

FACHHOCHSCHULE AACHEN, CAMPUS JÜLICH

FACHBEREICH 09 - MEDIZINTECHNIK UND TECHNOMATHEMATIK
STUDIENGANG ANGEWANDTE MATHEMATIK UND INFORMATIK

SEMINARARBEIT

Konzeptionierung einer Web-Anwendung zur Reorganisation der Ausbildung bei der PSI Software SE

Autor:

Pia Pflug, 3621293

Betreuer:

Prof. Dr. rer. nat. Alexander Voß
Maurice Fielau

Aachen, 15. Dezember 2025

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Seminararbeit mit dem Thema

Konzeptionierung einer Web-Anwendung zur Reorganisation der Ausbildung bei der
PSI Software SE


selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Ich verpflichte mich, ein Exemplar der Seminararbeit fünf Jahre aufzubewahren und auf Verlangen dem Prüfungsamt des Fachbereiches Medizintechnik und Technomathematik auszuhandigen.

Im Rahmen der Erstellung dieser Arbeit wurde das KI-System „KI.connect.nr“ unterstützend zur sprachlichen Überarbeitung sowie zur fachlichen Reflexion und Präzisierung eigenständig entwickelter Argumente genutzt. Eine Übernahme von KI-generierten Texten oder inhaltlichen Lösungsvorschlägen erfolgte nicht. Sämtliche fachlichen Aussagen, Bewertungen und Schlussfolgerungen wurden eigenständig erarbeitet und verantwortet. Die Nutzung erfolgte im Einklang mit der Zweckbestimmung des Systems sowie unter Beachtung datenschutz- und urheberrechtlicher Vorgaben.

Name: Pia Pflug

Aachen, den 15. Dezember 2025



Unterschrift der Studierenden

Zusammenfassung

Bei der PSI Software SE soll eine Anwendung zur Reorganisation der Ausbildung entwickelt werden. Diese soll einerseits das abgeschaltete Berichtsheft-Verwaltungstool der IHK Aachen ablösen und andererseits die internen Ausbildungsabläufe unterstützen. Daraus ergeben sich zwei Kernfunktionalitäten, die das System bereitstellen muss: Die Verwaltung von Ausbildungsnachweisen und eine Dokumentationsmöglichkeit für wöchentliche Statusmeetings.

In dieser Arbeit wird eine entsprechende Webanwendung konzeptioniert. Dafür wird zuerst eine Marktanalyse durchgeführt. Hierbei wird festgestellt, dass die zur Verfügung stehenden Systeme entweder den falschen Funktionsumfang bieten oder zu hohe Kosten mit sich bringen. Diese Erkenntnis begründet den Entschluss eine eigene Anwendung zu entwickeln. Um eine konkrete Konzeption zu ermöglichen, werden im Rahmen einer Anforderungsanalyse zunächst verschiedene User-Stories ermittelt. Aufbauend darauf wird dann ein auf der Client-Server-Architektur basierendes Umsetzungskonzept erarbeitet. Dieses beschreibt alle benötigte Komponenten und enthält zudem bereits ein Persistenzkonzept. In einem nächsten Schritt kann auf Basis des in dieser Arbeit vorgestellten Konzepts die Implementierung mit dem Framework Angular erfolgen.

Inhaltsverzeichnis

1	Einleitung	2
1.1	Motivation und Ziel der Arbeit	2
1.2	Herangehensweise	3
2	Stand der Technik	4
2.1	Ausbildungsnachweise	4
2.2	Statusmeetings	6
3	Anforderungsanalyse	8
3.1	Ausbildungsnachweise	8
3.2	Statusmeetings	10
3.3	Sonstiges	12
4	Konkrete Konzeptionierung	14
4.1	Systemarchitektur	14
4.1.1	Client-Server-Architektur	14
4.1.2	Kommunikationsprinzipien	15
4.2	Konkrete Systemkomponenten	15
4.2.1	Frontend-Komponenten	16
4.2.2	Backend-Komponenten	20
4.2.3	Datenbank- und Persistenzkomponenten	23
4.3	Zusammenfassung der Konzeption	25
5	Ausblick	27
6	Fazit	29
	Abbildungsverzeichnis	30
	Tabellenverzeichnis	31
	Literatur	32
A	Ausschnitt Ankündigung der Plattform Anubis	33
B	Request-Struktur der REST-API	34

1 Einleitung

1.1 Motivation und Ziel der Arbeit

Als wachsendes Unternehmen beabsichtigt die PSI Software SE zu expandieren. Mit der dualen MATSE-Ausbildung hat der Standort in Aachen eine sehr gute Möglichkeit, neue Mitarbeiter zu gewinnen. Daher wird das Ausbildungsprogramm in den letzten Jahren weiter gefördert und vergrößert.

Die Ausbildenden sind in der Verantwortung, die Auszubildenden so zu unterstützen, dass ein erfolgreicher Abschluss der Ausbildung möglich ist. Dazu müssen während der Ausbildung die Anforderungen der Industrie- und Handelskammer, kurz der IHK, erfüllt werden. Eine Vorgabe der IHK sind die Ausbildungsnachweise, die regelmäßig geführt werden müssen. In den Nachweisen sollen die Auszubildenden berichten, welche Aufgaben sie in dem Ausbildungsbetrieb übernommen haben und was sie dabei gelernt haben. Diese Ausbildungsnachweise müssen von den Ausbildenden geprüft werden. Bis zum 1. August 2025 hat die IHK Aachen selber ein Portal zum Verwalten dieser Ausbildungsberichte gewährleistet. Dies wurde allerdings eingestellt, was alle Ausbildungsbetriebe zu einer Umstellung nötigte. Bei der PSI wurde der Einfachheit halber entschieden, die Nachweise über einen simplen E-Mail-Verkehr zwischen den Ausbildenden und den Auszubildenden kontrollieren zu lassen. Dieser Prozess ist jedoch nicht optimal und führt zu dem Wunsch nach passenderen Anwendung.

Ein zusätzlicher Teil des Ausbildungsangebots bei der PSI sind wöchentliche Statusmeetings. Sie sind keine Vorgabe der IHK, aber ein von den Auszubildenden wertgeschätzter Termin. In diesem Meeting erhalten die Ausbildenden einen Überblick über die anstehenden und abgeschlossenen Aufgaben der Auszubildenden. Außerdem gibt das Meeting den Ausbildenden die Chance, wichtige Ereignisse zum Unternehmen oder innerhalb der Ausbildung anzusprechen und mit den Auszubildenden auszudiskutieren. Diese Statusmeetings werden von den Ausbildenden dokumentiert und im Anschluss an die Auszubildenden weitergeleitet, sodass alle Zugriff auf die Dokumentation erhalten. Auch dafür werden derzeit E-Mails versendet. Durch eine Anwendung soll die Dokumentation überarbeitet und auch der Umgang mit dieser verbessert werden.

Mit der immer größer werdenden Dimension der Ausbildung verstärkt sich das Bedürfnis nach neuen Strukturen, die sowohl die Ausbildenden als auch die Auszubildenden unterstützen sollen.

Ziel dieser Arbeit ist daher die Konzeptionierung einer Anwendung, die die Organisation der Ausbildungsnachweise und Statusmeetings bei der PSI Software SE unterstützt.

1.2 Herangehensweise

Zuerst wird in [Kapitel 2](#) untersucht, welche Anwendungen bereits von externen Anbietern existieren und ob diese den Prozessen bei der PSI entsprechen. Dabei wird zwischen der Ausbildungsnachweisverwaltung und einer Dokumentationsinstanz unterschieden.

Anschließend wird in [Kapitel 3](#) auf Basis der verglichenen Systeme und den bisherigen Abläufen genauer analysiert, welche Anforderungen an die Anwendung bestehen. Es werden funktionale und nicht-funktionale Anforderungen an die Anwendung aufgestellt und in Form von User Stories festgehalten.

In [Kapitel 4](#) wird eine zur Implementierung verwendbare Architektur vorgestellt. Weitergehend wird konkret auf die erarbeiteten Anforderungen eingegangen und ein Vorschlag zu Umsetzung der Anwendung in den vorgestellten Strukturen geliefert. Dabei wird auf die zentralen Frontend- und Backend-Komponenten sowie auf die Datenspeicherung und die Kommunikation zwischen den Komponenten eingegangen.

Zuletzt werden in [Kapitel 5](#) mögliche Erweiterungen der konzeptionierten Anwendung vorgestellt.

2 Stand der Technik

In diesem Kapitel wird der aktuelle Stand der Technik im Bereich der Ausbildungsverwaltung und Statusmeetings untersucht. Da es keine Anwendung gibt, die die beiden Hauptfunktionalitäten vereint, wird die Marktanalyse getrennt vorgenommen. Dabei wird aufgezeigt, welche Systeme und Anwendungen derzeit für die Verwaltung von Ausbildungsnachweisen existieren und wie die Abläufe der Durchführung und Dokumentation von Statusmeetings sind. Ziel ist es, die bestehenden Lösungen hinsichtlich ihrer Funktionalität, ihrer Vor- und Nachteile sowie ihrer Eignung für unsere spezifischen Anforderungen zu analysieren und eine Lösung zu finden.

2.1 Ausbildungsnachweise

Zu Beginn meiner Ausbildung gab es bereits ein etabliertes System. Als Teil der organisatorischen Web-Anwendung **AzubiOnline** gab es einen Bereich, der sich Berichtsheftführung nannte. Die einzige Funktionalität dieses Bereiches war die Verwaltung der Ausbildungsnachweise. AzubiOnline ist bis heute noch aktiv und zugänglich, da es sich um die offizielle Anwendung zur Verwaltung von Ausbildungen handelt. Was allerdings im Sommer 2025 abgeschaltet wurde, ist die digitale Berichtsheftführung. Als Alternative hat die IHK Aachen eine Reihe von anderen Anwendungen, die sich auf Berichtsheftführungen in Ausbildungen fokussieren, empfohlen. [1]

Durch diese Abschaltung waren auch wir gezwungen, eine Lösung zu finden. Als **Zwischenlösung** haben wir uns der Einfachheit halber gegen eine Anwendung entschieden. Stattdessen werden die Ausbildungsnachweise am Ende des Monats per E-Mail an die Ausbildenden gesendet. Auch das Feedback zu den Ausbildungsnachweisen erfolgt über den E-Mail-Verkehr. Für das Abspeichern der Nachweise sind die Auszubildenden eigenständig verantwortlich. Dieses System funktioniert zwar, ist allerdings nicht optimiert. Es verlangt viel Organisation auf beiden Seiten der Ausbildung. Die Auszubildenden müssen die Ausbildungsnachweise lokal organisiert abspeichern und daran denken, ihre Ausbildungsnachweise abzuschicken. Die Ausbildenden auf der anderen Seite erhalten am Anfang jeden Monats E-Mails mit Ausbildungsnachweisen im Anhang, die sie selber herunterladen und organisieren müssen. Das kann zum Beispiel zu Verwechslungen der Auszubildenden oder Vernachlässigung eines fehlenden Ausbildungsnachweises führen. Um diese Schwachstellen zu beseitigen, möchten wir uns von der Zwischenlösung entfernen und eine optimierte Anwendung nutzen.

Die von der IHK Aachen empfohlenen Anbieter verfügen alle über die grundlegenden Funktionalitäten. Die Auszubildenden können ihre Ausbildungsnachweise schreiben, die Ausbildenden haben Einsicht und können Feedback geben. Jedoch sind die Prozesse nicht immer vollständig nach unseren Bedürfnissen geregelt. So ist es beispielsweise bei **Azubiheft** nicht möglich,

einen Ausbildungsnachweis hochzuladen. Stattdessen kann nur online eine Vorlage ausgefüllt werden. Diese Vorlage ist für eine tägliche Ausfüllung ausgelegt und somit unbrauchbar für uns. Eine Umstellung auf tägliche Ausbildungsnachweise würde zu einem Übermaß an Organisation führen, und ist nicht für eine Ausbildung im Mathematischen und Informatischen Feld ausgelegt. Es kommt durchaus vor, dass eine Aufgabe über mehrere Tage andauert und dann nicht in verschiedenen Ausbildungsnachweisen erklärt werden sollte. [2]

Viele Anbieter bieten auch eine mobile Anwendung an, in der sowohl Auszubildende als auch Auszubildende alle Funktionen mobil nutzen können. So legt auch **Zubido** besonders viel Wert auf ihre Mobilität. Obwohl uns zur Ausbildung ein dienstliches Mobiltelefon zur Verfügung gestellt wird, liegt der Fokus der Ausbildung auf einer Arbeit an einem stationären Computer, der jederzeit über einen Internetzugang verfügt. Daher möchten wir eine Anwendung, die hauptsächlich über einen Desktop erreichbar ist. So scheidet auch Zubido mit seinem ausgeprägten Appangebot aus. [3]

Ein anderer vorgeschlagener Anbieter fokussiert sich auf einen spielerischen Ansatz in der Ausbildung. **Azubi:web** wirbt mit ausgeprägten Lernspielen und Duellfunktionen. Auch die Organisation einer solchen Funktionalität überschreitet den Rahmen der Ausbildung bei der PSI. Die Anwendung ist auf größere Unternehmen ausgelegt, in denen nicht jeder Auszubildende Kontakt mit seinen Mitazubildenden hat. [4]

Ähnliche Problematiken kommen bei Anbietern wie **BLok** und **u-form** auf. Sie legen zwar nicht viel Wert auf den spielerischen Aspekt, sind aber eindeutig für größere Ausbildungsbetriebe ausgelegt. Sie vermarkten die vollständige Ausbildung zu organisieren, wobei das Berichtsheft nur ein kleiner Teil des Produkts ist. Diese weiteren Funktionalitäten würden in unserer Anwendung wahrscheinlich ungenutzt bleiben, müssen jedoch trotzdem bezahlt werden. [5] [6]

Allgemein verlangen alle Anbieter mindestens eine regelmäßige Zahlung. Die Preise variieren in den Basisprodukten zwischen 1€ und 4€ pro Auszubildender pro Monat. Bei durchschnittlichen 6 Auszubildenden, die bei der PSI in Ausbildung sind führt das zu 72€ - 288€ an jährlichen Kosten. Im Vergleich zu den vorherigen kostenlosen Vorgängen würde uns eine günstigere Variante mehr Zusage.

Eine kostenlose Alternative der RWTH Aachen hat sich im Laufe der Arbeit ergeben. Von den MATSE-Diensten wurde am 9.12.2025 angekündigt, dass im ersten Quartal des Jahres 2026 die Plattform **Anubis** zur Verfügung stehen soll. Da bisher keine weiteren Informationen zu dem Umfang der Anwendung bekannt sind, lässt sich nicht einschätzen, ob die Anwendung den Anforderungen der PSI gerecht werden kann. Allerdings kann davon ausgegangen werden, dass die Anwendung von Auszubildenden des IT-Centers erstellt wurde, was bedeutet, dass auch Anubis sich an dem abgeschalteten Berichtsheft von AzubiOnline orientiert. Mit dem Versprechen, dass die Anwendung an den Start gehen wird kann mit Bugs zu rechnen sein, dessen Bearbeitung bei der RWTH und daher außer unserer Verantwortung liegt. Auch in dieser Anwendung werden die Server erneut extern gehandhabt. Da zu dem aktuellen Stand keine weiteren Informationen existieren, kann die Anwendung nicht weiter in Betracht genommen werden. [Anhang A]

In die Entscheidung gegen die vorgestellten Angebote fließt auch ein, dass selbst die umfangreicheren Anbieter keine passende Lösung für unsere Statusmeetings bereitstellen.

2.2 Statusmeetings

Die Statusmeetings finden wöchentlich freitags statt und folgen immer dem gleichen Ablauf. Alle Beteiligten der Ausbildung, was meist nur die Auszubildenden und Ausbildenden sind, treffen sich zusammen. Zu Beginn gibt es einen allgemeinen Themenblock der Ausbildenden, in dem Themen besprochen werden, die in der Woche aufgekommen sind, oder allgemein relevant sind. Anschließend werden alle Auszubildenden der Reihe nach gebeten zu berichten. Relevant sind die Aufgaben, die in dieser Woche geschafft wurden, die Aufgaben, die in der folgenden Woche anstehen und relevante Themen aus dem Studium. Zu den sogenannten Uni-Themen gehören Klausurtermine, aber auch persönliche Einschätzungen der Module. Die Uni-Themen werden allerdings nicht jede Woche besprochen, sondern nur sobald sich etwas ändert oder Bedarf zur Dokumentation besteht. Das vollständige Statusmeeting wird von den Ausbildenden auf einer OneNote-Seite dokumentiert. Am Ende des Meetings wird allen Teilnehmern ein Screenshot der Dokumentation via E-Mail zugeschickt.

Dieser Prozess ist ein wichtiger Teil der Ausbildung bei der PSI und ein großer organisatorischer Prozess. Die Ausbildenden müssen vor dem Statusmeeting die Dokumentation kopieren und eventuell Anpassungen machen. So kam der Wunsch nach einer Hilfestellung auf.

Als Alternative zu den One-Note Seiten gab es keine direkten Vorschläge. Bisher wird auch im Unternehmen für andere Projekte kein Tool verwendet. Die Projekte werden über das Ticketsystem **Jira** organisiert und in Meetings wird lediglich gemeinsam über die Tickets geschaut und anhand dieser werden dann auch die Aufgaben verteilt und protokolliert. Für die Ausbildung ist eine Adaption jedoch unpraktisch, da die Auszubildenden im Verlauf der Ausbildung in den Projekten mitarbeiten. Die Tickets müssten dann aus den Projekten geklont werden, um sie dem Ausbildungsprojekt zuordnen zu können. Das führt zu einer doppelten Auslastung der Auszubildenden und neue Möglichkeiten zur Verwirrung, welches Ticket sich nun bei wem und in welchem Status befindet. Diese Variante ist also keine plausible Alternative.

Mit ein wenig Recherche kommt man auf verschiedene Anbieter, die auf Statusmeetings und deren Dokumentation spezialisiert sind. Sie erlauben, Meetings zu planen und regelmäßig vorbereitete Vorlagen einzustellen. Allerdings sind die meisten Anwendungen darauf aus, das Meeting zu ersetzen. Das erreichen sie indem alle Teilnehmer anstelle von einem Meeting eigenständig ihren Fortschritt eintragen. Unsere Recaps dienen zwar auch den Ausbildenden dazu den aktuellen Status der Auszubildenden in ihren Aufgaben einzuordnen. Ihre zweite Funktion ist es jedoch den Auszubildenden eine Möglichkeit zu geben alles Mögliche an Themen mit den Ausbildern in einer geschlossenen Veranstaltung zu besprechen. Um diesen Aspekt nicht zu vernachlässigen, ist es keine Option das Statusmeeting ausfallen zu lassen und eine digitale Umfrage einzurichten. Anwendungen wie zum Beispiel **Friday** setzen hauptsächlich auf diesen Ansatz und scheiden daher in der Auswahl aus.[7]

Natürlich geht nicht jede Anwendung die digitale Meetings anbietet davon aus, dass das Meeting daraufhin ersetzt wird. **Parabol** als Beispiel bietet an, dass die Statusupdates ohne Meeting und asynchron ablaufen oder auch dass alle synchron ihre Daten eingeben. Anschließend können die Ergebnisse besprochen werden und daraus resultierende Aufgaben zugeteilt werden. Ein großer Vorteil von Parabol ist das kostenlose Angebot, was die Organisation von zwei Teams erlaubt. Zusätzlich wird angeboten, dass alle Beteiligten eine Zusammenfassung zugeschickt

bekommen, sobald das Meeting beendet wird, was eine Eigenschaft ist, die erhalten bleiben soll. Die Meetings lassen sich auch einplanen, sind jedoch sehr dynamisch angelegt. Jedes Meeting müssen alle Beteiligten die eingestellten Fragen erneut beantworten. Daten, die nur selten verändert werden, wie der Studienbereich, lassen sich nicht ohne Weiteres einstellen, und auch das Eintragen allgemeiner Besprechungsthemen ist umständlich. Damit ist Parabol eine gute Alternative, die jedoch gewisse Einschränkungen mitbringt.[8]

Ein alternativer Ansatz zum eigenen Jira Projekt wäre eine Anwendung die eine Kollaboration mit Jira eingeht. Dadurch würden alle Tickets die die Auszubildenden bearbeiten nur verlinkt werden. Ein solches Angebot hat zum Beispiel **weekdone**. Es bietet eine Vielzahl an Statistiken und verschiedenen Ansichten, die den Projektleitenden, also in unserem Anwendungsfall den Auszubildenden, helfen, den Überblick über das Team zu behalten. Die Problematik ist, dass es sich um eine kontinuierliche Ticketverwaltung handelt. Unser Bedarf ist jedoch eine Zustandsaufnahme des wöchentlichen Stands, der allen zur Verfügung gestellt wird. Zusätzlich müssen solche Anwendungen gewisse Sicherheitsvorgaben erfüllen, was dafür sorgt, dass sie kostenintensiver sind. Daher ist auch dieser Vorschlag nicht geeignet. [9]

Generell sind Sicherheit und der Datenschutz Aspekte, die in die Entscheidung einfließen. Externe Anbieter bieten eine erhöhte Ausfallsicherheit. Dagegen steht allerdings das Risiko, dass unklar ist, was mit den Daten in Falle eines Ausfalls passiert. Auch bei einem stabil laufenden Anbieter ist es unsicher, wo die Daten gespeichert werden. Bei einer In-House-Alternative würde diese Unsicherheit wegfallen. Obwohl es zu mehr Ausfällen kommen kann, ist jederzeit vergewissert, dass alle Daten firmenintern bleiben. Da in unserem Anwendungsfall die Datensicherheit stärker gewichtet wird als die Ausfallsicherheit liegt es nahe eine eigene Alternative zu entwickeln.

Allgemein besteht zusätzlich der Wunsch die Statusmeetings und die Ausbildungsnachweise zusammenzuführen, was von keiner der vorgestellten Anwendungen gewährt wird. Da sich weder für die Ausbildungsnachweise noch für die Statusmeetings ein perfekter Kandidat findet, haben wir uns entschieden, eine eigene Anwendung zu schaffen. Diese soll nun im Weiteren genauer geplant werden.

3 Anforderungsanalyse

Mit dem Wunsch nach einem eigenem Produkt ist es notwendig, eine eigene ausführliche Analyse durchzuführen, welche Anforderungen an eine Anwendung bestehen. Die Analyse und die daraus resultierenden Ergebnisse werden in diesem Kapitel aufgelistet. Dazu werden vorerst die speziellen Anforderungen, die an die Ausbildungsnachweis-Funktion und die Statusmeetings gestellt sind, erarbeitet und anschließend weitere Aspekte, wie zum Beispiel Benutzerverwaltungen, betrachtet.

3.1 Ausbildungsnachweise

Für die Anforderungen, die an die Ausbildungsnachweisanwendung bestehen, können die Abläufe aus dem abgeschalteten AzubiOnline übernommen werden. Die Hauptfunktionalitäten dieser Abläufe werden zur Veranschaulichung in einem Anwendungsfalldiagramm dargestellt.

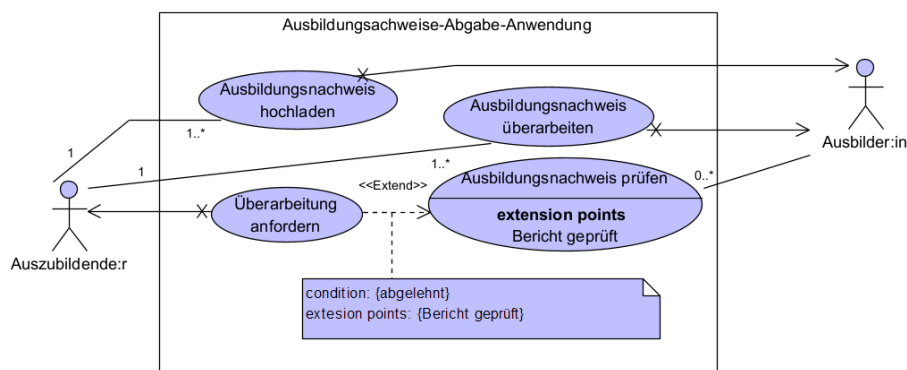


Abbildung 3.1: Visualisierung des Abgabeprozesses von Ausbildungsnachweisen

Die Auszubildenden sollen jederzeit einen neuen Ausbildungsnachweis hochladen können. Dieser Nachweis wird dann so abgespeichert, dass die Ausbildenden Zugriff erhalten. Die Ausbildenden sollen dann die Ausbildungsnachweise überprüfen können. Sie entscheiden, ob sie den Ausbildungsnachweis annehmen oder ablehnen. Wenn sie einen Nachweis ablehnen, sollen sie den Auszubildenden einen Kommentar hinzufügen müssen. So erhalten die Auszubildenden eine Rückmeldung, was sie an ihrem Ausbildungsnachweis zu verbessern haben. Sie sollen dann ihre Ausbildungsnachweise überarbeiten können, um die Fehler auszubessern. Sobald die Auszubildenden die Ausbildungsnachweise annehmen, sollen die Auszubildenden sie nicht

mehr bearbeiten können. Die Ausbildenden sollen trotzdem noch jederzeit die Möglichkeit erhalten den Nachweis abzulehnen, sollte ihnen im Nachhinein noch etwas auffallen.

Außerdem sollen die Ausbildenden informiert werden, wenn Auszubildende einen Ausbildungsnachweis abgeben oder eine Abgabe überarbeiten. Zusätzlich möchten auch die Auszubildenden informiert werden, sobald sich der Abgabestatus eines Ausbildungsnachweises ändert. Durch die Benachrichtigungen wird der Korrekturprozess beschleunigt.

Eine Anforderung, die nicht abgebildet wird, ist die Erinnerungen an ausstehende Ausbildungsnachweise. Da die Auszubildenden dazu verpflichtet sind die Nachweise regelmäßig abzugeben, möchten sie gerne daran erinnert werden, falls in einem Monat noch kein Ausbildungsnachweis abgegeben wurde. Allerdings unterscheidet sich, in welchen Abständen sie eine Erinnerung erhalten möchten. Daher sollen die Auszubildenden einstellen können, mit welcher Regelmäßigkeit sie die Erinnerungen erhalten.

Um die Nutzerfreundlichkeit zu erhöhen, sollen die hochgeladenen Ausbildungsnachweise in der Anwendung anschaulich dargestellt werden. Es soll nicht nötig sein, die Ausbildungsnachweise herunterzuladen, um einen Einblick in sie zu erhalten. Stattdessen soll es eine Vorschau geben, in der die hochgeladenen Ausbildungsnachweise angezeigt werden. Dennoch soll es sowohl für die Auszubildenden als auch für die Ausbildenden eine Möglichkeit geben, die Ausbildungsnachweise herunterzuladen und auszudrucken.

Zudem soll den Ausbildenden eine Möglichkeit zur Verfügung gestellt werden, die abgegebenen Ausbildungsnachweise filtern zu können. So sehen sie einfach alle Nachweise eines Auszubildenden übersichtlich ein. Alternativ können sie die Ausbildungsnachweise aller Auszubildenden auch nach ihrem Korrekturzustand filtern.

Eine Vorgabe der IHK-Aachen ist es, dass zu der Abschlussprüfung alle Ausbildungsnachweise in ausgedruckter Form mitzubringen sind. Außerdem kann jederzeit von der IHK beantragt werden, einen Einblick in die Nachweise zu erhalten. Aus diesen Vorgaben ergibt sich, dass die Auszubildenden jederzeit alle ihre Ausbildungsnachweise gesammelt ausdrucken können sollen. [1]

Aus datenschutzrechtlichen Gründen sollen die Auszubildenden nur Zugriff auf ihre eigenen Ausbildungsnachweise erhalten dürfen. Die Ausbildenden brauchen diesen Zugang jedoch zwingend.

ID	User Story
US1	Als Auszubildende:r möchte ich meine Ausbildungsnachweise hochladen, damit meine Ausbildenden Einblick erhalten und sie prüfen können.
US2	Als Auszubildende:r möchte ich meine Ausbildungsnachweise bearbeiten, um Fehler verbessern zu können.
US3	Als Ausbilder:in möchte ich die eingereichten Ausbildungsnachweise korrigieren können, um deren Qualität sicherzustellen.
US4	Als Auszubildende:r möchte ich, dass meine Ausbildenden mir einen begründenden Kommentar abgeben, wenn ein Ausbildungsnachweis abgelehnt wurde, damit ich weiß, was meine Fehler waren und ich sie verbessern kann.
US5	Als Ausbilder:in möchte ich eine Nachricht erhalten, sobald die Auszubildenden einen Ausbildungsnachweis hochgeladen haben, damit ich weiß, wann ich diesen überprüfen muss.

ID	User Story
US6	Als Auszubildende:r möchte ich eine Nachricht erhalten, sobald die Auszubildenden einen Ausbildungsnachweis überprüft haben, damit ich weiß, ob dieser genehmigt wurde oder ich ihn überarbeiten muss.
US7	Als Auszubildende:r möchte ich erinnert werden, falls ich einen Ausbildungsnachweis noch nicht abgegeben habe, damit ich es nicht vergesse.
US8	Als Ausbilder:in möchte ich die Ausbildungsnachweise online einsehen können, damit ich sie nicht herunterladen muss.
US9	Als Auszubildende:r möchte ich meine Ausbildungsnachweise einzeln herunterladen können, um sie lokal abzuspeichern.
US10	Als Ausbilder:in möchte ich die Ausbildungsnachweise filtern können, um einen besseren Überblick zu erhalten.
US11	Als Auszubildende:r möchte ich meine Ausbildungsnachweise gesammelt ausdrucken können, damit ich diese zur Abschlussprüfung mitbringen kann.
US12	Als Auszubildende:r möchte ich, dass nur meine Auszubildenden und ich meine Daten einsehen können, damit meine Nachweise und das Feedback vertraulich bleiben.

Tabelle 3.1: Erarbeitete User-Stories für die Anwendung zur Abgabe der Ausbildungsnachweise

3.2 Statusmeetings

Von den Statusmeetings sollen die Abläufe übernommen werden, die derzeitig aktiv sind. Das Meeting soll weiterhin wöchentlich stattfinden und die Inhalte sollen dokumentiert werden. Auch das Zukommen lassen der Dokumentation nach Abschluss des Meetings soll erhalten bleiben. Was jedoch verändert werden soll, ist OneNote als Dokumentationswerkzeug. Das inhaltliche Layout dieser Seiten hat sich allerdings bereits etabliert und soll deswegen übernommen werden. Ein Beispiel für eine solche Dokumentation sieht folgendermaßen aus.

1

2025-11-28 MoM Recap

Freitag, 28. November 2025 12:34

Teilnehmer MF, PP

Start: 12:30

Allgemeine Themen:

- Produktentwicklung
- Weihnachtsfeier nächste Woche

Pia

2

Verfügbarkeit

Uhrzeit	Mo	Di	Mi	Do	Fr
09:00					
10:00					
11:00					
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					

Uhrzeit	Mo	Di	Mi	Do	Fr
09:00					
10:00					
11:00					
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					

Fächer	Klausurtermin	Details/Rückmeldungen/Eindrücke/Note
BWL	28.01.2026	
Lernen mit Digitalen Medien		

3

Aufgaben

Was gemacht?	Bemerkung: (Was geschafft/erledigt/zufrieden?)	Nächste Woche	Was zu erledigen?	Bemerkung: (Hilfestellung/Betreuung?)
SMF	Autotest 3	Kleine Ziele	QM	Weiter Dev Systeme aufsetzen
QM	Dev Testserver aufsetzen	Erreichbare Meilensteine		Werksbesichtigung

3

Ende: 13:00

Abbildung 3.2: OneNote-Seite eines Statusmeetings

Auf dieser Vorlage findet sich ein organisatorischer Kopf des Statusmeetings (Nummer 1 in der [Abbildung 3.2](#)) in dem festgehalten wird, an welchem Tag und zu welcher Startzeit das Meeting stattfand. Außerdem wird festgehalten, wer an dem Statusmeeting teilgenommen hat. Dazu werden interne Abkürzungen verwendet, die nur im Rahmen der Ausbildung existieren. Sie werden meist aus dem Anfangsbuchstaben des Vor- und Nachnamens gebildet.

Aus diesem Bereich sollen alle Informationen weiterhin dokumentiert werden. Bei den Kürzeln soll zusätzlich gewährleistet sein, dass diese die Auszubildenden und auch die Ausbildenden eindeutig identifizieren, damit es nicht zu Verwechslungen kommt. Da die Ausbildenden diesen Kopf bereits vor dem eigentlichen Statusmeeting bearbeiten, sollen sie auch in der Anwendung diese Informationen einstellen können, bevor das Meeting eigentlich gestartet wird.

Anschließend wird für alle Auszubildenden ein persönlicher Bereich ausgefüllt (Nummer 2 in der [Abbildung 3.2](#)). Ein großer Teil ist dabei die „Verfügbarkeit“, die einen Kalender simuliert. Dazu wurde eine Tabelle angelegt, in der die Zellen farblich darstellen, ob die Auszubildenden in der Woche des Statusmeetings und der folgenden verfügbar sind. Zudem gibt es zwei weitere Tabellen: „Uni-Themen“ und „Aufgaben“. Die Tabelle „Uni-Themen“ beinhaltet die Informationen zu den Universitätsmodulen, die die Auszubildenden zu dem Zeitpunkt belegen. Falls vorhanden Klausurtermine und Kommentare, Noten oder anderen relevanten Anmerkungen werden diese vermerkt. In den „Aufgaben“ wird in dem linken Bereich festgehalten, welche Aufgaben abgeschlossen wurden und auf der rechten Seite welche Aufgaben in der folgenden Woche angegangen werden sollen. Zu beiden Seiten werden meist das Thema oder Projekt und in einer separaten Spalte eine detailliertere Beschreibung erfasst.

Von diesem Bereich sollen die Verfügbarkeiten nicht übernommen werden. Sie werden vom Sekretariat in den Firmenkalender eingetragen und die doppelte Notation schafft nur überflüssigen Organisationsaufwand und kann unvollständig sein. Was die anderen Teile des persönlichen Bereiches anbelangt, sollen diese übernommen werden. Die Ausbildenden sollen weiterhin die relevanten Informationen zu den Universitätsmodulen und die Aufgaben der Auszubildenden dokumentieren können.

Zuletzt wird einzig der Endzeitpunkt des Meetings dokumentiert (Nummer 3 in der [Abbildung 3.2](#)). Auch dieses Element soll bestehen bleiben, um die Dauer des Statusmeetings festzuhalten.

Mit dem Übergang in eine Anwendung sollen die persönlichen Bereiche dynamisiert werden. Die Bereiche sollen nur angezeigt werden, wenn die zugehörigen Auszubildenden auch an dem Meeting teilnehmen. Ein weiterer aufgekommener Wunsch ist es, die Reihenfolge der persönlichen Bereiche vorzugeben. Durch die Erweiterung der Ausbildung ist es zu einer immer wiederkehrenden Frage geworden, in welcher Reihenfolge die Auszubildenden ihren Status vortragen sollen. Dieses Problem soll durch eine Vorgabe behoben werden. Trotzdem soll es den Ausbildenden möglich bleiben Auszubildende vorzuziehen, sollte der Bedarf bestehen.

Zu den Anforderungen an das Layout der Dokumentation kommen Anforderungen an allgemeine Abläufe hinzu. Nach dem Abschließen eines Meetings sollen allen Teilnehmern die Dokumentation zugeschickt werden können. Außerdem sollen alle Statusmeetings in einem Archiv einsehbar sein. Dabei sollen den Ausbildenden alle Meetings zur Verfügung stehen und den Auszubildenden nur die Meetings, an denen sie teilgenommen haben.

ID	User Story
US13	Als Teilnehmer:in möchte ich den Zeitpunkt und die Dauer des Statusmeetings dokumentiert haben, damit ich die Dokumentation zeitlich zuordnen kann.
US14	Als Ausbilder:in möchte ich angeben können, wer an dem Statusmeeting teilgenommen hat, um die Anwesenheit zu dokumentieren.
US15	Als Ausbilder:in möchte ich die allgemeinen Themen vor dem Meeting einstellen können, um diese im Statusmeeting besprechen zu können.
US16	Als Auszubildende:r möchte ich, dass alle Informationen zu meinen Aufgaben und meinem Studium gesammelt dokumentiert werden, damit ich diese nicht zusammensuchen muss.
US17	Als Ausbilder:in möchte ich die aktuellen und abgeschlossenen Aufgaben der Auszubildenden und die Stimmung dazu dokumentieren, damit ich einen Überblick über die Auslastung und die Interessen erhalten kann.
US18	Als Ausbilder:in möchte ich die Studienmodule und zugehörige Informationen von den Auszubildenden dokumentieren, um einen Überblick über den Verlauf des Studiums zu erhalten.
US19	Als Ausbilder:in möchte ich sonstige Wünsche und Anmerkungen der Auszubildenden eintragen können, um den Auszubildenden eine Möglichkeit zu geben sie zu äußern.
US20	Als Ausbilder:in möchte ich eine Reihenfolge für die Auszubildenden vorgeschlagen bekommen können, damit ich diese nicht entscheiden muss.
US21	Als Auszubildende:r möchte ich die Dokumentation der Statusmeetings zugeschickt bekommen können, damit ich sie lokal abspeichern kann.
US22	Als Ausbilder:in möchte ich alle Statusmeetings jederzeit einsehen können, um auf vergangene Aufgaben der Auszubildenden eingehen zu können.
US23	Als Auszubildende:r möchte ich alle Statusmeetings an denen ich Teilgenommen habe einsehen können, um den Ausbildungsnachweis einfacher mit meinen Ergebnissen zu füllen kann.

Tabelle 3.2: Erarbeitete User-Stories für die Anwendung zur Dokumentation der Statusmeetings

3.3 Sonstiges

Zu den zwei Hauptfunktionalitäten bestehen noch weitere Anforderungen an die Anwendung. Wie in dem Stand der Technik bereits erarbeitet, gehört dazu keine mobile Anwendung, da der Ausbildungsplan einen Schreibtisch als Arbeitsplatz vorsieht. So haben die Ausbildenden und Auszubildenden immer einen Zugang zum Internet, was dazu führt, dass auch eine offline Anwendung nicht notwendig ist. Die Anwendung soll einfach und sicher zu erreichen sein. Das bedeutet, dass die Nutzenden sich nur einmal in einer Sitzung anmelden müssen. Trotzdem soll Datenschutz gewahrt werden und die Anwendung sicher mit den Nutzerdaten umgehen.

Da die Anwendung vorerst nur für die Ausbildung ausgelegt sein soll, ist es nicht notwendig explizite Administratoren zu ernennen. Im Rahmen der Ausbildung kann die Verwaltung der Anwendung in den Aufgabenbereich der Ausbildenden eingerechnet werden. Dennoch muss zwischen Ausbildenden und Auszubildenden unterschieden, da sie unterschiedliche Berechtigungen haben und differenzierte Ansichten benötigen. Zu der Verwaltung der Rollen

soll eine Benutzerverwaltung existieren. Die Ausbildenden sollen eine Möglichkeit haben neue Nutzende anzulegen, grundlegende Änderungen wie den Namen oder die Berechtigungen vorzunehmen und auch Nutzende zu entfernen. Außerdem sollen sie die Passwörter aller Nutzenden zurücksetzen können, falls diese Vergessen werden.

Das Passwort zurücksetzen sollen die Nutzenden auch selber aus der Anwendung können, sobald sie angemeldet sind. Allen Nutzenden sollen die Möglichkeit gegeben sein, einzustellen, ob sie die Dokumentation nach dem Abschluss eines Statusmeetings per E-Mail zugeschickt bekommen möchten oder ihnen die Einsicht im Archiv genügt. Für die Auszubildenden soll es zusätzlich noch Einstellungsmöglichkeiten für den Zeitraum der Erinnerungen an vergessene Ausbildungsnachweise geben. Ihnen sollen auch ihr Ausbildungsstart und ihr Ausbildungsende angezeigt werden. Diese Daten zu bearbeiten soll jedoch in der Verantwortung der Ausbildenden liegen.

ID	User Story
US24	Als Nutzer:in möchte ich die Anwendung einfach von meinem Arbeitsplatz erreichen können, damit ich alles Notwendige problemlos in meinen Arbeitsalltag integrieren kann.
US25	Als Nutzer:in möchte ich mein Passwort nur einmal eingeben müssen, um einfach durch die Anwendung navigieren zu können.
US26	Als Sicherheitsbeauftragte:r möchte ich eine sichere Anwendung, damit die Daten geschützt sind.
US27	Als Ausbilder:in möchte ich die Nutzenden verwalten können, um einen sicheren und zuverlässigen Betrieb gewährleisten zu können.
US28	Als Nutzer:in möchte ich mein Passwort zurücksetzen lassen können, damit ich, falls ich mein Passwort vergesse, noch an meine Daten kommen kann.
US29	Als Nutzer:in möchte ich mein Passwort zurücksetzen können, damit ich es ändern kann.
US30	Als Auszubildende:r möchte ich meinen Ausbildungszeitraum einsehen können, um mir über die organisatorischen Formalitäten im Klaren zu sein.

Tabelle 3.3: Sonstige erarbeitete User-Stories für die Anwendung

4 Konkrete Konzeptionierung

Mit den zuvor analysierten Anforderungen lässt sich nun eine Anwendung konkret konzeptuieren. Zunächst wird hierfür ein geeignetes Architekturkonzept vorgestellt. Im Anschluss wird dieses dann durch die Erarbeitung einzelner Komponenten konkretisiert. Dabei wird zwischen Client-, Server- und Persistenzkomponenten unterschieden.

4.1 Systemarchitektur

In der Softwareentwicklung ist es üblich, Software aufzuteilen und in Gruppen der gleichen Verantwortungsbereiche einzusortieren. Dieses Prinzip nennt sich Separation of Concerns und zielt darauf ab, Anwendungen testbarer und einzelne Komponenten leichter austauschbar zu machen. Da sich diese Trennung bereits in kleineren Anwendungen etabliert hat, ist sie auch in dieser Anwendung als Ziel sinnvoll. Jedoch ist die Trennung der Verantwortlichkeiten nur ein Grundprinzip, dessen Umsetzung in der Praxis von verschiedene Softwarearchitekturen unterstützt wird. [10]

Softwarearchitekturen geben strukturierte Modelle, Muster und Regeln vor, mit denen die Aufteilung definiert werden. Es gibt Vorgaben zu der Organisation der Komponenten, der Kommunikation zwischen den Komponenten und der Aufgabenbereiche, die von den Komponenten abgedeckt werden. Die verschiedenen Architekturen haben Vor- und Nachteile und sind für unterschiedliche Anwendungstypen geeignet. Für diese Anwendung bietet sich ein Client-Server-Modell an, dessen Strukturierung nun genauer erläutert wird. [11]

4.1.1 Client-Server-Architektur

In dem Client-Server-Modell wird das System aufgeteilt in den Client, der Ressourcen anfragt, und den Server, der die angefragten Daten bereitstellt. Viele Webanwendungen sind nach diesem Modell entwickelt, da es sich anbietet einen Server aufzusetzen, der über das Internet erreichbar ist und von verschiedenen Instanzen, den Clients, angesprochen werden kann.

Der **Server**, auch Backend genannt, hat verschiedene Aufgaben. Er ist verantwortlich die Anfragen von den Clients anzunehmen und zu deuten, welche Ressourcen verlangt werden und ob der Client berechtigt ist diese zu erhalten. Anschließend ist der Server für die Weiterverarbeitung zuständig. Er sorgt für die Ausführung der Hauptlogik. Dazu gehört auch der Zugriff auf Daten, falls erforderlich. Diese Daten können in Datenbanken oder anderen hinterlegten Speicherorten liegen. Zuletzt sorgt sich der Server um die Antwort, die an den Client zurückgeschickt werden soll. Er erstellt diese in einem Format, auf das sich geeinigt wird, und schickt sie ab.

Der **Client** oder auch das Frontend ist nun in der Verantwortung, die Benutzeroberfläche bereitzustellen und zu formatieren. Er muss auf Interaktionen von den Benutzenden reagieren und die Ansichten dementsprechend anpassen. Außerdem muss der Client Anfragen an den Server tätigen, sollten ihm Ressourcen fehlen. Auch der Client kann bereits simple Sicherheitsmaßnahmen übernehmen, um den Server nicht zu überlasten.

4.1.2 Kommunikationsprinzipien

Mit der Verteilung der Aufgaben bleibt die Frage, wie der Client und der Server kommunizieren. Um dies zu lösen müssen Serverseitig Schnittstellen definiert werden, die die CRUD-Operationen bereitstellen. CRUD steht für **C**reate, **R**ead, **U**ppdate und **D**eleate. Das Akronym fasst die am häufigsten verwendeten Operationen mit einer Datenquelle zusammen. Eine Technik zur Kommunikation zwischen Servern und Clients sind die sogenannten HTTP-Requests. [12]

Das Hypertext Transfer Protokoll (HTTP) ist eine Textbasiertes Kommunikationsprotokoll, dass in die folgenden Requests aufgeteilt wird:

- Post(Create): Hinzufügen neuer Daten
- GET(Read): Auslesen vorhandener Daten
- PUT/PATCH(Update): Teilweises Verändern bestehender Daten
- Delete>Delete): Löschen von Daten

Für die Kommunikation zwischen Server und Client muss sich mit dem Protokoll auf eine Datenstruktur geeinigt werden, mit der Daten übertragen werden können. Dazu bieten sich textbasierte Formate an, da sie leicht zu übertragen und verstehen sind. Ein oft verwendetes Format ist die Kommunikation mit JSON-Objekten (JavaScript Object Notation). Die Notation gibt vor, dass Daten in einem Schlüssel-Werte-Paar und mit einer bestimmten Syntax formuliert werden. [13]

Als Schnittstellendefinition bietet sich eine REST-API an. Sie nutzt HTTP Operationen zur Kommunikation, wobei die Daten oft als JSON-Objekte übertragen werden, und basiert auf der Client-Server-Architektur. Außerdem arbeitet sie zustandslos, was bedeutet, dass der Server keine Daten über vorherige Anfragen speichert. Dadurch muss ein Client mit jeder Anfrage alle benötigten Informationen teilen. Welche dieser Informationen benötigt werden und auch welche Daten der Server zur Verfügung stellt werden in der Definition festgehalten. [12]

4.2 Konkrete Systemkomponenten

Um diese allgemein formulierte Struktur anzuwenden wird nun für die einzelnen Komponenten eine konkrete Umsetzung für die Anwendung beschrieben. Für diese Umsetzung ist das Frontend-Framework Angular vorgegeben. Es wird firmenintern für die Produktentwicklung verwendet und soll daher auch für diese Anwendung verwendet werden. Für das Backend soll sich noch nicht auf ein konkretes Framework festgelegt werden. Stattdessen wird basierend

auf der Grundlage, dass das Framework eine REST-API implementiert, beschrieben, wie sich dieses aufbauen lässt.

4.2.1 Frontend-Komponenten

Angular ist ein Framework, dass von Google zur Erstellung von Single-Page-Applications (SPA) entwickelt wurde. Eine SPA ist eine Webanwendung, die beim Laden der Seite nur eine HTML-Seite lädt und den Inhalt anschließend dynamisch, basierend auf Anfragen der Nutzenden, austauscht und aktualisiert. Angular basiert auf TypeScript und Anwendungen werden in einer komponentenbasierten Architektur erstellt. Außerdem stellt Angular für viele Teile der Webentwicklung bereits eigene Lösungen bereit. Einer dieser Aspekte ist das Routing, dass besonders in Single-Page-Applications relevant ist. [14]

Routing

Das Routing einer Anwendung lädt abhängig von dem Zustand der URL unterschiedlichen Inhalt. Es ermöglicht die Navigation zwischen verschiedenen Bereichen übersichtlich zu gestalten. Eine Navigation steigert die Qualität einer Anwendung, da sie den Nutzenden die Bedienung vereinfacht. Auch in dieser Anwendung ist daher eine Navigation vorgesehen. Zur Umsetzung des Routings in der Anwendung kann das von Angular bereitgestellte Modul verwendet werden. Es erlaubt eine einfache Einbindung und Definition der Routen. Eine Sinnvolle Aufteilung der vorgesehenen Routen könnte folgendermaßen Aussehen. [14]

- **/home:** Diese Seite stellt eine Übersicht dar, auf der die Angemeldeten zum Beispiel eine Vorschau der Ausbildungsnachweise und Statusmeetings erhalten können.
- **/records:** Unter dieser Route werden alle Informationen zu allen Ausbildungsnachweisen angezeigt, die den Angemeldeten zur Verfügung stehen.
- **/meetings:** Auf dieser Ansicht können die Angemeldeten das letzte Statusmeeting oder auch andere Statusmeetings einsehen.
- **/settings:** In den Einstellungen haben die Angemeldeten die Möglichkeit ihre Daten einzusehen und verändern.
- **/users:** Die Benutzereinstellungen sollen nur den Ausbildenden zur Verfügung stehen, damit sie die Nutzenden Verwalten können. (Zu der Vergabe der Zugriffsberechtigungen von Routen wird unter [4.2.1 Authentifizierung](#) der Ablauf genauer Beschrieben)

Den Nutzenden wird durch die Routen das Gefühl vermittelt, dass eine neue Seite geladen wurde, weshalb die Routen auch als Ansichten oder Seiten bezeichnet werden können. Wie diese Seiten strukturiert werden können wird im folgenden dargestellt (siehe [Abbildung 4.1](#)). Dabei steht das Haus für eine Weiterleitung zu der Startseite und das Zahnrad zu den persönlichen Einstellungen, die jeder Nutzende tätigen kann.

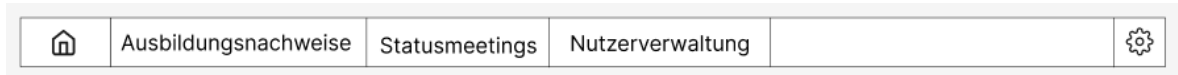


Abbildung 4.1: Navigationsleiste der Auszubildenden

Ansichten

Die Anwendung soll in die in den Routen genannten Seiten aufgeteilt werden. Im allgemeinen ähneln sich die Ansichten, die die Auszubildenden und Auszubildenden erhalten, zwar sehr, sind aber in ihrer Funktionalität unterschiedlich. Zur allgemeinen Navigation zwischen Ansichten kann jederzeit die zuvor gezeigte Navigation verwendet werden.

Durch die Komponentenbasierte Architektur von Angular lassen sich viele der Komponenten mehrfach verwenden. um dieses Potenzial auszunutzen werden die Komponenten so designed, dass sie möglichst für verschiedene Ansichten benutzt werden können, ohne stark verändert zu werden.

Herzlich Willkommen, Oscar!

Statusmeeting

31.10.2025

Allgemeine Themen:

- Christmas Party
- Ausbildungsberichte

Aaron

Diese Woche geschafft:

Was gemacht?	Bemerkung: (Was geschafft/erledigt?/Zufrieden?)
MiniPms	T4 & T5
Kaffee	Maschine gereinigt, entkalkt und aufgefüllt (Gerne wieder!)

Aufgaben nächste Woche:

Was zu erledigen?	Bemerkung: (Hilfestellung?/Betreuung?)
MiniPms	T6 & T7
AMASAP	Projekt einrichten

Uni:

Fächer	Klausurtermine	Details/Rückmeldungen/Note
Mathe Grundlagen	17.10.2025	Lief gut!
Lineare Algebra		
Analysis		
Java		
IT-Grundlagen		

Ausbildungsnachweise

Oktober 2025	AB - Warten auf Feedback
Oktober 2025	CD - Abgelehnt: August erneut abgegeben
Oktober 2025	EF - Angenommen
Oktober 2025	GH - Angenommen
Oktober 2025	IJ - Angenommen

Weitere Ausbildungsnachweise

Weitere Statusmeetings

Benutzereinstellungen

Abbildung 4.2: Startansicht der Auszubildenden

Eine wichtige Ansicht ist die Startseite (siehe [Abbildung 4.2](#)). Hier sollen die Nutzenden eine Übersicht erhalten zu sowohl ihren Ausbildungsnachweisen, als auch zu dem letzten

Statusmeeting. Dazu bietet es sich an die Ansicht aufzuteilen und dann die wichtigsten Elemente der Bereiche darzustellen.

Für die Übersicht der Statusmeetings ist von den Meetings, an denen der Nutzende teilgenommen hat, nur das aktuellste relevant. Dabei soll der Kopfbereich des Meetings immer angezeigt werden. Für die Auszubildenden soll danach ausschließlich ihr eigener Persönlicher Bereich angezeigt werden. So erhalten sie unabhängig von der Reihenfolge, in der die Auszubildenden in dem Meeting dran waren, direkten Zugriff auf ihre Daten. Da die Auszubildenden über keinen eigenen persönlichen Bereich in dem Statusmeeting verfügen wird ihnen das vollständige Statusmeeting angezeigt.

In der Komponente für die Ausbildungsnachweise wird dagegen nicht der neuste Ausbildungsnachweis dargestellt, sondern eine Liste der neusten abgegebenen Nachweise in einer Kurzfassung. Für die Auszubildenden besteht die angezeigte Kurzfassung aus dem Monat und dem Jahr, der von dem Ausbildungsnachweis abgedeckt wird und dem Status in dem sich der Ausbildungsnachweis befindet. Sollte der Ausbildungsnachweis über einen Kommentar verfügen wird auch dieser angezeigt. Die Auszubildenden erhalten zusätzlich in ihrer Kurzfassung noch einen Hinweis von welchem Auszubildenden der Ausbildungsnachweis abgegeben wurde. Über die beschriebene Komponente werden die Nutzenden weitergeleitet an die Ansicht der Ausbildungsnachweise, wo ihnen Einblick in die Nachweise gewährt wird.

Ausbildungsnachweise

Monat/Jahr	Name	Status
Oktober 2025	AB	Warten auf Feedback
Oktober 2025	CD	Abgelehnt: August erneut abgegeben
Oktober 2025	EF	Angenommen
Oktober 2025	GH	Angenommen
September 2025	AB	Angenommen
September 2025	GH	Angenommen
September 2025		

Oktober 2025

Aaron

Zustand: Warten auf Feedback

Gib einen Kommentar ab

Ausbildung zur/zum Mathematisch-technischen Softwareentwickler/in (MATSE) an der RWTH Aachen, im Forschungszentrum Jülich und kooperierenden Unternehmen


Ausbildungsnachweis

Bauer, Aaron	Oktober, 2025
Name, Vorname	Berichtszeitraum (Monat, Jahr)
Oscar Price	PSI Software SE
Name Betreuer/in	Institut / Firma
31.10.2025	
Datum	

Abbildung 4.3: Ansicht der Ausbildungsnachweise für die Auszubildenden

Auf dieser Ansicht (siehe [Abbildung 4.3](#)) kann die Komponente für die Ausbildungsnachweise aus der Startseite übernommen werden. Hier soll die Liste jedoch nicht ausschließlich informativ, sondern auch zur Auswahl des zu anzeigenden Nachweises fungieren. Den Auszubildenden wird hier auch die Möglichkeit gegeben, die Ausbildungsnachweise nach den Auszubildenden oder auch nach dem Korrekturzustand zu filtern. Der angezeigte Ausbildungsnachweis soll auch auf dieser Ansicht gezeigt werden. Dabei sollen alle Inhalte, aus der Kurzfassung und eine Vorschau

des Ausbildungsnachweises angezeigt werden. Außerdem sollen die Ausbildungsnachweise heruntergeladen werden können. Den Auszubildenden wird eine Möglichkeit gegeben den Status zu ändern und einen Kommentar abzugeben und den Auszubildenden wird das Hochladen der neuen Ausbildungsnachweise ermöglicht. Dazu wird sich eine Komponente öffnen, wo der Ausbildungsnachweis hochgeladen werden kann.

Statusmeeting am 07.11.2025 

Teilnehmer: AB, EF, OP

Allgemeine Themen:

• Klausureinsicht

Speichern

Abschließen

Aaron
Diese Woche geschafft:

Was gemacht?	Bemerkung: (Was geschafft/erledigt/Zufrieden?)
MiniPms	T4 & T5
Kaffee	Maschine gereinigt, entkalkt und aufgefüllt (Gerne wieder!)

Aufgaben nächste Woche:

Was zu erledigen?	Bemerkung: (Hilfestellung?/Betreuung?)
MiniPms	T6 & T7
AMASAP	Projekt einrichten

Uni:

Fächer	Klausurtermine	Details/Rückmeldungen/Note
Mathe Grundlagen	17.10.2025	Lief gut!
Lineare Algebra		
Analysis		
Java		
IT-Grundlagen		

Emil
Diese Woche geschafft:

Was gemacht?
MiniPms
Kaffee

Aufgaben nächste Woche:

Was zu erledigen?
MiniPms
AMASAP

Uni:

Fächer
Mathe Grundlagen
Lineare Algebra
Analysis
Java
IT-Grundlagen

Abbildung 4.4: Erstellung eines Statusmeetings

Wird von der Startseite aus mit der Komponente der Statusmeetings agiert, öffnet sich die Komponente der Statusmeetings, die analog zu der der Ausbildungsnachweise aufgebaut ist. Es gibt eine Liste in der die verschiedenen Statusmeetings untereinander angezeigt werden und eine Detailansicht, in der die Informationen des ausgewählten Meetings angezeigt werden. Jedoch wird hier nicht den Auszubildenden, sondern den Auszubildenden angeboten ein neues Element hinzuzufügen. Es öffnet sich eine andere Komponente, die die Eingabe der Teilnehmenden und der allgemeinen Themen bietet. Mit dem starten des Meetings, was die Komponente wieder schließt, öffnet sich eine neue Komponente (siehe [Abbildung 4.4](#)), die es den Auszubildenden erlaubt die Informationen aus dem Meeting einzugeben.

Zu den Daten die im Meeting eingegeben werden, gehören hauptsächlich die persönliche Bereiche der Auszubildenden. Alle Bereiche sollen gleich strukturiert werden, wobei trotzdem erkenntlich gemacht werden soll, wem der Bereich gehört. Für jeden Auszubildenden werden drei Tabellen zur Verfügung gestellt, die intuitiv editierbar sein sollen. In den Tabellen können die Auszubildenden die Inhalte, die besprochen werden, aufnehmen. Außerdem sollen die Auszubildenden das Meeting beenden können, wodurch automatisch die E-Mails mit der Dokumentation versendet werden.

Allen Nutzenden ersichtlich ist die Ansicht der persönlichen Nutzereinstellungen. Es können Daten, wie der hinterlegte Name, Nutzernamen und für Auszubildende der Ausbildungsstart bzw. das Ausbildungsende, eingesehen werden. Außerdem kann auch eingestellt werden, ob

die Nutzenden die Dokumentation eines Statusmeetings, an dem sie teilnahmen, zugeschickt bekommen möchte oder nicht. Für die Auszubildenden kann außerdem hier entschieden werden, mit welchem Abstand sie an einen vergessenen Ausbildungsnachweis erinnert werden möchten. Zusätzlich wird allen Nutzern hier die Möglichkeit gegeben ihr Passwort selber zurückzusetzen.

Ausschließlich den Ausbildenden wird die Ansicht der Benutzereinstellungen zur Verfügung gestellt. Dort werden alle Nutzenden in einer Liste aufgezeigt und, wie in den zuvor beschriebenen Ansichten, die Auswahl detailliert aufgeführt. Dabei gibt es Optionen zum Einstellen des Namens, des Nutzernamens und der Rolle des Nutzenden. Ist die Rolle auf „Auszubildender:r“ gestellt können zusätzlich noch Ausbildungsstart und Ausbildungsende eingestellt werden. Die Auszubildenden können von allen Nutzenden hier das Passwort zurücksetzen.

Zur Sicherstellung, dass nur angemeldete Nutzende Zugriff auf die Anwendung erhalten und auch dass die Auszubildenden keine Funktionen starten können, die nur den Ausbildenden gewährt sein sollen, ist die Authentifizierung verantwortlich.

Authentifizierung

Die Authentifizierung ist ein Problem, dass sowohl von dem Frontend als auch dem Backend behandelt wird. Im Frontend werden die Nutzenden über eine Eingabe gebeten ihre Anmeldeinformationen einzugeben. Diese Daten werden an das Backend weitergeleitet, wo es überprüft wird. Sind die Nutzenden Berechtig, wird an das Frontend ein sogenannter JSON Web Token (JWT) geschickt. Dieser JWT enthält Informationen zu dem authentifizierten Nutzenden, bis wann er gültig ist und eine Signatur. Der Client ist nun in der Verantwortung den JWT zu speichern. Dieser Token muss zu allen Anfragen, die das Frontend an das Backend sendet, angehängt werden. Dabei wird nicht zwischen Route oder API-Anfragen unterschieden (siehe [4.2.2 Controller-Schicht](#) für API-Anfragen). Das Backend kann mit dem JWT entscheiden ob die Anfrage berechtigt ist oder abgelehnt werden muss.

Für diese Anwendung werden zur Anmeldung ein Benutzername und ein selbst vergebenes Passwort verwendet. Der Benutzername setzt sich zusammen aus dem ersten Buchstaben des Vornamen und dem Nachname der Nutzenden zusammen und wird von den Ausbildenden vergeben. So wird der Benutzername für die Nutzerin Pia Pflug **ppflug**. Das Passwort kann zwar von den Nutzenden vergeben werden muss jedoch vorgegebene Richtlinien erfüllen. Diese Maßnahme soll die Sicherheit der Anwendung steigern.

4.2.2 Backend-Komponenten

In modernen Anwendungen wird das Backend als Komponente erneut in Schichten aufgeteilt. Dadurch sollen die einzelnen Aufgaben verteilt werden und das Austauschen und Testen der Element vereinfacht werden. Allgemein lassen sich drei Schichten erarbeiten: Controller-Schicht, Service-Schicht und Repository-Schicht.

In der Controller-Schicht werden die HTTP-Anfragen, die von dem Frontend gesendet werden, entgegengenommen und validiert. Anschließend werden die passenden Services aufgerufen.

Außerdem ist die Controller-Schicht nach dem Verarbeiten der Anfrage verantwortlich dafür die Antwort an das Frontend zu schicken.

Die Services, die in der Service-Schicht liegen, sind für die Geschäftslogik verantwortlich. Sie tätigen Berechnungen, Validierungen und stoßen weitere interne Abläufe an. Dabei koordinieren sie zwischen verschiedenen Repositories und anderen Services.

Die Repository-Schicht, die auch die Datenzugriffsschicht genannt werden kann, stellt den Services Datenzugriffsoperationen zur Verfügung. Sie trennt die eigentliche Geschäftslogik von den Datenquellen, was wichtig ist um die Datenbank zum Beispiel einfach austauschen zu können.

Controller-Schicht

Um in der Controller-Schicht mit den HTTP-Anfragen umgehen zu können, müssen die validen Anfragen definiert werden. Diese Definitionen folgen einem Muster: der HTTP-Methode (GET, POST, PUT, DELETE), einem Ressourcenpfad und einem Request-Body für Daten, die Übertragen werden. Dabei gibt der Ressourcenpfad an welche Adresse die Anfrage ansprechen soll. Optional können auch Daten wie die Identifikationsnummer eines Objektes über den Pfad übergeben, der wie ein Übergabeparameter behandelt wird. Für größere Datenmengen muss der Request-Body verwendet werden. Dort werden in der JSON-Schreibweise Informationen mitgegeben. Eine solche Definition kann folgendermaßen Aussehen. Dabei handelt es sich um einen Ausschnitt der vollständigen Definition, die im [Anhang B](#) zu finden ist.

Schnittstelle	Methode	Beschreibung	User Story
/records	POST	Legt einen neuen Ausbildungsnachweis an.	US1
/records	GET	Liefert alle Ausbildungsnachweise.	US8, US11
/records/{userID}	GET	Liefert alle Ausbildungsnachweise, die einem Nutzenden zugeordnet sind.	US12
/records/{state}	GET	Liefert alle Ausbildungsnachweise, die sich im übergebenen Korrekturzustand befinden.	US10
/records/{recordID}	GET	Liefert den angegebenen Ausbildungsnachweis.	US8, US9
/records/{recordID}	PUT	Speichert die Änderungen für den angegebenen Ausbildungsnachweis.	US2 - US4
/records/{recordID}	DELETE	Löscht den angegebenen Ausbildungsnachweis.	US2

Tabelle 4.1: Beispielhafte REST-API-Endpunkte des Backends

Zu jeder Schnittstelle wird zusätzlich noch ein Response-Body definiert, der das Format, in dem der Controller die Antwort verpackt, vorgibt. Auch hier wird wieder die JSON-Schreibweise verwendet.

Zu dem Implementieren der Schnittstellendefinitionen muss im Controller zusätzlich entschieden werden, welcher Service für die Bearbeitung der Anfrage verantwortlich ist.

Service-Schicht

In der Service-Schicht liegen die verschiedenen Services hierarchisch nebeneinander. Sie haben einen bestimmten Aufgabenbereich, den sie abdecken. Sollte eine Anfrage Funktionalitäten eines anderen Services beanspruchen, so kann dieser einfach aufgerufen werden. Für diese Anwendung lassen sich folgende Aufgabenbereiche und dadurch auch Services erarbeiten:

- **AuthService:** Dieser Service ist für alle Anfragen, die im Zusammenhang mit der Validierung von Nutzenden stehen, verantwortlich.
- **UserService:** Dieser Service ist für alle Anfragen, die im Zusammenhang mit der Verwaltung der Nutzenden stehen, verantwortlich.
- **RecordService:** Dieser Service ist für alle Anfragen, die im Zusammenhang mit den Ausbildungsnachweisen stehen, verantwortlich.
- **MeetingService:** Dieser Service ist für alle Anfragen, die im Zusammenhang mit den Statusmeetings stehen, verantwortlich.
- **NotificationService:** Dieser Service ist für Benachrichtigungen durch das Senden von E-Mails verantwortlich.

Das Besondere an dem **NotificationService** ist, dass es kein fachliches Modul ist, sondern von allen Modulen genutzt werden kann. Die Instanz entkoppelt die anderen Services von externen Systemen, die zur Versendung von E-Mails benutzt werden. Als anzusprechende E-Mail-Adresse wird der Benutzername, der auch zum Anmelden verwendet wurde und der Anhang der Firmendomäne genutzt.

Abgesehen von dem **NotificationService** werden alle anderen Services direkt von dem Controller angesprochen. Sie verarbeiten die Logik und holen sich benötigte Daten aus dem zugehörigen Repository.

Repository-Schicht

In der Repository-Schicht werden die Module ähnlich wie in der Service-Schicht nebeneinander angeordnet. Allerdings können die Repositories nicht auf ihre Nachbarn zugreifen. Sie sind in ihrem Aufgabenbereich beschränkt und können einzig mit den Services, die sie ansprechen, und mit den unterliegenden Datenquellen kommunizieren.

In einem Repository werden datenbankspezifische Anfragen formuliert, die die benötigten Daten liefern. Diese Daten werden in Objekte übertragen, mit denen die anderen Schichten arbeiten. Dieses sogenannte Object-Mapping kann von einem ORM (Object-Relational Mapper) übernommen werden oder manuell implementiert werden.

Da noch nicht definiert ist mit welchen Frameworks und Datenbanken gearbeitet wird, können die Repositories noch nicht genauer spezifiziert werden. Allerdings lassen sich folgende Repositories aus den Services ableiten:

UserRepository	→	AuthService, UserService
RecordRepository	→	RecordService
MeetingRepository	→	MeetingService

4.2.3 Datenbank- und Persistenzkomponenten

Auch die Persistenz ist eine wichtige Komponente einer Anwendung. Hier werden Daten, die über längere Zeit erhalten bleiben sollen abgelegt. Dabei ist es wichtig, dass die Daten nicht nur abgelegt, sondern auch schnell abrufbar sind.

In dieser Anwendung wird eine Datenbank verwendet, um die Daten der Nutzenden zu verwalten und in Beziehung zu einander zu setzen. Jedoch soll die Datenbank entlastet werden, indem die Ausbildungsnachweise, die als PDFs hochgeladen werden, nicht in der Datenbank selber, sondern in einem anderen Verzeichnis abgespeichert werden.

Datenbank

Viele der Projekte bei der PSI laufen über eine Microsoft SQL (MSSQL) Datenbank. Da nicht viele Anforderungen an die Datenbank bestehen wird für die Anwendung eine MSSQL Datenbank verwendet.

Durch die Anforderungsanalyse lassen sich die benötigten Daten, die in der Datenbank abgespeichert werden, erarbeiten. Dazu wurde gedeutet, welche Entitäten sich benötigt werden, und anschließend welche Informationen sie enthalten. Außerdem wurde betrachtet in welcher Beziehung die Entitäten zueinander stehen und ob zu diesen Beziehungen noch weitere Eigenschaften hinterlegt werden müssen. Eine mögliche Variante des Datenbankmodells sieht folgendermaßen aus:

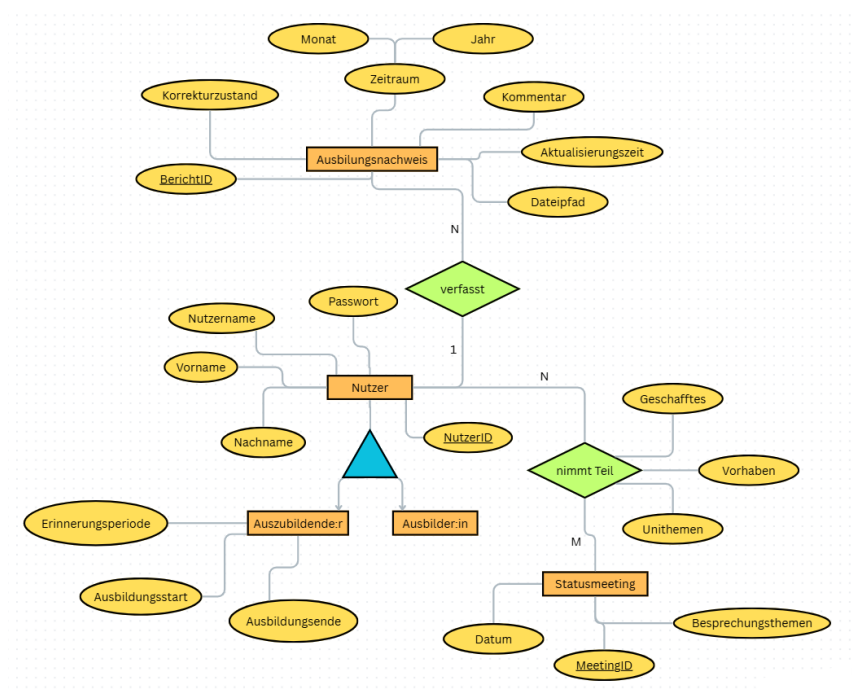


Abbildung 4.5: Entity Relation Diagramm

Zu jedem Nutzenden werden der Vor- und Nachname, der Nutzernamen und ein Passwort gespeichert. Der Nutzernamen und Passwort werden zur Anmeldung verwendet, wobei das Passwort verschlüsselt hinterlegt wird. Der Nutzernamen setzt sich aus dem ersten Buchstaben des Vornamens und dem vollständigen Nachnamen zusammen (Bsp: PPflug für die Nutzerin Pia Pflug). Der Vor- und Nachname wird verwendet, um den angemeldeten Nutzenden persönlicher und direkter anzusprechen. Die Nutzenden können in Auszubildende und Auszubildende unterschieden werden und müssen darauf basierend eventuell noch weitere Informationen speichern. Auszubildenden werden zusätzlich noch das Datum des Ausbildungsstart und des Ausbildungsende zugeteilt. Außerdem halten sie Informationen zu der eingestellten Erinnerungsperiode. Diese stellt dar mit welchem Abstand, in Wochen gemessen, die Auszubildenden eine Erinnerung über vergessene Ausbildungsnachweise erhalten.

Mit den Nutzenden sind die Ausbildungsnachweise verbunden. Jeder Bericht wird genau einem Nutzenden zugeordnet wobei Nutzende beliebig viele Berichte verfassen können. Diese Relation führt zu einer 1:N-Beziehung zwischen den Nutzenden und den Berichten. Eine solche Beziehung kann abgebildet werden, indem jedem Bericht zusätzlich die verfassende Person zugeordnet wird. Die Berichte erhalten eine eindeutige Identifikationsnummer, einen Bearbeitungszeitraum, der speichert zu welchem Monat in welchem Jahr der Ausbildungsnachweis zugeordnet werden kann, und einen Dateipfad an dem der Bericht auf dem Server befindet. Dazu wird die Speicherstrategie später genauer erläutert (siehe [4.2.3 Speicherstrategie](#)). Zusätzlich werden der Korrekturzustand, ein Kommentar von den Auszubildenden und der letzte Bearbeitungszeitpunkt gespeichert. Von dem Bericht wird somit nur der aktuelle Zustand abgebildet. Auf eine Versionierung wird damit vorerst verzichtet, da diese durch die Benachrichtigungen, die verschickt werden sobald sich der Status des Ausbildungsnachweises ändert, erfolgt.

Außerdem gibt es noch die Statusmeetings, die auch eine Identifikationsnummer erhalten und ein Datum zugeordnet bekommen, an dem das Meeting gehalten wurde. Zusätzlich werden zu den Meetings noch die allgemeinen Themen gespeichert, die esprochen werden. An einem Statusmeeting nehmen mehrere Nutzende teil, die alle folgende Daten speichern müssen: Geschaffte Aufgaben der letzten Woche, Aufgaben für die nächste Woche und relevante Themen in dem Studium. Da an einem Statusmeeting mehrere Nutzende teilnehmen und die Nutzenden an mehreren Meetings teilnehmen liegt eine N:M-Beziehung vor. Diese Beziehung führt zu einer eigenen Datenbanktabelle. Diese Entscheidung deckt sich mit der Anforderung, dass die Beziehung eigene Eigenschaften hat. Dort werden die Aufgaben und Uni-Themen der Auszubildenden abgespeichert.

Überführt man dieses Modell in eine MSSQL Datenbank kann diese folgendermaßen aussehen. Die Tabellen sind zur Veranschaulichung mit Beispieldaten gefüllt.

Nutzer	NutzerID	Benutzername	PasswortHash	Vorname	Nachname	Rolle	Ausbildungsstart	Erinnerungsperiode	Ausbildungsende		
	1	3	abauer	Hash123!	Aaron	Bauer	Azubi	2024-08-01	14	2027-08-01	
	2	4	aschmidt	Hash456!	Anna	Schmidt	Azubi	2023-09-01	7	2026-08-01	
	3	5	oprice	Hash789!	Oscar	Price	Ausbilder	NULL	NULL	NULL	
Ausbildungsnachweise	BerichtID	NutzerID	Monat	Jahr	Kommentar	Aktualisierungszeit	Dateipfad	Korrekturzustand			
	1	6	3	11	2025	2025-12-13 17:17:05.7740956	C:\Ausbildungsnachweise\abauer\2025_Nov.pdf	Offen			
	2	7	3	10	2025	2025-12-13 17:17:05.7740956	C:\Ausbildungsnachweise\abauer\2025_Okt.pdf	Angenommen			
	3	8	4	11	2025	Oktober erneut abgegeben.	2025-12-13 17:17:05.7740956	C:\Ausbildungsnachweise\aschmidt\2025_Nov.pdf	Abgelehnt		
Statusmeetings	RecapID	Datum	Besprechungsthemen								
	1	1	2025-12-05	Weihnachtsfeier!!							
	2	2	2025-11-28	Seminararbeit, Stundenbogen							
Meetingteilnahmen	NutzerID	RecapID	Vorhaben	Unitthemen		Geschafftes					
	1	3	1	Kundenprojekt: Ticket 123	Datenbanken, Softwaretechnik, BWL		Kundenprojekt: Einrichten				
	2	3	2	Kundenprojekt: Einrichten	Datenbanken, Softwaretechnik, BWL		Urlaub				
	3	4	1	MiniPMS: TT10	LA, Analysis, Algorithmen		MiniPMS: TT09				
	4	4	2	MiniPMS: TT09	LA, Analysis, Algorithmen		MiniPMS: TT08				

Abbildung 4.6: Datenbank gefüllt mit Beispieldaten

Speicherstrategie

Zusätzlich zu der Datenbank sollen auf dem Server auch Dateien abgelegt werden. Die in der Anwendung hochgeladenen Ausbildungsnachweise sollen auf dem Server hinterlegt werden. Dafür sollen diese nicht in der Datenbank, sondern als PDF-Datei gespeichert werden. Damit die Dateien dennoch zu den in der Datenbank gespeicherten Ausbildungsnachweisen zugeordnet werden können, wird in der Datenbank der Pfad zu der Datei gespeichert. Wichtig ist, dass die Dateien nicht unsortiert auf dem Server verweilen, sondern auch ohne die Anwendung einer verständlichen Struktur unterliegen.

Für alle Auszubildenden wird ein Ordner angelegt indem ihre Ausbildungsnachweise abgelegt werden. In diesen Ordnern werden die Nachweise dieses Auszubildenden abgelegt. Dabei haben diese ein bestimmtes Format: {Jahr}_{Monat}.pdf. Sollte ein Auszubildender den Ausbildungsnachweis bearbeiten, also eine Datei, die für einen bereits abgedeckten Zeitraum bestimmt ist, wird der im Server hinterlegte Nachweis in einen Archivierungsordner, welcher in dem Verzeichnis der Auszubildenden liegt, geschoben. Der neue Ausbildungsnachweis wird anschließend regulär gespeichert werden unter dem wieder freien Dateinamen.

4.3 Zusammenfassung der Konzeption

Das erstellte Konzept gibt eine Vorlage dafür, wie die Anwendung umgesetzt werden kann. Dabei wird zum einen auf die Frontendkomponenten eingegangen, in welche Routen diese unterteilt werden können und wie diese aufgebaut werden können. Außerdem wird die Authentifizierung der Nutzenden behandelt. Bei den Backendkomponenten wird die Aufteilung in die Controller-, die Service- und die Repository-Schicht erläutert, wobei auf die Aufgaben der Schichten, die Kommunikation dazwischen und die Strukturen beschrieben. Zuletzt wird noch die zugrundeliegende Persistenzkomponente dargestellt, welche in eine Datenbank und eine Dateiablage unterteilt ist. Für beide Speicherarten wird eine Strategie präsentiert.

Mit diesem Konzept wird jedoch nur die Anwendung beschrieben, die aus der Anforderungsanalyse hervorgeht. Dennoch ist die Anwendung vielseitig erweiterbar, was im Folgenden ausgeführt wird.

5 Ausblick

In diesem Kapitel werden Erweiterungsmöglichkeiten für die Anwendung und andere Einsatzmöglichkeiten dargestellt. Dabei wird jedoch nur kurz angeschnitten, was möglich ist, und nicht detailliert ausgearbeitet, wie solche Erweiterungen umgesetzt werden.

Automatisches Ausfüllen

Eine andere Erweiterungsmöglichkeit besteht in den Statusmeetings. Da die Auszubildenden in der Ausbildung viel in die Projekte eingebunden und oft Tickets aus den Jira-Projekten erhalten, kann es möglich gemacht werden, die abgeschlossenen Tickets beim erstellen eines Statusmeetings zu erwähnen. So können die Auszubildenden unterstützt werden und die formale Vollständigkeit wird gesteigert. Zweiteres wird begründet durch das menschliche Dasein der Auszubildenden und der damit verbundenen Vergesslichkeit. Zur Umsetzung könnte eine Automatisierung, die Jira von Haus aus anbietet, helfen.

Auch in der Ausfüllung der Ausbildungsnachweise können die Auszubildenden unterstützt werden. Da die Statusmeetings eine wöchentliche Zusammenfassung der abgeschlossenen Aufgaben und die Ausbildungsnachweise der gleichen Funktion in monatlichen Abständen dient, bietet es sich an die Ausbildungsnachweise mit den Inhalten der Statusmeetings zu füllen. Es sollte sich lediglich um eine Auflistung der Aufgaben handeln, die von den Auszubildenden selber auszuformulieren ist. Dabei ist zu beachten, dass eine Auswahl der zeitlich richtigen Statusmeetings und von diesen auch nur der persönliche Bereich des Auszubildenden betrachtet wird.

Künstliche Intelligenz

Als Fortsetzung der Auflistung der Aufgaben in den Ausbildungsnachweisen ist die Vervollständigung durch eine künstliche Intelligenz (KI) denkbar. Eine KI kann aus Stichpunkten einen Text formulieren, der die Aufgaben beschreibt. Sollte sich für eine solche Erweiterung entschieden werden, ist es Wert die Abläufe in den Statusmeetings zu überdenken, da es die Aufgabe der Auszubildenden ist ihre Ausbildungsnachweise auszufüllen. Mit den in der Arbeit beschriebenen Abläufen würden die Auszubildenden diesen Aufwand abnehmen.

Um die Auszubildenden zu entlasten kann überlegt werden, die Ausbildungsnachweise der Auszubildenden von einer KI im Vorhinein kontrollieren zu lassen. Dabei sollten die Auszubildenden trotzdem überprüfen müssen, aber ihnen kann eine Untersuchung der förmlichen Aspekte abgenommen werden.

Bei jeglichem Einsatz einer künstlichen Intelligenz ist die Datensicherheit mit höchster Priorität zu behandeln. Diese Diskretion wird sowohl von den Nutzenden selber, als auch dem Unternehmen erwartet.

Erweiterung des Nutzungsfeldes

Ein Wunsch, der im Laufe der Konzeptionierung dieser Anwendung bereits aufgekommen ist, verlangt die Funktionalität der Statusmeetings zu extrahieren und als alleinstehende Anwendung anzubieten. So können auch die firmeninternen Projektteams von der Anwendung profitieren und ihre Meetings dokumentieren.

6 Fazit

Ziel dieser Arbeit war es, die Konzeptionierung einer Anwendung zu erarbeiten, die zur Organisation der Ausbildungsnachweise und zur Dokumentation der Statusmeetings dient. Dabei sollte auf die Anforderungen, die in der Ausbildung bei der PSI existieren, eingegangen werden.

Eine zuvor angestellte Analyse hat ergeben, dass es keine Anwendung gibt, die dem entspricht. Sowohl für die einzelnen Funktionalitäten, als auch für die Kombination der beiden, werden von keiner betrachteten Anwendung die bestehenden Bedürfnisse vollständig abgedeckt. Dabei variieren die Gründe zwischen Kosten, Datenschutz oder den konkreten Abläufen, die verlangt werden. Basierend auf den alternativen Anwendungen und den vorherigen Prozessen wurden die Anforderungen an die Anwendung in der Form von User Stories festgehalten. Dabei werden sowohl funktionale Aspekte, wie die Abfolge der Prozesse, als auch die nicht-funktionalen Aspekte, wie die Nutzerfreundlichkeit betrachtet.

Durch diese Anforderungen konnte ein Konzept zur Entwicklung einer Anwendung erarbeitet werden. Es wurde eine passende Architektur etabliert und die einzelnen Komponenten der Strukturen in ihre Funktionalitäten unterteilt. In den Komponenten konnte weiter spezialisiert werden und konkrete Vorschläge zu einer möglichen Umsetzung formuliert.

Mit Hilfe dieser Konzeptionierung wird die Implementierung vorbereitet, um die Ausbildungsprozesse zu unterstützen.

Abbildungsverzeichnis

3.1	Visualisierung des Abgabeprozesses von Ausbildungsnachweisen	8
3.2	OneNote-Seite eines Statusmeetings	10
4.1	Navigationsleiste der Ausbildenden	17
4.2	Startansicht der Ausbildenden	17
4.3	Ansicht der Ausbildungsnachweise für die Ausbildenden	18
4.4	Erstellung eines Statusmeetings	19
4.5	Entity Relation Diagramm	23
4.6	Datenbank gefüllt mit Beispieldaten	25

Tabellenverzeichnis

3.1	Erarbeitete User-Stories für die Anwendung zur Abgabe der Ausbildungsnachweise	10
3.2	Erarbeitete User-Stories für die Anwendung zur Dokumentation der Statusmeetings	12
3.3	Sonstige erarbeitete User-Stories für die Anwendung	13
4.1	Beispielhafte REST-API-Endpunkte des Backends	21
B.1	Beispielhafte REST-API-Endpunkte des Backends	34

Literatur

- [1] IHK Aachen. *Ausbildungsnachweis (Berichtsheft)*. de. Publication Title: Ausbildungsnachweis (Berichtsheft). URL: <https://www.ihk.de/aachen/bildung/ausbildung/berichtsheft-6474264> (besucht am 22. 10. 2025).
- [2] Azubiheft. *Azubiheft – Das digitale Online-Berichtsheft*. URL: <https://www.azubiheft.de/> (besucht am 22. 10. 2025).
- [3] Zubido. *Zubido Azubi-App - Das bessere Digitale-Berichtsheft*. de-DE. URL: <https://zubido.de/> (besucht am 22. 10. 2025).
- [4] generation:l. *azubi:web*. de. URL: <https://generationl.de/azubiweb/> (besucht am 22. 10. 2025).
- [5] BLok. *Digitales Online Berichtsheft*. de. Publication Title: BPS GmbH. URL: <https://www.bps-system.de/blok-berichtsheft/> (besucht am 22. 10. 2025).
- [6] u-form. *Azubi-Navigator*. de. Publication Title: u-form Testsysteme. URL: <https://www.testsysteme.de/ausbildungsmanagement> (besucht am 22. 10. 2025).
- [7] Friday.app. *Work Productivity Tips, Tools, Templates & More*. URL: https://friday.app/?utm_source=chatgpt.com (besucht am 22. 11. 2025).
- [8] Parabol. *Free Agile Meeting Tool for Remote Teams*. en-US. URL: <https://www.parabol.co/> (besucht am 22. 11. 2025).
- [9] WEEKDONE. *Team Meetings with Purpose*. en-US. URL: <https://weekdone.com/teammeeting> (besucht am 22. 11. 2025).
- [10] Microsoft. *Architectural principles*. de. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/architectural-principles> (besucht am 03. 12. 2025).
- [11] Microsoft. *Common web application architectures*. en-us. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures> (besucht am 03. 12. 2025).
- [12] Roy T. Fielding, Mark Nottingham und Julian Reschke. *HTTP Semantics*. Request for Comments RFC 9110. Num Pages: 194. Internet Engineering Task Force, Juni 2022. DOI: [10.17487/RFC9110](https://datatracker.ietf.org/doc/rfc9110). URL: <https://datatracker.ietf.org/doc/rfc9110> (besucht am 06. 12. 2025).
- [13] JSON.org. *Introducing JSON*. URL: <https://www.json.org/json-en.html> (besucht am 06. 12. 2025).
- [14] Ferdinand Malcher. *Angular: Das große Praxisbuch – Grundlagen, fortgeschrittene Themen und Best Practices*. ger. überarbeitete und aktualisierte Auflage. iX Edition. Heidelberg: dpunkt.verlag, 2022. ISBN: 978-3-96910-862-8.

A Ausschnitt Ankündigung der Plattform Anubis

Von: matse@itc.rwth-aachen.de

Datum: Dienstag, den 9. Dezember 2025

Betreff: MATSE-Info 12/2025

[...]

1. Ausbildungsnachweise Voraussichtlich im ersten Quartal 2026 wird die Plattform Anubis an den Start gehen, mit der Sie die Ausbildungsnachweise verwalten können. Die Funktionalität orientiert sich an der bis Juli genutzten Plattform der IHK Aachen. Bis dahin bitten wir Sie, die Ausbildungsnachweise geeignet zwischenzuspeichern. Beim Verfassen Ihrer Ausbildungsnachweise möchten wir Sie darauf hinweisen, dass die Beschreibung des Theorieteils in den Ausbildungsnachweisen aufgrund der Einheitlichkeit auf das Nötigste (z.B. 3-4 Schlagworte pro Modul) beschränkt werden kann. Die Beschreibung des Praxisteils sollte hingegen aufgrund der Individualität ausführlicher beschrieben werden, vorzugsweise in ganzen Sätzen und unter Nennung des Projektnamens, der Projektziele, der selbst geleisteten Tätigkeiten im Zeitraum und der eingesetzten Technologien.

[...]

B Request-Struktur der REST-API

Schnittstelle	Methode	Beschreibung	User Story
/records	POST	Legt einen neuen Ausbildungsnachweis an.	US1
/records	GET	Liefert alle Ausbildungsnachweise.	US8, US11
/records/{userID}	GET	Liefert alle Ausbildungsnachweise, die einem Nutzenden zugeordnet sind.	US12
/records/{state}	GET	Liefert alle Ausbildungsnachweise, die sich im übergebenen Korrekturzustand befinden.	US10
/records/{recordID}	GET	Liefert den angegebenen Ausbildungsnachweis.	US8, US9
/records/{recordID}	PUT	Speichert die Änderungen für den angegebenen Ausbildungsnachweis.	US2 - US4
/records/{recordID}	DELETE	Löscht den angegebenen Ausbildungsnachweis.	US2
/meetings	POST	Fügt ein neues Statusmeeting hinzu.	US13 - US15
/meetings/{meetingID}/ {userID}	POST	Fügt einen Eintrag für den angegebenen Nutzenden in dem angegebenen Statusmeeting hinzu.	US16 - US18
/meetings	GET	Liefert alle Statusmeetings.	US22
/meetings/{userID}	GET	Liefert alle Statusmeetings, an denen der Nutzende teilgenommen hat.	US23
/meetings/{meetingID}	GET	Liefert das angegebene Statusmeeting.	US22
/meetings/{meetingID}/ {userID}	GET	Liefert von dem angegebenen Statusmeeting die Daten zum übergebenen Nutzenden.	US23
/meetings/{meetingID}	PUT	Speichert die Änderungen für das angegebene Statusmeeting.	US13 - US15
/meetings/{meetingID}	DELETE	Löscht das angegebene Statusmeeting.	US22
/users	POST	Legt einen neuen Nutzenden an.	US27
/users	GET	Liefert alle Nutzenden.	US27
/users/{userID}	GET	Gibt den angegebenen Nutzenden zurück.	US27
/users/{role}	GET	Liefert alle Nutzenden mit der übergebenen Rolle zurück.	US27
/users/{userID}	PUT	Speichert die Änderungen für den angegebenen Nutzenden.	US27
/users/{userID}	DELETE	Löscht den angegebenen Nutzenden.	US27

Tabelle B.1: Beispielhafte REST-API-Endpunkte des Backends