheinlichkeit	Schweregrad			
	Vernachlässigbar	Geringfügig	Kritisch	Katastrophal
Sicher				
Wahrscheinlich				
Möglich		RO01, RT03	RT01	
Unwahrscheinlich				
Selten				
Keine	RT02			

Table 24: Risikomatrix angepasst nach Prototyp

Nach dem erfolgreichen Prototypen konnten wir die Risiken für die Schnittstellen besser einschätzen und die Risikomatrix entsprechend anpassen.

6.3.3 Risikomatrix 30.11.2025

Wahrscheinlichkeit	Schweregrad			
	Vernachlässigbar	Geringfügig	Kritisch	Katastrophal
Sicher				
Wahrscheinlich				
Möglich			RT03	
Unwahrscheinlich		RO01		
Selten				
Keine	RT01, RT02			

Table 25: Risikomatrix angepasst nach Serverdeployment

Nach dem Serverdeployment konnten wir einige Risiken eliminieren oder Rückstufen. Das Single-Sign-On Risiko konnte nun auch besser eingeschätzt werden und wurde dementsprechend erhöht.

7 Persönlicher Bericht

7.1 Jeriel Frei

Diese Studienarbeit war für mich als Mitglied des IT-Supportteams der Berufsschule Lenzburg eine einzigartige Gelegenheit, ein Tool zu entwickeln, das direkt meinen eigenen Arbeitsalltag verbessert. Ich konnte mein Wissen aus der Praxis einbringen und gleichzeitig wertvolle Erfahrungen in der Softwareentwicklung sammeln.

Die Zusammenarbeit mit Simeon war sehr produktiv und sehr angenehm. Während ich die fachlichen Anforderungen aus dem Support einbringen konnte, hatte er oft einen frischeren Blick auf die technische Umsetzung. Diese Kombination war sehr wertvoll.

Die Feedbackschleifen mit meinen Kollegen aus dem Support-Team waren besonders aufschlussreich. Deren Hinweise wie der Bestätigungsdialog beim Passwort-Reset oder die Anzeige der Email-Nicknamen wären mir als "betriebsblinder" Entwickler vielleicht nicht aufgefallen.

Ich freue mich darauf, das Feedback meiner Kollegen im Alltag zu bekommen und zu sehen, wie das Dashboard den Support-Workflow verbessert. In Zukunft möchte ich weitere Module hinzufügen, um das Tool noch nützlicher zu machen.

7.2 Simeon Rhyner

Die Durchführung dieser Studienarbeit stellte für mich eine wertvolle und lehrreiche Erfahrung dar, da sie zahlreiche Module aus dem Studium mit einer realen Problemstellung aus der Praxis verband.

Technisch konnte ich mein Wissen in mehreren Bereichen deutlich vertiefen. Die Arbeit mit React im Frontend, .NET im Backend, der Anbindung von externen APIs sowie der Containerisierung mit Docker stellte mich wiederholt vor neue Herausforderungen. Insbesondere die unterschiedlichen Konfigurationen der Container in der Entwicklungs- und Produktionsumgebung erwies sich komplexer als zunächst erwartet.

Ein weiterer zentraler Lernaspekt für mich war die positive Zusammenarbeit mit Jeriel. Wir konnten technische Probleme gemeinsam analysieren, Lösungsansätze diskutieren und Entscheidungen fundiert treffen. Durch den Einsatz von Code Reviews habe ich zudem gelernt, strukturierter zu arbeiten und den eigenen Code kritisch zu hinterfragen.

Insgesamt hat mich diese Studienarbeit nicht nur fachlich, sondern auch methodisch und persönlich weitergebracht. Ich konnte meine Fähigkeiten in der Planung, Umsetzung und Dokumentation eines Softwareprojekts deutlich ausbauen. Gleichzeitig hat das Projekt gezeigt, wie anspruchsvoll, aber auch motivierend es ist, eine Lösung zu entwickeln, die einen echten Mehrwert für ihre Benutzer bietet.

8 Schlussfolgerungen und Ausblick

8.1 Schlussfolgerungen

Ziel dieser Studienarbeit war die Konzeption und Umsetzung eines modularen Support-Dashboards für das IT-Supportteam der Berufsschule Lenzburg, um wiederkehrende Supportprozesse zu vereinfachen, zu zentralisieren und effizienter zu gestalten. Ausgehend von einer klar definierten Ausgangslage mit hohem manuellem Aufwand konnte mit dem BSL Support Dashboard eine praxisnahe und produktiv einsetzbare Lösung realisiert werden.

Ein wesentliches Ergebnis der Arbeit ist die erfolgreiche Umsetzung aller definierten zwingenden Functional Requirements. Die drei Kernmodule AD Passwort Reset, PRTG Server Monitor und Power Automate Monitoring adressieren direkt die identifizierten Hauptprobleme im Supportalltag. Das AD-Modul reduziert den Aufwand für Passwort-Resets signifikant, da mehrere manuelle Schritte durch einen klar strukturierten, geführten Prozess ersetzt wurden. Das PRTG-Modul ermöglicht eine übersichtliche und schnelle Beurteilung des Serverzustands, während das Power-Automate-Modul eine zentrale Sicht auf fehlgeschlagene Flow-Runs bietet und damit die Fehlerbehebung beschleunigt.

Die Zielsetzung eines modularen und personalisierbaren Dashboards wurde ebenfalls erreicht. Benutzer können die Anordnung der Module individuell anpassen, wobei die Einstellungen persistent gespeichert werden. Dadurch eignet sich das System sowohl für Support-Arbeitsplätze als auch für Info-Screens. Die modulare Architektur bestätigt zudem die Annahmen aus der Technologie-Evaluation: Neue Module können mit klar definierten Schnittstellen integriert werden, ohne den bestehenden Kern wesentlich zu verändern.

Auch in Bezug auf Sicherheit und Benutzerfreundlichkeit erfüllt die Lösung die gesetzten Ziele. Das mehrstufige Authentifizierungskonzept mit JWT-Tokens sowie das separate Admin-Login für das Modul AD Passwort Reset entsprechen gängigen Sicherheitsstandards. Die durchgeführten Usability-Tests mit dem IT-Leiter der Berufsschule Lenzburg verliefen überwiegend positiv und lieferten wertvolles Feedback, welches iterativ in die Umsetzung eingeflossen ist. Dadurch konnte eine intuitive und praxistaugliche Benutzeroberfläche realisiert werden.

Nicht alle ursprünglich angedachten Funktionalitäten konnten jedoch vollständig umgesetzt werden. So wurden gewisse optionale Erweiterungen der Dashboard-Einstellungen (z. B. feinere Layout- oder Designanpassungen) bewusst zurückgestellt, da der Implementierungsaufwand den Rahmen der Studienarbeit überschritten hätte.

Zusammenfassend kann festgehalten werden, dass die Arbeit ihre Ziele weitgehend erreicht hat und eine stabile, erweiterbare und produktiv nutzbare Lösung hervorgebracht wurde.

8.2 Ausblick

Aus der Arbeit ergeben sich mehrere sinnvolle Ansatzpunkte für Weiterentwicklungen. Ein nächster Schritt ist die Erweiterung des modularen Dashboards um zusätzliche Support-Module. Dank der bestehenden Architektur können solche Erweiterungen gezielt und schrittweise umgesetzt werden.

Ein weiterer wichtiger Punkt betrifft die Vertiefung der Personalisierungsmöglichkeiten. Zukünftig könnten zusätzliche Layout-Optionen, modulare Konfigurationen oder rollenbasierte Standard-Dashboards eingeführt werden.

Abschliessend bietet das BSL Support Dashboard eine solide Grundlage für zukünftige Erweiterungen und zeigt exemplarisch, wie moderne Webtechnologien gezielt eingesetzt werden können, um administrative IT-Prozesse nachhaltig zu verbessern.

9 Time Tracking Report

In diesem Kapitel befindet sich die Übersicht über die tatsächlichen Arbeitsaufwände, die im Projekt erfasst wurden. Alle Arbeitsaufwände wurden in Asana auf die entsprechenden Kategorien gebucht und werden hier in Form von Grafiken dargestellt.

9.1 Zeit pro Person

Die folgende Grafik zeigt die aufgewendeten Stunden pro Person im Projekt.

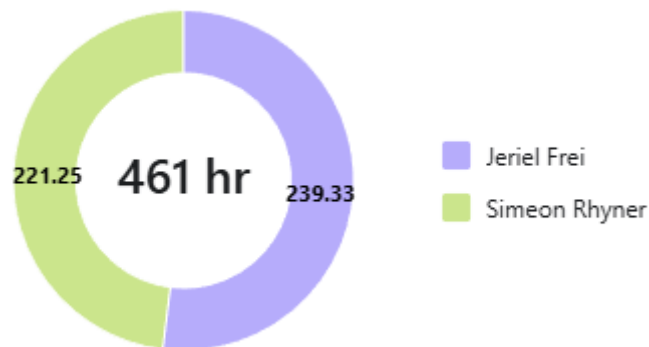


Figure 44: Arbeitsaufwände pro Person

9.2 Zeit pro Kategorie

Das Zeittracking wurde in verschiedene Kategorien unterteilt, um eine bessere Übersicht über die aufgewendeten Arbeitszeiten zu erhalten. Die folgende Grafik zeigt die aufgewendeten Stunden pro Kategorie.

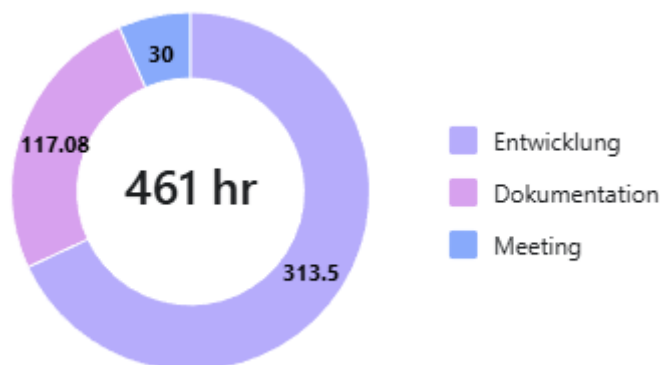


Figure 45: Arbeitsaufwände pro Kategorie

10 Glossary und Abkürzungen

Abkürzung	Bedeutung
PRTG	Paessler Router Traffic Grapher (Monitoring Software)
JWT	JSON Web Token
API	Application Programming Interface
BSL	Berufsschule Lenzburg

11 Abbildungsverzeichnis

Figure 1	Systemarchitektur des BSL Support Dashboards	4
Figure 2	AD Passwort Reset Modul	5
Figure 3	PRTG Server Monitor	6
Figure 4	Power Automate Monitoring	6
Figure 5	Einstellungen	7
Figure 6	BSL Support Dashboard Gesamtansicht	8
Figure 7	Use Case Diagram	13
Figure 8	Systemarchitektur des BSL Support Dashboards	22
Figure 9	C4 Diagramm der Container	28
Figure 10	Stages der Pipeline	29
Figure 11	Aktivitätsdiagramm des Logins und des Dashboardes	32
Figure 12	Mockup: Dashboard	33
Figure 13	Login der Applikation	34
Figure 14	Dashboard der Applikation	35
Figure 15	Aktivitätsdiagramm des Modules AD Passwort Reset	37
Figure 16	Mockup: Auswahl der Funktion des Benutzers	38
Figure 17	Mockup: Auswahl des Bereiches vom Benutzer	38
Figure 18	Mockup: Eingabe der Email-Adresse	39
Figure 19	Mockup: Anzeige des neuen Passworts	40
Figure 20	Login Seite des AD Modules	41
Figure 21	Benutzersuche im AD Modul	42
Figure 22	Suchergebnisse im AD Modul	43
Figure 23	Bestätigungsdiallog zum Zurücksetzen des Passworts	43
Figure 24	Anzeige des neuen Passworts	44
Figure 25	Backend Endpoints des AD Modules	44
Figure 26	Aktivitätsdiagramm: PRTG Server Monitor Modul	46
Figure 27	Mockup: PRTG Server Monitor Modul in Figma	47
Figure 28	PRTG Server Monitor Modul	48
Figure 29	PRTG Server Monitor Modul mit erweitertem Device	49
Figure 30	PRTG Filtermöglichkeiten	49
Figure 31	PRTG Filter auf Sensoren mit Warnung	50
Figure 32	Backend Endpoints des PRTG Server Monitor Modules	50
Figure 33	Aktivitätsdiagramm des Modules AD Passwort Reset	52
Figure 34	Mockup: Powerautomate Flow Runs Fehler Modul aus Figma	53
Figure 35	Modul ohne aktiven Filter	54
Figure 36	Modul mit Einträgen aktivem Filter	55
Figure 37	Modul nach dem ausführen der Aktion "als abgeschlossen markieren"	56
Figure 38	Backend Endpoints des Powerautomate Flow Runs Fehler Modules . .	56
Figure 39	Ablaufdiagramm: Einstellungen	58
Figure 40	Mockup: Einstellungen im Figma	59
Figure 41	Screenshot der Einstellungen	60
Figure 42	Postgres Tabelle der Einstellungen	60

Figure 43 Backend Endpoints der Einstellungen	61
Figure 44 Arbeitsaufwände pro Person	86
Figure 45 Arbeitsaufwände pro Kategorie	86

12 Tabellenverzeichnis

Table 1	Liste aller Functional Requirements, die aus dem Use Case UC01 abgeleitet wurden	14
Table 2	Liste aller Functional Requirements, die aus dem Use Case UC02 abgeleitet wurden	15
Table 3	Liste aller Functional Requirements, die aus dem Use Case UC03 abgeleitet wurden	16
Table 4	Liste aller Functional Requirements, die aus dem Use Case UC04 abgeleitet wurden	17
Table 5	Liste aller Functional Requirements, die aus dem Use Case UC05 abgeleitet wurden	18
Table 6	Functionality Non-Functional Requirements	19
Table 7	Usability Non-Functional Requirements	20
Table 8	Reliability Non-Functional Requirements	20
Table 9	Supportability Non-Functional Requirements	21
Table 10	Mapping zwischen den Suchfeldern und den AD Feldern	42
Table 11	Branch Prefixes	63
Table 12	Commit Types	64
Table 13	Legende Testbewertung	66
Table 14	Testprotokoll Part Functionality - 12.12.2025	66
Table 15	Testprotokoll Part Usability - 12.12.2025	66
Table 16	Testprotokoll Part Reliability - 12.12.2025	67
Table 17	Testprotokoll Part Supportability - 12.12.2025	67
Table 18	Milestones	76
Table 19	Milestones	77
Table 20	Risiko-Liste Teil Technisch	79
Table 21	Risiko-Liste Teil Organisatorisch	80
Table 22	Legende	81
Table 23	Risikomatrix	81
Table 24	Risikomatrix angepasst nach Prototyp	81
Table 25	Risikomatrix angepasst nach Serverdeployment	82

Part V

Anhang

Hilfsmittelverzeichnis

Im Rahmen dieser Arbeit wurden folgende KI-Assistenten und Tools eingesetzt:

Aufgabenbereich	Tools
Recherche	Google, ChatGPT, Claude, YouTube, Stack Overflow,
Datenanalyse und Visualisierung	Drawio, ChatGPT, Claude
Prototyping	Figma, Goodnotes
Coding	Visual Studio Code, JetBrains, ChatGPT, Copilot, React, .Net,
Texterstellung, Textoptimierung, Rechtschreib- und Grammatikprüfung	Copilot, ChatGPT, Claude, typst
Zusammenarbeit und Projektmanagement	Outlook, Teams, GitLab, Asana, WhatsApp, typst
DevOps	Docker, GitLab,