


Konzipierung und Implementierung
einer Lernplattform im Schulumfeld
mit Strukturen ähnlich sozialen Netzwerken
und Fokus auf Datenschutz
und Privatsphäre

Besondere Lernleistung
Otto-Hahn-Gymnasium Bensberg
Q2 Informatik Abitur 2020
Betreuende Lehrkraft: 



31. März 2020

Inhaltsverzeichnis

I	Zielbestimmung	5
1	Thema der besonderen Lernleistung	5
1.1	Definition	5
1.2	Motivation	5
1.2.1	Problematik von Nachhilfe	5
1.2.2	Mangel an digitalem Unterricht	6
1.2.3	Konflikte mit Datenschutz	6
2	Kernkonzepte	7
2.1	Privatsphäre	7
2.2	Datenschutz	7
2.3	Sicherheit	8
2.4	Intuitive Bedienung	8
2.5	Transparenz	8
3	Kriterien	9
3.1	Musskriterien	9
3.1.1	Lernhilfe	9
3.1.2	Kontakte	9
3.1.3	Chat	10
3.1.4	Nutzersystem	10
3.1.5	Moderation der Nutzerbasis	10
3.2	Sollkriterien	11
3.2.1	Nutzerprofile	11
3.2.2	Lernangebote	11
3.2.3	Admin-Bereich	11
3.2.4	Zuordnung zu Fächern und Themen	11
3.2.5	Hochladen von Dateien	11
3.3	Kannkriterien	12
3.3.1	Benachrichtigungen	12
3.3.2	Meldung von Nutzern/Beiträgen	12
3.3.3	Lerngruppen	12
3.3.4	Bewertung der Hilfestellung	12
3.3.5	Personalisierter Feed	13

3.4	Abgrenzungskriterien	13
3.4.1	Soziales Netzwerk	13
3.4.2	Gruppen im Chat	13
3.4.3	Zahlungsabwicklung	14
II	Einsatz	15
4	Zielgruppe	15
5	Technische Voraussetzungen	15
5.1	Server	15
5.2	Datenbank	16
5.3	Nutzer	16
III	Software-Architektur	17
6	Struktur	17
6.1	REST	17
6.2	Vorteile	18
7	Client	19
7.1	Eingesetzte Technologien	19
7.1.1	Angular	19
7.1.2	Struktur	20
8	Server	22
8.1	Eingesetzte Technologien	22
8.1.1	Node.js	22
8.1.1.1	Node Package Manager	23
8.1.2	LoopBack 4	23
8.1.2.1	Struktur	23
8.1.3	Weitere Libraries	25
8.1.3.1	Multer	25
8.1.3.2	Nodemailer	25
8.1.3.3	Sharp	25
8.1.3.4	JsonWebToken	25
8.1.3.5	CSV-Parser	25
8.1.3.6	Request	25
8.2	Datenbank	26
8.3	Implementierungen	26

8.3.1	Nutzer- und Postsystem	26
8.3.2	Personalisierter Feed	27
9	Chat	29
9.1	WebSocket	29
9.2	Verschlüsselung	30
9.3	Funktionsweise	32
IV	Funktionen	33
10	User Stories	33
10.1	Anfrage erstellen	33
10.2	Anfrage beantworten	34
10.3	Chat	34
10.4	Hilfe bewerten	34
10.5	Login	34
10.6	Registrierung	35
10.7	Profil-Einstellungen	35
10.8	Account-Einstellungen	35
10.9	Melden	36
10.10	Suchen	36
V	Bedienungsanleitung	37
11	Installation & Einrichtung	37
11.1	Konfiguration	37
11.2	Client	38
11.2.1	Ordnerstruktur	38
11.3	Server	39
11.3.1	Ordnerstruktur	39
12	Verwaltung	39
12.1	Nutzer hinzufügen	39
12.2	Nutzer verwalten	40
12.3	Fächer & Themen	40
12.4	Meldungen	41
12.5	Feedback	41
12.6	Klassen	41

VI	Schluss	42
13	Fazit	42
13.1	Ausblick	42
14	Libraries und Frameworks	44
15	Literatur- und Linkverzeichnis	45
16	Eigenständigkeitserklärung	47

Teil I

Zielbestimmung

1 Thema der besonderen Lernleistung

1.1 Definition

Diese besondere Lernleistung umfasst die Konzipierung und Implementierung einer Lernplattform. Diese soll Schülern einer Bildungsinstitution das Lernen vereinfachen und insbesondere Hilfe bei Schwierigkeiten anbieten.

Dabei basiert das Konzept auf Gegenseitigkeit: Anstatt einer zentralen Einrichtung, welche Lerninhalte zusammenstellt und anbietet, helfen sich die angemeldeten Schüler gegenseitig. In diesem Sinne ließe sich die Lernplattform auch als Nachhilfeplattform sehen.

Jedoch klingt bei dem Begriff *Nachhilfe* sehr stark das Außerunterrichtliche mit; die Lernplattform soll allerdings auch in den regulären Unterricht integriert werden können, indem digitale Lernmaterialien einfach zur Verfügung gestellt werden können.

Insbesondere beschränkt sich die angebotene Hilfe auf der Lernplattform nicht nur auf den aktuellen Unterricht eines Schülers, sondern kann auch aus seinen besonderen Interessen heraus entstehen: Beispielsweise ist es denkbar, dass ein mathematisch interessierter Schüler von einem älteren Schüler komplexere Konzepte der Mathematik erklärt bekommen möchte.

1.2 Motivation

Die Idee dieser Lernleistung ist aus drei verschiedenen, eigens erlebten Gegebenheiten entstanden.

1.2.1 Problematik von Nachhilfe

Grundsätzlich ist das Konzept von Nachhilfe nachvollziehbar: Schüler, die ein bestimmtes Thema nicht verstanden haben, sollten keinesfalls zurückgelassen werden, während der Unterrichtsstoff weiterläuft, sondern individuelle Hilfestellung erhalten. Dadurch, dass häufig andere Schüler die Quelle dieser maßgeschneiderten Hilfe sind, können sie womöglich auf einem anderen Niveau Sachverhalte erklären als die Lehrkraft. Gegebenenfalls hilft sogar diese andere Erklärungs- und Herangehensweise beim Verständnis des Problems.

Allerdings findet diese Form der Nachhilfe meist gegen Bezahlung statt. Zudem existieren professionelle Nachhilfeagenturen, deren Geschäftsmodell darin besteht, Schülern gegen Geld außerschulische Weiterbildung anzubieten. Insgesamt werden in Deutschland jährlich circa 879 Millionen Euro für Nachhilfe ausgegeben; viele Eltern geben sogar über 50 Euro pro Monat für die zusätzlichen Bildungsangebote aus (Klemm & Hollenbach-Biele, 2016). Dies steht der Schulgeldfreiheit (Art. 9 Verfassung für das Land Nordrhein-Westfalen) entgegen, welche grundsätzlich die Möglichkeit der kostenlosen Bildung für jeden Schüler vorsieht.

Aus diesem Anlass entsprang die Idee einer Onlineplattform zum kostenlosen Austausch von Schülern. Solange sich die Kommunikation lediglich im Digitalen aufhält, ermöglicht dies auch eine zeitlich deutlich flexiblere Lösung, wodurch ein Entgelt ebenfalls nicht zwangsweise erforderlich ist. Zudem besteht die Möglichkeit, dass sich ein Schüler von mehreren Personen Hilfe sucht und letztendlich das bestmögliche Gesamtbild erhält.

1.2.2 Mangel an digitalem Unterricht

Schon seit Langem existieren zahlreiche Möglichkeiten für digitale Unterrichtsgestaltung. Als Beispiel lässt sich hier die freie Lernplattform Moodle nennen, welche bereits seit 2001 existiert (Moodle Pty Ltd., 2019) und das Verteilen von Materialien, Aufgaben, Bewertungen und mehr vereinfacht.

Jedoch findet der Einsatz von solcher Software erfahrungsgemäß nur sehr eingeschränkt statt und das auch nur in wenigen Fächern. Denkbar ist, dass die Einrichtung und Bedienung solcher Dienste vielen Lehrern zu aufwendig oder zu schwierig im Verhältnis zum erwarteten Nutzen erscheint.

Diese Ergebnisse erzielte auch eine Studie der Bitkom Research GmbH 2014. Zwar werden vermehrt digitale Geräte für den Unterricht eingesetzt, allerdings stehen dabei die besonderen Möglichkeiten von diesen Medien, wie beispielsweise digitale Lernprogramme, oftmals im Hintergrund. Dabei steht die deutliche Mehrheit von 95 Prozent der befragten Lehrer dem Einsatz von digitalen Medien im Unterricht positiv gegenüber (Bitkom Research GmbH, ARIS Umfrageforschung GmbH, BITKOM e.V., 2015).

Daraus ergibt sich der Bedarf einer sowohl simplen, intuitiven als auch praxistauglichen digitalen Unterrichtsplattform, welche den Austausch von Lehr- und Lernmaterialien mit den Schülern ermöglicht.

1.2.3 Konflikte mit Datenschutz

Seit der Einführung der Datenschutzgrundverordnung (DSGVO) ist der Einsatz vieler Dienste im Rahmen des Unterrichts schwierig. Beispielhaft ist auch der Einsatz von Microsoft Office 365, was im schulischen Rahmen breiten Einsatz

finden könnte, datenschutzrechtlich bedenklich (Der Hessische Beauftragte für Datenschutz und Informationsfreiheit, 2019).

Zumal noch erschwerend hinzu kommt, dass die Schüler gewissermaßen das Recht auf Selbstbestimmung über ihre Daten verlieren, wenn sie im Rahmen des Unterrichts gezwungen werden, einen digitalen Dienst jeglicher Art zu verwenden.

Jedoch wäre es akzeptabel einen solchen Weg zu gehen, wenn die eingesetzte Plattform nicht mehr Daten der Schüler speichert, als ohnehin im schulischen Umfeld der Fall ist. Immerhin werden zum Beispiel die Namen der Schüler in analogen Klassenbüchern *gespeichert*.

Somit benötigt man eine Möglichkeit für digitalen Unterricht, welche so sparsam wie möglich Daten sammelt. Allerdings müsste man jedem Schüler die Option geben, die Datenschutzerklärung nicht zu akzeptieren und dennoch jegliche Unterrichtsmaterialien zu erhalten. An dieser Stelle bleibt jedoch das Problem, dass diese wichtige Entscheidungsfreiheit seitens der Schüler erheblich die Motivation sowie Möglichkeiten der Lehrkraft, digitale Medien einzusetzen, beschränkt. Dieses Problem wird sich so leicht auch nicht lösen lassen.

2 Kernkonzepte

Bei der Entwicklung der Lernplattform sind bestimmte Konzepte besonders eingeflossen. Hierzu gehören Privatsphäre, Sicherheit, intuitive Bedienung sowie Transparenz, welche im Folgenden detaillierter erläutert werden.

2.1 Privatsphäre

Aus den bestehenden Komplikationen mit dem Datenschutz (s. 1.2.3) geht der Bedarf für erhöhten Datenschutz und darüber hinaus für Privatsphäre hervor. Im Rahmen einer Anwendung, welche gezielt auch im Schulunterricht verwendet werden soll und somit gewissermaßen Schülern *aufgezwungen* würde, muss der Schutz der persönlichen Daten sehr hohe Priorität haben.

Daraus ergibt sich die Verantwortung, die Lernplattform in einer Weise zu strukturieren und entwickeln, die es ermöglicht, dass die Privatsphäre jedes Schülers und selbstverständlich auch jeder Lehrkraft jederzeit nach eigenem Belieben des Nutzers geschützt ist.

2.2 Datenschutz

Da die Lernplattform grundlegend auf die Nutzung im schulischen Rahmen ausgelegt ist, folgert sich daraus ein besonderes Maß an Datenschutz. Es ist das Ziel der besonderen Lernleistung, möglichst wenige Daten über den einzelnen Nutzer zu sammeln. Es können keine Daten gestohlen werden, die nicht existieren, was

gleichzeitig auch zum Kriterium Sicherheit (s. 2.3) gehört. So fordert es auch der Grundsatz der Datenminimierung in der DSGVO (Art. 5/1c DSGVO).

Natürlich geht auch einher, dass es unvereinbar ist, Daten an Drittanbieter, vor allem nicht an Nutzungsanalyse-Tools oder Werbedienste, weiterzugeben. Alle Daten müssen auf dem Server der Lernplattform bleiben.

2.3 Sicherheit

Aus dem Schutz der Privatsphäre geht auch ein hohes Maß an Sicherheit hervor, welches erfüllt werden muss. Dazu gehört beispielsweise die Sicherheit der gespeicherten Daten, d.h. Außenstehende sollten sich keinesfalls Zugang zum Speicherort der Lernplattform verschaffen können. Zudem sollten die Datenschnittstellen nicht für Unbefugte zugänglich sein und für die Nutzer sollten lediglich die Daten verfügbar sein, die für sie vorgesehen sind. Weiterhin ist die besondere Sicherung sensibler Daten zu beachten.

Allerdings gibt es noch einen weiteren Sicherheitsaspekt auf der Lernplattform zu beachten: Die Schüler müssen sich nicht nur in puncto Datenschutz sicher fühlen, sondern auch in sozialen Aspekten. Dazu gehört, dass Mobbing, Beleidigungen und Ähnliches auf der Plattform geahndet werden. Außerdem müssen besonders jüngere Schüler vor dem Kontakt mit Personen mit böswilligen Absichten geschützt werden.

2.4 Intuitive Bedienung

Aufgrund der breiten und vielfältigen Nutzerbasis sowohl bezüglich Alter, Erfahrung als auch digitaler Affinität ergibt sich die Notwendigkeit einer intuitiven Bedienung, damit die Lernplattform praktikabel eingesetzt werden kann. Dazu gehören beispielshalber Erklärungen einzelner Funktionen, ein übersichtliches Layout und eine zugängliche Bedienung.

Sowohl frisch eingeschulte Fünftklässler als auch Lehrer kurz vor dem Ruhestand sowie jegliche Abstufungen dazwischen sollen in der Lage sein, die ihnen gebotenen Funktionen vollständig und möglichst ohne langwierige Anweisung zu verwenden.

2.5 Transparenz

Um Vertrauen zu schaffen benötigt man vor allem Transparenz. Es muss dem Nutzer klar ersichtlich sein, was mit seinen Daten passiert und wie sie verarbeitet bzw. gespeichert werden. Nur so ist es möglich, dass sowohl Schüler ungestört die Plattform nutzen können als auch Eltern keine Bedenken über die Sicherheit eben dieser hegen.

Dieser Punkt ist ebenfalls sehr essenziell, da die Funktionsfähigkeit der Plattform darauf basiert, dass genügend aktive Nutzer vorhanden sind.

3 Kriterien

Zur Strukturierung eines Software-Projekts ist eine detaillierte Planung essenziell. Insbesondere ist eine genaue Beschreibung der gesetzten Ziele notwendig, um festzustellen, ob alle benötigten sowie gewünschten Funktionen implementiert wurden. Diese lassen sich häufig in drei Prioritätsniveaus unterteilen:

1. die Musskriterien, welche essenziell für die Nutzbarkeit der Anwendung sind;
2. die Sollkriterien, die sich auf weiterführende, wichtige Funktionen beziehen,
3. und die Kannkriterien, die im Zweifelsfall weggelassen werden können, da sie lediglich einzelne Aspekte der Anwendung verbessern würden, ohne dass ihnen eine zentrale Rolle zukommt.

3.1 Musskriterien

3.1.1 Lernhilfe

Die Nutzer der Plattform sind dazu in der Lage, Anfragen für Hilfe zu erstellen und Anderen zugänglich zu machen. Diese Anfragen können sich auf ein Fach beziehen, optional auch auf ein bestimmtes Thema in diesem Fach (s. 3.2.4), und enthalten eine genauere Beschreibung des vorliegenden Problems sowie der gewünschten Hilfe.

Andere Nutzer sind in der Lage, ihren Fähigkeiten und Schwächen entsprechend, diese Anfragen anderer Nutzer einzusehen und zu filtern. Sie können nach eigenem Ermessen eine Antwort auf die Beiträge verfassen, um ihre Unterstützung anzubieten. Der ursprüngliche Autor des Beitrags kann sie annehmen oder ablehnen und somit den Kontakt starten (s. 3.1.2).

Hierbei ist wichtig, dass die Anfragen und Angebote anonym gestellt werden, sodass alle Beiträge ohne Nutzernamen angezeigt werden und somit nicht einem bestimmten Schüler zugeordnet werden können. Dies soll verhindern, dass sich irgendjemand davor fürchten muss, nach Hilfe zu fragen oder gar gezieltem Mobbing ausgesetzt zu sein.

3.1.2 Kontakte

Sobald erfolgreich eine Antwort auf einen Beitrag verfasst und akzeptiert wurde, wird beiden Nutzern ein neuer Kontakt hinzugefügt. Dieser Vorgang ermöglicht die Kommunikation über den Chat (s. 3.1.3) und soll ausschließlich im Rahmen des ursprünglichen Beitrags stattfinden.

3.1.3 Chat

Jeder Nutzer besitzt einen Chat mit jedem seiner Kontakte. Der Chat stellt das zentrale Kommunikationsmittel für das Lernen und Helfen dar und wird ausschließlich durch das Hinzufügen von Kontakten (s. 3.1.2) im Anschluss an Antworten zu Angeboten (s. 3.2.2) und Anfragen freigeschaltet.

Beide Chatteilnehmer können normale Text-Nachrichten senden. Sie können ihre eigenen Nachrichten im Nachhinein löschen.

Nach Möglichkeit sind alle gesendeten Nachrichten verschlüsselt, aber jederzeit und dauerhaft für beide beteiligten Nutzer zugänglich. Jedoch darf die Kommunikation keinesfalls für Dritte zugänglich sein, inklusive der Lehrer und Verwalter bzw. Administratoren, so wie es das Briefgeheimnis (Art. 10 GG) für die analoge Kommunikation vorsieht.

3.1.4 Nutzersystem

Zur individuellen Hilfestellung verfügt die Plattform über ein Nutzersystem, wie es weit verbreitet ist. Das bedeutet, dass sich jeder Nutzer ein eigenes Konto mithilfe seiner E-Mail-Adresse erstellen kann. Mithilfe des selbst festgelegten Passworts ist der Nutzer jederzeit in der Lage, auf seine Beiträge und Chats zuzugreifen. Über die Einstellungen hat der Nutzer die Möglichkeit, auch nach der Registrierung sein Konto zu modifizieren, beispielsweise durch das Ändern seines Passworts.

Jeder Nutzer verfügt zudem über einen einzigartigen, selbst ausgewählten Nutzernamen. Über diesen ist bei der Anmeldung eine eindeutige Zuordnung möglich.

3.1.5 Moderation der Nutzerbasis

Da es sich bei der Lernplattform um eine sichere Umgebung handelt, muss gewährleistet sein, dass die Nutzerbasis aus vertrauenswürdigen Personen besteht. Dementsprechend ist sicherzustellen, dass jeder beigetretene Nutzer auch über die Berechtigung dazu verfügt.

Einerseits können Schüler, Lehrer und andere Personen, die direkt von der Schule eingeladen werden, über einen einzigartigen Zugangscode die Berechtigung zum Erstellen eines Kontos erhalten. Andererseits kann die Registrierung von all jenen, die nicht über einen solchen Code verfügen, durch Administratoren und Lehrer moderiert werden: das bedeutet, die Registrierungsanfrage kann akzeptiert oder abgewiesen werden.

Zusätzlich ist auch eine kontinuierliche Moderationsoption für bereits registrierte Nutzer wichtig. Nutzer, die sich nicht gemäß allgemeiner Umgangsregeln verhalten, können durch Administratoren und Lehrer gesperrt bzw. entfernt werden.

3.2 Sollkriterien

3.2.1 Nutzerprofile

Zum genaueren Kennenlernen eines Nutzers, der einem Hilfe angeboten hat, ist eine Profilübersicht sinnvoll. Diese enthält alle Eckdaten zu der Person, wie den Namen, die ausgewählten Fähigkeiten, die Klassenstufe etc., und im besten Fall auch eine Zusammenfassung seiner vergangenen Hilfe für andere Nutzer.

3.2.2 Lernangebote

Als Ergänzung der regulären Hilfsanfragen sind Lernangebote vorgesehen, welche umgekehrt funktionieren: Anstelle eines Hilfesuchenden, der eine gezielte Anfrage verfasst, können all diejenigen, die sehr gut in einem Thema oder Fachbereich sind, eigeninitiativ ihre Hilfe anbieten.

Andere Nutzer können dann ebenso wie bei Lernanfragen darauf reagieren und somit Kontakt aufnehmen. Der anschließende Ablauf des Chattens ist derselbe.

3.2.3 Admin-Bereich

Ein grundlegendes Prinzip der Lernplattform ist die einfache Bedienung (s. 2.4) für Jeden. Somit ist es wichtig, dass auch Lehrern und der Verwaltung intuitive oder zumindest leicht verständliche Mittel zur Verfügung stehen, die ihnen die Verwaltung der Plattform, Nutzer und Beiträge ermöglicht. Es kann nicht von jeder Lehrkraft erwartet werden, sich intensiv mit der Datenbank und der zusammenhängenden Datenstruktur zu beschäftigen, um einzelne Beiträge zu löschen, einen Nutzer zu sperren oder ähnliche Aktionen, die für die Funktionsfähigkeit der Plattform unabdingbar sind, durchzuführen.

3.2.4 Zuordnung zu Fächern und Themen

Zur vereinfachten Auffindbarkeit bietet sich eine Zuordnung der Lernanfragen zu Fächern und Themen an, welche generell von der Schule festgelegt werden könnten. Dies ist traditioneller Nachhilfe nachempfunden, die sich meist auf ein Fach beschränkt.

Wichtig für den Nutzen dieser Funktion ist, dass alle Nutzer ihre Fähigkeiten und Schwächen angeben, sodass personalisierte Feeds (s. 3.3.5) generiert werden können. Jedoch ermöglicht die klare Einteilung der Beiträge auch ohne diese Personalisierung ein schnelleres Überfliegen und Einschätzen der Lernanfragen, wodurch Nutzer eher Beiträge finden, bei denen sie ihre Hilfe anbieten können.

3.2.5 Hochladen von Dateien

Beim Lernen sind vielfältige Wege und Sinnesaufnahmen höchst bedeutsam und Sachverhalte können häufig durch Grafiken, Videos und Co. deutlich besser ver-

anschaulicht werden als durch puren Text. Daher sollte es im Rahmen der Lernplattform möglich sein, die eigenen Beiträge sowie Chat-Nachrichten durch das Hochladen und Anhängen eigener Dateien in dieser Weise zu erweitern.

3.3 Kannkriterien

3.3.1 Benachrichtigungen

Individualisierte Benachrichtigungen für die Nutzer dienen dem Zweck, Neuigkeiten und Änderungen in Themenbereichen, die für sie persönlich von Interesse sind, zu übermitteln. Bezogen auf die Lernplattform beziehen sich diese Änderungen insbesondere auf die eigenen Beiträge der Schüler.

Es ist beispielsweise eindeutig von hoher Priorität für einen Hilfesuchenden, wenn sie eine Antwort auf ihr Problem erhalten. Dementsprechend sollten Nutzer über diese Ereignisse informiert werden.

3.3.2 Meldung von Nutzern/Beiträgen

Ein Teil des Kriteriums Sicherheit (s. 2.3) besteht im Schutz vor Mobbing, Beleidigungen und anderweitig schädlichen Inhalten bzw. Nutzern. Dementsprechend muss die Möglichkeit bestehen, derartige Inhalte zu entfernen.

Da die Entwicklung eines automatisierten Systems zur Erkennung solcher Fälle sehr komplex und zu umfangreich für diese besondere Lernleistung wäre, beruht die Moderation der Plattform auf den Nutzern: Einerseits können alle Nutzer Beiträge melden, andererseits können Administratoren oder Lehrer die Meldungen einsehen und gegebenenfalls eingreifen.

3.3.3 Lerngruppen

Um die intensive Einbindung der Lernplattform in den Unterricht zu unterstützen, sind Lerngruppen eine Möglichkeit, die Kurse und Klassen aus der realen Welt in den digitalen Raum zu bringen. Lehrer könnten Gruppen für ihre Kurse erstellen, die entsprechenden Schüler hinzufügen und im Anschluss zum Beispiel Aufgaben, Erklärungen und Lernmaterialien verteilen, welche ausschließlich für die Schüler der Lerngruppe zugänglich sind.

3.3.4 Bewertung der Hilfestellung

Als Mittel der Qualitätskontrolle und -sicherung können Nutzer die Kommunikation und empfangene Hilfestellung von anderen Nutzern bewerten. Diese Bewertungen werden aufbereitet auf den jeweiligen Nutzerprofilen angezeigt und ermöglichen anderen Nutzern eine vorzeitige Einschätzung des potentiellen Gesprächs- und Lernpartners.

Der Bewertungsvorgang ist sehr simpel zu halten, damit eine übersichtliche, anonyme Aufbereitung möglich ist. Daher wäre die Bewertung in einer geringen Anzahl an Kategorien mittels positiv oder negativ vorstellbar.

3.3.5 Personalisierter Feed

Damit Schüler auch die Hilfe angeboten bekommen können, die sie erfragt haben, bedarf es einer Möglichkeit, diese Anfragen an andere Schüler weiterzuleiten.

Eine sehr grundlegende Option wäre an dieser Stelle die zeitlich absteigend sortierte Auflistung aller Beiträge. Jedoch hätte dies zum Nachteil, dass alte, unbeantwortete Anfragen vernachlässigt würden. Zudem würden häufig unpassende Beiträge angezeigt: Selbst ein Nutzer, der ein bestimmtes Fach oder Thema als Schwäche ausgewählt hat, würde Hilfeanfragen dieser Rubrik angezeigt bekommen.

Stattdessen wäre eine personalisierte Rangfolge der Beiträge sinnvoll, die die Auswahl an Posts auf die ausgewählten Stärken und Schwächen maßschneidert. Dies würde eine deutlich bessere Möglichkeit bieten, dass passende Nutzer die eigene Hilfeanfrage sehen und deutlich wahrscheinlicher auch antworten.

3.4 Abgrenzungskriterien

3.4.1 Soziales Netzwerk

Die Lernplattform soll kein soziales Netzwerk sein, sondern weist lediglich ähnliche Strukturen auf. Das bedeutet, die soziale Interaktion abseits des vorgesehenen Zwecks steht nicht im Fokus. Stattdessen dient die Plattform ausschließlich zum Lernen und Helfen jeglicher Art. Aus diesem Grund gibt es keine Möglichkeit *Freunde*, *Follower* oder ähnliches zu sammeln und diese auf dem öffentlichen Profil zur Schau zu stellen.

Das beschriebene Chat-System soll zudem nicht für anderweitige Kommunikation abseits vom Lernen eingesetzt werden, allerdings werden keine plattformseitigen Maßnahmen ergriffen, um eine derartige Nutzung zu unterbinden. Ohnehin wäre ein solcher Eingriff ein Verstoß gegen die erwünschte Privatsphäre eines jeden Chats.

3.4.2 Gruppen im Chat

Der Chat ist darauf fokussiert, individuelle Hilfe an die Schüler zu liefern. Dem stehen Chat-Gruppen fundamental entgegen, da sie niemals den gleichen Grad personalisierter Unterstützung bieten können. Natürlich sind einige Vorteile von gemeinsamem Lernen und dem gegenseitigen Austausch innerhalb einer Gruppe, beispielsweise eines Kurses oder einer Klasse, nicht zu verleugnen. Allerdings ist

es fragwürdig, inwiefern ein digitales Gruppenkonstrukt dieselben Effekte erzeugen würde. Zumal in solchen Situationen mitunter ein Gruppendruck auf Einzelne ausgeübt werden könnte und sich Manche womöglich daran gehindert fühlen, frei ihre Probleme zu äußern.

Als Ersatz für den gegenseitigen Austausch von Hilfe gibt es zumindest das Bereitstellen von Lernmaterial. Über diesen Weg können auch Schüler Hilfsmittel und Informationen anbieten, wie es in einer Gruppe geschehen könnte.

3.4.3 Zahlungsabwicklung

Oftmals gibt es bei traditioneller Nachhilfe ein Entgelt für die Schüler, die ihre Hilfe zur Verfügung stellen. Allerdings beruht die Plattform grundsätzlich eher auf der gegenseitigen Unterstützung, sodass Schüler jeweils in den Bereichen, in denen sie Hilfe benötigen, diese erhalten. Gleichzeitig können sie dann in ihren besseren Fächern diese Hilfe an andere Schüler weitergeben. Dementsprechend soll der finanzielle Anreiz zur Nachhilfe eher durch die Gegenseitigkeit der Unterstützung ersetzt werden.

Dennoch ist diese Gegenseitigkeit womöglich nicht in allen Fällen gegeben. In diesen Fällen wäre eine private Abmachung einer eventuellen Bezahlung denkbar, jedoch soll in diesem Prozess der Plattform keine Kontrolle, Verantwortung oder Aufsicht zukommen.

Teil II

Einsatz

4 Zielgruppe

Die entwickelte Lernplattform soll begleitend zum Schulalltag der individuellen Weiterbildung und Verbesserung dienen. Dementsprechend ist eine recht enge Verknüpfung des analogen Unterrichts mit dem digitalen Austausch einzelner Nutzer vorgesehen. Dies ließe sich beispielsweise durch freigegebene Lernmaterialien aus dem Unterricht auf der Lernplattform realisieren.

In erster Hinsicht richtet sich die Lernplattform an Schüler einer Schule, welche diese Software einsetzt. Diese können, wie bereits beschrieben, nach eigenem Ermessen und Bedarf Hilfe anbieten und entgegennehmen.

Weiterhin ließe es sich vorstellen, dass auch Lehrer auf der Plattform registriert sind und verschiedene Aufgaben übernehmen. Einerseits besteht das Risiko von schädlichen, illegalen oder unerwünschten Inhalten: Ein automatisierter Algorithmus zur Bekämpfung eben solcher Beiträge befindet sich außerhalb des Rahmens dieser Lernleistung, sodass eine Lösung wäre, Lehrer oder Verwaltungsmitglieder dazu anzuhalten, gemeldete Inhalte zu entfernen. Andererseits eröffnet die Anwesenheit der Lehrkräfte auch die Möglichkeit, wie bereits beschrieben, Lehrmaterialien bereitzustellen.

Zuletzt wäre es naheliegend, in einer kontrollierten Weise Dritten Zugang zu der Plattform zu verschaffen. Diese könnten unter anderem Eltern und Verwandte von den Schülern, Studierende sowie private Nachhilfe-Instrukteure umfassen. Dabei ist es allerdings von Bedeutung, dass sichergestellt wird, dass ausschließlich vertrauenswürdige Personen in Kontakt mit den Schülerinnen und Schülern gelangen. Immerhin ließe sich die Lernplattform in diesem Umfang als Erweiterung des Schulumfelds sehen, was einen erhöhten Schutz des Einzelnen erfordert. Auch Werbetreibende dürfen in diesem Zuge keinesfalls Teil der Lernplattform werden.

5 Technische Voraussetzungen

5.1 Server

Zur Bereitstellung der Lernplattform wird ein Server benötigt, welcher Node.js 12 ausführen kann sowie über einen Webserver, beispielsweise Apache 2, verfügt.

Weiterhin wird das Tool Angular CLI Version 9+ benötigt, um die Webseite für den Client zu veröffentlichen.

5.2 Datenbank

Eine MariaDB 10.3.22+-Datenbank wird zur Speicherung der Daten der Lernplattform benötigt.

5.3 Nutzer

Aufgrund der Strukturierung der Plattform als Webseite benötigt der Nutzer lediglich einen modernen Web-Browser. Dabei ist die Art des Endgeräts egal: sowohl auf Desktop-PCs, Notebooks als auch Smartphones lässt sich die Webseite mithilfe eines Browsers aufrufen. Getestet wurde die Verwendung der Lernplattform mit Mozilla Firefox Version 73, Google Chrome Version 79.

Teil III

Software-Architektur

6 Struktur

Die Lernplattform lässt sich in zwei grundlegende Hälften unterteilen: den Client und den Server.

Genauer gesagt stellt der Server eine API (*Application Programming Interface*) dar. Diese weit verbreitete Struktur weist zahlreiche Vorteile auf, abhängig von der genauen Umsetzung. Es handelt sich bei der Lernplattform um eine dynamische Implementierung vom REST-Design. Dies bedeutet, dass Daten und Inhalte asynchron während der Laufzeit der Webseite geladen werden.

6.1 REST

REST, ein Akronym für *Representational State Transfer*, stellt ein Paradigma zur Gestaltung von APIs dar. Erstmals wurde es von Roy Fielding im Jahr 2000 definiert, wobei er sechs Anforderungen an eine REST-API festlegte (Fielding, 2000).

1. Client-Server: Das Nutzerinterface und die Daten sind getrennt zwischen Client und Server.
2. Zustandslosigkeit: Jede Datenanfrage vom Client an den Server muss alle Informationen beinhalten, die zur Verarbeitung benötigt werden.
3. Caching: Die Antworten des Servers müssen enthalten, ob diese beim Client gespeichert werden dürfen, um zukünftige Anfragen zu reduzieren.
4. Einheitliche Schnittstelle: Eine einheitliche Schnittstelle dient der Schaffung eines global uniformen Designs von REST-APIs. Mitunter werden Daten von den Ressourcen über eine URI (*Uniform Resource Identifier*) per HTTP-Methoden¹ abgefragt. Außerdem muss die Antwort jegliche Informationen beinhalten, die der Client zur Nutzung benötigt.
5. Mehrschichtiges System: Die Architektur des Systems soll mehrschichtig sein, um beispielsweise verschiedene Versionen von Client beziehungsweise

¹Zu den HTTP-Methoden gehören in erster Linie *GET* für das Erhalten von Daten, *POST* für das Hinzufügen von Einträgen, *PATCH* zum Bearbeiten von Daten und *DELETE* zum Löschen von Einträgen.

Server auseinander zu halten oder serverseitige Caches zu nutzen. Dabei hat eine Schicht lediglich Zugriff auf benachbarte Ebenen.

6. Code-on-Demand: Optional erlaubt REST das Hinzuladen von Code zur lokalen Ausführung auf dem Client.

REST APIs bestehen aus Ressourcen, welche einzelne Informationseinheiten beschreiben. Diese können über Endpoints aufgerufen oder manipuliert werden. Eine Route gibt den URL-Pfad an, der zu einem bestimmten Endpoint führt.

6.2 Vorteile

Bei der Lernplattform werden konstante sowie dynamische Daten getrennt. Layout und Design verwaltet der Client und über asynchrone Abfragen an den Server werden die dynamischen Inhalte nachgeladen. Dies geschieht zu dem Zeitpunkt und in dem Ausmaß, wie es erforderlich ist.

Das Gegenteil dazu wäre eine Struktur, bei der die wechselnden Inhalte bei jedem Seitenaufruf direkt vom Server mitgesendet werden. Dies schließt alle konstanten Daten, wie Styling und Layout, mit ein.

Flexibilität Während beim dargestellten Gegenstück der verwendeten Client-Server-Architektur jegliche Daten zusammen mit den Informationen für die letzte Darstellung direkt an den Nutzer gesendet werden, stellt der Server bei einer API-Struktur ausschließlich die rohen Daten bereit. Die visuelle Aufbereitung übernimmt der Client, welcher diese optimal an die Hardware-Gegebenheiten anpassen kann. Sobald man Änderungen an dieser Darstellung vornehmen möchte, wird dies durch die strikte Trennung von Daten und Aussehen simplifiziert.

Geringere Bandbreite Die API-Struktur benötigt deutlich geringere Mengen an Bandbreiten, was insbesondere auf Mobilgeräten von Vorteil ist. Da nicht jegliches Styling und Layout bei jedem einzelnen Seitenwechsel geladen werden muss, ergibt sich eine erheblich geringere Datenmenge, die zu übertragen ist. Außerdem werden die anzuzeigenden Daten in einem stark komprimierten Format übertragen: Es muss nicht das Aussehen etc. für diese Daten jedes Mal mit übertragen werden. Stattdessen ist der Client selbst für die Verarbeitung der rohen Daten verantwortlich.

Integrationsfähigkeit Derzeit besteht die Lernplattform ausschließlich aus einer Web-Applikation, wobei auch keine Erweiterungen geplant sind. Jedoch ist ein weiterer Vorteil der unabhängigen Entwicklung von Client und Server, dass man in Zukunft je nach Bedarf auch mobile Apps für Android und iOS oder ein Programm für Windows, Linux und macOS entwickeln könnte. In diesem Fall müsste

man keinen neuen Server entwickeln, sondern könnte sich lediglich auf die neue Client-Architektur konzentrieren, was die Entwicklung signifikant vereinfacht.

Auch wäre es denkbar, dass die API Kunden zur Verfügung gestellt werden könnte, sodass sie einen eigenen Client entwickeln können, der ihren individuellen Bedürfnissen gerecht wird.

7 Client

7.1 Eingesetzte Technologien

Der Client basiert auf dem Angular-Framework und wurde mit TypeScript entwickelt.

Als fundamentalster Ansatz zum Erstellen eines Web-Clients bietet sich die Entwicklung einer klassischen Webseite mittels der Standardsprache im Webdesign *Hypertext Markup Language* (HTML) an. Die Inhalte des HTML-Layouts werden dann mittels JavaScript manipuliert, um dynamischen Daten gerecht zu werden.

Zur Vereinfachung der Veränderung der DOM-Struktur (*Differential Object Marking*) stehen verschiedene Bibliotheken zur Verfügung. Als weit verbreitetes Beispiel ließe sich hier jQuery nennen.

Daraus gingen später dedizierte Webapplikationsframeworks für das Frontend hervor. Diese vereinfachen die Entwicklung von dynamischen Webseiten und unterstützen den Entwickler in vielseitiger Hinsicht. Zusätzlich zeigte sich eine Umstrukturierung hin zu Komponenten, also abgeschlossenen Paketen von HTML. Diese weisen insbesondere eine bessere Wiederverwendbarkeit auf und somit muss Code seltener doppelt geschrieben werden.

Die beschriebenen Ansätze spiegeln grob auch die Entwicklungsphasen des World Wide Webs wider, welches sich von statischen Texten zu multimedialen Webapplikationen wandelte (Nutter, 2016).

In diesem Fall war die Entscheidung für eine Methode recht schnell getroffen, da das selbstständige Abfragen und Anzeigen von Inhalten mit nativem JavaScript erheblich komplexer ist als jegliches Webframework. Darüber hinaus können diese Frameworks auch viele weitere Prozesse einfacher gestalten, wie zum Beispiel das Senden von HTTP-Anfragen an den Server.

7.1.1 Angular

Das Webapplikationsframework Angular ist ein quelloffenes Projekt, initiiert von Google, und ermöglicht die Entwicklung in TypeScript.

Zudem wird für die Lernplattform Angular Material verwendet, um viele Aspekte des Designs und Layouts schneller umzusetzen. Angular Material ist

eine Implementierung der Material Design-Guidelines von Google.

Neben dem Kern von Angular, dem Erstellen dynamischer Webseiten, werden verschiedene weiterführende Funktionen mitgeliefert. Dazu gehören das Senden von HTTP-Anfragen, was die Einbettung von APIs vereinfacht; von Anfang an integrierte Unit-Tests sowie eine Library zum Verwenden asynchroner Daten.

Die Unit-Tests werden mit Jasmine¹ definiert und beziehen sich jeweils auf alle Einzelteile, die in Angular vorhanden sind. Mittels Karma², einem Tool zum Ausführen von Jasmine-Tests, lassen sich auf diese Weise Teile der Applikation schnell auf ihre Funktionalität testen.

Für asynchrone Daten setzt Angular RxJS ein, welches einen alternativen Ansatz zum nativen `async/await` in JavaScript bietet. Anstatt, dass die Code-Ausführung in einer Funktion gestoppt wird, bis die asynchrone Methode ein Ergebnis zurückgibt, wie es in JavaScript der Fall ist, basiert RxJS auf sogenannten Observables. Ein Observable stellt ein Datenkonstrukt dar, welches dynamisch mit Daten gefüllt werden kann. Dabei benötigt es nicht unbedingt einen Startwert. Es kann sich vorgestellt werden wie ein Array, dessen Einträge nach und nach gefüllt werden. Dabei stellt der hinterste Eintrag den aktuellen Wert dar.

Mithilfe von Subscriptions kann überall Code ausgeführt werden, sobald sich der Wert eines Observables ändert. Währenddessen kann jeglicher Code außerhalb dieser Subscription ungehindert weiterlaufen. Darüber hinaus stehen Pipes zur Verfügung, welche unterschiedliche Operationen auf einen oder mehrere Observables anwenden kann, um die Daten bei jeder Veränderung zu manipulieren. Es ist jedoch festzuhalten, dass RxJS auf den Promises von JavaScript, womit auch `async/await` funktioniert, aufbaut und somit lediglich eine Schnittstelle auf einem höheren Programmlevel bietet.

Ebenfalls positiv hervorzuheben ist die gute Performance von Angular (Borrelli, 2019).

7.1.2 Struktur

Angular basiert auf Komponenten. Dies sind kleine, abgegrenzte Bestandteile des Nutzer-Interfaces und umfassen jeweils Layoutinformationen in HTML, Styling in CSS sowie Logik in TypeScript, beispielsweise zur Implementierung von Nutzerinteraktionen.

Die Architektur basiert dabei auf dem Model-View-Viewmodel-Paradigma (MVVM).

¹Jasmine ist ein Framework zum Testen von JavaScript-Code und wird in Angular für Unit-Tests eingesetzt.

²Karma dient zum Ausführen von in Jasmine definierten Tests. Dazu wird der Code in realen Browserumgebungen getestet.



Abbildung 7.1: Ausschnitt der Komponentenstruktur von Angular anhand der Startseite und Post-Details.

Durch diese Verbindung von Views, also Interface-Elementen, und Models, wie zum Beispiel vom Server abgefragte Datensätze, ist eine simple und schnelle Änderung der angezeigten Inhalte möglich. Besonders ist hierbei, dass Daten beidseitig in die Views eingebunden werden können: Änderungen in einem Model spiegeln sich automatisch in einer Änderung der Anzeige wieder und Nutzerinteraktionen, wie das Drücken eines Buttons, können auch direkt die Änderung eines Models auslösen. Views und Viewmodels werden wie beschrieben über HTML, CSS und TypeScript in den Komponenten vereint.

Sogenannte Services ermöglichen zudem die gleichzeitige Aktualisierung von Daten auf der gesamten Webseite sowie das Verarbeiten von Daten.

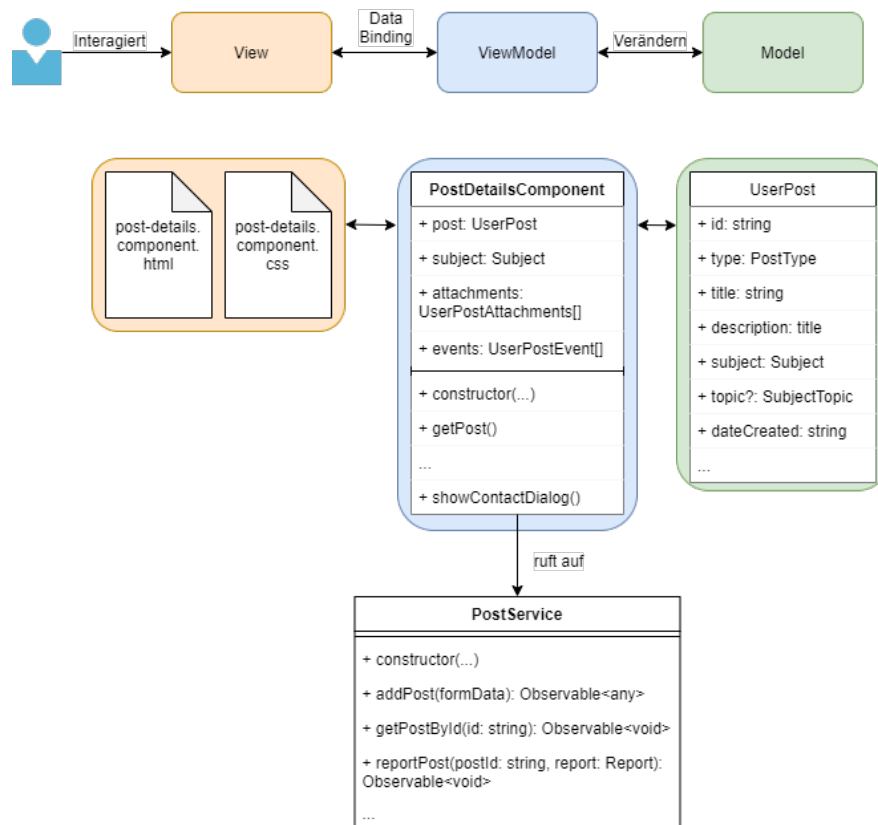


Abbildung 7.2: Struktur von MVVM basierend auf der Anzeige von Post-Details.

8 Server

Für die Entwicklung des Servers wurde das Framework Loopback 4 eingesetzt, welches ebenso wie Angular auf der Programmiersprache TypeScript basiert. Es ist mittels Node.js entwickelt worden.

8.1 Eingesetzte Technologien

8.1.1 Node.js

Mithilfe der Software-Umgebung Node.js lässt sich serverseitig JavaScript ausführen. Somit stellt es eine Alternative dar zu anderen Sprachen zur Realisierung von Server-Anwendungen wie beispielsweise PHP.

Im Rahmen der Lernplattform bot sich Node.js als sinnvolle Möglichkeit an, den Webserver zu erstellen, da es die Entwicklung in einer einzigen Programmiersprache ermöglicht, welche sowohl für den Client als auch den Server eingesetzt wird - TypeScript.

Weiterhin weist Node.js eine recht gute Performance auf. Basierend auf der V8-Engine von Google für ihren Chrome-Webbrowser wird Node.js kontinuierlich weiterentwickelt und dabei auch die Leistung verbessert (Warncke, 2020).

8.1.1.1 Node Package Manager

Der Node Package Manager (NPM) ist ein weit verbreitetes Tool zum Veröffentlichen und Herunterladen von Software-Paketen für Node.js. Es stellt eine umfangreiche Datenbank an Libraries, Programmen und Frameworks dar. Mithilfe der NPM CLI (*Command Line Interface*; Befehlszeilenschnittstelle) können diese Code-Pakete heruntergeladen sowie als Basis für weitere Software-Projekte verwendet werden. Alle im Folgenden aufgelisteten Libraries wurden über NPM installiert.

NPM wird bei der Installation von Node.js mitinstalliert.

8.1.2 LoopBack 4

LoopBack 4 ist die vierte Version eines umfangreichen, quelloffenen, von Strongloop geführten Frameworks für Node.js. Es fokussiert sich auf das Kreieren von REST APIs. Das Framework basiert auf Express.js, einem weit verbreiteten Node.js-Framework für Web-Server. Die primär verwendete Sprache ist der JavaScript-Dialekt TypeScript.

Loopback ermöglicht durch die direkte Einbindung diverser Systeme und Datenbanken die einfache Entwicklung eines Backends. Im Falle der Lernplattform handelt es sich hierbei um eine MariaDB-Datenbank (s. 8.2).

8.1.2.1 Struktur

Die Struktur besteht im einfachsten Fall aus einer DataSource, einem Model, einem Repository und einem Controller. Als Erweiterung sind beispielsweise noch Services möglich.

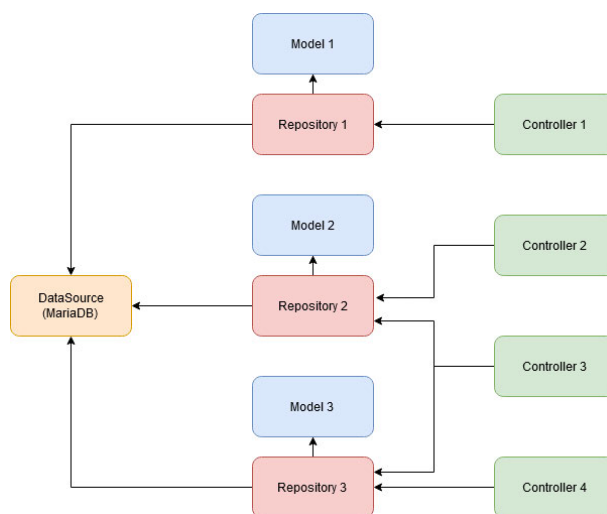


Abbildung 8.1: Beispielhafter Aufbau einer Loopback-Applikation.

DataSource Eine DataSource ist eine Verbindung zu einer externen Quelle von Daten, wie beispielsweise einer Datenbank, anderen Diensten oder APIs. In diesem Fall gibt es lediglich eine DataSource, namentlich die MariaDB-Datenbank.

Model Ein Model ist ähnlich wie bei objektorientierter Programmierung eine Klassifizierung von Daten. Beispiele hierfür sind die Models für Nutzer oder Posts. Es enthält immer alle Attribute, welche eine Instanz dieser Klasse eindeutig kennzeichnen.

Repository Ein Repository stellt das Verbindungsstück zwischen einem definierten Model und der Datenbank dar. Es sorgt für die initiale Erstellung einer Tabelle in der Datenbank oder einer anderen DataSource und fügt die Attribute des Models als Spalten hinzu. Später ermöglicht das Repository verschiedene Operationen, die den SQL-Befehlen der Datenbank entsprechen: Unter anderem lassen sich Daten abfragen, hinzufügen, bearbeiten oder löschen.

Controller In einem Controller werden Endpoints der API (Beispiel: s. Code-Ausschnitt 8.1) definiert, welche über jeweils spezifizierte Routen zugänglich sind. Die Controller repräsentieren somit ein Verbindungsstück zwischen dem Nutzer beziehungsweise dem Client und den Repositories oder Services.

```
1 @authenticate('jwt') // Nutze JWT-Authentisierung
2 @authorize(['ViewChat:Self']) // Spezifiziere benötigte
  Berechtigungen
3 @post('/chats/connect', { // Definiere neue POST-Ressource mit
  angegebener URL
4   responses: { // Dokumentiere mögliche Antworten dieses
    Endpoints
5     '200': { // 200 = HTTP Code Success
6       description: 'Checks whether the provided token is
          correct and returns the user profile if so',
7     }
8   }
9 })
10 async connect( // Beginn der asynchronen Funktion
11   @inject(AuthenticationBindings.CURRENT_USER) currentUser:
    UserProfile, // Erhalte Profildaten des derzeitigen Nutzers
12 ): Promise<UserProfile> { // Gebe den Rückgabewert an
13   return currentUser; // Gebe die Profildaten zurück
14   // Alle Rückgaben werden automatisch in das gewünschte Format
    gebracht und zurückgesendet
15 }
```

Code-Ausschnitt 8.1: Beispiel-Endpoint aus dem Chat-Controller (`server/src/controllers/chat.controller.ts`).

Services Services stellen ein übergeordnetes Konstrukt zu Repositories dar. Sie können jegliche Art von lokalen Operationen auf Datensätze enthalten oder auch Verbindungen zu anderweitigen APIs herstellen.

8.1.3 Weitere Libraries

Zur Vereinfachung vieler Prozesse auf der Lernplattform werden verschiedene Libraries eingesetzt. Diese ersetzen die eigene Entwicklung allgemeiner Aufgaben, wie das Manipulieren von Bildern.

8.1.3.1 Multer

Multer ist eine Library für Node.js, welche die Verarbeitung hochgeladener Dateien ermöglicht und die rohen Daten in eine besser nutzbare Form bringt. Unter anderem ermöglicht Multer es, auf einfachem Wege die erlaubte Dateigröße, -anzahl sowie -typen zu konfigurieren. Die Library wird beispielsweise für das Hochladen von Anhängen für Posts, Profilbildern und Dateien im Chat eingesetzt.

8.1.3.2 Nodemailer

Die Node.js-Library Nodemailer vereinfacht das Senden von E-Mails über einen SMTP-Server signifikant. Das Senden von Benachrichtigungen an die Nutzer, zum Beispiel zum Zurücksetzen des Passworts oder Bestätigen der E-Mail-Adresse, fundiert auf dieser Library.

8.1.3.3 Sharp

Als Node.js-Modul für effizientes Verarbeiten von Bildern wird Sharp hauptsächlich eingesetzt, um hochgeladene Bilder wie die Profilbilder in eine kleinere Auflösung und ein einheitliches Bildformat zu konvertieren.

8.1.3.4 JsonWebToken

JsonWebToken ist eine Node.js-Implementation des *JSON Web Token*-Standards (JWT) (Jones, Bradley & Sakimura, 2015). JWT ermöglicht die Authentifizierung von eingeloggten Nutzern. Dazu wird ein Zugriffstoken beim Einloggen generiert, der neben verschiedenen Informationen über den eingeloggten Nutzer auch eine Signatur aufweist, um zu verifizieren, dass es sich um einen validen Token handelt. In der verwendeten Implementation wird dazu mitunter ein geheimes Secret eingesetzt, welches serverseitig gespeichert ist.

8.1.3.5 CSV-Parser

Mithilfe der Library CSV-Parser werden *.csv*-Dateien (*Comma-separated values*) verarbeitet und in eine leichter zu verwendende Form gebracht. Das Einsatzgebiet hierfür liegt beim Erstellen von mehreren Nutzern gleichzeitig.

8.1.3.6 Request

Request ist ein HTTP-Client für Node.js, welcher im Rahmen der Lernplattform verwendet wird, um von der WebSocket-Verbindung mit der API zu kommuniziere-

ren. Dies geschieht, indem Anfragen an den Server gesendet werden, wie es auch durch den Client geschieht.

Es ist anzumerken, dass die Library mittlerweile als deprecated¹ deklariert wurde und somit in Zukunft keine Aktualisierungen mehr erhalten wird. Dementsprechend wäre es in Zukunft notwendig, eine Alternative hierfür zu finden und integrieren.

8.2 Datenbank

Zu Zwecken der Datenspeicherung und -verwaltung kommt MariaDB zum Einsatz, was ein open-source, relationales Datenbank-Verwaltungssystem ist und als Abspaltung von MySQL entstand. Dennoch ist MariaDB größtenteils kompatibel mit MySQL und wird auch weiterhin dahingehend entwickelt (MariaDB Corporation Ab, o.J.).

Ausschlaggebend für die Wahl von MariaDB war, dass es einerseits eine kostenlose sowie frei anwendbare SQL-Lösung darstellt und andererseits weit verbreitet ist sowie über eine große Entwickler-Basis verfügt, wobei es auch stark von der Popularität von MySQL profitiert (solid IT GmbH, o.J.).

8.3 Implementierungen

8.3.1 Nutzer- und Postsystem

Wie in Abbildung 8.2 dargestellt, besteht der Kern der Applikation aus dem Nutzersystem sowie der Möglichkeit, Posts sowie Antworten auf Posts zu verfassen. In dem Diagramm wurden wichtige andere Bestandteile, wie beispielsweise Anhänge von Beiträgen, Passwörter, Berechtigungen und Weiteres, zwecks Übersichtlichkeit weggelassen.

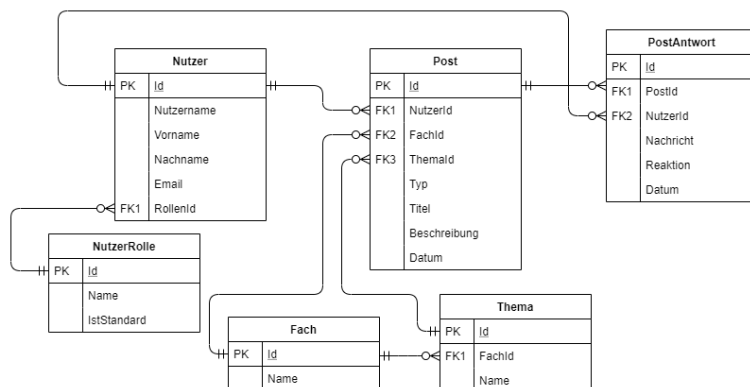


Abbildung 8.2: ER-Diagramm der simplifizierten Datenstruktur der Kern-Applikation mit Nutzern und Posts.

¹Die Entwicklung wird nicht weitergeführt und vom weiteren Einsatz wird abgeraten.

Post-Antworten sind zudem stark mit den Chat-Nachrichten (s. 9) verknüpft. Die Kontaktaufnahme erfolgt meist über dieses Schema:

1. Der Hilfesuchende (im Folgenden Nutzer A) verfasst einen Post des Typs Anfrage und legt sein Problem mithilfe von Text und gegebenenfalls Anhängen dar.
2. Ein Helfer (im Folgenden Nutzer B) bekommt den Beitrag von Nutzer A angezeigt, möglicherweise aufgrund seiner eingestellten Fähigkeiten.
3. Nutzer B liest sich das Problem von Nutzer A durch und antwortet im Anschluss auf die Anfrage, um Kontakt aufzunehmen. Dabei geht er darauf ein, wie er Nutzer A grob helfen kann beziehungsweise was ihn dafür qualifiziert.
4. Sowohl per E-Mail als auch per Benachrichtigung auf der Lernplattform erhält Nutzer A eine Mitteilung über die neue Antwort auf seine Anfrage.
5. Nutzer A kann nun entscheiden, ob er die Kontaktaufnahme von Nutzer B akzeptiert und somit einen Chat startet oder den Kontakt ablehnt. In letzterem Fall passiert nichts.
6. Sollte Nutzer A den Kontakt ebenfalls aufnehmen wollen, wird beiden Nutzern ein neuer Kontakt hinzugefügt sowie eine Chat-Konversation erstellt. Diese bildet eine Verknüpfung des Chats mit der ursprünglichen Anfrage von Nutzer A sowie der Antwort von Nutzer B.
7. Die Nutzer können fortan den Chat nutzen, um jederzeit und von überall miteinander zu kommunizieren und das Problem von Nutzer A zu lösen.
8. Sobald die Hilfe abgeschlossen ist - also entweder das Problem gelöst wurde oder Nutzer B nicht mehr weiterhelfen kann - steht Nutzer A die Option frei, die Konversation zu beenden sowie zu bewerten. Die Bewertung fließt in das Nutzerprofil von Nutzer B ein. Umgekehrt besteht eine solche Möglichkeit nicht.

8.3.2 Personalisierter Feed

Als simplen Ersatz von komplexen künstlichen Intelligenzen zum personalisierten Empfehlen von Beiträgen wird jedem Post ein individueller Wert zugewiesen. Anschließend können mithilfe dieses Werts die Posts sortiert werden.

Dabei startet jeder Post mit einem Wert von 50 und bekommt anschließend entweder Punkte abgezogen oder addiert. Allgemein erhalten Beiträge Abzüge, wenn sie vom Nutzer selbst stammen oder bereits archiviert sind.

```

1 let value = 50;
2
3 // Bestraft Beiträge vom Nutzer selbst
4 if (post.user.id === currentUser.id) {
5     value -= 50;
6 }
7
8 // Bestraft archivierte/beendete Beiträge
9 if (post.isArchived) {
10    value -= 50;
11 }

```

Code-Ausschnitt 8.2: Allgemeiner Abschnitt des Algorithmus (server/src/services/post.service.ts).

Der nachfolgende Prozess ist individuell je nach Typ des Beitrags. An dieser Stelle bezieht sich der Algorithmus lediglich auf die Variante für Anfragen.

```

1 // Klassen vergleichen
2 if (schoolClassYear && post.classYear) {
3     if (schoolClassYear.orderPosition < post.classYear.
4         orderPosition) {
5         value -= 25;
6     } else {
7         value += (schoolClassYear.orderPosition - post.classYear.
8             orderPosition) * 10;
9     }
10 }

```

Code-Ausschnitt 8.3: Wertung von Anfragen basierend auf Klassenstufen (server/src/services/post.service.ts).

Alle Anfragen, die von Nutzern aus höheren Klassenstufen verfasst wurden, erhalten 25 Punkte Abzug. Dahingegen werden Posts 10 Punkte für jede Klassenstufe, die der aktuelle Nutzer höher ist als der Verfasser der Anfrage, zugeschrieben. Dies bewirkt, dass ältere Schüler deutlich eher Anfragen aus niedrigeren Klassen empfohlen bekommen.

```

1 // Filtert die Fähigkeiten des Nutzers nach Fach und Thema des
2   Posts
3 const skillsIntersection = skills.filter(skill =>
4     skill.topic.subjectId === post.subject.id ||
5     (post.topic && skill.topic.id === post.topic.id)
6 );
7 // Hat der Nutzer das Thema des Posts als Fähigkeit?
8 skillsIntersection.forEach(skill =>
9     value += (skill.ability * Math.abs(skill.ability)) * 15
10 );

```

Code-Ausschnitt 8.4: Bepunktung einer Anfrage anhand von Thema und Fach (server/src/services/post.service.ts).

Weiterhin beeinflussen die ausgewählten Fähigkeiten des Nutzers stark die Einstufung der Posts. Es wird je für Thema und Fach der Anfrage überprüft, ob der Nutzer diese in seinen Fähigkeiten und Schwächen festgelegt hat. Daraufhin bekommt die Anfrage Boni, sollte der Nutzer ein Fähigkeitsniveau größer als Null ausgewählt haben. Andernfalls werden Punkte abgezogen. Dabei entspricht ein Fähigkeitsniveau 15 Punkten.

```
1 // Bonus für unbeantwortete Fragen, die länger existieren
2 const dayBonus = Math.min(getElapsedDays(post.dateCreated), 28);
3 const replyCount = await this.postResponseRepository.count({
4   postId: post.id,
5 });
6 value += Math.round(Math.sqrt(dayBonus * 10) - replyCount.count
   ** 2 * 10);
```

Code-Ausschnitt 8.5: Sortierung in Hinsicht auf unbeantwortete Anfragen (server/src/services/post.service.ts).

Als letzte Instanz fließen das Datum sowie die Anzahl an Antworten in die Wertung der Posts ein. Dabei werden mehr Punkte vergeben, je älter der Beitrag ist, jedoch maximal bis zu 4 Wochen. Abgezogen werden Punkte für bereits beantwortete Fragen. Unbeantwortete Beiträge erhalten einen Bonus. Bereits bei einer Antwort werden teils Punkte abgezogen, teils geschieht nichts. Ab zwei Antworten werden immer Punkte subtrahiert, damit unbeantwortete Fragen in den Vordergrund rücken.

9 Chat

9.1 WebSocket

WebSocket ist ein digitales Protokoll zur beidseitigen Kommunikation über eine einzelne TCP-Verbindung. Ebenso wie das *Hypertext Transfer Protocol* (HTTP) ist es im OSI-Modell¹ auf der Anwendungsschicht einzuordnen.

Im Rahmen der Lernplattform dient eine WebSocket-Verbindung zwischen jedem eingeloggten Client und dem Server zum Echtzeit-Austausch von Chat-Nachrichten und mehr.

Dabei wird Socket.IO eingesetzt, um die WebSocket-Verbindungen zu vereinfachen sowie fehlerresistenter zu gestalten. Die JavaScript-Library basiert für die bidirektionale Kommunikation hauptsächlich auf WebSocket, fällt jedoch auf periodische HTTP-Anfragen zurück, sollte diese Verbindung nicht aufgebaut werden können (SocketIO, o.J.).

¹Das *Open Systems Interconnection*-Modell, kurz OSI, dient zur Charakterisierung von Kommunikationsprozessen und weist verschiedene Abstraktionsschichten auf.

9.2 Verschlüsselung

Hinsichtlich der Verschlüsselung der Chat-Nachrichten muss ein Mittelweg zwischen absoluter Sicherheit und vollkommenem Komfort gefunden werden: Völlig ohne Verschlüsselung ist die Plattform sehr einfach zu bedienen sowie zu entwickeln jedoch höchst anfällig gegenüber jeglichem Angriff. Umgekehrt kann ein zu hohes Maß an Sicherheit signifikant das Nutzungserlebnis beeinträchtigen.

Die sicherste Möglichkeit, um Chat-Nachrichten zu verschlüsseln, besteht in der Ende-zu-Ende-Verschlüsselung (E2E). Allerdings lässt sich diese Art der Verschlüsselung im Kontext einer Web-Applikation nur schwierig umsetzen: Die Kodierung der Nachrichten basiert auf einem Paar von Schlüsseln, welche für jeden Nutzer einzigartig sind.

Einer der Schlüssel ist öffentlich zugänglich und kann ohne Probleme beispielsweise in der vorhandenen Datenstruktur gespeichert werden. Mit diesem Schlüssel könnte jeder Nutzer seine Nachricht so verschlüsseln, dass lediglich der Eigentümer des zugehörigen privaten Schlüssels in der Lage ist, den Inhalt zu entschlüsseln. Dieser andere Schlüssel darf auf keinen Fall anderen Nutzern zugänglich sein, da die gesamte Sicherheit des Verfahrens auf der Geheimhaltung des privaten Schlüssels basiert.

Jedoch kann der Schlüssel, im Gegensatz zu beispielsweise einer Smartphone-App, nicht dauerhaft auf dem Gerät gespeichert werden. Einerseits ist die Möglichkeit zum geschützten Speichern sensibler Daten lokal im Browser nur begrenzt möglich.

Andererseits besteht der fundamentale Vorteil einer Web-Applikation in der Flexibilität, sich jederzeit und überall einzuloggen und vollen Zugriff auf seine Daten zu haben. Würde jedoch der private Schlüssel sowie die bereits empfangenen Chat-Nachrichten in einem spezifischen Browser speichern, ginge diese Eigenschaft verloren.

Daher bietet sich lediglich ein symmetrisches Verschlüsselungsverfahren mit einem auf dem Server gespeicherten Schlüssel an, um die Nachrichten nicht im Klartext in der Datenbank zu speichern. Dabei folgt die Sicherheit aus

- der Sicherheit des verwendeten Verfahrens,
- der Geheimhaltung sowie Komplexität des Schlüssels und
- der Unbekanntheit des eingesetzten Verschlüsselungsverfahrens.

In Abbildung 9.1 ist die Verschlüsselung einer Chat-Nachricht schematisch dargestellt. Als Eingaben werden ein Codeword, welches einmalig für den Server

ist; die Nutzer-Id des Senders sowie des Empfängers, die sich nie ändert für einen Nutzer, und die gesendete Nachricht benötigt.

Dabei ist anzumerken, dass die Chat-Nachrichten über WebSocket nicht zusätzlich verschlüsselt an den Server gesendet werden. An dieser Stelle basiert die Sicherheit des Systems auf der *Transport Layer Security* (TLS), wofür als Beispiel HTTPS genannt werden kann.

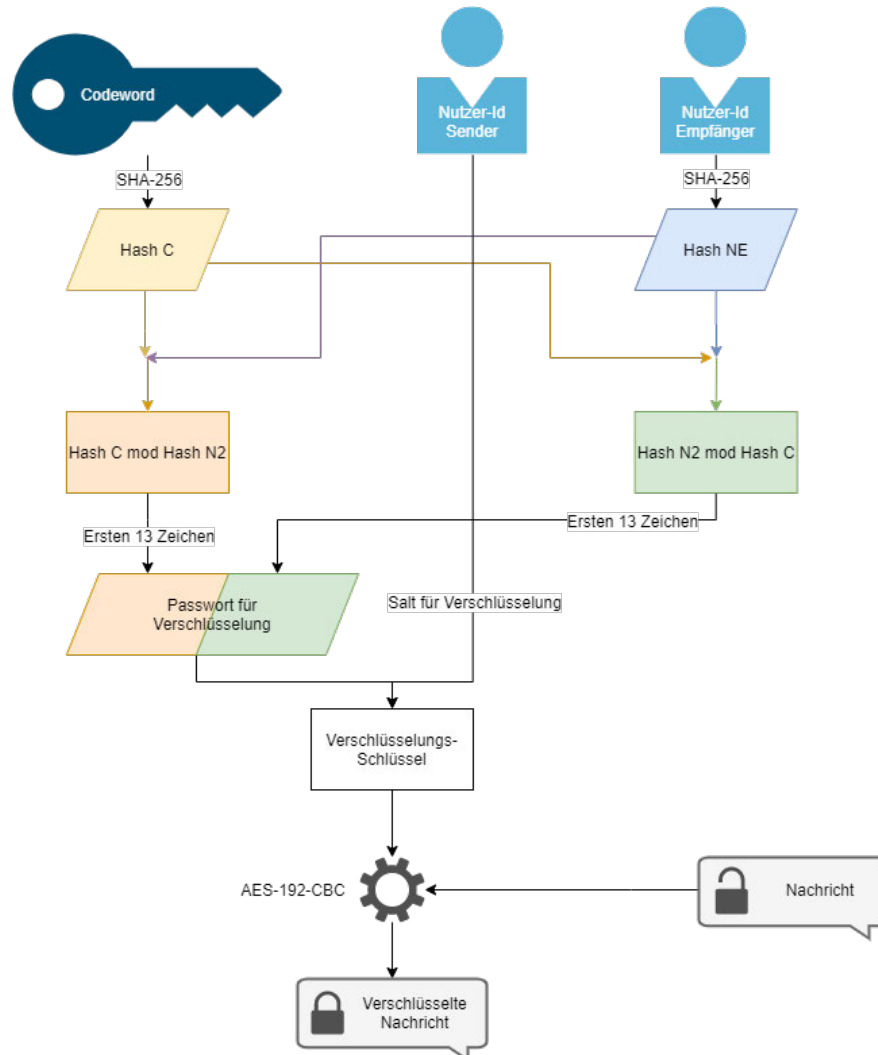


Abbildung 9.1: Schema der Verschlüsselung von Chat-Nachrichten.

Zwar ist es mit dieser Verschlüsselung dennoch möglich für den Besitzer des Codewords, alle gesendeten Chat-Nachrichten zu entschlüsseln. Dennoch ermöglicht die Einbindung der Nutzer-Ids, dass keinesfalls gleichzeitig alle Chat-Nachrichten mit demselben Schlüssel entschlüsselt werden können.

Zudem schützt diese serverseitige Verschlüsselung vor dem Stehlen von Chat-Nachrichten, sollte ein Angreifer Zugriff auf die Datenbank erhalten.

9.3 Funktionsweise

Ebenso wie die allgemeine Struktur der Lernplattform besteht auch der Chat mitunter aus einem Client und aus einem Server. Jedoch kommt an dieser Stelle der bereits beschriebene WebSocket zum Einsatz, welcher das Verbindungsstück dieser beiden Teile ist.

Die gesamte Kommunikation zwischen allen Clients, also den Webbrowsern der eingeloggtten Nutzer, findet über diesen Kanal statt. Auf der Client-Seite übergibt ein Angular-Service die geschriebene Chat-Nachricht mit Metadaten, wie dem Empfänger, an den WebSocket. Nachdem die Nachricht den Server erreicht, übergibt die WebSocket-Verbindung die Daten an die eigentliche Chat-API, welche mittels Loopback entwickelt wurde.

Dort wird die Chat-Nachricht verarbeitet, in der Datenbank gespeichert und anschließend zurückgegeben, damit sie der WebSocket an alle derzeitigen Anmelde-Instanzen von Sender beziehungsweise Empfänger senden kann. Von dort übernimmt der Client wieder und kümmert sich um Darstellung, Gruppierung, Benachrichtigungen etc.

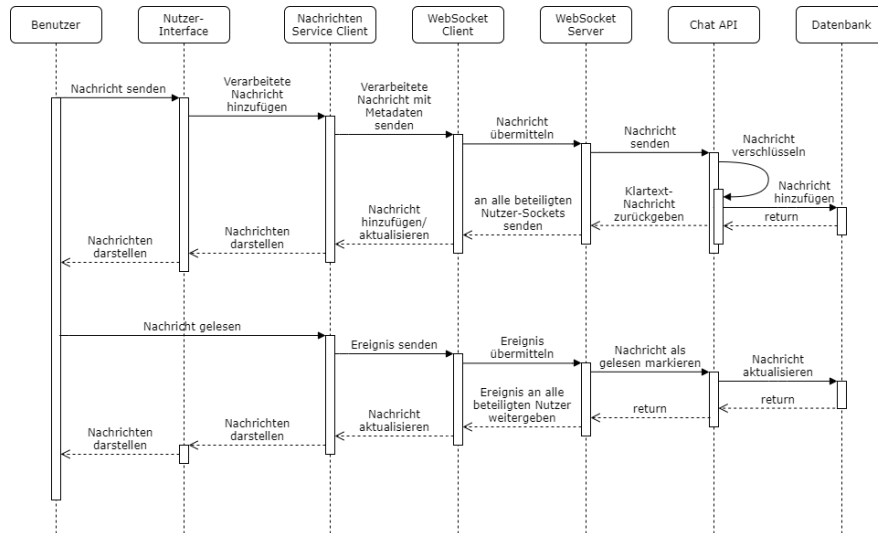


Abbildung 9.2: Ablauf des Senden und Empfangen einer Chat-Nachricht inklusive Markieren als gelesen.

Teil IV

Funktionen

10 User Stories

10.1 Anfrage erstellen

Als Schüler kann ich eine Anfrage mit einem Titel, einer Beschreibung, einem Fach und einem Thema erstellen, um Hilfe bei einem Problem zu erhalten.

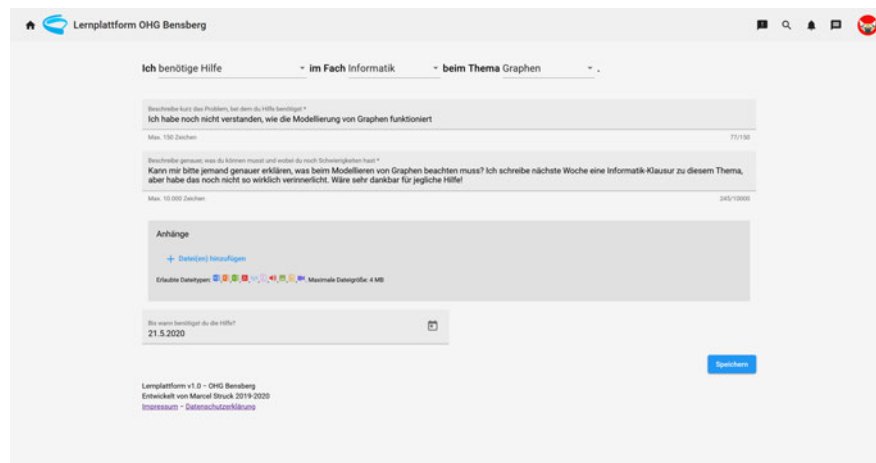


Abbildung 10.1: Erstellen einer Anfrage mit Thema und Fälligkeitsdatum.

Akzeptanzkriterien

- Der Nutzer muss mindestens einen Titel, eine Beschreibung, einen Typ und ein Fach ausgewählt haben.
- Die Anfrage wird umgehend veröffentlicht.

10.2 Anfrage beantworten

Als Schüler kann ich Details einer Anfrage einsehen und darauf antworten, um dem Nutzer zu helfen.

Akzeptanzkriterien

- Ein Schüler kann nur eine einzige Antwort pro Post verfassen.
- Der Hilfesuchende wird über die Antwort informiert.



Abbildung 10.2: Beispiel einer Antwort auf eine Anfrage.

10.3 Chat

Als Schüler kann ich mit hinzugefügten Kontakten Nachrichten austauschen, um Lernhilfe zu geben oder entgegenzunehmen. Diese kann ich jederzeit und auf allen Geräten einsehen und lesen.

Akzeptanzkriterien

- Es ist möglich, Chat-Nachrichten zu verfassen, welche beim Kontaktpartner angezeigt werden.
- Der Nachrichtenaustausch findet in Echtzeit statt.
- Bereits verschickte Nachrichten können gelöscht werden.
- Es ist ersichtlich, welcher Nutzer die Nachricht verfasst hat.

10.4 Hilfe bewerten

Als Schüler kann ich die erhaltene Hilfe von einem anderen Nutzer bewerten.

Akzeptanzkriterien

- Eine Hilfestellung kann lediglich einmal bewertet werden.
- Die Bewertung ist anonym.
- Auf dem Nutzerprofil ist die Gesamtbewertung einsehbar.

10.5 Login

Als Nutzer kann ich mich von jedem Gerät durch die Eingabe von Nutzernamen oder E-Mail-Adresse und Passwort einloggen, um auf meine Account-Daten zuzugreifen.

Akzeptanzkriterien

- Man kann sich als eingeloggter Nutzer wieder abmelden.
- Für den Verlauf der Sitzung bleibt man angemeldet.
- Bei einem vergessenen Passwort gibt es Hilfestellungen.

10.6 Registrierung

Als Schüler kann ich mich durch die Eingabe von Nutzernamen, Name, Passwort, E-Mail-Adresse und optional meinen Fähigkeiten registrieren. Im Anschluss kann ich meine E-Mail-Adresse bestätigen.

Akzeptanzkriterien

- Nutzernamen und E-Mail-Adresse werden überprüft, ob sie bereits verwendet wurden.
- Es kann ein Einladungscode angegeben werden, welcher den Nutzer automatisch bestätigt.
- Alle Eingaben werden auf ihre Validität überprüft.
- Die E-Mail-Adresse muss bestätigt werden.
- Es gibt bestimmte Kriterien zum Erstellen eines sicheren Passworts.
- Optional können Fähigkeiten ausgewählt werden.

10.7 Profil-Einstellungen

Als Schüler kann ich mein Nutzerprofil einrichten, indem ich beispielsweise ein Profilbild oder meine Fähigkeiten auswähle.

10.8 Account-Einstellungen

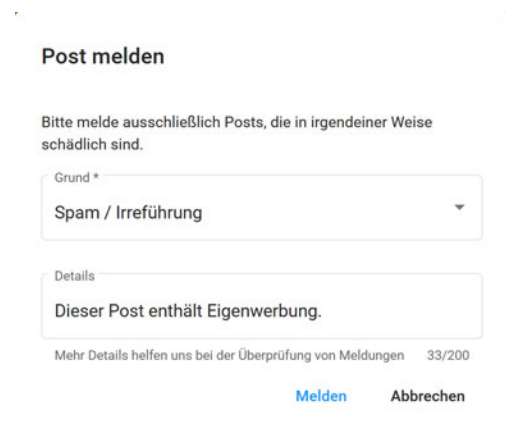
Als Nutzer kann ich meine Account-Einstellungen wie Nutzernamen, Name, Passwort und E-Mail-Adresse ändern.

10.9 Melden

Als Nutzer kann ich eine Anfrage oder einen Nutzer melden, um auf einen Regelverstoß aufmerksam zu machen.

Akzeptanzkriterien

- Es kann ein Grund angegeben werden.
- Die Meldungen werden Administratoren angezeigt.



Post melden

Bitte melde ausschließlich Posts, die in irgendeiner Weise schädlich sind.

Grund *
Spam / Irreführung

Details
Dieser Post enthält Eigenwerbung.

Mehr Details helfen uns bei der Überprüfung von Meldungen 33/200

[Melden](#) [Abbrechen](#)

Abbildung 10.3: Dialog zum Melden eines Beitrags.

10.10 Suchen

Als Nutzer kann ich alle Anfragen nach Titel, Fach, Typ u.ä. durchsuchen.

Akzeptanzkriterien

- Mehrere Filter können kombiniert werden.

Teil V

Bedienungsanleitung

11 Installation & Einrichtung

Basierend auf dem Fokus der leichten Bedienung existiert eine Cloud-Version der Lernplattform unter <https://lernen.kingofdog.de>. Über diese ist es jedem möglich, sich online zu registrieren und von überall die Plattform zum Lernen zu verwenden. Als Alternative zur Cloud-Instanz ist es möglich, die Lernplattform lokal zu installieren und verwenden.

Die Installation setzt sich aus dem Client und dem Server sowie der Datenbank zusammen. Im Anschluss müssen die einzelnen Komponenten zur Verbindung eingerichtet werden. Es werden hierfür die definierten technischen Voraussetzungen (s. 5) benötigt.

11.1 Konfiguration

Im Hauptordner der Lernplattform ist eine `config.json`-Datei zu finden. Diese beinhaltet die meisten verfügbaren Einstellungen bezüglich der Lernplattform. Unter anderem können hier der Name der Schule, die zugängliche Adresse der Webseite sowie die zu verwendende E-Mail-Adresse konfiguriert werden.

Diesbezüglich lassen sich auch unter `client/src/assets` die Schullogos einfügen. Hierfür können `logo_large.svg` und `logo_small.svg` ersetzt werden. Dabei bezieht sich die große Variante auf das Logo, welches auf der Startseite für nicht eingeloggte Nutzer angezeigt wird, und die kleine Variante wird in der Navigationsleiste sowie als Favicon genutzt.

Es gibt drei Ports, die entweder auf den Standard-Werten belassen oder neu zugewiesen werden können. Diese sind jeweils für den Client, die API und den WebSocket. Außerdem kann eine URL festgelegt werden, unter der die Webseite öffentlich zugänglich ist. Dies kann beispielsweise von einem Apache-Webserver über einen Proxy geändert werden. Die eingetragene URL dient unter anderem dazu, in E-Mails auf die richtige Adresse zu verlinken.

Unter `mail` ist ein E-Mail-Account zu hinterlegen, der über SMTP zugänglich ist. Über diese E-Mail-Adresse werden Benachrichtigungen an die Nutzer versendet. Es sind der Host, also der Mail-Server, sowie der zugehörige Port, der Nutzernamen beziehungsweise die E-Mail-Adresse und das Passwort erforderlich.

Optional können erlaubte Dateiformate und -größen konfiguriert werden, welche beispielsweise für die Anhänge von Posts gelten.

Zuletzt muss in `server/src/datasources/db.datasource.json` die Datenbank hinzugefügt werden. Dafür werden mindestens der Host, der Nutzernamen, das Passwort und der Name der MariaDB-Datenbank benötigt, gegebenenfalls auch der Port, sollte dieser vom Standard-Port abweichen.

11.2 Client

Zuerst müssen im Ordner `client` die benötigten NPM-Pakete installiert werden. Dazu muss in der Befehlszeile `npm install` ausgeführt werden. Die Befehlszeile lässt sich unter Windows 10 auf verschiedene Wege öffnen; der einfachste ist, innerhalb des Ordners Rechtsklick auf eine leere Fläche zu drücken und dann *PowerShell-Fenster hier öffnen* auszuwählen.

Weiterhin wird das Angular Command Line Interface (CLI) benötigt, welches über `npm install -g @angular/cli` installiert werden kann.

Im Anschluss kann entweder mit `ng serve` eine Test-Instanz gestartet werden oder per `ng build` eine kompilierte Version des Clients erstellt werden, welche mit einem beliebigen Webserver bereitgestellt werden kann. Bei letzterer Variante ist zu beachten, dass bei jeder Aktualisierung der `config.json`-Datei der Befehl erneut ausgeführt werden muss. Die kompilierte Webseite befindet sich in `dist/Nachhilfeplattform`.

11.2.1 Ordnerstruktur

Der gesamte Quellcode des Clients befindet sich in `src`. Dabei sind in `src/assets` Grafiken, wie die Logos oder Dateicons, zu finden, in `src/app` sind die Komponenten definiert. Der Übersichtlichkeit halber sind die Komponenten wiederum in drei Teile unterteilt: `src/app/admin` umfasst alle Komponenten sowie Services, die sich lediglich auf den Administratorbereich beziehen. In `src/app/main` befindet sich die Hauptapplikation, welche genauso wie der Admin-Ordner in Unterordner für die Komponenten gegliedert ist. Weiterhin gibt es mehrere `shared`-Ordner, welche unter anderem Models, Services und weitere Code-Abschnitte, die an mehreren Stellen gebraucht werden, beinhalten.

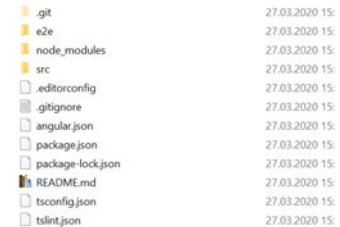


Abbildung 11.1: Öffnen der Befehlszeile in einem Ordner.

11.3 Server

Ebenso wie der Client muss der Server zuerst initialisiert werden, indem alle benötigten Pakete installiert werden. Dazu muss im `server`-Ordner derselbe Befehl `npm install` ausgeführt werden.

Daraufhin ist es möglich, den Server mit `npm start` erstmals zu starten. Sollten alle Konfigurationen abgeschlossen sein, initialisiert der Server automatisch die Datenbank mit den verwendeten Tabellen und einem Admin-Nutzer. Dieser kann sich nun registrieren mit dem Einladungscode `admin`.

Unter der URL `/explorer` auf dem Server ist ein API-Explorer von LoopBack 4 hinterlegt. Damit ist es möglich, die einzelnen Endpoints der API zu betrachten.

11.3.1 Ordnerstruktur

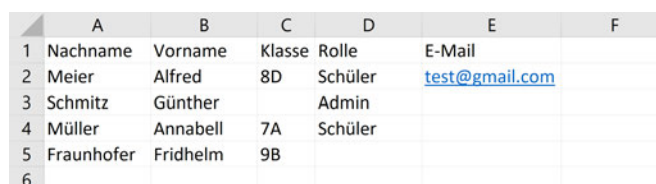
Ebenso wie beim Client ist in `src` der Quellcode zu finden. Abseits dessen gibt es in `assets` Vorlagen für Mails sowie ein Standard-Profilbild.

Der Quellcode ist in zahlreiche Ordner unterteilt, wobei `controllers`, `models` und `repositories` die Definitionen der zentralen Grundelemente von Loopback beinhalten. Des Weiteren sind in `services` verschiedene Dienste, zum Beispiel für E-Mails, Authentifizierung, Passwörter und mehr, lokalisiert. `authorization` enthält die Logik hinter der Autorisierung basierend auf Berechtigungen.

12 Verwaltung

Zur Verwaltung der Plattform steht ein Admin-Bereich unter der URL `/admin` zur Verfügung. Hier können unter anderem die Nutzer, Beiträge, Fächer und Themen verwaltet werden.

12.1 Nutzer hinzufügen



	A	B	C	D	E	F
1	Nachname	Vorname	Klasse	Rolle	E-Mail	
2	Meier	Alfred	8D	Schüler	test@gmail.com	
3	Schmitz	Günther		Admin		
4	Müller	Annabell	7A	Schüler		
5	Fraunhofer	Fridhelm	9B			
6						

Abbildung 12.1: Eine `.csv`-Datei in einem Tabellenkalkulationsprogramm mit Vorname, Nachname, E-Mail-Adresse, Rolle und Klasse.

Neben der Funktion, einzeln Nutzer hinzuzufügen, können auch mehrere gleichzeitig erstellt werden, indem eine `.csv`-Datei hochgeladen wird. Diese benötigt bestimmte Informationen über jeden Schüler als Einträge von beschrifteten Spalten (s. Abbildung 12.1). Erforderlich sind der Vor- und Nachname, optional kön-

nen Klasse, Rolle, E-Mail-Adresse und ein individueller Einladungscode angegeben werden. Sollte Letzteres leer gelassen werden, generiert die Lernplattform automatisch EinladungsCodes. Es ist zu empfehlen, zumindest teils zufällige EinladungsCodes zu verwenden, da diese direkten Zugriff auf die Lernplattform gewähren.

Wird keine E-Mail-Adresse hinzugefügt, kann der Nutzer später selber bei der Registrierung seine E-Mail-Adresse angeben. Datenschutzrechtlich dürfte dieser Weg deutlich sicherer sein.

Wichtig ist, dass die Datei in der UTF8-Zeichenkodierung gespeichert wurde. Andernfalls kann es zu Problemen aufgrund von Umlauten kommen.

Hinzufügen von 4 Nutzern bestätigen

Es werden **4** ausstehende Nutzer erstellt. Aus der Datei konnten **0** Personen nicht importiert werden.

Vorname	Nachname	E-Mail-Adresse	Klasse	Rolle	Einladungs-Code
Alfred	Meier	test@gmail.com	8D	Schüler	5823-alf.mei
Günther	Schmitz			Admin	6598-gün.sch
Annabell	Müller		7A	Schüler	3813-ann.mül
Fridhelm	Fraunhofer		9B	Schüler	3577-fri.fra

[Verwerfen](#) [Nutzer erstellen](#)

Abbildung 12.2: Übersicht der importierten Nutzer. Der Administrator kann nun überprüfen, ob die Daten korrekt sind.

Nach dem Hochladen und Verarbeiten der Datei wird eine Übersicht der anzulegenden Nutzer gegeben. Auf Bestätigung des Dialogs werden ausstehende Nutzer generiert, welche sich nachfolgend mit ihrem persönlichen Einladungscode registrieren können.

12.2 Nutzer verwalten

Der Admin-Bereich bietet verschiedene Möglichkeiten, Nutzer zu verwalten. In der Nutzerliste werden alle Nutzeraccounts sortiert gelistet und durch Anklicken werden Details bereitgestellt.

Außerdem können in der Detail-Ansicht verschiedene Aktionen durchgeführt werden: Beispielsweise können Nutzer gesperrt oder gelöscht beziehungsweise deren Posts entfernt werden.

12.3 Fächer & Themen

Ein Administrator kann Fächer hinzufügen und bearbeiten. Den Fächern lassen sich jeweils individuelle Farben zuordnen, welche die Darstellung von Posts des

jeweiligen Fachs beeinflussen.

Außerdem können Fächer über Themenbereiche verfügen. Diese können möglicherweise den Kernthemen der Lehrpläne entsprechen. Sie stehen später den Nutzern bei der Erstellung von Posts sowie bei der Auswahl ihrer Fähigkeiten zur Verfügung.

12.4 Meldungen

Alle Nutzer sind dazu in der Lage, unangemessene Posts sowie Nutzer zu melden. Der Admin-Bereich listet alle diese Meldungen auf und bietet verschiedene Interaktionen, um die gemeldeten Inhalten gegebenenfalls zu entfernen.

Unter anderem ist es möglich, den gemeldeten Post zu löschen, den Nutzer zu sperren oder den Nutzer zu löschen.

Diese Aktionen sind erreichbar, indem man den Namen oder den Titel des Posts in der Melde-Tabelle anklickt.

12.5 Feedback

Zur stetigen Verbesserung der Lernplattform verfügt die Seite über einen dedizierten Button in der Navigationsleiste, der es Nutzern ermöglicht, Feedback zu verfassen.

Das gesendete Feedback ist im Admin-Bereich einzusehen. Nachdem es entweder zur Kenntnis genommen oder bearbeitet wurde, kann es über den Button auf der rechten Seite in jeder Zeile als *fertig* markiert werden.

12.6 Klassen

Damit Schüler bessere Posts vorgeschlagen bekommen können, ist ihre Klassenstufe erforderlich. Es ist möglich, verschiedene Klassenstufen zu erstellen und in der Reihenfolge durch Drag & Drop zu verändern. In den Details zu einer Klassenstufe können die einzelnen Klassen dieser Stufe eingetragen werden.

Schüler erhalten zum Beispiel eher Anfragen aus Stufen unter ihnen sowie Angebote aus Stufen über ihnen (s. 3.3.5).

Teil VI

Schluss

13 Fazit

Um die entwickelte Lernplattform zu bewerten, lässt sich festhalten, dass sowohl alle Muss- und Sollkriterien erfüllt wurden und auch die meisten der Kannkriterien umgesetzt sind. Die Lernplattform ist vollkommen funktionsfähig.

Besonders hervorzuheben ist hier das vollkommen von Grund auf eigens entwickelte, vielfältige Nutzersystem inklusive Registrierung, Login, Einstellungen und Profilen. Dabei wurde besonderen Wert gelegt auf die eingangs definierten Kernkonzepte.

Weiterhin verfügt die Lernplattform über ein ebenfalls selbst konstruiertes Chat-System mit modernster Verschlüsselung. Die Umsetzung mittels WebSockets war eine sinnvolle Entscheidung, da es ermöglicht, Nachrichten in Echtzeit zu verschicken. Das vorzeitige Planen der Plattform half zudem enorm bei der Umsetzung, insbesondere beim komplexen Chat.

Es wurden zahlreiche originelle Konzepte eingebracht; als Beispiel ließe sich hier die Chat-Konversationen nennen, welche den Nutzern einen guten Überblick über ihre Kontakte sowie Hilfe bieten. Diese wurden passend ergänzt mit Libraries, um bereits existierende Funktionen in einem besseren Umfang einzubinden.

Auch ist die Entscheidung, einen Administrator-Bereich mit Verwaltungstools zu implementieren, sehr gewinnbringend gewesen. Ohne eine Möglichkeit, die Nutzerbasis zu moderieren, wäre die Lernplattform nicht einsatzfähig.

Das Motiv der Nachhilfe deckt die Lernplattform definitiv zufriedenstellend ab. Die gegenseitige Hilfe unter Schülern wird durch die Kommunikation auf der Plattform erfolgreich digitalisiert. Lernmaterialien können einfach miteinander geteilt werden und über den Chat können Schüler zielgerichtete und individuelle Hilfe entgegen nehmen.

Bereits durchgeführte Tests der Lernplattform innerhalb des Q2-Informatikkurses zeigten ebenfalls positive Ergebnisse. Zwar bezogen sich diese Testläufe vor allem auf die Prüfung der Funktionalität, jedoch zeigten sich auch sehr gute Kommunikationsmöglichkeiten für die Schüler.

13.1 Ausblick

Im nächsten Schritt soll die Lernplattform testweise in Klassen eingesetzt werden, um herauszufinden, ob auch unterrichtsbezogene Ziele erreicht werden können.

Für den noch besseren praktischen Einsatz bedarf es noch weiterer Ergänzungen, damit das alltägliche Nutzen noch angenehmer und sicherer wird.

Hierzu gehört beispielsweise die Möglichkeit seitens der Nutzer, individuelle Datenschutz- sowie Privatsphäre-Einstellungen zu treffen. Unter anderem müssten Nutzer wählen können, ob ihr Profil mit ihren Fähigkeiten und der Bewertung ihrer bisherigen Hilfe öffentlich einsehbar sein soll, um den entsprechenden Kernkriterien noch besser gerecht zu werden (s. 2.1 & 2.2).

Mit einher geht auch das Verfassen einer Datenschutzerklärung, um mehr Transparenz über die genauen Abläufe der Datenspeicherung im Rahmen der Lernplattform zu schaffen, wie es eines der Kernkonzepte fordert.

Dementsprechend wäre es auch angebracht, einen Cyber Security-Spezialisten die Sicherheit der API analysieren zu lassen, um Datenlecks auszuschließen.

Als weniger dringliche Erweiterung der Lernplattform bietet sich eine künstliche Intelligenz an, die automatisch Beiträge der Nutzer auf Beleidigungen, Anfeindungen oder anderweitig schädliche Inhalte überprüft. Derzeit basiert dieser Prozess auf Nutzern, die derartige Posts melden. Dabei soll jedoch eigentlich verhindert werden, dass insbesondere jüngere Schüler Solches überhaupt zu Gesicht bekommen.

Weiterhin bedarf es weiteren Tests und Verbesserungen in puncto Performance sowie User Experience. Leider war es aufgrund der COVID-19-Pandemie nicht weiter möglich, die Lernplattform intensiv und kontrolliert in realen Anwendungsgebieten einzusetzen, um verschiedene Design-Entscheidungen sowie die Geschwindigkeit zu testen und gegebenenfalls Änderungen vorzunehmen.

Diesbezüglich ist es ebenfalls wichtig, zukünftig intensiver von Beginn an Unit-Tests zu entwickeln, um stets die Funktionsfähigkeit der Applikation zu gewährleisten. Ansonsten müssen viele Bugs und Fehler erst durch Nutzer gefunden werden, was den Entwicklungsprozess erheblich verlangsamt.

Des Weiteren muss das Admin-Panel in seinem Funktionsumfang noch erweitert werden. Unter anderem müsste es noch ein Dashboard mit den aktuellen Informationen zur Plattform geben. Auch sollte es eine grafische Oberfläche mit den Optionen geben, die derzeit ausschließlich über die Datei `config.json` geändert werden können.

14 Libraries und Frameworks

Angular <https://angular.io/docs>

Angular Material <https://material.angular.io/>

CSV-Parser <https://github.com/mafintosh/csv-parser>

Icons <https://github.com/file-icons/icons>

Jasmine <https://jasmine.github.io/>

JsonWebToken <https://github.com/auth0/node-jsonwebtoken>

Karma <https://karma-runner.github.io/4.0/index.html>

Loopback 4 <https://loopback.io/doc/en/lb4/>

Multer <https://github.com/expressjs/multer>

Node.js <https://nodejs.org/dist/latest-v12.x/docs/api/>

Nodemailer <https://github.com/nodemailer/nodemailer>

NPM <https://docs.npmjs.com/>

Request <https://github.com/request/request>

RxJS <https://rxjs.dev/>

Sharp <https://github.com/lovell/sharp>

15 Literatur- und Linkverzeichnis

- [1] Bitkom Research GmbH, ARIS Umfrageforschung GmbH, BITKOM e.V. (2015). *Digitale Schule – vernetztes Lernen. Ergebnisse repräsentativer Schüler- und Lehrerbefragungen zum Einsatz digitaler Medien im Schulunterricht*. Abgerufen 17. Februar, von <https://www.bitkom.org/sites/default/files/pdf/noindex/Publicationen/2015/Studien/Digitale-SchulevernetztesLernen/BITKOM-Studie-Digitale-Schule-2015.pdf>
- [2] Borrelli, P. (2019, August 29). *Angular vs. React vs. Vue: A performance comparison*. Abgerufen 21. März 2020, von <https://blog.logrocket.com/angular-vs-react-vs-vue-a-performance-comparison#dommanipulation>
- [3] Der Hessische Beauftragte für Datenschutz und Informationsfreiheit. (2019, Juli 9). *Stellungnahme des Hessischen Beauftragten für Datenschutz und Informationsfreiheit zum Einsatz von Microsoft Office 365 in hessischen Schulen* [Pressemeldung]. Abgerufen 27. März 2020, von <https://datenschutz.hessen.de/pressemitteilungen/stellungnahme-des-hessischen-beauftragten-f%C3%BCr-datenschutz-und>
- [4] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Abgerufen 20. März 2020, von https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [5] Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token*. Abgerufen 28. März 2020, von <https://tools.ietf.org/html/rfc7519>
- [6] Klemm, K., & Hollenbach-Biele, N. (2016). *Nachhilfeunterricht in Deutschland: Ausmaß - Wirkung - Kosten*. Abgerufen 17. Februar 2020, von https://www.bertelsmann-stiftung.de/fileadmin/files/Bst/Publicationen/GrauePublicationen/Nachhilfeunterricht_in_Deutschland_160127.pdf
- [7] MariaDB Corporation Ab. (o. J.). *MariaDB versus MySQL - Compatibility*. Abgerufen 28. März 2020, von <https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/>
- [8] Moodle Pty Ltd. (2019, Dezember 17). *About Moodle - Our Mission and Values*. Abgerufen 17. Februar 2020, von <https://moodle.com/about/>

- [9] Nutter, M. (2016, Februar 4). *Modern Web Development*. Abgerufen 30. März 2020, von <https://hackernoon.com/modern-web-development-bf0b2ef0e22e>
- [10] SocketIO. (o. J.). *Engine.io: the realtime engine*. Abgerufen 23. März 2020, von <https://github.com/socketio/engine.io/blob/master/README.md>
- [11] solid IT GmbH. (o. J.). *MariaDB vs. MySQL vs. PostgreSQL Comparison*. Abgerufen 28. März 2020, von <https://db-engines.com/en/system/MariaDB%3bMySQL%3bPostgreSQL>
- [12] Warncke, E. (2020, Februar 15). *Benchmarking Node.js: v12 vs v10, v8, v6, v4 and Go*. Abgerufen 26. März 2020, von <https://medium.com/@ersun.warncke/benchmarking-node-js-v12-vs-v10-v8-v6-v4-and-go-89d28eee603d>

16 Eigenständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche gekennzeichnet.

Ort, Abgabedatum

Unterschrift des Verfassers