



Erstprüfer:

Zweitprüfer:



Kurzfassung:

Die vorliegende Bachelor-Thesis beschäftigt sich mit den Benutzeranforderungen, der technischen Konzeption und der prototypischen Entwicklung einer Webapplikation unter Einsatz von verschiedenen Werkzeugen der Informationstechnologie, z.B. der Web-Development-Plattform Angular¹. Das Konzept der Applikation richtet sich in erster Linie an Kinder (in Abhängigkeit des Alters in verschiedene Zielgruppen aufgeteilt) und stellt eine potentielle Hilfestellung im Erlernen und Verbessern der Deutschkenntnisse dar.

Der Kern der Applikation besteht darin, den Nutzern die Möglichkeit zu bieten, unter Anwendung der Methode des kreativen Schreibens Kurzgeschichten zu verfassen und so die literarischen Fähigkeiten und die Lese- und Rechtschreibkompetenz potentiell zu verbessern.

Neben der Anwendung skalierbarer Web-Development-Technologien wird der Service einer Künstlichen Intelligenz (KI) zum Zweck der Text-zu-Bild-Generierung eingesetzt, um den folgenden Gamification-Mechanismus zu erzielen: Der Nutzer kann sich für jeden verfassten Abschnitt seiner Kurzgeschichten vier Bilder generieren lassen, die den Inhalt des geschriebenen Textes visualisieren.

Diese Form der Live-Illustration ist an Märchenbücher² sowie an Kinderbücher³ angelehnt und soll die literarische Welt durch die Visualisierung des Textes lebendiger erscheinen lassen und damit die Motivation der Schüler, Texte zu verfassen, erhöhen. Dabei hängt die Qualität der Bilder bei diesem Prozess davon ab, welche Qualität der verfasste Text aufweist (Präzision der Formulierung, Struktur, Korrektheit der Grammatik, Kreativität), was dazu anregen kann, die Qualität der Texte zu verbessern.

Die vorliegende wissenschaftliche Arbeit legt damit den Grundstein für den praktischen Einsatz der vorgestellten technischen Werkzeuge innerhalb des Schulfaches Deutsch und bei entsprechender Anpassung auch für den Fremdsprachenunterricht.

Schlagwörter

Kreatives Schreiben, Web-Development, Angular, TypeScript, Künstliche Intelligenz, Text-zu-Bild-Generierung, Technische Infrastruktur von Schulen, PISA-Studien.

¹ Wilken (2018). S. 1.

² z.B. Jacob und Wilhelm Grimm (1825). Kinder- und Hausmärchen, Kleine Ausgabe.

³ z.B. Ali Mitgutsch (1989). Meine Seeräuberinsel.

Abstract:

This bachelor thesis deals with the requirements analysis, the technical conception and the prototypical development of a web application using different computer technology tools, amongst which is the web development platform Angular. The concept of the application is primarily aimed at children (divided into different target groups depending on age) and represents a potential aid in learning and improving German language skills.

The core of the application is to offer users the opportunity to write short stories by means of the creative writing method, thus enhancing literary skills and reading and spelling competencies.

In addition to the utilization of scalable web-development-technologies, an artificial intelligence service is integrated for the purpose of text-to-image generation to achieve the following gamification mechanism: Users are allowed to have an image generated for each written section of their short stories, visualizing the content of the written text.

This form of live illustration is based on fairy tale books and children's books and is intended to make the literary world appear more vivid through the visualization of the text and thus increase the students' motivation to write texts. The quality of the pictures in this process depends on the quality of the written text (precision of wording, structure, correctness of grammar, creativity), which can encourage the user to improve the quality of the text.

This academic work lays the foundation for the practical use of the technical tools presented within the school subject of German and, with appropriate adaptation, also for teaching different languages.⁴

Keywords

Creative Writing, Web-Development, Angular, TypeScript, Artificial Intelligence, Text-to-Picture-Generation, Technical Infrastructure of schools, PISA-Studies.

⁴ Die englische Übersetzung wurde in Teilen mit Hilfe von <https://www.deepl.com/translator> generiert sowie von dem Autor der Thesis auf Fehler überprüft.

Danksagung:

Der Elemente vier,
Der Wege schier, unendlich viele
in die Gier, In die Wut und in den Hass,
doch wenn du einen Text verfasst,

Der deine Bilderwelt vereint,
In flotten Sprüngen durch die Zeit,
Mit Kafka durch die Gänge eilt,
Auf Hesses Stufen nicht verweilt,

Dank Platon auch dem Tod verzeiht,
In Prousts Salon den Geigen lauscht,
Durch die Carusos Stimme haucht,
Und selig von Unendlichkeit,
Ein Stück schon vor dem Ende teilt,

Wie May, der hinter Gittern wacht,
Und malerische Welten schafft,
Den Pfaden einen Sinn vermacht,
Der Hennens Zauber neu entfacht.

Dann wird dir klar sein wer du bist,
ein Mensch aus Fleisch und Blut wie ich,
Der traute Seelen um sich weiß,
Die ihn verlassen mit der Zeit,
Damit er auch allein bereit und würdig neue Sätze schreibt.

Mannigfaltiger Dank gilt...

... den genannten Schriftstellern.

... Vater, Mutter, Max, Jenn, Dorothea Feth, Eduard Feth, Martha Goth, Anton Goth.

... Don, Rick, Markus, Karl-Heinz, Raphael, Lee und den 24 Räufern.

... David und WG, Sven und Familie, Anna und Verein.

... Matthias Adler, Sibel Aktikkalmaz.

... Prof. Dr.-Ing. Oliver Kretschmar, Willi Schorrig und Stephan Dauscher (Oberstudienrat).

Ehrenwörtliche Erklärung:


Hiermit versichere ich, Clemens Feth, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel: „Konzeption und prototypische Entwicklung einer Webapplikation für Kinder zum Verfassen von Kurzgeschichten“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich erkläre hiermit, dass ich alle Textpassagen, die durch den Einsatz von KI-gestützten Schreibwerkzeugen generiert wurden, unter Nennung des eingesetzten KI-gestützten Schreibwerkzeugs gekennzeichnet habe.

Bei der Erstellung der vorgelegten Studienleistung habe ich durchgehend eigenständig und beim Einsatz KI-gestützter Schreibwerkzeuge steuernd gearbeitet.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.

Stuttgart, den 23.02.2024



Clemens Feth

Hinweis:

Die in dieser wissenschaftlichen Arbeit verwendeten Personenbezeichnungen beziehen sich immer gleichermaßen auf männliche, weibliche und diverse Personen.

Auf eine doppelte/dreifache Nennung und auf gegenderte Bezeichnungen wird zugunsten einer besseren Lesbarkeit verzichtet.

Projekt:

<https://gitlab.mi.hdm-stuttgart.de/cf061/kurzgeschichten-thesis>.

Demonstrationsvideos:

<https://drive.google.com/drive/folders/1bnhNYNmstiXmFQdcxtpXt4wAdrznNJZf?usp=sharing>.

Inhaltsverzeichnis

Vorwort.....	10
1 Einführung.....	11
1.1 Motivation.....	11
1.1.1 Lesen und Schreiben: grundlegende Kulturtechniken.....	11
1.1.2 Stand der Digitalisierung der Schulen in Deutschland.....	14
1.2 Fragestellungen.....	17
1.3 Ziele der Arbeit.....	18
1.4 Aufbau der Arbeit.....	18
2 Anforderungsanalyse.....	21
2.1 Zielgruppen.....	21
2.1.1 Weiterführende Schulen: 10+ Jahre alt.....	21
2.1.2 Grundschulkind: 6-10 Jahre alt.....	21
2.1.3 Vorschulkind: 3-6 Jahre alt.....	22
2.2 Bedarfsanalyse.....	22
2.2.1 Kreatives Schreiben.....	22
2.2.2 Digitales Schreiben.....	26
2.2.3 Gamification.....	27
2.2.4 Technische Infrastruktur von Schulen: Ist-Status.....	28
2.2.5 Lokales Hosting vs. Cloud-Hosting.....	31
2.2.6 Datenschutz und Datensicherheit.....	34
2.3 Benutzer-Anforderungen.....	37
2.3.1 Funktionale Benutzer-Anforderungen.....	37
2.3.1.1 Die Liste mit den Templates anzeigen lassen.....	38
2.3.1.2 Kurzgeschichten verfassen.....	38
2.3.1.3 Die Rechtschreibprüfung durchführen lassen.....	38
2.3.1.4 Bilder zum Text generieren lassen.....	38
2.3.2 Nicht-funktionale Benutzer-Anforderungen.....	39
2.3.2.1 Benutzerfreundlichkeit.....	39
2.3.2.1.1 Leicht verständliche Sprache und erklärender Charakter.....	39
2.3.2.1.2 Responsiveness.....	39
2.3.2.1.3 Barrierefreiheit.....	40
2.3.2.2 Text-zu-Bild-Generator-Dienst.....	40
2.3.2.2.1 Schutz vor nicht jugendfreien Inhalten.....	40
2.3.2.2.2 Qualität der generierten Bilder.....	40
2.3.2.2.3 API-Zugang.....	40
2.3.2.2.4 Kosten.....	40
2.3.2.3 Dynamisches Benutzerinterface.....	41
2.3.2.4 Nutzerzahl.....	41

2.3.2.5 Anpassbarkeit in Abhängigkeit des Alters des Nutzers (Laufzeit).....	42
2.3.2.6 Anpassbarkeit und Erweiterbarkeit der Features (Entwicklung).....	42
3 Technische Konzeption.....	43
3.1 Client-Server-Architektur.....	43
3.2 Webapplikationen.....	44
3.2.1 Software-Architektur.....	44
3.2.2 Kommunikationsprotokoll HTTP.....	44
3.2.3 Deployment.....	45
3.2.4 Performanz.....	45
3.3 Backend.....	45
3.3.1 Web-Server-Technologien.....	45
3.3.1.1 Node.js.....	46
3.3.1.2 Apache.....	47
3.3.2 Datenbanken.....	48
3.4 System-Anforderungen des Prototypen (Proof-of-Concept).....	49
3.4.1 Prozesse (UML-Diagramme).....	49
3.4.2 Anwendungsfalldiagramm.....	50
3.4.3 Aktivitätsdiagramm.....	51
3.4.4 Arbeitspakete.....	53
4 Technische Grundlagen.....	58
4.1 Client (Frontend).....	58
4.1.1 Web-Development-Plattform Angular.....	58
4.1.1.1 TypeScript.....	58
4.1.1.2 Komponenten.....	59
4.1.1.3 Das reaktive Programmierparadigma.....	60
4.1.1.4 Domain-specific Language (DSL).....	62
4.1.1.5 RxJS.....	62
4.1.1.6 Node-Package-Manager (npm).....	62
4.2 Leonardo.AI (Text-zu-Bild-Generator).....	63
5 Ergebnisse.....	65
5.1 Dokumentation.....	65
5.2 Demonstrationsvideos.....	65
5.3 Templates (angefangene Kurzgeschichten).....	66
5.4 Implementierung.....	66
Aspekt 1: Div-Struktur und @media-flags (Responsiveness).....	66
Aspekt 2: Grafisches Design.....	68
Aspekt 3: HTML-Alt-Attribut (Barrierefreiheit: visuelles Handicap).....	68
Aspekt 4: Model-View-Controller-Prinzip (Dynamisches Benutzerinterface).....	69
6 Fazit & Ausblick.....	70

6.1 Fazit.....	70
6.2 Ausblick.....	74
6.2.1 Vorschläge für Nutzertests.....	74
6.2.2 Vorschläge für weitere Features.....	75
6.2.2.1 Reglementierte Einbindung intelligenter Chatbots.....	75
6.2.2.2 PDF-Exportfunktion.....	76
6.2.2.3 Gamification (Verstärkung).....	76
6.2.2.3.1 Kooperative Mini-Spiele.....	76
6.2.2.3.2 Kreieren von Sammelkarten.....	77
6.2.2.3.3 Text-zu-Musik-Generator (Künstliche Intelligenz).....	79
6.2.2.4 Barrierefreiheit (Verbesserung).....	79
6.2.2.4.1 Text-to-Speech-Support.....	79
6.2.2.4.2 Speech-to-Text-Support.....	79
Literaturverzeichnis.....	80
Anhang.....	85
A Glossar.....	85
B Abbildungsverzeichnis.....	86

Vorwort

„Es gibt keine großen Entdeckungen und Fortschritte, solange es noch ein unglückliches Kind auf Erden gibt.“ - Albert Einstein (*14. März 1879; † 18. April 1955).

Der deutsche Physiker Albert Einstein unterzeichnete am 11. April 1955 zusammen mit zehn weiteren namhaften Wissenschaftlern das sogenannte Russell-Einstein-Manifest zur Sensibilisierung der Menschen für die Abrüstung. Damals war der zweite Weltkrieg noch nicht lange her und dessen Schrecken war den Menschen noch mehr vor Augen als heute.

Leider sind die Kriege und Krisenherde auf der Erde in den Jahren danach aber nicht weniger geworden, sondern setzen sich bis heute ununterbrochen fort. Dabei sind die allermeisten Menschen, die auf dieser Erde leben, friedlich eingestellt. Das ist natürlich nur eine Vermutung, die auf der Erfahrung des Zusammentreffens mit den Menschen beruht, die ich bislang getroffen habe. Jedenfalls stellt sich, auch wenn diese These nicht zutrifft, für mich die Frage, was getan werden kann, damit ein dritter Weltkrieg eine Schreckensvision bleibt und nicht Wirklichkeit wird, und damit auch die bereits existierenden Kriege und Krisenherde beendet werden. Ich kenne die Antwort auf diese Frage nicht. Ich kann nur sagen, dass ich an die positive Kraft und Phantasie des einzelnen Menschen glaube und aus diesem Grund eine Thesis verfasst habe, die Kindern, letztlich aber jeder Person die Möglichkeit gibt, ihre Erfahrungen und Ideen aufzuschreiben und mit anderen zu teilen.

Bücher mit ihren Buchstaben, Bildern und Geschichten waren und sind seit meiner Kindheit ein ständiger und zuverlässiger Begleiter für mich. Sie füllen Räume, sie füllen aber auch mein und unser aller Inneres mit Wissen und Informationen, mit Gefühlen und Kreativität. Das Schreiben hat sich innerhalb der heutigen Gesellschaft allerdings schon längst in Richtung von digitalen, sozialen Plattformen wie z.B. X (früher Twitter) verlagert. Leider bilden sich in solchen Plattformen inzwischen polarisierte, teils verfeindete Gruppen, die sich gegenseitig beleidigen und denen es darum geht, „Recht zu haben“ bzw. eine so oder so gestaltete ideologisierte Meinung zum Besten zu geben. Da ich scheinbar gerade demselben Schema ver falle, beende ich das Vorwort und danke nochmal allen Menschen, die mich nicht aufgegeben haben und mir gezeigt haben, dass Selbsthass keine Liebe ist.

1 Einführung

1.1 Motivation

1.1.1 Lesen und Schreiben: grundlegende Kulturtechniken

Dass das gute Beherrschen der Sprache eine der besten Voraussetzungen ist, Lesen und Schreiben zu lernen, und dass die erfolgreiche Anwendung dieser Kulturtechniken auch mittel- und langfristig für gute Leistungen in der Schule und in den darauf folgenden Lebensabschnitten der Kinder mitverantwortlich ist, wurde mehrfach nachgewiesen.⁵

Dabei ist das Lesen- und Schreibenlernen in der Grundschule ein komplexer und schwieriger Prozess. Viele Faktoren spielen dabei eine Rolle, u.a. der Grad der phonologischen Bewusstheit und deren Förderung. Das heißt konkret, dass das Sprachliche von den Kindern verarbeitet werden muss, indem sie verstehen lernen, dass Sätze aus Wörtern, Wörter aus Silben und Silben aus angeordneten Lauten, sprich Buchstaben, bestehen.⁶

Wie aber steht es um das Verständnis der geschriebenen Sprache unter deutschen Schülern?

Die PISA-Studien (Programm for International Student Assessment) untersuchen regelmäßig die Leistung der Schüler in den funktionalen Kern-Bildungsbereichen Mathematik, Naturwissenschaften und Lesen, indem die Fähigkeit geprüft wird, das bestehende Wissen auf diverse Problemstellungen anzuwenden. Die dazugehörige Studie aus dem Jahr 2022 zeigt, dass die Lesekompetenz deutscher Fünfzehnjähriger im OECD-Vergleich (The Organisation for Economic Co-operation and Development) zwar nur knapp unterhalb des Durchschnitts liegt. Allerdings liegt der Mittelwert der von den deutschen Schülern erreichten Leistungen des Jahres 2022 im Vergleich zu den gemessenen Leistungen der geprüften deutschen Schüler in den PISA-Studien 2018 und 2012 signifikant niedriger.⁷

Seit 2011 werden zudem deutsche Grundschüler, die sich zeitlich am Ende der vierten Jahrgangsstufe befinden, in regelmäßigen Abständen vom Institut zur Qualitätsentwicklung im Bildungswesen (IQB) getestet, ob sie die altersgerechten bzw. gewünschten Kenntnisse in den Gebieten des Lesens, Schreibens und der Mathematik besitzen und anwenden können. Die Kenntnisse im Fach Deutsch sind dabei aufgeteilt in die Bereiche:

- Lesen sowie das Verständnis von und das Arbeiten mit Texten und Medien.
- Sprechen und Zuhören.

⁵ Huettig et al. (2018). S. 275-277.

⁶ Küspert (1998). S. 68-70.

⁷ Lewalter et al. (2023). S.4, S.19.

- Schreiben (und Orthografie), Sprachverständnis und Sprachgebrauch.

In den Jahren 2021 und 2022 wurden neue Ergebnisse veröffentlicht, die mit den Befunden des Negativtrends, der aus den genannten PISA-Studien hervorgeht, scheinbar korrelieren. Seit dem letzten Test im Jahr 2016 ist ein negativer Trend bei den deutschlandweiten Mittelwerten der Leistungen der Kinder am Ende der 4. Jahrgangsstufe in den Kompetenzbereichen des Faches Deutsch zu verzeichnen. Im Lesen sank der Mittelwert im Jahr 2021 im Vergleich zu 2016 klar um 22 Punkte, was zeitlich circa ein Drittel eines Jahres Unterschied ausmacht, gemessen an den Kompetenz-Fortschritten, die ein Schüler dieses Alters innerhalb dieser Zeit machen sollte (nachfolgend „Lernzeit“). Der Mittelwert der Kompetenz des Zuhörens fiel im genannten Zeitraum sogar um 28 Punkte, der Mittelwert in der Kompetenz der Orthographie (Rechtschreibung und Sprachverständnis) um 27 Punkte, was auf circa ein halbes Schuljahr Lernzeit abzubilden ist. Der Vergleich 2011 zu 2016 zeigte ebenso einen Negativtrend, was die Differenz der Mittelwerte von 2011 und 2021 vergrößert.⁸

Für den Kontext dieser Thesis ist der Kompetenzbereich des Schreibens und der Orthographie (Rechtschreibung und Sprachverständnis) hervorzuheben. Darüber kann zusammenfassend gesagt werden, dass der IQB-Bildungstrend 2011-2021 zu dem Ergebnis kommt, dass deutschlandweit mehr als 20 Prozent der Kinder am Ende der vierten Jahrgangsstufe die Mindeststandards der Bildungspläne innerhalb des genannten Bereichs nicht erreichen, mit steigender Tendenz. Während bei den Tests des Jahres 2011 „nur“ ca. 12% der Kinder das dem definierten Bildungsstandard dieser Jahrgangsstufe entsprechende Leistungsniveau nicht erreichen konnten, waren es im Jahr 2016 schon ca. 22%.⁹

Die Ursachen für die Negativ-Trends bei den Rechtschreibleistungen, dem Lesen und dem Sprachverständnis von Kindern sind vielschichtig: Soziale Herkunft¹⁰, Zuwanderung¹¹ und motivational-emotionale Negativ-Trends¹² sind mögliche Faktoren. Die zwei erstgenannten Gründe können an dieser Stelle aus zeitlichen Gründen nicht näher beleuchtet werden.

Der dritte Punkt der „motivational-emotionalen Negativ-Trends“ hinsichtlich des Schulfaches Deutsch stellt sich aber im Kontext des in der Thesis entwickelten Konzepts der Webapplikation als wichtig heraus. Dieses Konzept schließt nämlich (neben den Vorteilen der digitalen Verwaltbarkeit der Texte - auch für die Lehrer) den Versuch mit ein, durch die Integration der Methoden des kreativen Schreibens ([Kapitel 2.2.1](#)), des digitalen Schreibens ([Kapitel 2.2.2](#)) sowie der Gamification ([Kapitel 2.2.3](#)) näher an die Lebenswelt von Kindern und Jugendlichen zu kommen und den Schulunterricht dadurch interessanter zu machen.

⁸ Wittig und Schneider. (2022) S. 82-90. Referenz: (Herausgeber) Stanat et al.

⁹ Kuhn (2021). Deutsches Schulportal der Robert-Bosch-Stiftung.

¹⁰ Sachse et al. (2022) Kapitel 7. S. 151-179. Referenz: Stanat et al.

¹¹ Henschel et al. (2022) Kapitel 8. S. 181-219. Referenz: Stanat et al.

¹² Schneider et al. (2022) Kapitel 9. S. 221-231. Referenz: Stanat et al.

In der Analyse des genannten IQB-Bildungstrends 2011-2021 wird dargelegt, dass eine korrekte Rechtschreibung und ein gutes Sprachverständnis zu den Basiskompetenzen zählen, die notwendig sind, damit die Kinder sowohl den Schulabschluss als auch das darauf folgende Studium bzw. die Ausbildung erfolgreich meistern können. Insofern könnte es als ein schlechtes Zeichen angesehen werden, dass bereits in der Primarstufe bei den Viertklässlern neben den Leistungsdefiziten auch ein signifikanter Negativ-Trend, was das sogenannte Selbstkonzept und das Interesse im Fach Deutsch betrifft, nachgewiesen werden konnte.

Das mangelnde Interesse der Kinder am Lesen und Schreiben und die zu geringen Motivationsanreize, dieses Interesse anzuheben, konnten dementsprechend als Mitursache für die Negativ-Trends der Lese- und Schreibkompetenzen der Kinder identifiziert werden.¹³

Das Selbstkonzept nach Rogers bezeichnet dabei z.B. die Meinungen, die das Kind von sich selbst hat und die sich in Abhängigkeit der (negativen und positiven) Erfahrungen, die das Kind im Laufe seines Lebens macht, entwickeln.¹⁴ Der genannte Negativ-Trend des Selbstkonzepts bedeutet also, dass das Selbstvertrauen der Schüler, was das Vertrauen in die eigenen Fähigkeiten in den genannten Kompetenzbereichen des Schulfachs Deutsch angeht, durchschnittlich betrachtet innerhalb der Zeitspanne zwischen den genannten Tests des IQB-Bildungstrends gesunken ist.

Das sinkende Selbstkonzept der Kinder steht dem IQB-Bildungstrend zufolge, wie bereits angeführt, mit dem sinkenden Interesse an dem Schulfach Deutsch in enger Korrelation. Bei der sukzessiven Vermittlung von Lese- und Schreibstrategien sollte allerdings der Grad der Motivation der Kinder, produktiv am Unterricht teilzunehmen, nicht vernachlässigt werden. Das Entstehen der negativen Einstellung und der angstbesetzten Hemmnisse der Kinder gegenüber dem Lesen und Schreiben sollte von Anfang an vermieden werden.¹⁵

Letzteres im Blick werden in der Bedarfsanalyse die Themen „kreatives Schreiben“, „digitales Schreiben“ und „Gamification“ ([Kapitel 2.2.1](#) bis [2.2.3](#)) eingeführt.

¹³ Schneider et al. (2022). S. 221–231. Referenz: Stanat.

¹⁴ Rogers (1961). Kapitel: Personality Change in Psychotherapy, S. 13.

¹⁵ Jambor-Fahlen. (2018). S. 11.

1.1.2 Stand der Digitalisierung der Schulen in Deutschland

Solange die Schulträger sowie die Kultusministerien sich bei der Ausstattung der Lehrkräfte, der Schüler, und der Infrastruktur der Schulen generell auf die analogen Werkzeuge Kreide, Tafel, Stifte, Füller sowie gedruckte Bücher beschränken konnten, sind sie, auch im Blick auf die in der Regel existierende Lernmittelfreiheit, nicht unbedingt an finanzielle und fachlich-didaktische Grenzen gestoßen.

Dieses Kapitel soll zeigen, dass sich diese Lage durch den „Digitalisierungs-Druck“, der sich durch die Corona-Pandemie und den dadurch forcierten Distanzunterricht noch verschärft hat, verändert hat. In den Blick genommen werden die Haltung der deutschen Politik, der Grad der qualitativen und quantitativen Nutzung von digitalen Werkzeugen und der Zustand der digitalen Ausstattung der Schulen.

Die deutsche Bundesregierung und die Länder der Bundesrepublik haben in der Verwaltungsvereinbarung „DigitalPakt Schule 2019 bis 2024“ (fortan Digitalpakt) dringenden Handlungsbedarf festgestellt. Folgender Wortlaut findet sich in der Präambel:

„Die fortschreitende Digitalisierung aller Lebensbereiche stellt eine zentrale strukturelle Herausforderung für die Bildung junger Menschen am Bildungsstandort Deutschland dar. Es ist eine der großen Zukunftsaufgaben, die Schülerinnen und Schüler an den Schulen in Deutschland umfassend auf die Digitalisierung in allen Lebensbereichen vorzubereiten.

Die Bundesregierung und die Regierungen der Länder der Bundesrepublik Deutschland einschließlich der Kommunen arbeiten bei dieser Zukunftsaufgabe zusammen und setzen einen abgestimmten Innovationsimpuls. Damit sollen die bestehenden Entwicklungen an den Schulen entscheidend unterstützt werden, um die Voraussetzungen für Bildung in der digitalen Welt bundesweit und nachhaltig spürbar zu verbessern.“¹⁶

Das heißt konkret, dass der Bund die Länder, d.h. letztlich Kommunen und Gemeinden, bei deren eigenen Projekten finanziell unterstützen kann, sofern die Länder dabei auch einen finanziellen Eigenanteil stemmen. Die genannten Projekte sollen außerdem den Zweck verfolgen, die digitale Infrastruktur der Schulen zu stärken.¹⁷

Wie aber geht es voran mit der Umsetzung des Digital-Pakts?

¹⁶ Bundesministerium für Bildung und Forschung. „DigitalPakt Schule 2019-2024“. S. 1-3.

¹⁷ Bundesministerium für Bildung und Forschung. „DigitalPakt Schule 2019-2024“. S. 1-10.

Entwicklung Mittelbindung und Mittelabfluss: Basis-DPS und ZV Administration

Stand: 30.06.2023; Angaben in Mio. Euro; alle Länder

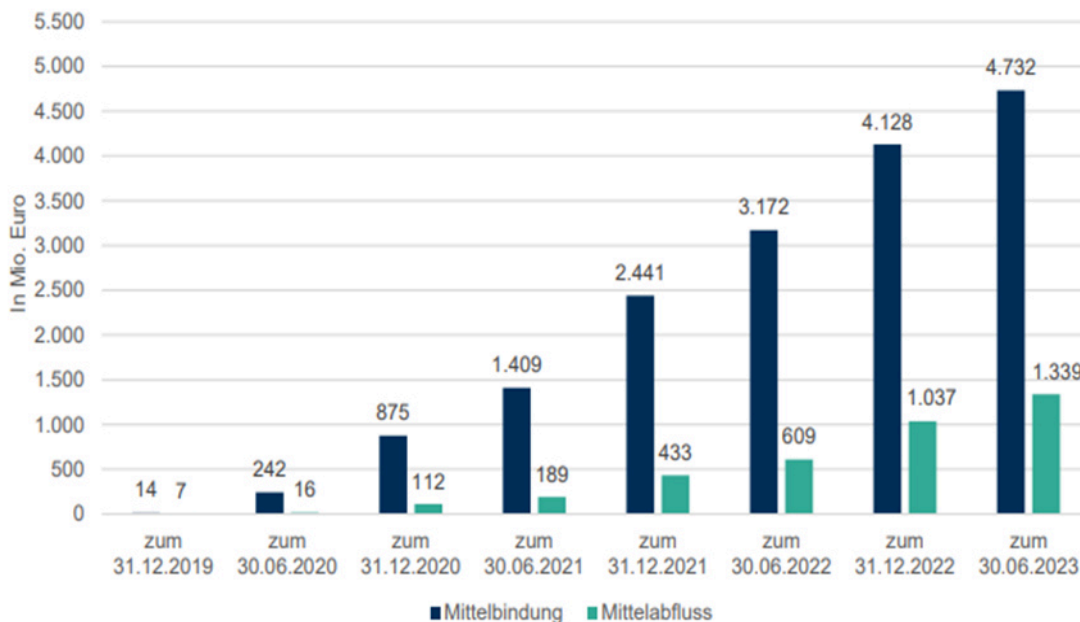


Abbildung 1: Entwicklung der Mittelbindung und des Mittelabflusses aus dem Sondervermögen des Bundes für den „DigitalPakt Schule 2019-2024“. Quelle: Bundesministerium für Bildung und Forschung.

Die dargestellte Graphik veranschaulicht den Mittelabfluss aus dem Sondervermögen des Bundes von Dezember 2019 an. Hierbei handelt es sich um den Vorgang der finanziellen Abrechnung zwischen dem Bund und den Ländern. Deswegen lässt eine Analyse der Graphik keinen eindeutigen Rückschluss darüber zu, auf welche Art und Weise die abgerechneten finanziellen Mittel eingesetzt wurden und inwiefern oder ob dieser Einsatz zu den oben genannten Zielen produktiv beigetragen hat. Des Weiteren werden die bereitgestellten finanziellen Mittel nur zaghaft genutzt. Die negative Interpretation dessen könnte lauten, dass die Länder und Kommunen den innerhalb des Digitalpakts definierten Zielen der Digitalisierung und Verbesserung der schulischen IT-Infrastruktur hinterherhinken. Andererseits gibt es Länder, die zum Teil ihre eigenen Projekte bereits selbst vorfinanziert haben, und deswegen zu den dargestellten Zeitpunkten womöglich schlichtweg nicht auf weitere finanzielle Mittel angewiesen waren.¹⁸

¹⁸ Bundesministerium für Bildung und Forschung.
<https://www.digitalpaktschule.de/de/die-finanzen-im-digitalpakt-schule-1763.html>.
 Datum des Abrufs: 28.12.2023.

Da die dargestellten „blanken“ Zahlen dementsprechend keine durchschlagende Aussage über den Zustand bzw. einen etwaigen qualitativen und quantitativen Fortschritt der digitalen Infrastruktur deutscher Schulen zulassen, sollen nachfolgend einige wesentliche Forschungsergebnisse zu der genannten Thematik aufgezählt werden.

- Deutliche Unterschiede zwischen den deutschen Schulen sind im Grad der Ausstattung und der digitalen Orientierung, d.h. der Form der jetzigen Nutzung und der Planung der zukünftigen Nutzung von digitalen Werkzeugen, festzustellen. Dabei verfügen ca. 60% der Schulen in Deutschland über eine kleine bis sehr kleine digitale Ausprägung und Ausstattung.¹⁹ Derweil nutzen vor allem Lehrer an Gymnasien (20% der Lehrer) und Gesamtschulen (29%) digitale Lernformate auf vielfältige Art und Weise, Lehrer an Grundschulen (2%) dagegen fast gar nicht.²⁰
- Die ICILS-Studien (International Computer and Information Literacy Studies, www.iea.nl/studies/icils) aus den Jahren 2013 und 2018 haben gezeigt, dass die Nutzung digitaler Medien im Unterricht in deutschen Schulen im internationalen Vergleich signifikant unterdurchschnittlich ist.²¹
- Der Grad der Verfügbarkeit von internetbasierten Anwendungen, die zum gemeinschaftlichen Arbeiten dienen können, ist für Schüler sowie Lehrkräfte in Deutschland sehr niedrig: Er beträgt lediglich 16,5 Prozent, während der internationale Mittelwert 63.1% beträgt. Dieser Wert liegt in Finnland sowie Dänemark bei ca. 97%, während der Wert in den USA bei ca. 93% liegt.²²

Die dargestellten Forschungsergebnisse weisen darauf hin, dass in Deutschland ein Nachholbedarf bei den für das digitale Lernen von Kindern und Jugendlichen notwendigen Rahmenbedingungen besteht.

Das Kapitel der Einführung ([Kapitel 1](#)) hat den Kontext der vorliegenden Thesis dargelegt, indem zunächst das zentrale Thema, d.h. die Problemstellung des Negativ-Trends der Kompetenzen von Kindern und Jugendlichen in den Bereichen Lesen, Schreiben und Orthografie belegt wurde ([Kapitel 1.1.1](#)). Zudem wurden die bildungspolitischen Entwicklungen der letzten Jahre, was das Thema Digitalisierung an deutschen Schulen betrifft, thematisiert ([Kapitel 1.1.2](#)). In den nächsten beiden Kapiteln werden die Fragestellungen ([Kapitel 1.2](#)) und die Ziele der Thesis ([Kapitel 1.3](#)) konkretisiert. Danach wird der strukturelle Aufbau des nachfolgenden Teils der Thesis ([Kapitel 1.4](#)) erklärt.

¹⁹ Mußmann et al. (2021). S. 83-93.

²⁰ Schmid et al. (2017). S. 29-33.

²¹ Autorengruppe Bildungsberichterstattung (2020). S. 253-260.

²² Eickelmann et al. (2019) S. 14-16.

1.2 Fragestellungen

Wie bereits dargelegt werden konnte, befindet sich der Prozess der Digitalisierung der deutschen Schulen im Vergleich zu anderen Industrienationen noch an seinem Anfang ([Kapitel 1.1.2](#)). Als Kontrast dazu wurde der klare Wille der deutschen Bundesregierung sowie der Landesregierungen dokumentiert, große finanzielle und politische Anstrengungen zu unternehmen, um den Schulen zu einer Verbesserung der IT-Infrastruktur zu verhelfen. Diese IT-Infrastruktur solle des Weiteren so eingesetzt werden, dass die Schülerinnen und Schüler gut vorbereitet sind auf eine (Arbeits-)Welt, in der das Beherrschen der produktiven und problemlösungsorientierten Anwendung von digitalen Endgeräten und Technologien ein wichtiger Faktor für persönlichen und wirtschaftlichen Erfolg sei.

Die Tatsache, dass die deutsche Bundesregierung und die Landesregierungen mit dem Digitalpakt die Verbesserung der digitalen Infrastruktur der Schulen als klares Ziel definiert ([Kapitel 1.1.2](#)) und Gelder dafür bereitstellt, führt zu der Frage: Auf welche Art und Weise, d.h. mit welchen digitalen Werkzeugen und mit welchen zugehörigen fachlich-didaktischen Konzepten, kann der Plan der fortlaufenden Digitalisierung der Schulen in Deutschland in einer für das Lernen der Kinder und Jugendlichen positiven und produktiven Richtung durchgeführt werden?

Im Rahmen der vorliegenden wissenschaftlichen Arbeit kann nur ein Teil der vielfältigen Möglichkeiten, die sich bieten, analysiert und getestet werden. Bei dem Versuch, die angeführte Frage zu beantworten, wird der Fokus auf das Schulfach Deutsch sowie auf das Zusammenspiel der Methoden des kreativen Schreibens ([Kapitel 2.2.1](#)) und des digitalen Schreibens gelegt ([Kapitel 2.2.2](#)).

Mit der Thesis soll dementsprechend ein Beitrag geleistet werden, um die Frage zu beantworten, ob eine digitale Anwendung für das kreative Schreiben mit den eingesetzten Technologien realisiert werden kann und dementsprechend auch ähnliche Anwendungen.

Der Weg zur Konkretisierung des genannten Teilbereichs und zur Realisierung der zu konzipierenden digitalen Anwendung wird innerhalb der folgenden Kapitel der Anforderungsanalyse ([Kapitel 2](#)), der technischen Konzeption ([Kapitel 3](#)) sowie der notwendigen technischen Grundlagen ([Kapitel 4](#)) beschriftet. Die Ergebnisse finden sich in [Kapitel 5](#) sowie an den dort angegebenen Stellen.

Der Ausblick zeigt schließlich Möglichkeiten auf, den Prototypen in seiner fachlich-didaktischen Eignung mit Nutzertests ([Kapitel 6.2.1](#)) zu prüfen und evaluiert Verbesserungsvorschläge und Ideen für die Erweiterung der Webapplikationen ([Kapitel 6.2.2](#)).

1.3 Ziele der Arbeit

In der vorliegenden wissenschaftlichen Arbeit wird eine Web-Applikation konzipiert und als Prototyp zur Verfügung gestellt, mit dem Schüler auf kreative Weise eigene Kurzgeschichten verfassen und damit potentiell ihre Rechtschreibkompetenz und Literalität verbessern können.

Mit der Webapplikation können sich die Schüler darüber hinaus Bilder generieren lassen, die mit Hilfe der Einbindung der API (Application Programming Interface, siehe [Glossar](#)) einer Text-zu-Bild-KI ([Kapitel 4.2](#)) passend zum verfassten Text erstellt werden. Dementsprechend setzt die Web-Applikation die Künstliche Intelligenz als Werkzeug ein, mit dem das Konzept der Gamification innerhalb des Deutschunterrichts realisiert werden kann ([Kapitel 2.2.3](#)).

Damit wird den Deutsch-Lehrern ein technisches Werkzeug an die Hand gegeben, mit dem sie den Aspekt des kreativen Schreibens ([Kapitel 2.2.1](#)) effizienter, motivierender und um den Aspekt des digitalen Schreibens ([Kapitel 2.2.2](#)) erweitert abdecken können.

Die vorliegende wissenschaftliche Arbeit zeigt demnach ein Beispiel für eine technische Anwendung, die in der Schule eingesetzt werden kann und - in diesem konkreten Fall - den Deutschunterricht erweitern kann. Dabei wird neben dem Einsatz der vorgestellten Technologien ([Kapitel 4](#)) der Weg von der Anforderungsanalyse ([Kapitel 2](#)) zur technischen Konzeption ([Kapitel 3](#)) bis hin zur Programmierung ([Kapitel 5](#)) der Webapplikation aufgezeigt. Das Fazit fasst die Erkenntnisse, die sich aus der Konzeption und der Programmierung des Prototypen ergeben, kritisch zusammen ([Kapitel 6](#)). Der Ausblick stellt zusätzlich, wie bereits angeführt, den Entwurf für den Nutzertest vor und fasst Vorschläge für weitere Features (Begriffserklärung: [Glossar](#)) der Webapplikation zusammen ([Kapitel 6.2](#)).

1.4 Aufbau der Arbeit

Die folgende Auflistung stellt einen kurzen Einblick in die Inhalte dar, welche in den nachfolgenden Kapiteln der wissenschaftlichen Arbeit diskutiert werden.

➤ **Kapitel 2: Anforderungsanalyse**

In [Kapitel 2](#) wird nach der Definition der Zielgruppen zunächst die Bedarfsanalyse, was die in der Einführung genannten Themen angeht, durchgeführt. Daraufhin werden die funktionalen und nicht-funktionalen Benutzeranforderungen der Webapplikation aufgelistet und beschrieben. Die funktionalen Benutzeranforderungen präzisieren die Funktionen und die Features, die sich der Stakeholder (meist der Kunde bzw. Auftraggeber) für die Zielgruppe wünscht und die aus der Nutzersicht der Webapplikation erklärt werden. Die nicht-funktionalen Benutzeranforderungen

dagegen legen Standards für die Nutzerfreundlichkeit, das grafische Design des Interfaces, die bereits vorhandene IT-Infrastruktur des Systems, falls ein solches bereits existiert, und die Anforderungen an Datenschutz und Datensicherheit u.a. fest.

➤ **Kapitel 3: Konzeption**

Die Benutzeranforderungen werden in diesem Kapitel auf Systemanforderungen abgebildet, d.h. welche technischen Voraussetzungen und Werkzeuge sind nötig bzw. geeignet, die Benutzeranforderungen effizient und auch langfristig umzusetzen? Des Weiteren ist zu definieren, welche Arbeitspakete notwendig sind, um die Systemanforderungen programmiertechnisch zu realisieren.

➤ **Kapitel 4: Technische Grundlagen**

In [Kapitel 4](#) werden die Grundlagen der Technologien dargelegt, die basierend auf der Anforderungsanalyse und der Konzeption für die programmiertechnische Realisierung der Webapplikation ausgewählt wurden.

Dabei wurden die folgenden Technologien zum Programmieren des Prototypen der Webapplikation gewählt: Die Web-Development-Plattform Angular (Frontend), die Laufzeitumgebung Node.js, der Paket-Manager Node Package Manager sowie der Text-zu-Bild-Generator Leonardo.AI.

➤ **Kapitel 5: Ergebnisse**

Danach werden wichtige Stellen ausgewählter Implementierungen erklärt. Somit kann die programmiertechnische Realisierung von einzelnen festgelegten funktionalen und nicht-funktionalen Benutzeranforderungen gezeigt werden ([Kapitel 2.3](#)).

Die Möglichkeit der Nutzung der Webapplikation kann zum Zeitpunkt der Abgabe der Thesis nur durch das manuelle Setup der Webapplikation gewährleistet werden. Die Anleitung für das Setup findet sich in der Datei README.md des Projektordners:

<https://gitlab.mi.hdm-stuttgart.de/cf061/kurzgeschichten-thesis>.

Des Weiteren wird in dem schriftlichen Teil des Kapitels ein Ausschnitt aus den Templates (Deutsch: Vorlagen) gezeigt, die vollständigen Templates sind während der Nutzung der Webapplikation einsehbar. Bei den Templates handelt es sich um

angefangene Kurzgeschichten, die der Nutzer zur Verfügung gestellt bekommt, die fortgeschrieben werden können und dabei als Schreibimpuls dienen sollen ([Kapitel 2.2.1](#)).

➤ **Kapitel 6: Fazit & Ausblick**

Das Fazit fasst die Ergebnisse auf kritische Art und Weise zusammen.

Der Ausblick zeigt Möglichkeiten auf, die in der Thesis erzielten Erkenntnisse nutzbar zu machen, z.B. für eine Weiterentwicklung des Prototyps. Neben dem Konzept für Nutzertests wird die Einführung weiterer Features diskutiert, die auf den vorhandenen Features (Begriffserklärung Feature, siehe [Glossar](#)) aufbauen.

2 Anforderungsanalyse

In diesem Kapitel werden zunächst die Zielgruppen der zu konzipierenden digitalen Anwendung konkretisiert ([Kapitel 2.1](#)). Daraufhin werden in der Bedarfsanalyse die bildungs-didaktischen Grundlagen erörtert ([Kapitel 2.2](#)), deren Verständnis notwendig ist, um die Bedürfnisse der Zielgruppen herauszuarbeiten. Diese Bedürfnisse werden nachfolgend als funktionale und nicht-funktionale Benutzer-Anforderungen spezifiziert ([Kapitel 2.3](#)).

Die Benutzer-Anforderungen, die an die zu konzipierende Anwendung gestellt sind, werden an späterer Stelle in System-Anforderungen übertragen, indem zugehörige technische Rahmenbedingungen mit einbezogen werden ([Kapitel 3.4](#)).

2.1 Zielgruppen

Die Zielgruppen der in der vorliegenden Thesis konzipierten Webapplikation sind nach dem Alter von Kindern aufgeteilt, da die vorhandenen Fähigkeiten in den Feldern Lesen und Schreiben je nach dem Alter der Kinder differieren. Die Altersgrenzen überlappen sich, da die Fähigkeiten des Weiteren je nach Person variieren und die hier vorgenommene Einteilung demnach nur als grobes Raster verstanden werden sollte. Zusätzlich zeigen Forschungsergebnisse, dass es nicht die eine perfekte didaktische Lösung gibt, die für alle Kinder gleichermaßen gilt, sondern dass Kinder eben auf unterschiedliche Weise gut lernen können und dementsprechend ihre individuellen Lernpfade ins Gewicht fallen sollten.²³

2.1.1 Weiterführende Schulen: 10+ Jahre alt

Kinder sowie Jugendliche der weiterführenden Schulen könnten die Webapplikation weitgehend eigenständig nutzen und ihre Geschichten und Bilder in anschließenden Diskussionen mit den anderen Schülern in der Klasse sowie mit den Lehrern teilen.

Dabei könnte die dargelegte Methode des kreativen Schreibens ([Kapitel 2.2.1](#)), verbunden mit den Vorteilen des digitalen Schreibens ([Kapitel 2.2.2](#)), eingesetzt werden.

2.1.2 Grundschulkinder: 6-10 Jahre alt

Grundschüler dagegen sind noch dabei, das Lesen und das korrekte Schreiben sowohl handschriftlich als auch mit dem Computer zu lernen ([Kapitel 1.1.1](#) sowie [Kapitel 2.2.2](#)).

Trotzdem kann sich die Nutzung der Webapplikation anbieten, indem z.B. das Schreiben von

²³ Rose et al. (2011). S. 223.

durch den Lehrer vorgegebenen Wörtern und Sätzen praktiziert wird oder die Lieblingsgeschichten der Kinder abgeschrieben werden (mit den generierten Bildern als Belohnung). Zusätzlich könnten die Kinder z.B. in Gruppenarbeit kleine eigene Kurzgeschichten entwerfen (mit der Unterstützung des Lehrers, [Kapitel 2.2.1](#)).

2.1.3 Vorschulkinder: 3-6 Jahre alt

Nicht ausgeklammert werden soll die mögliche Nutzung der Applikation als eine Art Bilderbuch für Geschichten, die von Erziehern entweder selbst verfasst oder von anderer Stelle in die Textfelder der Applikation ([Kapitel 3.4](#)) eingefügt werden könnten. Weiterhin könnte auch schon in diesem Alter in Erwägung gezogen werden, Kurzgeschichten in Zusammenarbeit bzw. mit Einbezug der Ideen der Kinder zu schreiben.

2.2 Bedarfsanalyse

Das Kapitel ermittelt die spezifischen Bedürfnisse der Zielgruppen ([Kapitel 2.1](#)) und der potentiellen Stakeholder (Kultusministerien und Lehrerverbände) unter Einbezug aktueller wissenschaftlicher Erkenntnisse der bildungstheoretischen Forschung sowie der in der Einführung ([Kapitel 1](#)) dargelegten Themen hinsichtlich der Entwicklung einer digitalen Anwendung, deren regelmäßige Nutzung potentiell die Rechtschreibfähigkeit von Kindern und Jugendlichen verbessern sowie deren Motivation für das Lernen, primär innerhalb des Schulfaches Deutsch, steigern soll. Die herausgearbeiteten Bedürfnisse an eine solche digitale Anwendung werden daraufhin genutzt, indem daraus die funktionalen und nicht-funktionalen Benutzer-Anforderungen ([Kapitel 2.3](#)) abgeleitet werden.

2.2.1 Kreatives Schreiben

Bis in die 1970er Jahre hinein dominierte in der Deutschdidaktik das Diktat und der Aufsatz. Das Ziel dieser Herangehensweise war, die schriftlichen Darstellungsformen wie den Bericht, die Beschreibung, die Erzählung, die Erörterung etc. in formal korrekter Schreibweise zu vermitteln. Auch ein Erlebnisbericht hatte strengen Normen zu genügen, nicht zuletzt aus dem Grund, dass damit eine unkomplizierte Benotung möglich war. Fürsprecher des kreativen Schreibens haben dies kritisiert und auf die Gefahr von Schreibhemmungen der Kinder infolge des stark normierten Unterrichts hingewiesen. In den 1980er und 1990er Jahren wurden die traditionellen Aufsatzformen schließlich schrittweise durch eine Vielfalt von

Methoden und Konzepten erweitert, in denen kreative Schreibverfahren als methodische Elemente im Schulunterricht ihren Platz fanden.²⁴

Grundvoraussetzung für die Ausübung des kreativen Schreibens ist, dass dafür notwendige Fertigkeiten der Kinder im Lesen und Schreiben schlechthin vorhanden sein müssen. Deshalb soll an dieser Stelle kurz auf das 4-Säulen-Modell zum Lesen- und Schreibenlernen eingegangen werden, das von den Erziehungswissenschaftlern Erika Brinkmann und Hans Brügelmann entworfen wurde und sich in der Deutschdidaktik der Grundschulen etabliert hat.

Die vier methodisch-didaktische Säulen sind:

- Das systematische Behandeln von Komponenten und Verfahren der Schriftsprache (z.B. das Alphabet kennenlernen, Schreib- und Lesestrategien kennenlernen).
- Das freie Verfassen von Texten (z.B. der Erlebnisbericht, das kreative Schreiben).
- freie Lesezeiten der Kinder und das (Vor-)Lesen von Kinderliteratur neben der Nutzung von Hörbüchern und Filmen.
- Das Erforschen, Sammeln, Sortieren und Praktizieren bzw. Austesten von Wörtern sowie Texten (z.B. Wörter einer Liste alphabetisch ordnen sowie schwierige Wörter und Sätze wiederholt schreiben und vorlesen).

Dementsprechend gilt das freie Schreiben von Texten, das in Säule zwei genannt wird, als wichtiger Faktor des Deutschunterrichts in der Grundschule.²⁵

Dabei seien Fehler in der Rechtschreibung und Grammatik zunächst zuzulassen und mit der Hilfe des Lehrers schließlich so gut es geht in der Eigenregie des Kindes auszubessern. Damit wird der Ansicht widersprochen, dass freies Schreiben schädlich sei, nur weil es kurzfristig zu Rechtschreib- und Grammatikfehlern sowie zu Formulierfehlern führen kann.

Das freie Schreiben wird als motivierendes und mit Spaß verbundenes Element des Schreibenlernens betont, da Kinder damit die Chance haben, ihre eigene Meinung und Lebenswirklichkeit auszudrücken.²⁶

Die Methoden des freien Schreibens können neben der Lernmotivation zusätzlich das Sprachbewusstsein fördern und die Kinder in ihrer Schreibkompetenz nachhaltig verbessern, da die individuellen Vorerfahrungen und Interessen der Kinder beim Schreibenlernen berücksichtigt werden.²⁷

²⁴ Abraham et al. (2014). S. 364–370.

²⁵ Landesbildungsserver Baden-Württemberg.

²⁶ Andresen et al. (2006). S. 9-10.

²⁷ Brügelmann, Brinkmann (2012). S. 1-3.

Statt also dem Kind das „fehlerlose Schreiben“ aufzuzwingen und so Verunsicherung zu erzeugen, sollte der große Wert des Lesens erkannt werden (sowohl von Sachtexten als auch von fiktionaler Literatur). Denn mit dem Lesen findet automatisch die Auseinandersetzung des Kindes mit den korrekten Regeln und Formen der Rechtschreibung statt, vorausgesetzt, die gelesenen Texte sind formal korrekt verfasst.²⁸

Kreatives Schreiben als deutsch-didaktisches Konzept wird heutzutage meist mit Inhalten des Sachunterrichts verbunden. Das sollte allerdings nicht dazu führen, dass im Sinne einer konstruierten Ganzheitlichkeit persönliche Themen und Sachinformationen ohne klare Zielsetzung zusammengeworfen werden.²⁹

Stattdessen sollen durch die Verbindung der Fächer neue Anreize und Perspektiven für die Kinder entstehen. Kreatives Schreiben soll Möglichkeiten des Selbstausdrucks für Kinder hinsichtlich gesellschaftlicher Themen (z.B. Armut, Klimawandel) bieten und/oder den individuellen Lebens- und Gefühlswelten der Kinder in kindgerechter Sprache und Erzählweise (z.B. spannende Erlebnisse oder auch phantastische Geschichten) Ausdruck verleihen. Das kann gerade leistungsschwächere Kinder dazu motivieren, nicht die Lust am Lernen zu verlieren.³⁰

Demzufolge kann das kreative Schreiben als Methode charakterisiert werden, die einen tiefgreifenden Bezug zu den Lebenswelten der Kinder inkorporiert. Das kreative Schreiben sei aus den genannten Gründen als Methode des freien Schreibens nicht etwa als Ersatz, sondern als Ergänzung zu den weiteren Formen des freien Schreibens wie der Erörterung, der Nacherzählung und des thematischen Aufsatzes zu betrachten.

Die Methode des kreativen Schreibens ist didaktisch betrachtet dadurch gekennzeichnet, dass das Kind durch einen Schreibimpuls, z.B. eine begonnene Geschichte oder ein Bild, beim Finden eigener Ideen zur Fortführung der Geschichte unterstützt wird. Das kreative Schreiben ist demzufolge als Prozess zu charakterisieren, an dessen Startpunkt zunächst eine aktiv in Szene gesetzte Situation einer Schreibaufgabe steht.

Dieses Konzept wird von der in dieser Thesis prototypisch entwickelten Webapplikation genutzt ([Kapitel 1.3](#), [Kapitel 3.4](#)).

Der Schreibimpuls generiert dabei sozusagen Ideen beim Kind, die zu einem flüssigen Schreibprozess führen können. Dabei hat sich herausgestellt, dass die wesentliche Funktion eines Schreibimpulses darin liegt, durch Irritation und Spannung die Phantasie sowie die

²⁸ Andresen et al. (2006). S. 6. S. 10.

²⁹ Meiers (2008). S.174-178.

³⁰ Andresen et al. (2006), S. 9-10.

Neugier der Kinder darauf anzuregen, wie das vorgegebene Szenario des Schreibimpulses, z.B. die Handlung einer Geschichte, fortgeführt werden kann.³¹

Zusätzlich hat die Forschung gezeigt, dass für Kinder das Schreiben fiktionaler Geschichten leichter ist, verglichen mit dem Verfassen einer Erlebniserzählung oder einer Sacherörterung. Die Kinder können mit dem Schreiben fiktionaler Geschichten nämlich mehr oder weniger ungefiltert ihr ganzes Wissen sowie ihre persönlichen Erfahrungen (Freizeit, Familie, Natur, Medien usw.) verarbeiten und der Geschichte in kreativ veränderter Form hinzufügen.³²

Damit passt das Konzept des kreativen Schreibens zu den generellen Erkenntnissen der heutigen Unterrichtsforschung, nach denen der Unterricht nicht mehr als Prozess angesehen wird, in dem das Wissen, die Fertigkeiten und die Interessen der Schüler nur das Produkt des durch den Lehrer vermittelten Lehrstoffes sind. Stattdessen sollten die bereits vorhandenen Ressourcen der Schüler miteinbezogen werden, damit zusammen mit dem Lehrer ein produktiver Raum der Lernangebote geschaffen wird, der individuell genutzt werden kann. Demnach handelt es sich um ein Prinzip von Angebot und Nutzung.³³

In der Folge stellt sich die Frage, mit welchen didaktischen Lernkonzepten der Unterricht so attraktiv gestaltet werden kann, dass dieses Angebot von den Schülern auch gerne und produktiv genutzt wird. Dieser Frage wird in der Thesis nachgeforscht, jedoch sollte abschließend zu diesem Kapitel erwähnt werden, dass der Prozess aufgrund des Kontextes auf das Konzept „kreatives Schreiben in der schulischen Schreibdidaktik“, kombiniert mit dem Konzept des „digitalen Schreibens“ ([Kapitel 2.2.2](#)), beschränkt ist.

Weitere Konzepte innerhalb des Schulfaches Deutsch und Konzepte aus anderen Fächern können aus zeitlichen Gründen nicht abgedeckt werden.

Des Weiteren gibt es im deutschsprachigen Raum noch weitere Hauptkonzepte des kreativen Schreibens, z.B. in der Therapie, in der autobiographischen Selbstreflexion und sogar in der Wissenschaft.³⁴ Aufgrund dessen ist relativierend zu ergänzen, dass der Versuch einer allgemeingültigen Definition dessen, was kreatives Schreiben ist bzw. umfasst, nicht sinnvoll und praktikabel ist, da die methodischen Ansätze sich zum Teil sehr unterscheiden.³⁵

³¹ Spinner (2015). S. 108-126.

³² Dehn et al. (2015). S. 4-9.

³³ Kleinbub (2018). S. 12.

³⁴ von Werder (2004). S. 75, S. 286-289, S. 79.

³⁵ von Werder (2004). S. 17-23, S.68ff.

2.2.2 Digitales Schreiben

Schreiben, auch kreatives Schreiben, kann man sowohl handschriftlich mit Stift als auch digital mit Tastatur bewerkstelligen, sofern man die Werkzeuge dafür hat und die notwendigen Kompetenzen bei den Kindern, aber auch bei den Lehrern vorhanden sind. Zunächst stellt sich die Frage: Ist es überhaupt wichtig für Kinder, dass sie neben dem handschriftlichen Schreiben schon früh das Tastatur-Schreiben und die Kniffe digitaler Textverarbeitung kennenlernen? Und wenn ja, welche Faktoren sprechen dafür?

In einer mehr und mehr durch die Digitalisierung geprägten Welt stellt das Beherrschen von digitalen Arbeitstechniken und von digitalen Medien eine sehr wichtige Kompetenz dar, sowohl privat als auch beruflich im Leben gut zurechtzukommen.³⁶ Deshalb ist die Frage grundsätzlich mit ja zu beantworten. Es ist anzuraten, dass Kinder bereits im Grundschulalter grundlegende Fertigkeiten in der Nutzung von digitalen Schreibwerkzeugen erwerben. Eine generell vorhandene gute Schreibkompetenz bietet generell positive Lernmöglichkeiten, nicht nur im Fach Deutsch, sondern auch in den meisten anderen Fächern.³⁷

Sollte die handschriftliche Schreibfertigkeit bereits vorhanden sein, lässt sich diese aber nicht ohne Weiteres auf das Schreiben am Computer übertragen. Zudem kann nicht von jedem Kind (und Elternhaus) erwartet werden, dass das Lernen des Tastaturschreibens ausschließlich in der Freizeit erfolgt.³⁸

Das Schreiben mit der Tastatur verlangt von den Kindern neben dem Verständnis der Anordnung der Buchstaben und Satzzeichen auf der Tastatur sowie der Mechanismen zu deren Erzeugung die Kenntnis weiterer Funktionen wie der des Löschens, des Verschiebens, des Kopierens und des Einsetzens von Textbausteinen.³⁹

Dementsprechend stellt die Kompetenz, Texte mit dem Computer zu verfassen, einen wichtigen Teilaspekt des generellen Ziels dar, Kinder altersadäquat an die zielorientierte Nutzung digitaler Arbeitsmethoden, Medien und Programme heranzuführen, das die Kultusministerkonferenz in ihrer Strategie „Bildung in der digitalen Welt“ postuliert hat.⁴⁰

Weitere Vorteile des digitalen Schreibens mit der Tastatur, verglichen mit dem handschriftlichen Schreiben, werden nachfolgend angeführt:

- Studien legen nahe, dass der Prozess des Tastaturschreibens das Arbeitsgedächtnis nicht so sehr in Anspruch nimmt wie das handschriftliche Schreiben.

³⁶ Bildung in der digitalen Welt - Strategie der Kultusministerkonferenz. (2016). S. 9-10.

³⁷ Becker-Mrotzek et al. (2015). S. 8-14.

³⁸ Reble et al. (2020). S. 55.

³⁹ Grabowski et al. (2007). S. 54.

⁴⁰ Bildung in der digitalen Welt - Strategie der Kultusministerkonferenz. (2016). S. 9-10. S. 13.

Schüler könnten dadurch mit der Tastatur eine schnellere Schreibgeschwindigkeit erzielen und hätten mehr Zeit zum Planen, Formulieren und Korrigieren der Texte, was zu einer besseren quantitativen und qualitativen Schreibleistung führen kann.⁴¹

- Das digitale Schreiben ist mit einem sauberen Schriftbild und einer besseren Lesbarkeit verbunden, für den Autor und für andere Lesende (Mitschüler, Lehrer).
- Der Text kann so oft wie nötig korrigiert, erweitert oder gelöscht werden.
- Texte und Bilder können ausgedruckt und vervielfältigt werden.
- Das Nutzen von Textverarbeitungs-Software kann den Kindern helfen, z.B. die Rechtschreib- und Grammatikkontrolle in Eigenregie durchzuführen.⁴²
- Die Vermutung liegt zudem nahe, dass das Tastaturschreiben in hohem Maße die Lebenswelt von Kindern und Jugendlichen mit einbezieht, da sie mehrheitlich digitale Medien und technische Geräte nutzen und damit auch in ihrer Freizeit Texte digital lesen und schreiben. Eine Anknüpfung daran in der Schule könnte sich insgesamt positiv auf die Lernmotivation auswirken.⁴³

2.2.3 Gamification

In der vorliegenden Thesis soll der Begriff der Gamification wie folgt definiert sein: Gamification sei der Einsatz von mindestens einem der Hauptelemente von Computerspielen (englisch: Computer Games), namentlich Game-Mechaniken, Game-Kunst (Grafik, Musik und Soundeffekte) und Game-Design innerhalb von ursprünglich nicht-spielerischen Systemen,⁴⁴ um Menschen, die diese Systeme nutzen, zum Lösen von Problemen anzuregen.⁴⁵

Das Spielen an sich sei definiert als die Bereitschaft des Spielers, sich freiwillig in ein solches System zu begeben, um zu versuchen, Herausforderungen zu bewältigen, die von außen betrachtet unnötig erscheinen können. Unnötig deshalb, weil diese Herausforderungen ja von den Game-Entwicklern künstlich konstruiert wurden. Betrachtet man zum Beispiel das analoge Spiel Fußball, wäre es für alle Beteiligten mit weniger Schmerzen und Aufwand verbunden, wenn sich die beiden Mannschaften verbal auf ein Ergebnis einigen und den Ball ein paar Mal in die Tore tragen würden.⁴⁶ Ziel des Game-Entwicklers sollte es demnach sein,

⁴¹ Bulut (2019). S. 5.

⁴² Becker-Mrotzek (2007). S. 11-13.

⁴³ Statistik zur Mediennutzung von Kindern und Jugendlichen.

[https://de.statista.com/themen/2662/mediennutzung-von-jugendlichen/#:~:text=Laut%20einer%20Umfrage%20zur%20Mediennutzung,Musik%20h%C3%B6ren%20\(66%20Prozent\).](https://de.statista.com/themen/2662/mediennutzung-von-jugendlichen/#:~:text=Laut%20einer%20Umfrage%20zur%20Mediennutzung,Musik%20h%C3%B6ren%20(66%20Prozent).) Datum des Abrufs: 13.01.2024.

⁴⁴ Stampfl (2012). S.16.

⁴⁵ Kapp (2012). S.10-13.

⁴⁶ Suits (1978). S. 43.

mit den Methoden des Game-Designs ein Spiel zu kreieren, das über gewisse ästhetische Eigenschaften, Game-Mechaniken sowie Regeln verfügt, die dem Spieler den Prozess des Spielens so attraktiv erscheinen lassen, dass er freiwillig Zeit, Kraft und ggfs. Geld investiert, um solche Herausforderungen bewältigen zu dürfen.

Der Prozess der Gamification kann zusammenfassend als Methode charakterisiert werden, bei der die Prinzipien der (Computer-)Spielewelt in ihr bis dahin abgewandten Bereichen eingesetzt werden. Bei der Gamification wird dementsprechend kein ganz neues System mit neuen, implementierten Herausforderungen und den dazugehörigen, einen ästhetischen oder sozialen Wert darstellenden Belohnungen, die man nach der erfolgreichen Bewältigung der Herausforderungen ggfs. erhält, geschaffen.

Stattdessen wird ein schon existierendes System mit existierenden Herausforderungen und Mechanismen herangezogen (z.B. der Schulunterricht, der schließlich auch ein von Menschen entworfenes System darstellt), dem nun im Nachhinein die genannten Prinzipien hinzugefügt werden, meist um die Motivation der menschlichen Akteure innerhalb des Systems, daran produktiv teilzunehmen, zu steigern.

Im Kontext der vorliegenden Thesis handelt es sich bei dem bestehenden, grundsätzlich gesehen „spielabgewandten“ System um den Deutschunterricht an deutschen Schulen, für den eine Anwendung konzipiert werden soll, welche neben der Schulung der Grundelemente der digitalen Kompetenz ([Kapitel 2.2.2](#)) auch den motivierenden Faktor der Gamification zu integrieren versucht ([Kapitel 1.3](#), [Kapitel 3.4](#)). Dabei soll die konzipierte Anwendung nicht etwa Anlass sein, die klassischen Formen des Deutschunterrichts ([Kapitel 1.1.1](#), [Kapitel 2.2.1](#)) außer Acht zu lassen, sondern diese gewinnbringend und zeitgemäß weiterzuführen.

Das dargelegte Konzept der Gamification wird umgesetzt, indem die von den Schülern verfassten Texte abschnittsweise in Bilder transformiert werden können. Dafür wird ein Text-zu-Bild-Generator verwendet. Die nicht-funktionalen Anforderungen, die an den Text-zu-Bild-Generator (Künstliche Intelligenz) gestellt sind, finden sich in [Kapitel 2.3.2.2](#).

2.2.4 Technische Infrastruktur von Schulen: Ist-Status

Die vorhandene technische Infrastruktur der deutschen Schullandschaft sowie ein eventuell geplanter Ausbau sind Themen, die in diesem Kapitel behandelt werden. Die Ergebnisse sollen dazu beitragen, dass eine fundierte Wahl für die Technologien getroffen werden kann, die in der konzipierten digitalen Anwendung und in einer potentiellen Weiterentwicklung im Nachgang der Thesis verwendet werden.

Wie in [Kapitel 1.1.2](#) angeführt, kommt der Ausbau der Digitalisierung deutscher Schulen insgesamt nur schleppend voran, obwohl die deutsche Bildungspolitik bereits 2017 auf der Kultusministerkonferenz das Ziel formuliert hat, dass jeder deutsche Schüler bis ca. 2021 zu jedem Zeitpunkt des Schulunterrichts, zu dem es aus pädagogischer Sicht sinnvoll ist, eine digitale Lernumgebung sowie einen Zugang zum Internet nutzen können sollte.⁴⁷

Für den nachfolgenden Abschnitt soll eine positive, lernförderliche IT-Infrastruktur so definiert sein, dass sie die folgenden Punkte miteinschließt:

- Der Zugang dazu und die Nutzung sind einfach und durch alle Nutzer (Schüler, Lehrer, Verwaltung) problemlos zu gewährleisten.
- Neue Lernszenarien, die für Schüler und Lehrer vorteilhaft sind, werden geschaffen.
- Die Nutzungsmöglichkeit in der Schule, von zu Hause sowie mobil ist gewährleistet.
- Gesetzliche Anforderungen an Datenschutz, Jugendschutz, Urheberrecht sind erfüllt.

Die in Deutschland schulrechtlich festgelegte Aufgabenteilung zwischen, Bund, Ländern, Kommunen und den einzelnen Schulen bringt für die Schaffung einer solchen fortschrittlichen und lernförderlichen digitalen Infrastruktur allerdings spezifische Probleme der Abstimmung und kompatiblen Realisierbarkeit mit sich, da für die notwendigen Komponenten die Schulen selber, die Schulträger, 16 Bundesländer und zentral die Bundesregierung Verantwortung tragen. Das führt dazu, dass es bei der Geräteausstattung und bei dem IT-Support von Schulen zwischen den Ländern und sogar innerhalb der Länder selbst erhebliche Unterschiede gibt.⁴⁸

Analysiert man die Lage in Deutschland, sollte der Vergleich zu anderen Ländern wie Südkorea herangezogen werden, wo bereits 1990 mit der Einführung digitaler Medien in das Bildungssystem des ganzen Landes begonnen wurde. Flächendeckend wurden binnen von ca. 15 Jahren belastbare, rechenstarke IT-Infrastrukturen an allen Schulen aufgebaut, freie und ortsunabhängige Zugänge zu Lehr- und Lernmaterialien geschaffen und IT- und Didaktik-Fortbildungen für Lehrkräfte durchgeführt. Begleitend erfolgt zur Überprüfung des Qualitätsstandes eine kontinuierliche Evaluation.⁴⁹

Was den Zugang zum Internet angeht, stehen die deutschen Schulen zwar nicht so schlecht da. Eine beispielhafte Umfrage zur IT-Ausstattung und Medienbildung der Schulen innerhalb eines Bundeslandes aus dem Jahr 2018 ergab, dass dort fast alle Schulen über einen Internetzugang verfügten. Die meisten Schulen verfügten zudem über WLAN-Netzwerke.⁵⁰

⁴⁷ Mußmann (2021). S. 84.

⁴⁸ Breiter et al. (2015). S.24-27.

⁴⁹ Hwang et al. (2010). S. 6-7. S. 22-24.

⁵⁰ Statistiken zur IT-Ausstattung der Schulen - Deutscher Bildungsserver.

Wenn man jedoch die genannten komplexen Voraussetzungen einer guten digital-technischen Schul-Infrastruktur als Ziel und Notwendigkeit unterstellt, ergibt sich ein negativeres Bild. Dabei kommt eine Studie der GEW (Gewerkschaft für Erziehung und Wissenschaft) aus dem Jahr 2021 zu dem Ergebnis, dass nur ca. 23% aller Schulen in Deutschland über eine ausreichende IT-Infrastruktur verfügen.⁵¹

Wichtig wäre für eine nachhaltige Verbesserung der technisch digitalen Infrastruktur der Schulen, dass alle Komponenten bedarfsdeckend vorhanden sind, nämlich die passende Hardware für Lehrende und Lernende, ausreichend Präsentationsgeräte, eine widerstandsfähige Gebäudeinfrastruktur aus Strom- und Netzwerkanschlüssen und eine skalierbare technische Ausstattung mit Servern und WLAN sowie ein fachspezifisches Benutzer- und Gerätemanagement. Darüber hinaus braucht es lernort-vernetzende Lernplattformen und online-gestützte Kommunikationssysteme. Letztlich ist ein breites Angebot an digitalen Anwendungen und lernunterstützenden Medien notwendig.⁵²

Die nächsten wesentlichen Komponenten der digitalen Infrastruktur in Schulen, die in vielen gesellschaftlichen Debatten bisher wenig Berücksichtigung fanden, sind der IT-Support und die fachliche IT-Beratung. Es hat sich gezeigt, dass diese Aufgaben meist nur mangelhaft durch Lehrkräfte, z.B. aufgrund des großen Zeitaufwands, bewältigt werden können. Dementsprechend könnte es häufig notwendig werden, externe IT-Dienstleister zu beauftragen, die das fachliche Know-How ortsnahe einbringen können, wenn die Lehrkräfte und auch die IT-Spezialisten der Kommunen an ihre Grenzen stoßen.⁵³

Denn nach der bereits referenzierten Studie des „Education Campus“ aus dem Jahr 2022 können lediglich ein Drittel der Lehrkräfte auf einen geschulten IT-Betreuer zurückgreifen.⁵⁴

Die von der deutschen Regierung angestrebte Verbesserung der genannten Situation ist mit großen finanziellen Kosten verbunden. Im Auftrag der Bertelsmann-Stiftung wurde in der Abhandlung „Szenarien lernförderlicher IT-Infrastrukturen in Schulen“ aus dem Jahr 2015 eine Schätzung vorgenommen, wie hoch die Gesamtkosten sind, um eine hinreichende Integration digitaler Medien in Lern- und Lehrprozesse von Schulen zu gewährleisten. Die Gesamtkosten würden in Anbetracht der Gesamtschülerzahl allein der weiterführenden allgemeinbildenden Schulen in Deutschland mindestens 538 Mio. Euro betragen, wenn das

<https://www.bildungsserver.de/Statistiken-zur-IT-Ausstattung-der-Schulen-11499-de.html>

⁵¹ Mußmann (2021). S. 83-93.

⁵² Stangl et al. (2021). S.21.

⁵³ Dahlmann (2021). S. 16f.

⁵⁴ Szenarien lernförderlicher IT-Infrastrukturen in Schulen.

<https://www.bertelsmann-stiftung.de/de/publikationen/publikation/did/szenarien-lernfoerderlicher-it-infrastruktur-en-in-schulen/>

5:1-Modell angewandt wird, d.h. ein Endgerät für 5 Schüler. Rechnet man dasselbe Szenario mit dem 1:1-Modell durch, ergeben sich sogar ca. 2,62 Mrd. Euro.

Diese Schätzungen sind allerdings lediglich eine Orientierungshilfe und basieren auf Annahmen einer Modellschule der Sekundarstufe. Deshalb sind sie nicht ohne weiteres auf andere Schulformen übertragbar. Hierzu müssten abhängig von den besonderen Bedingungen und Anforderungen der einzelnen Schulen einige Kostenpunkte angepasst werden. Deutlich wird jedoch, dass die Bereitstellung von WLAN und Breitbandanbindung (Basisinfrastruktur) sowie die Betriebskosten der zentralen Dienste, der Lernplattformen und der Serverstruktur den Großteil der Kosten ausmachen. Insofern sind Bund, Bundesländer und Kommunen auch finanziell enorm gefordert.⁵⁵

2.2.5 Lokales Hosting vs. Cloud-Hosting

Neben der Bereitstellung der Hardware in den Schulen, neben der Bewältigung des Fortbildungsbedarfs und neben den anspruchsvollen Aufgaben des Supports, gilt es noch ein weiteres Handlungsfeld zu „beackern“, nämlich die Schaffung eines flächendeckenden WLAN- bzw. Glasfaser-Netzes in Deutschland. Politisches Ziel ist es, bis zum Jahr 2025 jeden Haushalt mit einem Gigabit-Anschluss zu versorgen, damit auch die Schulen sich vernünftig vernetzen können und damit für Lehrer und Schüler ein effektives Arbeiten in der Schule und zuhause möglich wird.

Deswegen wird von diversen kommunalen Dienstleistern für die meisten Fälle empfohlen, einheitliche digitale Lernplattformen in Schul-Clouds zu schaffen, anstatt auf lokale Server in Schulen zurückzugreifen. Durch den Einsatz von Cloud-Lösungen könnten zusätzlich Kosten gespart, der Wartungsaufwand verringert und höhere Sicherheitsstandards erfüllt werden. Dahinter steht die Überzeugung, dass eine sinnvolle und effektive Digitalisierung der Schulen am ehesten mit zentralisierter Hard- und Software gelingen kann.⁵⁶

Der Einsatz von Lern-, Informations- und Kommunikationsplattformen für den Erziehungs- und Bildungsauftrag von Schulen ist durch §1 des Schulgesetzes abgedeckt.⁵⁷ Aktuell verfügen die Bundesländer weitestgehend über solche Lernplattformen. Eine nationale Bildungsplattform, die öffentliche Lösungen der Länder und Lösungen privater Anbieter zusammenführt, steckt jedoch noch in den Anfängen. Es ist laut der Vorsitzenden der

⁵⁵ Breiter et al. (2015). S.44-45.

⁵⁶ Koch. S.18-19.

⁵⁷ Kultusministerium Baden-Württemberg (März 2018). S. 1.

Kultusministerkonferenz, Britta Ernst, geplant, eine digitale „Schicht“ über die Länder-Plattformen zu legen, um deren Inhalte zu vernetzen.⁵⁸

Im Land Baden-Württemberg hat das Kultusministerium in Abstimmung mit den Mitarbeitervertretungen im Jahr 2018 eine Rahmendienstvereinbarung zum Einsatz einer landeseinheitlichen digitalen Bildungsplattform abgeschlossen. Sie regelt u.a.

- die Einführung, den Zugang und den Service der landeseinheitlichen Lern-, Informations- und Kommunikationsplattform für die Beschäftigten,
- die Einführung, den Einsatz und die Nutzung einer sicheren dienstlichen E-Mail-Adresse für die Lehrkräfte,
- die Nutzung von digitalen Formaten der Lehrerfortbildung (E-Learning und Blended-Learning, d.h. die Verzahnung von Präsenz- und E-Learning),
- die Bereitstellung und die Nutzung von „Open Educational Resources“ auf der digitalen Bildungsplattform (Bildungsmedien),
- den Einsatz von WLAN in der Schule.⁵⁹

Die Diskussion über eine Zentralisierung durch Cloud-Computing kam anlässlich der Notwendigkeit des Distanz-Unterrichts in der Corona-Zeit verstärkt in Gang. Seitdem existiert z.B. der Vorschlag, kommunale Rechenzentren für das Hosten von Schul-Clouds zu nutzen. Mit dieser Lösung könnte die Systemlast auf mehrere Server verteilt und die Systemkapazität nach Bedarf skaliert werden.⁶⁰

Insbesondere die Skalierbarkeit sei für die digitale Infrastruktur von Schulen wichtig, da die Benutzerbasis sich entsprechend der Schulzeiten signifikant verändern könne und es dadurch notwendig sei, die Leistungsfähigkeit und Kapazität der Server entsprechend zu erhöhen oder zu senken. Um flexibel auf die Nachfrage nach Serverkapazität zu reagieren, sei insofern eine vertikale Skalierung in Form einer „Skalierung in der Cloud“ besonders geeignet.⁶¹

Die webbasierte Arbeitsplattform „Logineo“ kann als Beispiel einer digitalen Anwendung für den Einsatz in Schulen genannt werden. Diese Anwendung hat das Bundesland Nordrhein-Westfalen schon geraume Zeit vor der Pandemie eingeführt. Sie bietet für Lehrende und Lernende digitale Kommunikationsmöglichkeiten, ein Kalendertool, Lehr- und Lernmaterial in einem geschützten Cloud-Bereich und eine Datenbank mit lizenzierten audiovisuellen Bildungsmedien. Die Plattform beinhaltet auch ein Lern-Management-System,

⁵⁸ Interview mit der Vorsitzenden der Kultusministerkonferenz Britta Ernst. Vitako aktuell, 01.2021, S.23.

⁵⁹ Ministerium für Kultus, Jugend und Sport Baden-Württemberg, Az.: 53-6534.42/149, 06.02.2018.

⁶⁰ Coenen (2021). S. 7.

⁶¹ Digitales Institut, 2023, <https://digitales-institut.de/was-bedeutet-skalierbarkeit/>

um Unterrichtsmaterialien zu verschicken, Aufgaben zu bearbeiten und individuelles Feedback (Lehrer-zu-Schüler) zu geben. Seit Einführung im Jahr 2019 arbeiteten Anfang 2021 bereits knapp 2000 Schulen in Nordrhein-Westfalen mit dieser Bildungsplattform.⁶²

Zunehmend werden webbasierte Anwendungen als digitale Werkzeuge im Unterricht deutscher Schulen genutzt. Einige der am weitest verbreiteten Lernplattformen sind:

- „Moodle“, eine Lernplattform, die weltweit ca. 340 Millionen Nutzer hat und in einigen Bundesländern den Schulen über Bildungsserver bereitgestellt wird.
- „itslearning“, ein skandinavisches Produkt, das ein vollwertiges, mobil nutzbares Lernmanagementsystem darstellt.
- „Classtime“, eine Anwendung, mit der Schülern in digitaler Form Aufgaben und Fragen gestellt werden können. Auch kollaborative Herausforderungen, die von einer Gruppe von Schülern im Team gemeinsam bewältigt werden können, sowie Lernfortschrittskontrollen durch die Lehrkräfte sind mit diesem Werkzeug möglich.⁶³

Nicht unerwähnt bleiben darf jedoch ein nicht unwesentlicher Vorteil eines lokalen Hostings in deutschen Schulen. Weil die Daten vor Ort auf lokalen Servern liegen, bietet das lokale Hosting mehr Kontrolle und Sicherheit für sie als ein Cloud-Hosting, bei dem die Daten auf entfernten Servern gespeichert und von Dritten verwaltet werden. Dies kann Unsicherheiten mit sich bringen. Ein weiterer Vorteil des lokalen Hostings ist, dass Vor-Ort-Speicher ohne Internetverbindung genutzt werden können und dadurch Kosten gesenkt werden können.⁶⁴

Datenschutz und Datensicherheit sind demnach mit dem lokalen Hosting auf den ersten Blick einfacher zu gewährleisten als mit Cloud-Hosting. Doch aufgrund der gesetzlichen Anforderungen an Informationssicherheit, Datenschutz, Jugendschutz und Urheberrecht, die in den letzten zehn Jahren durch die zunehmende Mediatisierung des Unterrichts immer komplexer wurden, ist davon auszugehen, dass der Aufbau von „Insellösungen“ in jeder einzelnen Schule mit Hilfe des lokalen Hostings nicht so wirtschaftlich und nicht so praktikabel ist wie die Nutzung von Cloud-Lösungen. Ausnahmefälle könne es zwar geben, in der Regel jedoch könnten die Anforderungen an Datenschutz und Datensicherheit aber nicht von jeder Schule allein verantwortet werden.⁶⁵

⁶² Resch et al. (2021). S.13.

⁶³ Rust. Digitale Tools & Apps für den Unterricht.

⁶⁴ Hagel. Vor-Ort- vs. Cloud-Speicher.

⁶⁵ Breiter et al. (2015). S. 24.

2.2.6 Datenschutz und Datensicherheit

Datenschutz und Datensicherheit sind wichtige Aspekte bei der Digitalisierung der Schulen. Denn selbst innerhalb des geschützten Raums der Schule gelten die europäischen Datenschutzvorgaben der DSGVO (Datenschutz-Grundverordnung). Es gilt im Grundsatz zu gewährleisten, dass die personenbezogenen Daten von Schülern und Lehrern geschützt sind. Hierzu gehören nicht nur Kontaktdaten von Schülern, Lehrern und Eltern, sondern auch Informationen über Schulnoten, Fehlzeiten, Förderbedarfe, Fotos, Aufsätze, Prüfungen u.a., sofern sie in Beziehung zu den Akteuren in der Schule gebracht werden können. Die DSGVO legt fest, dass alle personenbezogenen Daten nur für bestimmte Zwecke erhoben und verarbeitet werden dürfen. Noch dazu muss vorher die Einwilligung der betroffenen Personen eingeholt werden.

Im Schulgesetz ist in den §§ 120-122 geregelt, dass die Schulleitung für die Einhaltung der gesetzlichen Bestimmungen des Datenschutzes und der Datensicherheit verantwortlich ist. Hierzu gehört insbesondere auch, entsprechende Auftragsdatenverarbeitungsverträge mit den beteiligten IT-Firmen abzuschließen, sofern personenbezogene Daten in einer Software erfasst und abgelegt werden. Folgende wesentliche Aspekte sind von der Schulleitung im Blick auf die digitalen Endgeräte, die eingesetzten Software-Anwendungen und vor allem, sofern genutzt, die Schul-Cloud zu bedenken:

- Wo und auf welchen Servern werden die Daten verarbeitet?
- Für wen sind die Server zugänglich?
- Zu welchem Zweck werden die Daten gespeichert?
- Können die Daten für andere Zwecke weiterverarbeitet werden?

Unabhängig davon sollte das Thema Datenschutz auch regelmäßig im Unterricht behandelt werden, verbindlicher Bestandteil von Lehrerfortbildung sein und im Schulalltag von den Verantwortlichen ernst genommen werden.⁶⁶

Ferner ist die Schulleitung durch die Artikel 15-21 der DSGVO verpflichtet, folgende Betroffenenrechte zu wahren: Auskunftsrecht, Recht auf Berichtigung und Löschung, Recht auf Einschränkung der Verarbeitung, auf Datenübertragbarkeit und Widerspruchsrecht, die personenbezogenen Daten betreffend. Eine Verweisung der Schüler, Eltern oder Lehrer an den jeweiligen externen Dienstleister ist nicht zulässig.⁶⁷

Gerade weil Software in Schulen zunehmend webbasiert verwendet wird und standardisierte Bildungsangebote und Applikationen externer Anbieter, z.B. als Public-Cloud-Angebote,

⁶⁶ Wawrzyniak (2020). S. 1.

⁶⁷ Kultusministerium Baden-Württemberg. Leitfaden für den Betrieb von Lern-, Informations- und Kommunikationsplattformen, März 2018, S.1.

genutzt werden, ist eine datenschutzrechtliche Prüfung von besonderer Bedeutung. Dies gilt vor allem für Software und Server, deren Leistungserbringung außerhalb der Europäischen Union, also außerhalb des Geltungsbereichs der DSGVO liegt. Das betrifft insbesondere auch Office-Produkte, z. B. „Microsoft Office 365“ oder „Google Docs“, die bereits verbreitet in Schulen eingesetzt sind, aber auch Standardsoftware, z.B. für Bildbearbeitung, oder sonstige Lernsoftware.⁶⁸

Personenbezogene Daten von Akteuren in der Schule dürfen also nicht ohne weiteres auf US-amerikanischen Servern gespeichert werden. Grundlage dafür ist das Urteil des Europäischen Gerichtshofs (EuGH) vom 16.07.2020, in dem das „Privacy Shield“ zwischen der Europäischen Gemeinschaft und den Vereinigten Staaten für ungültig erklärt wurde.⁶⁹

Deshalb empfiehlt das Kultusministerium Baden-Württemberg, in den Schulen ausschließlich Tools auf Servern mit deutschen oder europäischen Standorten zu nutzen. Konkret wird die vom Datennetz der wissenschaftlichen Einrichtungen des Landes Baden-Württemberg BeLWue (Baden-Württemberg extended Lan) gehostete Software „Moodle“ als Lern-, Informations- und Kommunikationsplattform empfohlen. Diese ist aus datenschutzrechtlicher Sicht ohne Einschränkung zulässig.⁷⁰

Als Hauptgrund für das obige Urteil nannten die Richter des EuGH die Massenüberwachung durch US-amerikanische Sicherheitsbehörden, wie z.B. der NSA (National Security Agency). Der Europäische Datenschutzausschuss hat daraufhin Empfehlungen zur Ausgestaltung von Schutzmaßnahmen erarbeitet, was den internationalen Datentransfer anbelangt.⁷¹

Um die Komplexität und Vielschichtigkeit der Anforderungen für Schulen und Schulträger, was den Datenschutz und die Datensicherheit betrifft, zu verdeutlichen, soll abschließend eine summarische Aufzählung der Arbeitsbereiche erfolgen, für die das Kultusministerium Baden-Württemberg einheitliche Standards empfiehlt:

- Eine Vereinbarung betreffend Lernmanagementsysteme, die deren Nutzung für pädagogische Zwecke, z.B. für Unterrichtsvorbereitung und Unterrichtsdurchführung durch Lehrer und Schüler, regelt.
- Die Einführung eines landeseinheitlichen Messengers für die Lehrkräfte, um eine sichere Kommunikation im schulischen Kontext zu gewährleisten.

⁶⁸ Breiter et al. (2015). S. 33-40.

⁶⁹ Gerichtshof der Europäischen Union. Pressemitteilung Nr. 91/20. Luxemburg. 2020.
<https://curia.europa.eu/jcms/upload/docs/application/pdf/2020-07/cp200091de.pdf>.

Datum des Abrufs: 30.01.2024.

⁷⁰ Rust. Digitale Tools & Apps für den Unterricht.

⁷¹ Pressestelle des Landesbeauftragten für den Datenschutz Baden-Württemberg. 20.11.2020.
<https://www.baden-wuerttemberg.datenschutz.de/dsgvowirkt/>. Datum des Abrufs: 30.01.2024.

- Eine Empfehlung zur Verschlüsselung von Speichermedien, um Festplatten, USB-Sticks und andere Speichermedien sowie PC's, Laptops, Tablets u.ä. vor fremdem Zugriff zu schützen, z.B. durch den Einsatz der Software VeraCrypt.
- Das Verbot des Einsatzes „Sozialer Netzwerke“ an Schulen (z.B. Facebook, Twitter, Instagram und Snapchat), da diese Foren fast ausschließlich auf Servern außerhalb der Europäischen Union betrieben werden und nicht europäischen Standards entsprechen.
- Der Abschluss einer datenschutzrechtlichen Vereinbarung zur Schulverwaltungs-Cloud mit der Kommunalen Informationsverarbeitung Baden Franken (KIVBF), wodurch nicht jede einzelne Schule einen Vertrag zur Auftragsverarbeitung abschließen muss, um webbasierte Anwendungen in einer Cloud zu nutzen.
- Die Vorgabe zur verbindlichen Nutzung von Passwörtern beim Arbeiten mit allen Anwendungen und Dateien, die personenbezogene Daten enthalten, z.B. mit Hilfe der Open-Source-Software „KeePass“.
- Eine Empfehlung, ein von der Stiftung Warentest für gut befundenes Virenschutz-Programm einzusetzen.
- Das Verbot von E-Mail-Verkehr für digitale Nachrichten, die personenbezogene Daten enthalten, es sei denn, diese stehen in einem verschlüsselten Anhang. Das Passwort ist allerdings dann über einen anderen Kommunikationskanal mitzuteilen, z.B. telefonisch.⁷²

Bis zu diesem Punkt der Anforderungsanalyse wurden bereits die Zielgruppen definiert ([Kapitel 2.1](#)). Daraufhin wurden die zum Verständnis der Zielgruppen und des Kontexts der Schule und des Deutschunterrichts notwendigen didaktischen und bildungstheoretischen Grundlagen erläutert sowie die Bedürfnisse der Zielgruppen und der Stakeholder (Bundesregierung, Kultusministerien) ermittelt, indem bereits vorhandene Ressourcen dargestellt und mit den Zielen der Stakeholder verglichen wurden ([Kapitel 2.2.1](#) bis [2.2.6](#)).

Der Ist-Status der technischen Infrastruktur der Schulen in Deutschland ([Kapitel 2.2.4](#)) wurde dabei als nicht ausreichend im Vergleich zu anderen führenden Industrienationen bewertet. Der Vergleich von lokalem Hosten von Softwareprodukten mit dem Hosten in der Cloud im Kontext Schule fiel zugunsten der Cloud aus ([Kapitel 2.2.5](#)), da die Anforderungen an den Datenschutz und die Datensicherheit zu komplex sind, um von jeder einzelnen Schule im Alleingang technisch geregelt zu werden. Des Weiteren verfolgt z.B. das Kultusministerium

⁷² Datenschutz und Urheberrecht in der Kultusverwaltung BW.
<https://it.kultus-bw.de/,Lde/Startseite#page>. Datum des Abrufs: 30.01.2024.

Baden-Württemberg das Ziel des Aufbaus einer landeseinheitlichen digitalen Bildungsplattform. Allerdings ist zu beachten, dass für die Cloud-Lösung Server mit dem Standort Europa zu bevorzugen sind, da andere Serverstandorte in der Vergangenheit Probleme mit dem Datenschutz und der Datensicherheit zur Folge hatten ([Kapitel 2.2.6](#)).

2.3 Benutzer-Anforderungen

Die Benutzer-Anforderungen stellen die Anforderungen aus Sicht der Stakeholder und aus der Sicht der Zielgruppen dar. Das nachfolgende Kapitel stellt die Benutzer-Anforderungen dar, die aus der Bedarfsanalyse abgeleitet werden.

Dabei werden zunächst die funktionalen Benutzer-Anforderungen aufgelistet und beschrieben ([Kapitel 2.3.1](#)). Anschließend werden die nicht-funktionalen Benutzer-Anforderungen dargelegt ([Kapitel 2.3.2](#)). Die Benutzer-Anforderungen werden dann mit Einbezug der technischen Rahmenbedingungen auf System-Anforderungen abgebildet ([Kapitel 3.4](#)).

2.3.1 Funktionale Benutzer-Anforderungen

Die funktionalen Anforderungen eines Softwareprodukts sind die Schnittstellen und Bedienelemente, die dem Nutzer durch den Entwickler zur Verfügung gestellt werden müssen, damit der Nutzer die Ziele, die durch die Verwendung des Softwareprodukts erreichbar sein sollen, auch wirklich erreichen kann. Dementsprechend handelt es sich um Funktionen, welche der Nutzer des Softwareprodukts für das lösungsorientierte Arbeiten mit der Applikation benötigt. Darunter fallen sowohl die funktionalen Aspekte des User-Interfaces, die das Bedienen der Benutzeroberfläche und das Navigieren innerhalb der verschiedenen Komponenten des Softwareprodukts ermöglichen, als auch die implementierten Input-Output-Operationen, die dem Nutzer Eingaben möglich machen sowie Daten zur Verfügung stellen, damit er die gewünschten Ziele erreicht.⁷³

In Abgrenzung dazu stehen nicht-funktionale Anforderungen, die in [Kapitel 2.3.2](#) definiert und dargestellt werden.

Nachfolgend werden als erstes die Benutzeranforderungen, die der Prototyp der in dieser Thesis vorgestellten Webapplikation erfüllen soll, anhand von “Use Cases” (engl. Anwendungsfälle) aufgelistet. Jeder Use Case stellt damit einen Anwendungsfall dar, der aus Anwendersicht die Art und Weise beschreibt, wie ein Nutzer das Softwaresystem nutzen können soll, damit er die von den Stakeholdern (meistens die Kunden, die das

⁷³ Balzert, S. 486-489.

Softwareprodukt in Auftrag geben) festgelegte Ziele erreichen kann. In der Software-Entwicklung werden Use-Cases seit ihrer erstmaligen Vorstellung bei der OOPSLA-Konferenz (Object-Oriented Programming, Systems, Languages & Applications) im Jahr 1987 häufig genutzt, um die Funktionen eines Systems und damit die Anforderungen, die daran gestellt sind, aus der Anwendersicht zu beschreiben und im Verlauf des Entwicklungsprozesses schrittweise zu erweitern.⁷⁴

Die Use-Cases werden in der technischen Konzeption auf System-Anforderungen abgebildet, damit hiernach der Prototyp als Proof-of-Concept programmiert werden kann ([Kapitel 3.4](#)). Das Proof-of-Concept kann anschließend evaluiert werden, indem z.B. lokale Nutzertests durchgeführt werden ([Kapitel 6.2.1](#)).

2.3.1.1 Die Liste mit den Templates anzeigen lassen

Der Nutzer kann sich eine Liste anzeigen lassen, die die existierenden Templates zeigt. Templates sind angefangene Kurzgeschichten, die als Schreibimpuls, der in der Methode des kreativen Schreibens Anwendung findet ([Kapitel 2.2.1](#)), dienen sollen. Die Templates enthalten somit einen Textabschnitt und ein dazu generiertes Bild.

2.3.1.2 Kurzgeschichten verfassen

Der Nutzer kann sich ein Template heraussuchen und die darin enthaltene Kurzgeschichte weiter schreiben. Dies geschieht abschnittsweise, d.h. es existieren Textfelder, die untereinander angeordnet sind und in denen eine begrenzte Anzahl an Zeichen geschrieben werden kann. Statt mit den gefüllten Templates zu arbeiten, kann der Nutzer das leere Template auswählen und den Start der Kurzgeschichte selbst verfassen.

2.3.1.3 Die Rechtschreibprüfung durchführen lassen

Der Nutzer kann für jeden Textabschnitt einzeln die automatische Rechtschreibprüfung durchführen. Wird diese beendet, sind grammatikalisch falsche Textstellen rot unterstrichen.

2.3.1.4 Bilder zum Text generieren lassen

Der Nutzer kann sich für jeden Textabschnitt vier Bilder generieren lassen, die zum Inhalt des verfassten Textes passen sollen.

⁷⁴ Jacobson, S. 4.

2.3.2 Nicht-funktionale Benutzer-Anforderungen

Nicht-funktionale Anforderungen eines Softwareprodukts fokussieren sich auf generelle Anwendungseigenschaften und Softwarequalitätsmerkmale, welche für die reibungslose Nutzbarkeit des Systems wichtig sind.

2.3.2.1 Benutzerfreundlichkeit

2.3.2.1.1 Leicht verständliche Sprache und erklärender Charakter

Angesichts der dargestellten Zielgruppen der zu konzipierenden digitalen Anwendung ([Kapitel 2.1](#)) sollten dargestellte Texte sowohl in Textfeldern als auch Buttons leicht zu verstehen sein und keine unnötig komplizierte Sprache verwenden. Diese Schnittstellen der grafischen Benutzeroberfläche sollten des Weiteren kindergerecht sein, indem sie einen „erklärenden“ Charakter inhabitieren.

Dieser erklärende Charakter soll kurz anhand des vierten Use-Cases „Der Nutzer kann sich Bilder zum Text erstellen lassen“ ([Kapitel 2.3.1.4](#)) erklärt werden. Sollte in der technischen Konzeption erwogen werden, pro Textabschnitt einen Button für das Generieren von Bildern neben den Textabschnitten zu platzieren, sollte der Text dieses Buttons kurz und präzise die Funktion beschreiben, die mit dem Klicken des Buttons einhergeht, z.B. „Bilder generieren.“

Des Weiteren sollte dieses Feature einen kleinen, aber präzisen Erklärtext mit sich bringen. Z.B. sollte in der Spalte über den Buttons ein Erklärtext folgender Art stehen: „Du willst dir doch sicher Bilder zu deiner Kurzgeschichte generieren lassen? Kein Problem! Klick einfach auf die Lila Buttons und du erhältst dann die Bilder für den Textabschnitt!“ Zudem sollten Warnmeldungen klar formuliert sein, aber nicht zu sehr ins Details gehen, falls ein Fehler auftritt, z.B. wenn der Button-Klick der Bildgenerierung nicht funktioniert hat: „Ups, da hat wohl etwas nicht geklappt. Bitte versuch doch nochmal, das Bild zu generieren!“

2.3.2.1.2 Responsiveness

Der Begriff Responsiveness meint, dass eine digitale Anwendung sich in der Größe der einzelnen visuellen Elemente an die Bildschirmgröße des Benutzers und im Falle einer Webapplikation zusätzlich an die Größe des Browser-Fensters anpasst. Die Responsiveness sollte für eine reibungslose Nutzung der Webapplikation sichergestellt sein.

2.3.2.1.3 Barrierefreiheit

Die Inklusion spielt eine wichtige Rolle in der Schullandschaft. Die Webapplikation sollte so programmiert sein, dass auch Menschen mit beeinträchtigter Sehleistung die Webapplikation navigieren und nutzen können (Resultat: [Aspekt 3: Barrierefreiheit: visuelles Handicap](#)).

Ebenso vorteilhaft wäre der Text-to-Speech-Support, das heißt, dass Hilfstexte sowie verfasste Texte von einem Sprach-Computer vorgelesen werden können ([Kapitel 6.2.2.4.1](#)).

Nicht vernachlässigt werden sollte des Weiteren der Speech-to-Text-Support, d.h. dass Nutzer der Webapplikation den Text, den sie schreiben möchten, in das Mikrofon des jeweiligen Endgerätes sprechen können, statt ihn einzutippen ([Kapitel 6.2.2.4.2](#)).

2.3.2.2 Text-zu-Bild-Generator-Dienst

2.3.2.2.1 Schutz vor nicht jugendfreien Inhalten

Aus dem Alter der Zielgruppen resultiert die nicht-funktionale Anforderung, dass die Webapplikation keine Bilder mit nicht jugendfreien Inhalten, z.B. Gewaltdarstellungen und Pornographie, zeigt. Dies muss auch dann sichergestellt sein, sollte der Nutzer einen Text mit derartigen Inhalten in den Textfeldern verfassen oder dort reinkopieren.

2.3.2.2.2 Qualität der generierten Bilder

Die Qualität der Bilder sollte von der Qualität der Texte des Nutzers (Präzision der Formulierung, Struktur, Korrektheit der Grammatik, Kreativität) abhängen. Dieses Prinzip wird in der Datenwissenschaft als „Garbage in, Garbage out“-Prinzip bezeichnet. Der Ausdruck „Garbage“ heißt auf deutsch „Müll“. Dementsprechend impliziert das Prinzip die Prämisse, dass die Daten, die ein System ausgibt, in ihrer Qualität nur so nützlich sein können, wie die Daten es waren, die in ein System hineingegeben wurden.⁷⁵

2.3.2.2.3 API-Zugang

Der Text-zu-Bild-Generator-Dienst sollte über eine API (Programmierschnittstelle) verfügen, damit er in die konzipierte Webapplikation eingebunden werden kann.

2.3.2.2.4 Kosten

Angesichts der am Ende von [Kapitel 2.2.4](#) dargestellten Kostenschätzung der Digitalisierung der schulischen IT-Infrastruktur erscheinen Kosteneinsparungen bei neuen digitalen Werkzeugen wünschenswert. Deswegen wäre es sinnvoll, bei der Wahl des Text-zu-Bild-Generators auf den Kostenfaktor (je niedriger, desto besser) zu achten.

⁷⁵ Rose et al. (2004). S. 222-224.

Andererseits wurden die in [Kapitel 1.1.2](#) angeführten Geldmittel, die die Bundesregierung zur Digitalisierung bereitgestellt hat, noch kaum ausgeschöpft, was auf eine gegenteilige Aussage verweisen würde. Der Faktor Kosten kann in seiner Gewichtung in der vorliegenden Thesis ohne Interviews mit den potenziellen Stakeholdern (Kultusministerien, Kommunen, Schulen) nicht ohne Weiteres bewertet werden.

2.3.2.3 Dynamisches Benutzerinterface

Das grafische Benutzerinterface sollte dynamisch aktualisiert werden, sobald der Nutzer Text schreibt, Bilder generiert, die Rechtschreibprüfung durchführt und neue Textfelder erstellt u.a. Das heißt konkret, dass Teile des Benutzerinterfaces variabel sichtbar und unsichtbar gemacht werden sollten oder erstellt bzw. gelöscht werden, abhängig von den Benutzereingaben.

2.3.2.4 Nutzerzahl

Die Schülerzahl in Deutschland des Schuljahres 2022/2023 beträgt circa 11,1 Millionen.⁷⁶ Diese Zahl stellt die potentielle Nutzerzahl in Schulen dar, sofern die Relevanz des literarischen und kreativen Schreibens nicht auf einige wenige Klassenstufen begrenzt wird. Sollte der in [Kapitel 2.1.3](#) beschriebene Einsatz der Webapplikation in Kindertageseinrichtungen (Kitas) erwogen werden, wäre die geschätzte Anzahl der Erzieher an deutschen Kitas mit einzubeziehen, da die Kinder die Webapplikation dort nicht selbst nutzen können müssen, sondern die Erzieher als Nutzer angesehen werden könnten. Dabei sind circa 700000 Personen in der pädagogischen Kinderbetreuung in Kindertageseinrichtungen tätig.⁷⁷ Zusammengerechnet mit der Anzahl der Schüler muss also mit der Nutzerzahl von circa 11,8 Millionen Personen gerechnet werden.

Diese generelle Nutzerzahl von 11,3 Millionen spielt allerdings nur eine Rolle, sollte die Webapplikation auf ausgelagerten Servern in der Cloud gehostet werden und bundesweit zum Einsatz kommen. Wird die Webapplikation dagegen lokal an einzelnen Schulen oder Kitas gehostet, spielt nur die Nutzerzahl pro Schule bzw. pro Kita eine Rolle. Die Anzahl der allgemeinbildenden Schulen in Deutschland betrug zum Schuljahr 2022/2023 circa 32600.⁷⁸ Die Anzahl der beruflichen Schulen beträgt circa 8700.⁷⁹ Zusammen ergibt sich also die Zahl

⁷⁶ https://www.destatis.de/DE/Presse/Pressemitteilungen/2023/03/PD23_105_211.html.

Datum des Abrufs: 14.01.2024.

⁷⁷ https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Soziales/Kindertagesbetreuung/_inhalt.html.

Datum des Abrufs: 14.01.2024.

⁷⁸ <https://de.statista.com/statistik/daten/studie/476776/umfrage/allgemeinbildende-schulen-in-deutschland/>.

Datum des Abrufs: 23.01.2024.

⁷⁹ https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Bildung-Forschung-Kultur/Schulen/_inhalt.html.

Datum des Abrufs: 23.01.2024.

von 41300 Schulen in Deutschland. Wird die Zahl der Schüler jetzt durch die Anzahl der Schulen geteilt, ergeben sich circa 270 Personen als Nutzer der Anwendung pro Schule.

Die Zahl der Kindertageseinrichtungen in Deutschland betrug 2023 circa 60.000⁸⁰. Teilt man die Anzahl der Personen in der Kinderbetreuung durch die Anzahl der Kindertageseinrichtungen, ergibt sich die Nutzerzahl von 11 Personen pro Kita.

2.3.2.5 Anpassbarkeit in Abhängigkeit des Alters des Nutzers (Laufzeit)

Da es sich bei den Nutzern um Kinder und Jugendliche mit großer Altersspanne handelt, sollte die Webapplikation die Funktion haben, dass das grafische Benutzer-Interface und die Business-Logik in den bereitgestellten Funktionen, Features und Bedienelementen zur Laufzeit an das Alter des aktuellen Nutzers anpassbar sind. Das heißt konkret, dass es mit dem ausgewählten Web-Development-Framework möglich sein soll, die nötige Business-Logik für diese Abstufung je nach altersabhängiger Zielgruppe zu implementieren.

Beispielsweise sollten die angezeigten Texte und sonstige Elemente je nach Alter des Nutzers leicht variieren. Der Grad des in [Kapitel 2.3.2.1.1](#) vorgestellten erklärenden Charakters der Benutzerelemente und der Grad der kindgerechten Sprache sollte mit dem zunehmenden Alter des Nutzers abnehmen. Weiterhin könnte die korrekte Grammatik der verfassten Texte für ältere Schüler obligatorisch sein, damit überhaupt Bilder generiert werden dürfen, während diese Funktion für jüngere Schüler zu hart sein kann und zu Schreibhemmungen führen kann, was mit den in [Kapitel 2.2.1](#) angeführten Forschungsergebnissen gezeigt werden konnte. Innerhalb des Prototypen wird aus zeitlichen Gründen der Fokus während der Programmierung nur auf eine der Zielgruppen gelegt werden können.

2.3.2.6 Anpassbarkeit und Erweiterbarkeit der Features (Entwicklung)

Der Teil der Ergebnisse der Recherche für die Erarbeitung des fachlich-didaktischen Konzepts der Webapplikation wurde in den Kapiteln [2.2.1](#) bis [2.2.3](#) diskutiert.

Der Teil der Ergebnisse, der nicht zum Proof-of-Concept gehört ([Kapitel 3.4](#)) und deshalb nicht innerhalb der Thesis programmiert wird, ist in [Kapitel 6.2.2](#) dargestellt und macht deutlich, dass eine große Fülle an Konzepten für zusätzliche Features bereits vorhanden ist und weiter anwachsen kann. Dementsprechend sollte die entwickelte digitale Applikation technisch in ihren Funktionen und Features anpassbar und erweiterbar sein.

⁸⁰ <https://de.statista.com/statistik/daten/studie/1194803/umfrage/anzahl-der-kitas-in-deutschland/>.
Datum des Abrufs: 14.01.2024.

3 Technische Konzeption

3.1 Client-Server-Architektur

Der in dieser Thesis entwickelte Prototyp der konzipierten Web-Applikation basiert auf der Client-Server-Architektur. Dabei handelt es sich um eine Netzwerkarchitektur.

Der Begriff Client meint die Anwendung bzw. das Computerprogramm, das auf der Hardware des Nutzers der Web-Applikation ausgeführt wird und das mit einem Server kommuniziert. Häufig wird jedoch das Endgerät selbst, über das der Nutzer des Endgeräts mit dem Server in Kontakt tritt, um dessen Dienste in Anspruch zu nehmen, Client genannt.

Der Server ist in diesem Zusammenhang das Computerprogramm, das auf dem Zentralrechner ausgeführt wird, mit dem der Client in Verbindung steht. Ähnlich wie im Falle des Clients kann auch hier der Zentralrechner selbst als Server bezeichnet werden.⁸¹

Client und Server kommunizieren über ein Netzwerk miteinander, im Falle einer schlüsselfertigen Webapplikation meist über das Internet. Die Client-Server-Architektur dient vor allem dem Datenaustausch zwischen Client und Server. Eine Client-Anwendung kann dabei so aufgebaut sein, dass sie nicht nur mit einem einzelnen, sondern mit mehreren Servern in Verbindung steht, um alle in der Client-Anwendung programmierten Funktionen erfüllen zu können. Genauso kann ein einzelner Server mit einer Vielzahl von Client-Anwendungen in den Datenaustausch treten.

Die Client-Anwendung stellt dabei eine Anfrage an einen Server, der diese Anfrage verarbeitet und - wenn der Server für diese Art von Anfrage programmiert wurde - versucht, die Aufgaben, die die Anfrage enthält, zu erfüllen. Daraufhin sendet der Server eine Antwort an den Client zurück, die neben einer Statusmeldung z.B. Daten enthalten kann, die der Client benötigt, um seine Funktionen zu erfüllen.⁸²

⁸¹ Geisler, S. 341-350.

⁸² Held et al. (2000). Kapitel 1: The Client/Server Architecture, S. 3.

3.2 Webapplikationen

3.2.1 Software-Architektur

Webapplikationen nutzen das Prinzip der Client-Server-Architektur ([Kapitel 3.1](#)). Dabei gibt es, verglichen mit nativen Applikationen, wichtige Unterschiede zu beachten. Die Architektur einer Webapplikation orientiert sich an dem Funktionsprinzip von dynamischen Webseiten, nur dass die Webapplikation zusätzlich dazu Business-Logik aufweist und dass die Nutzung der Webapplikation zu einer Änderung des Business-State (in Angular: Application State) des Systems führen kann ([Kapitel 4.1.1.2](#)).

Des Weiteren kann eine Webapplikation als System definiert werden, das aus den folgenden Komponenten zusammengesetzt ist: Browser, Netzwerk, HTTP-Protokoll und Web-Server. Dabei spielen in Webapplikationen die Eingaben des Benutzers eine wichtige Rolle, da durch diese Eingaben, welche sowohl Daten-Eingaben als auch die Navigation zwischen den verschiedenen Webseiten einschließen, der genannte Business State verändert wird, dessen Wesen und Nutzen innerhalb des Clients detaillierter in [Kapitel 4.1](#) dargestellt werden.

Die Modellierung der Business-Logik von Webapplikationen wie auch von nativen Applikationen kann mit der Modellierungssprache UML ([Kapitel 3.4.1](#)) erfolgen, die eingesetzt wird, um die Prozesse, die innerhalb des Softwaresystems vonstatten gehen, klar verständlich zu planen und zu dokumentieren.⁸³

3.2.2 Kommunikationsprotokoll HTTP

Das Kommunikationsprotokoll, das von Webapplikationen hauptsächlich für den Datenaustausch zwischen Client und Server genutzt wird, wird Hypertext Transfer Protocol (HTTP) genannt. Dieses Protokoll ist in erster Linie darauf ausgelegt, stabile Verbindungen mit großer Fehlertoleranz aufrechtzuerhalten.

Des Weiteren arbeiten Webapplikationen in ihrer Client-Server-Architektur mit Webseiten. Neben der Navigation zwischen verschiedenen Webseiten dient dieses Prinzip der Kommunikation zwischen Client und Server innerhalb von Webapplikationen.

Das Kommunikationsprinzip kann darauf heruntergebrochen werden, dass sogenannte Instanzen von Webseiten (englisch: Entities), d.h. letztlich Datenpakete, mit Hilfe von sogenannten HTTP-Anfragen von dem Client an den Server gesendet werden können. Der Server kann dabei mit entsprechenden Antworten Daten zurückschicken.⁸⁴

⁸³ Conallen (1999). S. 63-64.

⁸⁴ Conallen (1999). S. 63-64.

3.2.3 Deployment

Das Deployment einer Webapplikation erfordert v.a. die Einrichtung der serverseitigen Elemente des Systems, während auf der Client-Seite lediglich eine Internetverbindung bzw. - falls es sich um ein anderes Netzwerk handelt - die Verbindung zu dem genutzten Netzwerk sowie ein installierter Browser nötig sind, um Zugang zur Webapplikation zu haben.⁸⁵

3.2.4 Performanz

Studien aus dem Jahr 2019 zeigen, dass Webapplikationen imstande sind, mit der Performance und Rechengeschwindigkeit von nativen Desktop-Anwendungen (welche häufig Programmiersprachen wie z.B. Java sowie C++ nutzen) zu konkurrieren. Dies gelingt durch den Einsatz der V8-JavaScript-Engine (Begriffserklärung Engine: [Glossar](#)). Die Engine wird in zahlreichen JavaScript-Frameworks eingesetzt.

Weiterhin kann die V8-JavaScript-Engine in serverseitigen Web-Development-Plattformen wie Node.js genutzt werden. Durch diese Technologien kann JavaScript neben dem Einsatz als Script-Sprache für Webseiten zusätzlich für die Programmierung von Webservern verwendet werden.⁸⁶

3.3 Backend

3.3.1 Web-Server-Technologien

Die Entscheidung für eine spezifische Web-Server-Technologie kann nur unter Einbezug der potenziellen Nutzerzahl ([Kapitel 2.3.2.4](#)), des Ist-Status der IT-Infrastruktur und des IT-Personals an Schulen ([Kapitel 2.2.4](#)), der gegebenen Anforderungen an den Datenschutz ([Kapitel 2.2.6](#)) sowie in Zusammenarbeit mit den potenziellen Stakeholdern (Kultusministerien, einzelne Kommunen oder einzelne Schulen) der Webapplikation getroffen werden, sollte im Nachgang der Thesis deren Weiterentwicklung erwogen werden. Für diesen Entscheidungsprozess wichtig ist auch das Verständnis für den Unterschied zwischen dem Hosten der Webapplikation in der Cloud und dem Hosten auf lokalen Servern an der jeweiligen Schule ([Kapitel 2.2.5](#)).

Das innerhalb der Thesis implementierte Proof-of-Concept, mit dem die Benutzer-Anforderungen der ersten Entwicklungs-Iteration ([Kapitel 2.3](#)) getestet werden können, ist lediglich auf das Frontend angewiesen und benötigt keinen Web-Server.

⁸⁵ Conallen (1999). S. 63-64.

⁸⁶ Xing et al. (2019). S. 1.

Nichtsdestotrotz werden nachfolgend exemplarisch zwei verschiedene Web-Server-Technologien vorgestellt, die für die Weiterentwicklung der in der vorliegenden Thesis konzipierten Webapplikation infrage kommen könnten, da Webserver ein Kernstück der Client-Server-Architektur darstellen ([Kapitel 3.1](#)).

Webserver sind Computer, die über Adressen, sogenannte URLs (Uniform Resource Locator) erreicht und angesprochen werden können. Die Aufgabe eines Webserver kann wie folgt definiert werden: Der Webserver kann zum Einen die URLs, die er vom Client über das Internet zugesendet bekommt, in Dateinamen übersetzen und die dazugehörigen Dateien dann als Antwort zurücksenden. Zum Anderen kann es sich bei diesen Dateien um Programme handeln, die der Webserver auszuführen hat und deren Ausgabe er als Antwort an den Client zurücksendet. Dabei wird zur Kommunikation das Protokoll HTTP verwendet ([Kapitel 3.2.2](#)). Die genannten URLs sind nach dem folgenden Muster strukturiert:

`<scheme>://<host>/<path>`

Die Angabe des `<scheme>` teilt dem genutzten Webbrowser (Browser siehe [Glossar](#)) demnach mit, welches Kommunikationsprotokoll verwendet werden soll, in diesem Fall HTTP.

Der `<host>` ist z.B. die Suchmaschine mit dem Namen www.google.de.

Das letzte Element `<path>` gibt schließlich an, welche Webseite des Hosts, d.h. welche der Ressourcen, die der Host auf seinen Webservern bereitstellt, aufgerufen werden soll.⁸⁷

3.3.1.1 Node.js

Node.js ist eine auf Netzwerkoperationen ausgelegte Laufzeitumgebung, mit der Web-Server programmiert werden können. Des Weiteren verfügt Node.js über folgende Merkmale.

Node.js ist Open-Source und wird dementsprechend unter einer Lizenz veröffentlicht, die es den Nutzern des Softwareprogramms erlaubt, den Quellcode zu lesen und ihn auf individuelle Art und Weise zu nutzen, d.h. ihn zum Beispiel zu verändern und in neuer Form zu veröffentlichen.⁸⁸

Zudem kann Node.js plattformübergreifend (engl. cross-platform) verwendet werden und verfügt durch den Einsatz der V8-JavaScript-Engine, die auch von Google Chrome genutzt wird, über eine große Performanz. Des Weiteren nutzen Node.js-Bibliotheken überwiegend nicht-blockierende Programmier-Paradigmen, die auf den asynchronen Input- und Output-Operationen der Node.js Standard-Bibliothek aufbauen. Durch diese Features kann in den meisten Fällen, wenn genannte Operationen durchgeführt werden müssen, auf das

⁸⁷ Laurie et al. (2002). S. 16.

⁸⁸ Crowston et al. (2012). Seite 1.

Blockieren des Threads verzichtet und damit CPU-Leistung eingespart werden. Trotzdem kann ein einzelner Node.js-Server tausende von Client-Server-Verbindungen managen.⁸⁹

Dabei steigt die Performanz von Node.js bei Software- und Benchmark-Tests mit einer größer werdenden Anzahl an Input-Output-Operationen im Vergleich zu anderen Web-Development-Technologien wie PHP und Python. Das bedeutet, dass Node.js im Vergleich zu den genannten Technologien mehr Anfragen innerhalb eines gegebenen Zeitabschnitts bearbeiten kann. Dementsprechend ist der Einsatz von Node.js gerade für Webapplikationen mit einer großen Anzahl aktiver Nutzer und damit einer großen Zahl an Input-Output-Operationen sinnvoll.⁹⁰

Die angesprochenen nicht-blockierenden Programmierparadigmen haben den Vorteil, dass pro Client-Server-Verbindung nur ein einzelner Thread (Prozess) erstellt werden muss, wodurch der Overhead von Node.js sehr gering ist. Erreicht eine HTTP-Anfrage den Node.js-Server mit der Aufforderung, eine Operation durchzuführen, leitet der Server diese Aufgabe an die Ereignisschleife weiter, in der sie abgearbeitet wird. Der Server registriert lediglich eine Callback-Funktion und kann mit der Bearbeitung anderer Operationen weiter verfahren.

Demnach handelt es sich um einen asynchronen Vorgang und der Main-Thread des Webservers fungiert sozusagen als Mittelsmann, der verschiedene Aufgaben (Rechenoperationen, Datenbankabfragen) an verschiedene Hintergrund-Threads delegiert.

Der jeweilige Hintergrund-Thread bearbeitet die erhaltene Aufgabe, führt den angegebenen Callback aus und gibt nach dessen Abschluss das Ergebnis an den Main-Thread zurück. Demnach ist Node.js für Webapplikationen geeignet, in denen eine große Anzahl von Input-Output-Operationen abgearbeitet werden muss, ohne den Main-Thread und damit die Nutzung des Clients durch den Benutzer der Webapplikation zu blockieren.⁹¹

3.3.1.2 Apache

Die Webserver-Technologie Apache ist kostenlos und open-source nutzbar ([Kapitel 3.3.1.1](#)).

Die große Anzahl an Personen, die an der Weiterentwicklung von Apache mitwirken, arbeiten sowohl an neuen Funktionen als auch an der Verbesserung und Aufrechterhaltung von Themen wie der IT-Sicherheit. Apache kann für Websites verschiedener Größen und Typen verwendet werden. Dabei werden die genannten typischen Anwendungsfälle von Webservern abgedeckt, wie das Versenden von statischen Dateien über das Internet ([Kapitel 3.3.1](#)).

⁸⁹ <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>. Datum des Abrufs: 29.11.2023.

⁹⁰ Lei et al. (2014). S. 661-663.

⁹¹ Chhetri (2016). S. 9-10.

Dabei ist Apache wie auch Node.js für das Betreiben von lokalen Testservern geeignet, mit denen Softwareentwickler ihren Programmcode in einer lokalen Umgebung testen können.⁹²

Wie bereits dargelegt, nutzt Node.js pro Client-Server-Verbindung einen einzelnen Thread, der in einer Ereignisschleife abläuft und die Anfragen (HTTP-Requests) des Clients mit Hilfe von nicht-blockierenden Programmierparadigmen bearbeitet. Dagegen erzeugt Apache für jede Anfrage des Clients an den Server einen neuen Thread, der dementsprechend seinen eigenen Zustand hat.⁹³ Nachfolgend sollen einige Vorteile von Apache aufgezählt werden:

Stabilität: Das thread- bzw. prozessbasierte Modell von Apache kann in Umgebungen mit begrenzten Ressourcen stabiler sein. Jeder Thread oder Prozess behandelt eine Anfrage separat, was zu einer klaren Trennung der Zustände führt und potenzielle Probleme isoliert.

Kompatibilität: Des Weiteren hat sich Apache seit langer Zeit bewährt und ist demnach ein weit verbreiteter Webserver. Viele bestehende Webapplikationen und -frameworks wurden für die Verwendung mit Apache optimiert und sind möglicherweise besser kompatibel.

Flexibilität: Apache unterstützt verschiedene Module und Konfigurationsoptionen, die es ermöglichen, den Server an unterschiedliche Anforderungen anzupassen. Dies umfasst Funktionen wie Lastausgleich, Authentifizierung, Verschlüsselung und mehr, die in der Konfiguration des Servers aktiviert werden können.

Integration: Apache ist oft Teil von Software-Stacks und -Lösungen, die eng mit anderen Open-Source-Technologien wie PHP, MySQL und Perl kooperieren. Diese Integration kann die Entwicklung, Bereitstellung und Verwaltung von Webanwendungen vereinfachen.⁹⁴

3.3.2 Datenbanken

Datenbanken sind innerhalb einer IT-Infrastruktur v.a. dafür verantwortlich, Daten sowie Informationen über die Zusammenhänge zwischen diesen Daten zu sichern. So könnte eine Datenbank innerhalb eines Unternehmens z.B. die Rechnungsnummern einzelner Aufträge speichern, indem jeder Rechnungsnummer gleichzeitig der jeweilige Kunde, das Datum des Eingangs des Auftrags, das Datum der Lieferung usw. zugeteilt ist. Werden diese Daten jeweils getrennt voneinander betrachtet, handelt es sich lediglich um einzelne Fakten bzw. Zeichenfolgen, erst der Kontext und der Zusammenhang zwischen den Daten machen dieses Datenpaket zu einer wertvollen Information.⁹⁵

⁹² Laurie et al. (2002). S.18.

⁹³ Chhetri (2016). S. 9-10.

⁹⁴ Laurie et al. (2002). S. 18-44.

⁹⁵ Geisler (2014). S. 19-24.

3.4 System-Anforderungen des Prototypen (Proof-of-Concept)

Der zur Demonstration des Konzepts der Webapplikation programmierte Prototyp wird nach dem Prinzip des sogenannten „Proof-of-Concept“ entwickelt. Dabei werden die notwendigen Kern-Features eines Softwareprodukts implementiert, um daraufhin, z.B. in Nutzertests ([Kapitel 6.2.1](#)) überprüfen zu können, ob die Bedürfnisse der Nutzer, d.h. die postulierten Benutzer-Anforderungen, die an das Softwareprodukt gestellt wurden, erfüllt werden. Während des Prozesses der Erstellung des Proof-of-Concept kann außerdem die Eignung verschiedener Technologien für das Erreichen der Design-Ziele überprüft werden.⁹⁶

Die System-Anforderungen legen die Design-Ziele der ersten Iteration der Entwicklung der digitalen Anwendung fest, die angestrebt werden. Konkret heißt das, dass diejenigen Benutzer-Anforderungen ([Kapitel 2.3](#)), die für das Proof-of-Concept notwendig sind, für das Erstellen von Arbeitspaketen sowie zum Zwecke des besseren und langfristigen Verständnisses der technischen Prozesse, die dahinter stecken, in diesem Kapitel in technische Sprachen übersetzt werden und eine klar strukturierte Vorgehensweise für die Entwicklung des Prototypen erarbeitet wird. Die genannten Prozesse werden zunächst mit der Modellierungssprache UML dargestellt ([Kapitel 3.4.1](#)).

3.4.1 Prozesse (UML-Diagramme)

Die Modellierungssprache UML (Unified Modeling Language) eignet sich für die Planung von Webapplikationen sowie für die Dokumentation des Programmier-Prozesses, da UML das Verständnis für die Funktionsweise von komplexen und software-intensiven Systemen durch Visualisierung und Abstraktion von Komponenten und Prozessabläufen verbessern kann.

Dabei bietet UML nicht nur eine Form der Modellierung, sondern je nach dem Blickwinkel, aus dem das System betrachtet werden soll, viele verschiedene Abstraktions-Möglichkeiten. Das Software-System kann mit UML dementsprechend sowohl sehr oberflächlich als auch sehr detailliert betrachtet werden. Das Ziel der Modellierung ist generell jedoch, gerade die Details des Software-Systems so verständlich wie möglich darzustellen.

Zusätzlich zu den UML-Diagrammen, die auch für die Entwicklung von nativen Softwareapplikationen genutzt werden, kann für Webapplikationen die sogenannte Sitemap eingesetzt werden, um die einzelnen Webseiten und deren Navigations-Routen darzustellen.

Für die Entwicklung von Webapplikationen ist die Trennung zwischen der rein grafischen Darstellung, d.h. dem Layout, das meist in den Programmiersprachen HTML und CSS

⁹⁶ Prasanna et al. (2021). S. 1.

implementiert wird, und der Business-Logik (Programmiersprachen: JavaScript, TypeScript) hervorzuheben. Diese Trennung sollte auch in der Modellierung von Webapplikationen mit UML beachtet werden. UML ist vor allem für die Modellierung der Business-Logik geeignet, das Layout und die grafische Darstellung sollten aber nicht vernachlässigt werden.⁹⁷

3.4.2 Anwendungsfalldiagramm

Zu den für das Proof-of-Concept entscheidenden Benutzer-Anforderungen, die in [Kapitel 2.3](#) als Anwendungsfälle (Use-Cases) aus Sicht des Nutzers erklärt wurden, zählen der Text-Editor mit der Möglichkeit, Bilder daraus generieren zu lassen sowie die Rechtschreibprüfung durchführen zu lassen und die Listenansicht mit der Möglichkeit, die bereits vorhandenen Kurzgeschichten in einer Liste zu betrachten und weiterzuschreiben.

Diese Anwendungsfälle werden an dieser Stelle zum besseren Verständnis mit einem Anwendungsfall-Diagramm (englisch: Use-Case-Diagram) dargestellt. Das heißt konkret, dass darin die verschiedenen Aktionen gezeigt werden, die ein Nutzer in der Applikation ausführen kann. Jede der funktionalen Benutzer-Anforderungen ([Kapitel 2.3](#)) wird dementsprechend als separater Anwendungsfall innerhalb des Anwendungsfall-Diagramms repräsentiert. Dieses UML-Diagramm stellt allerdings ausschließlich die Perspektive des Nutzers der Webapplikation dar und nicht die Form der technischen Funktionalität innerhalb des Software-Systems.⁹⁸

Die folgenden Benutzer-Anforderungen werden dargestellt:

- „Der Nutzer kann sich die Liste mit den Templates anzeigen lassen.“ Dieser Anwendungsfall beinhaltet die Möglichkeit, auf eine Reihe der Liste zu klicken und damit in den Text-Editor zu wechseln, in dem der Inhalt des angeklickten Templates angezeigt wird. Diese Art von Templates sind angefangene Kurzgeschichten. Zusätzlich darf der Nutzer leere Templates erstellen.
- „Der Nutzer kann Kurzgeschichten verfassen.“ Dieser Anwendungsfall schließt die Fälle ein, dass Templates bzw. Kurzgeschichten verändert werden können, d.h. Text gelöscht sowie Text geschrieben werden kann.
- „Der Nutzer kann die Rechtschreibprüfung durchführen lassen.“ Dieser Anwendungsfall schließt mit ein, dass die Textstellen mit Grammatikfehlern markiert werden, z.B. rot unterstrichen werden, was für den Nutzer sichtbar ist.

⁹⁷ Conallen (1999). S. 64-65.

⁹⁸ Eriksson (2004). S. 24.

- „Der Nutzer kann sich Bilder zum Text erstellen lassen.“ Diese Bilder erscheinen danach und dürfen betrachtet werden.

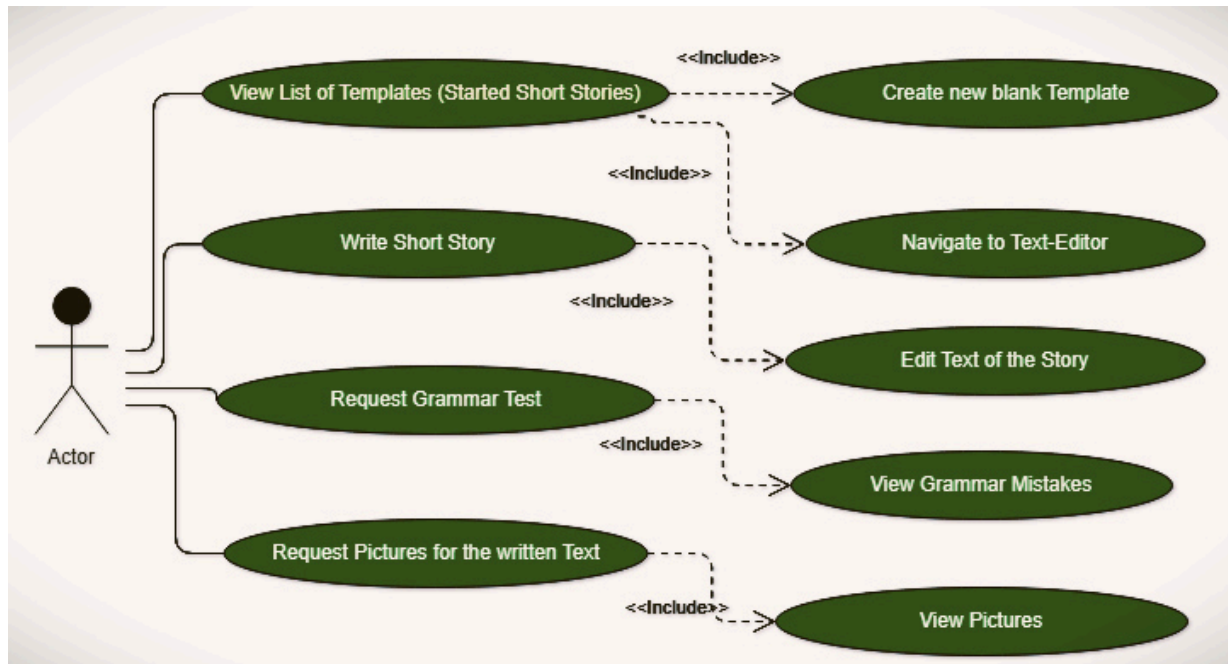


Abbildung 2: Anwendungsfalldiagramm (englisch: Use-Case-Diagramm). Darstellung von: Autor der Thesis.

3.4.3 Aktivitätsdiagramm

Aktivitätsdiagramme (englisch: Activity Diagrams) zeigen die Abfolge der Prozesse der funktionalen Anforderungen. Der Ablauf dieser Prozesse (englisch: Workflow) verdeutlicht, welche Schritte technisch innerhalb des Systems erfolgen müssen, damit ein Nutzer z.B. eine Kurzgeschichte verfassen oder die Rechtschreibprüfung durchführen lassen kann.

Das Aktivitätsdiagramm wird dabei von Aktionen zusammengesetzt, die sozusagen in einer Folge hintereinander geschaltet sind. Dieser Kontrollfluss des Software-Systems kann mit Hilfe von sogenannten Tokens dargestellt werden. Damit kann z.B. gezeigt werden, wann und wie das System auf gewisse Ereignisse und Nachrichten reagiert, sobald diese das System von außerhalb erreichen. Das System kann darauf reagieren, indem es zum Beispiel seinerseits Nachrichten oder Objekte versendet. Die Prozesse sind zusätzlich als Arbeitspakete, die für den Prototypen programmiert werden müssen, in Textform dargelegt ([Kapitel 3.4.4](#)).⁹⁹

⁹⁹ Eriksson et al. (2004). S. 28-29.

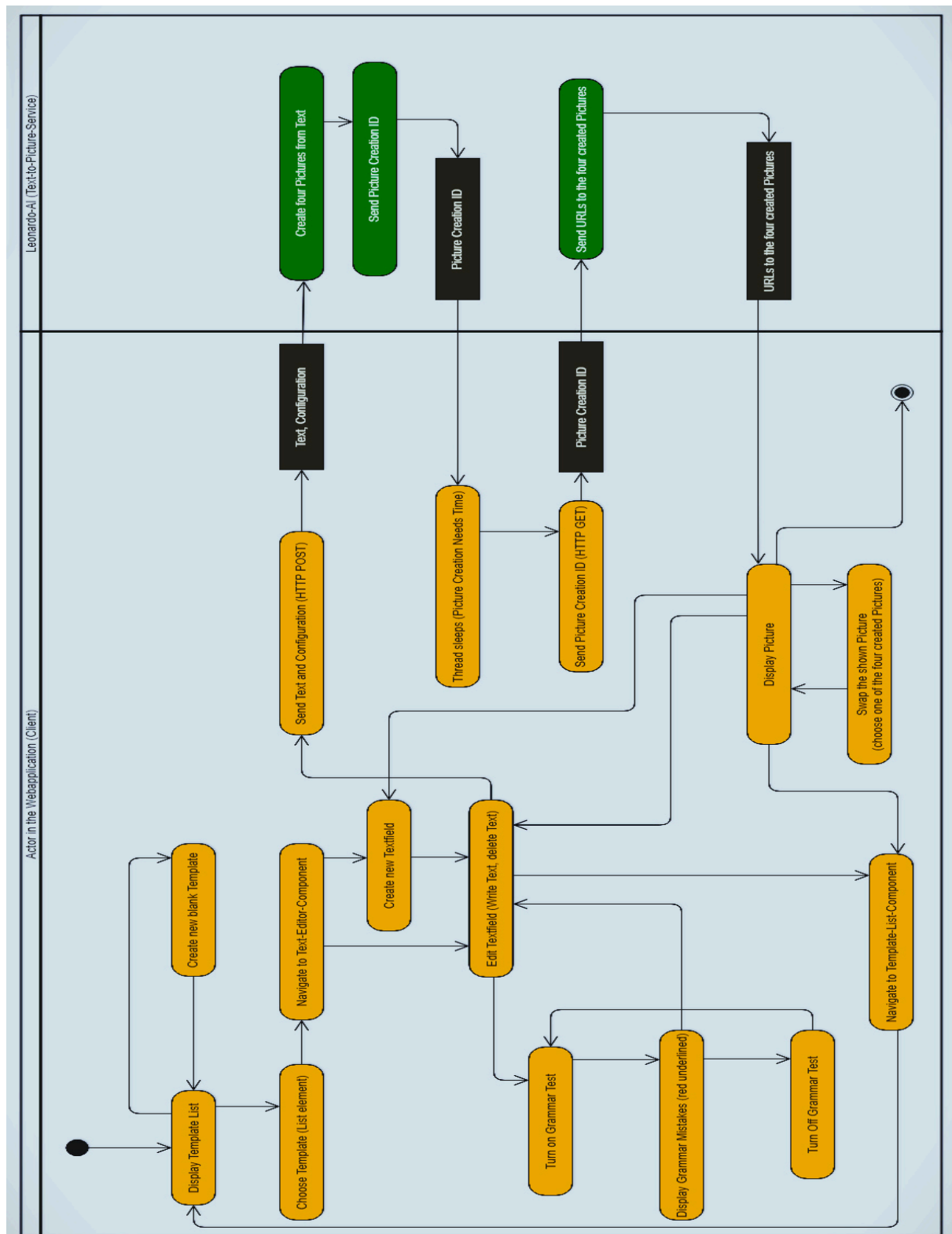


Abbildung 3: Aktivitätsdiagramm (englisch: Activity-Diagram). Darstellung: Autor der Thesis.

3.4.4 Arbeitspakete

Der Entwicklungs-Prozess zum Erreichen der in den Benutzeranforderungen spezifizierten Ziele ([Kapitel 2.3](#)) lässt sich auf klar definierte technische Arbeitsschritte abbilden.

Diese technischen Arbeitsschritte definieren dabei die innerhalb des Aktivitätsdiagramms dargestellten Prozesse ausführlich in Textform. Diese Prozesse werden für die Programmierung des Prototypen Schritt für Schritt in Programmcode umgesetzt.

Jedes Arbeitspaket ist mit einer eindeutigen ID gekennzeichnet, die in dem Programmcode, der in dem dazugehörigen Git-Repository gespeichert und einsehbar ist, an den Stellen der Implementierung referenziert wird ([Kapitel 5.3](#)).

Die Arbeitspakete der System-Anforderungen sind mit Nummern gekennzeichnet, die den deklarativen Zusatz der technischen Kategorie, zu der sie gehören, enthalten.

Kategorie: GUI - Graphical User Interface (Begriffserklärung: [Glossar](#))

Listenanzeige der Templates bzw. Kurzgeschichten (GUI.001)

Template-List-Komponente:

Die Template-List-Komponente dient als Startpunkt der Webapplikation. Darin wird eine Liste der Templates bzw. Kurzgeschichten angezeigt. Diese in der Liste zum Start vorhandenen Daten sind lokal in den Dateien (englisch: Assets) der Webapplikation verpackt.

Hinzufügen von neuen Kurzgeschichten (GUI.002)

Template-List-Komponente:

Die Liste in der Template-List-Komponente, die die vorhandenen Templates für Kurzgeschichten anzeigt, enthält zusätzlich einen Button, der mit dem Plus-Symbol und zusätzlich mit dem Text „Neue Kurzgeschichte starten“ dargestellt wird. Klickt der Nutzer diesen Button, erscheint ein Pop-Up-Fenster, in dem der Nutzer mit einem Info-Text aufgefordert wird, den Namen seiner Kurzgeschichte in das Text-Eingabefeld einzutragen und mit dem Button „OK“ den Prozess zu bestätigen. Mit dieser Funktion kann der Nutzer sich ein neues, leeres Template erstellen lassen, für das er lediglich den Namen einzutippen hat. Wird der Button „OK“ geklickt, wird das Pop-Up-Fenster geschlossen und der Name des

neuen Templates erscheint räumlich nach seiner Zeilennummer als neue Reihe in der Liste. Der Button mit dem Plus-Icon verschiebt sich dafür eine Zeile nach unten.

Das Pop-Up-Fenster enthält weiterhin den Button „Abbrechen“.

Der Klick auf den Button „Abbrechen“ schließt schlichtweg das Pop-Up-Fenster.

Funktionsnamen: createNewTemplate.

Exkurs: Damit kann letztendlich die komplette Geschichte in Eigenregie verfasst werden und es wird, zumindest innerhalb der Webapplikation, auf den ersten Schreibimpuls ([Kapitel 2.2.1](#)) verzichtet. Diese Funktion (leeres Template) gibt den Nutzern bzw. auch den Lehrern die Chance, den didaktischen Rahmen festzulegen und den Schreibimpuls zunächst unabhängig von der Webapplikation vorzugeben. Z.B. kann es sein, die Klasse war in der Natur spazieren oder hat einen Kinofilm angeschaut und will nun diese Erlebnisse als Schreibimpuls nutzen. Zusätzlich könnte in diesem Fall in Teams gearbeitet werden, indem der eine Schüler die Kurzgeschichte mit seinem Text anfängt, und der andere Schüler diesen Text als Schreibimpuls nutzt und weiter schreibt.

Navigation in die Text-Editor-Komponente (GUI.003)

Template-List-Komponente:

Die in Template-List-Komponente dargestellten Kurzgeschichten sind anklickbar. Wird die Reihe mit dem Namen einer Kurzgeschichte angeklickt, navigiert die Webapplikation in die Text-Editor-Komponente, in der die ausgewählte Kurzgeschichte editiert werden kann.

Funktionsname: navigateToTextEditor.

Editieren von Kurzgeschichten (GUI.004)

Text-Editor-Komponente:

Die Text-Editor-Komponente enthält zum einen feste Textfeldern, die jeweils einen bereits verfassten Abschnitt der Kurzgeschichte beinhalten. Von diesen festen Textfeldern kann es pro Geschichte je nach Länge der Geschichte eine Vielzahl geben.

Hat der Nutzer sich für ein Template mit einer Vorlage entschieden, sind darin (pro Geschichte) bereits einige solche Textabschnitte mit Bildern vorhanden, d.h. werden durch den Entwickler bereitgestellt. Das soll dem Nutzer der Schreibimpuls bzw. die Inspiration sein, die in der Methode des kreativen Schreibens eingesetzt wird ([Kapitel 2.2.1](#)).

Neben den Textfeldern sind jeweils Buttons mit dem Layout des Stift-Icons platziert.

Wird der Stift-Button geklickt, wechselt das feste Textfeld in den dynamischen Modus und ist damit editierbar. Der Nutzer kann zu diesem Zeitpunkt den Text darin verändern. Zudem darf kein anderes Textfeld gleichzeitig verändert werden. Statt des Stift-Icon-Buttons erscheint zusätzlich neben dem dynamisch gewordenen Textfeld ein Button mit Check-Mark-Icon. Wird der Button geklickt, wird das Textfeld fest und kann nur durch Klicken des Stift-Icons, das dann wieder sichtbar ist, erneut editiert werden.

Die Stift-Buttons neben den anderen Textfeldern sind nicht sichtbar (sind dem aktuell sichtbaren, d.h. zuletzt gerenderten HTML-Template nicht hinzugefügt worden), solange sich ein anderes Textfeld innerhalb des dynamischen Modus befindet (englisch: Edit-Mode).

Funktionen: startEditMode, stopEditMode, saveTextField.

Generieren von Bildern (GUI.005)

Text-Editor-Komponente, Text-To-Picture-Service:

Zusätzlich zum Stift-Icon-Button ist weiter rechts davon neben jedem Textfeld der Rahmen für die Bilder positioniert. Daneben ist der Button mit dem Text „Bilder generieren.“ platziert. Wird der Button geklickt, wird der Text des dazugehörigen Text-Abschnitts (d.h. derselben Reihe) an den Server des genutzten Text-zu-Bild-Generator-Dienstes geschickt. Zu diesem Zweck wird die API (Programmierschnittstelle, [Glossar](#)) dieses Dienstes genutzt, indem eine HTTP-Post-Anfrage an den Dienst mit dem Text des Nutzers gesendet wird.

Der Dienst antwortet im Idealfall, indem er vier Bilder generiert, die auf dem Text des Nutzers basieren, und die ID des Prozesses der Bildgenerierung an den Client bzw. das Frontend sendet. (Falls etwas nicht funktioniert, kann es nämlich sein, dass der Dienst mit einer Fehlermeldung antwortet.)

Daraufhin wird eine HTTP-Get-Anfrage an den Server des Dienstes gesendet, die die ID des genannten Prozesses enthält und die Links zu den vier kreierten Bildern anfordert. Diese Links werden nun in das zugehörige Textfeld-Objekt gespeichert und dem Nutzer wird dann

ein Bild davon innerhalb des grafischen Benutzer-Interfaces angezeigt, und zwar rechts von dem zugehörigen Textfeld, aber nicht so weit rechts wie der Button „Bilder generieren.“, d.h. in der Mitte der beiden anderen Elemente. Der Nutzer kann außerdem zwischen den vier Bildern wechseln, indem er den Pfeil neben dem Bilderrahmen klickt. Der Button “Bilder generieren.” verschwindet derweil, d.h. ist nicht mehr sichtbar, nachdem er geklickt wurde. Er wird allerdings wieder sichtbar, wenn das zugehörige Textfeld erneut editiert wird, d.h. dem Nutzer wird die Chance gegeben, neue Bilder generieren zu lassen, falls er den Text ändert. Wird der Button „Bilder generieren.“ geklickt, wird außerdem eine Ladeanimation sichtbar.

Funktionsnamen:

createPicturesFromText (Text-Editor-Komponente).

createPicturesfromTextwithLeonardoAI (Text-To-Picture-Service).

getCreatedPicturesFromLeonardoAI (Text-To-Picture-Service).

Rechtschreibprüfung (GUI.006)

Text-Editor-Komponente:

Direkt neben dem Button mit dem Stift-Icon befindet sich ein weiterer Button mit dem Text „Grammatik-Test einschalten“. Mit dem Klick auf diesen Button wird die Rechtschreibprüfung für genau dieses Textfeld durchgeführt und nur in diesem Textfeld werden dann grammatikalisch falsche Wörter rot unterstrichen. Der Button wandelt sich derweil und heißt, solange die Funktion eingeschaltet ist, „Grammatik-Test abschalten.“, damit der Test zielgerichtet nach dem Bedarf des Nutzers ein- und ausgeschaltet werden kann.

Exkurs: Je nach Alter des Nutzers, könnte die korrekte Rechtschreibung als Voraussetzung dienen, überhaupt Bilder generieren lassen zu dürfen. Das heißt konkret, sind noch Rechtschreibfehler innerhalb eines Textfeldes vorhanden und der Nutzer klickt auf den diesem Textfeld zugehörigen Button „Bilder generieren.“, könnte eine Fehlermeldung erscheinen mit dem Text: „Ups. Da haben sich wohl noch Rechtschreibfehler eingeschlichen. Bitte korrigiere die Rechtschreibfehler, damit das Bild korrekt generiert werden kann.“

Diese sozusagen strenge Version der Webapplikation könnte den Nutzer der Zielgruppe „Weiterführende Schulen: 10+ Jahre alt“ ([Kapitel 2.1.1](#)) gelten, während die „softe“ Version der Webapplikation, in der die Rechtschreibprüfung lediglich eine freiwillige Zusatz-Hilfe darstellt, den Nutzern der Zielgruppe „Grundschulkinder: 6-10 Jahre alt“ ([Kapitel 2.1.2](#)) gilt.

Gänzlich wegfallen könnte dieses Feature dagegen für die Kleinsten, die Zielgruppe „Vorschulkinder: 3-6 Jahre alt“ ([Kapitel 2.1.3](#)). Das Thema der Anpassbarkeit der grafischen Benutzeroberfläche und der Business-Logik an das Alter des Nutzers zur Laufzeit der Webapplikation ([Kapitel 2.3.2.4](#)) und das Thema Anpassbarkeit und Erweiterbarkeit der Features während der Entwicklung ([Kapitel 2.3.2.5](#)) wurden bereits angeschnitten.

Funktionsnamen: checkGrammar.

Navigation in die Template-List-Komponente (GUI.007)

Text-Editor-Komponente:

Der Nutzer kann den Button „Zurück ins Hauptmenü“ klicken. Der Client navigiert damit in die Template-List-Komponente. Dieser Button ist oben rechts sowie unten rechts an den Rändern des Bildschirms platziert.

Funktionsnamen: navigateToTemplateList.

Exkurs: Wird der Button geklickt, gehen die Änderungen am Inhalt der Textfelder, die der Nutzer innerhalb der Kurzgeschichte auf seinem Endgerät vorgenommen hat, verloren, da für den Prototypen auf das Aufsetzen eines Webserver und einer Datenbank verzichtet wurde.

Sofern eine Weiterentwicklung der Webapplikation, die auf den in der vorliegenden Bachelor-Thesis erarbeiteten Grundlagen basiert, angestrebt wird, könnten die Änderungen, die der Nutzer an der Kurzgeschichte vorgenommen hat und die neuen Textpassagen, die er oder sie verfasst hat, jedoch an das Backend gesendet und von dort aus in einer Datenbank gespeichert werden. Genauso könnten die generierten Bilder dort gespeichert werden.

Die Datenstruktur könnte dann wie folgt aussehen: Jede einzelne Kurzgeschichte ist ein Objekt, das sowohl eine Liste aus Textpassagen als auch eine Liste aus Bildern enthält, während jedes Listenelement der Bilderliste zusätzlich die Information gespeichert hat, zu welcher Textpassage es gehört. Der Name des Kurzgeschichtenobjekts ergibt sich dabei aus dem Namen der Kurzgeschichte innerhalb des Clients.

4 Technische Grundlagen

4.1 Client (Frontend)

4.1.1 Web-Development-Plattform Angular

Der in der vorliegenden Arbeit entwickelte Prototyp nutzt für die Programmierung die Development-Plattform Angular. Diese setzt sich aus drei großen Teilen zusammen.

1. Das Framework, das es dem Entwickler erlaubt, aus Komponenten zusammen gesetzte und dadurch modularisierte und skalierbare Web-Applikationen zu realisieren.
2. Des Weiteren enthält die Plattform Bibliotheken mit Funktionalitäten für z.B. Routing, Formular-Management und Layout-Design (z.B. Material Design).
3. Gleichzeitig gibt es eine Vielzahl von Werkzeugen - eingebaut in die Angular CLI (Command Line Interface) - für Web-Entwickler, die u.a. das Programmieren, das Testen und das Instandhalten von Webapplikationen unterstützen.^{100, 101}

Nachfolgend werden einzelne Elemente des Angular-Frameworks näher beleuchtet.

4.1.1.1 TypeScript

Die in Angular verwendete Programmiersprache ist TypeScript und wurde von Microsoft entwickelt. TypeScript ist ein sogenanntes “Superset” von JavaScript, d.h. dass TypeScript die Eigenschaften von JavaScript umfasst und es gleichzeitig um Features wie z.B. Typsicherheit, Klassen und Annotationen erweitert.¹⁰²

Der Einsatz der von TypeScript eingeführten Typisierung von Variablen ist zwar optional, kann Entwicklern aber einige Vorteile bieten. Durch das Anwenden der Typisierung können einfache Fehler in der Syntax schon während des Programmier-Prozesses erkannt werden (und nicht erst zur Laufzeit innerhalb des Browsers), womit die Fehleranfälligkeit des Programmiercodes und damit der Zeitaufwand für die Fehlersuche reduziert werden können. Zusätzlich sorgt die Typisierung von Variablen für ein schnelles Verständnis des Nutzens der Variablen, z.B. steht der Typ „number“ für einen Zahlenwert. TypeScript-Code wird für den

¹⁰⁰ <https://angular.io/guide/what-is-angular>, Datum des Abrufs: 3.12.2023.

¹⁰¹ Wilken (2018). S. 5-6.

¹⁰² Höller (2019). S. 33.

Browser zu JavaScript-Code kompiliert und kann dabei mit jeder JavaScript-Version ab ECMAScript 3 aufwärts genutzt werden.¹⁰³

4.1.1.2 Komponenten

Die Kernstruktur einer Angular-Anwendung lässt sich als Baum von Komponenten beschreiben. Die Komponenten unterliegen einer hierarchischen Ordnung. Dabei ist die sogenannte “AppComponent” der Startpunkt dieser Ordnung und damit der Startpunkt der Angular-Anwendung. Dies wird dadurch sichergestellt, dass die “AppComponent” eine bootstrap-Eigenschaft inhabitiert, die sie als Startpunkt ausweist.¹⁰⁴

Jede dieser Komponenten in Angular-Applikationen besitzt grundsätzlich die folgenden vier Teile, deren (Programmcode-)Inhalt von Komponente zu Komponente meistens variiert:

- Das Template, in dem die Layout-Struktur der zur Komponente gehörenden Webseite, die dargestellt werden soll, mit der Programmiersprache HTML definiert ist.
- Die TypeScript-Klasse, in der die Logik und das Verhalten der Frontend-Schnittstellen enthalten sind.
- Den CSS-Selektor, in dem die Art und Weise der Nutzung der Komponente innerhalb eines Templates definiert wird.
- Die CSS-Datei, die für die grafische Manipulation des Templates genutzt werden kann, aber nicht unbedingt genutzt werden muss.¹⁰⁵

Anhand der hierarchischen Komponentenstruktur kann ein zentrales Architekturprinzip von Angular erläutert werden. Das Framework hat Kenntnis bzw. Daten darüber, welche Verbindungen zwischen den Komponenten einer Applikation bestehen und kann den Entwickler so dabei unterstützen, Änderungen des Programmcodes effizient durchzuführen und z.B. zu warnen, falls eine geplante Änderung des sogenannten „Application State“ (Zustand der Applikation) einen anderen Teil der Applikation (negativ) betrifft.

Darüber hinaus dient der Komponentenbaum der Steuerung und der Übersicht des Datenflusses einer Angular-Applikation. Daten fließen sozusagen an der hierarchischen Struktur des Komponentenbaums herauf- oder hinab. Dementsprechend wird von einer „Kind-Komponente“ gesprochen, wenn diese Komponente Daten von einer in der Hierarchie über ihr (d.h. näher an der Root-Komponente) stehenden Komponente erhält. Dieses Konzept

¹⁰³ Wilken (2018). S. 23.

¹⁰⁴ Höller (2019). S. 99-100.

¹⁰⁵ <https://angular.io/guide/component-overview>. Datum des Abrufs: 14.12.2023.

der Datenübertragung von einer Eltern-Komponente zu einer Kind-Komponente wird „Input-Binding“ genannt.¹⁰⁶

4.1.1.3 Das reaktive Programmierparadigma

Zum Verständnis des reaktiven Programmierparadigmas, das durch die Programmbibliothek RxJS ([Kapitel 4.1.1.5](#)) in das Angular-Framework integriert ist, wird an dieser Stelle das dazu differierende imperative Programmierparadigma erklärt. Dadurch können Unterschiede klar aufgezeigt werden und der Charakter des reaktiven Programmierens wird deutlicher. Daraufhin wird das reaktive Programmierparadigma anhand seiner Ursprünge (deklaratives, funktionales und datenfluss-orientiertes Programmierparadigma) dargestellt.

Das imperative Programmierparadigma steht im Gegensatz zu den restlichen oben genannten Paradigmen, dient aber als Grundlage viel genutzter Programmiersprachen und soll als erstes erläutert werden:

Bei imperativen Programmiersprachen wie z.B. Java und C++ wird der Status einer Applikation dadurch entscheidend bestimmt und verändert, dass den in dem zugehörigen Programmcode definierten Variablen bestimmte Werte zugewiesen werden (auch mehrmals), während der Programmfluss u.a. durch Kontrollmechanismen wie z.B. if-Abfragen gesteuert wird. Somit können für gegebene Input-Daten Berechnungen durchgeführt werden und die Ergebnisse wiederum als Output-Daten zurückgegeben werden, indem Variablen erstellt werden und ihnen ein Zustand bzw. Wert zugewiesen wird, der dann durch eine neue Zuweisung wieder verändert werden kann. Der Programmcode gleicht insgesamt einer in zeitlich diskreten Abschnitten definierten Schritt-für-Schritt-Anleitung, nur dass es sich um Befehle an den Computer handelt.¹⁰⁷

Das reaktive Programmierparadigma dagegen setzt nicht (nur) auf die genau festgelegte zeitliche Abfolge von Instruktionen, es wird also nicht durchgehend vorgeschrieben, in welcher Reihenfolge und zu welchem Zeitpunkt das Programm bzw. letztlich die Hardware eine Funktion, z.B. eine Berechnung auszuführen hat. Stattdessen steht die Beschreibung und die Lösung eines Problems durch Funktionen im Vordergrund, weshalb die reaktive Programmierung (RP) eng verwandt mit dem deklarativen und mit dem funktionalen Programmierparadigma ist. Dementsprechend können reaktive Programme zu einem großen Teil aus Funktionen zusammengesetzt werden, die mit Eingabewerten Rechenprozesse durchführen und Ausgabewerte zurückgeben. Die RP ist dabei gut geeignet für Frontend-Anwendungen, in denen häufig auf Nutzer-Eingaben wie z.B. Touch- und

¹⁰⁶ Höller (2019). S.100-101.

¹⁰⁷ Watt et al. (1990). S.1-3.

Mausklicks reagiert werden muss. Der klassische imperative Programmfluss ist dafür weniger geeignet, da dort eine Vorhersage der zeitlichen Abfolge der Benutzereingaben getroffen wird, was nicht praktikabel bzw. effizient ist.¹⁰⁸

Denn die oft große Anzahl an System-Events und Veränderungen des Application State ([Kapitel 4.1.1.2](#)), deren Auslöser neben den genannten Eingaben des Nutzers im Frontend auch Netzwerk-Nachrichten und die Verarbeitung von Sensordaten sein können, müssen vom Computer-System durch Rechenprozesse verarbeitet werden. Das Ergebnis dieser Rechenprozesse kann wiederum sein, dass der Application State aktualisiert werden muss. Dies kann weitere System-Events und Rechenprozesse auslösen. Diese angedeutete enge Verzahnung und Vernetzung von Prozessen innerhalb von Computerprogrammen macht das Design und die Wartung reaktiver Programme sehr herausfordernd. Der Programmcode sollte deswegen streng modularisiert, d.h. in einzelne Komponenten aufgeteilt werden, damit die zahlreichen Interaktionen zwischen den einzelnen Variablen, Klassen und Entities des Programmcodes zu überblicken sind und wartbar bleiben.¹⁰⁹

Diese Modularisierung erreicht Angular durch die Komponenten-Hierarchie ([Kapitel 4.1.1.2](#)). Die genannten zeitlich veränderlichen Werte und deren Abhängigkeiten voneinander werden in der Reaktiven Programmierung automatisch von der Laufzeitumgebung der Programmiersprache aktualisiert. Damit wird das Szenario vermieden, dass der Programmierer es versäumt, solche Abhängigkeiten in Werten und Variablen manuell anzupassen. Zusätzlich zu der Zeitersparnis, die durch diese Debugging-Hilfe erzielt werden kann, bietet der Programmierstil der Reaktiven Programmierung durch den deklarativen Ansatz einen verständlichen Programmcode.¹¹⁰

Das dargestellte Konzept der Repräsentation von diskreten Werten als zeitlich veränderlich in Abhängigkeit von der Änderung anderer Werte sowie von auftretenden Events stammt aus der Datenflussprogrammierung und wird in der reaktiven Programmierung in angepasster Form angewandt. Beispielsweise kann die Struktur des Datenflusses in der reaktiven Programmierung zusätzlich zur Laufzeit verändert werden.¹¹¹

Die Programmier-Bibliotheken für die reaktive Programmierung sind häufig mit Hilfe einer domänen-spezifischen Sprache (DSL) in eine Programmiersprache integriert.¹¹² Dieses Vorgehen ist deshalb sinnvoll, da sich die Anwendungsfälle der reaktiven Programmierung meist zielgerichtet auf einzelne Problem-Domänen fokussieren.¹¹³

¹⁰⁸ Bainomugisha et al. (2012). S.1-3.

¹⁰⁹ Salvaneschi et al. (2014). S. 1.

¹¹⁰ Salvaneschi et al. (2016). S.1.

¹¹¹ Bainomugisha, (2012). S. 1-3.

¹¹² Salvaneschi et al. (2016). S.6.

¹¹³ Bainomugisha, (2012). S. 8-9.

4.1.1.4 Domain-specific Language (DSL)

Eine Domain-specific Language (DSL) ist in ihrer Notation auf einen oder eine begrenzte Zahl von Anwendungsfällen (zusammenfassend bezeichnet als Domäne) spezialisiert. Die DSL ist damit passgenau auf die relevanten Konzepte und Eigenschaften ihrer Domäne zugeschnitten. Dies kann Vorteile für die Produktivität in genau dieser Domäne haben.¹¹⁴

Auch Angular selbst nutzt intern DSLs, z.B. für die Domains Formulare, Animationen und Navigation.¹¹⁵

4.1.1.5 RxJS

Die Features der reaktiven Programmierung müssen nicht unbedingt Teil der Programmiersprachen selbst sein. Stattdessen stellen häufig Bibliotheken und Spracherweiterungen die nötigen Funktionen für die zugehörigen Programmiersprachen bereit. Die in Angular integrierte Programmbibliothek für reaktive Programmierung wird RxJS (Reactive Extensions for JavaScript) genannt. In RxJS werden sogenannte „Observables“ eingesetzt, damit der Programmierer Features der reaktiven Programmierung nutzen und damit asynchrone und event-orientierte Programme implementieren kann.¹¹⁶

Die Angular-Klasse HTTPClient kann zum Beispiel genutzt werden, HTTP-Anfragen an einen Server zu stellen, indem ein Observable erstellt wird, das sozusagen „abonniert“ werden kann (von englisch: subscribe). Daraufhin wird grundsätzlich jedes Mal, wenn der Programmierer das Observable mit Hilfe der HTTPClient().get-Methode abonniert, eine neue Anfrage an den Server gestellt, von dem z.B. Daten angefordert werden.¹¹⁷

4.1.1.6 Node-Package-Manager (npm)

Für die Wartung und die Entwicklung von Angular-Webapplikationen wird der Einsatz der Laufzeitumgebung Node.js sowie des Node-Package-Managers (npm) empfohlen.¹¹⁸

Der Node-Package-Manager (npm) ergänzt Node.js ([Kapitel 3.3.1.1](#)) dadurch, dass er die sogenannten „Dependencies“ (engl. Abhängigkeiten) einer Web-Applikation verwaltbar macht. Der Begriff Dependencies meint in diesem Fall die externen Software-Pakete, die in npm enthalten sind und die ein Entwickler in Web-Applikationen verwenden kann.

¹¹⁴ Kosar et al. (2008). S. 390.

¹¹⁵ <https://angular.io/guide/glossary#domain-specific-language-dsl>. Datum des Abrufs: 14.12.2023.

¹¹⁶ <https://angular.io/guide/rx-library>. Datum des Abrufs: 18.12.2023.

¹¹⁷ <https://angular.io/api/common/http/HttpClient>. Datum des Abrufs: 18.12.2023.

¹¹⁸ <https://angular.io/guide/setup-local>. Datum des Abrufs: 13.01.2024.

Mit Hilfe des Node-Package-Managers kann der Entwickler dabei die Notwendigkeit einer Wartung und Aktualisierung der in der Web-Applikation genutzten Software-Pakete überprüfen und deren (Neu-)Installation durchführen.¹¹⁹ Dieser Schritt kann durch Nutzung der npm-CLI (Command-Line-Interface) mit der Eingabe des Befehls „npm install“ innerhalb der Ordnerstruktur der zugehörigen Web-Applikation ausgeführt werden.¹²⁰

Dabei bietet die Webseite <https://www.npmjs.com/> eine Auflistung und Dokumentation aller von npm inkorporierten Pakete. Auch viele Angular-Pakete, wie z.B. das Angular Command Line Interface (CLI) und Angular Material, können mit npm installiert und gewartet werden.¹²¹

4.2 Leonardo.AI (Text-zu-Bild-Generator)

In [Kapitel 4.1](#) wurden die technischen Grundlagen, die zum Verständnis der Programmierung des Prototypen der konzipierten Web-Applikation notwendig sind, beschrieben. Doch wie werden die Eingaben bzw. Texte, die der Benutzer tätigt bzw. schreibt, verarbeitet? Woher kommen die Bilder, die aus den verfassten Textabschnitten generiert werden sollen?

Die Antwort darauf ist, dass die Webapplikation mit Hilfe der bereits eingeführten Klasse HTTPClient ([Kapitel 4.1.1.5](#)) einen HTTP-POST-Call (mit dem Text, den der Nutzer verfasst hat) an den Server des Text-zu-Bild-Generators Leonardo.AI schickt, der den verfassten Text zu vier Bildern verarbeitet und eine Identifikationsnummer (ID) dieses Prozesses zurück die Webapplikation schickt, damit diese vier Bilder wiedergefunden werden können.

Daraufhin sendet die Webapplikation einen HTTP-GET-Request an den Text-zu-Bild-Generierungs-Service, diesmal wird dabei die ID des eben genannten Prozesses mitgesendet, woraufhin der Service die URLs der Bilder, die mit dieser ID verknüpft sind, sucht und an die Webapplikation als Antwort schickt. Der Teil dieses Prozesses, der innerhalb der Webapplikation stattfindet, ist Teil der Arbeitspakete der Programmierung ([Kapitel 3.4](#)).

Leonardo.AI wurde ausgewählt, da der Service die drei nachfolgenden nicht-funktionalen Anforderungen, die an den Text-zu-Bild-Generator gestellt wurden, erfüllt:

- Der Schutz vor nicht jugendfreien Inhalten erfolgt mit verschiedenen Algorithmen.
- Die Bildqualität hängt von der Qualität des verfassten Textes ab, kann aber mit Hilfe der genutzten Stable Diffusion vielfältige Ergebnisse mit scharfer Grafik erzielen.¹²²
- Die API (Programmierschnittstelle) funktioniert und wurde ordentlich dokumentiert.¹²³

¹¹⁹ Höller (2019). S. 29f.

¹²⁰ <https://docs.npmjs.com/cli/v10/commands/npm>, Abrufdatum 11.12.2023.

¹²¹ <https://docs.npmjs.com/about-npm>, Abrufdatum 11.12.2023.

¹²² Yang et al. (2023). S. 1-3. MMA-Diffusion.

¹²³ <https://docs.leonardo.ai/reference/getuserself>. Datum des Abrufs: 19.02.2024.

Mit den Kosten verhält es sich bei Leonardo.AI ähnlich wie bei den Konkurrenten MidJourney¹²⁴ und Dall-E 3¹²⁵. Die private Nutzung mit einer begrenzten Anzahl an Bildern, die pro Tag generiert werden können, ist grundsätzlich kostenlos. Für eine ausgiebige Nutzung mit der API fallen dagegen Kosten an.¹²⁶ Während die Firmen hinter MidJourney (das wissenschaftliche Labor Midjourney, Inc.) und Leonardo.AI (Leonardo Interactive) auf Abonnenten-Modelle setzen, setzt die Firma OpenAI (Dall-E 3) auf einen festgelegten Preis pro Bild (\$0.016 bis \$0.120 pro Bild je nach Qualität).

Dabei bietet Leonardo Interactive zusätzlich an, das Unternehmen direkt zu kontaktieren, um ein individuelles Abonnement auszuhandeln.

API Basic	API Standard	API Pro	API Custom
\$9 / month <small>ex. tax</small>	\$49 / month <small>ex. tax</small>	\$299 / month <small>ex. tax</small>	API Plan
Manage Subscription	Upgrade to API Standard	Upgrade to API Pro	Contact Us
3.500 API credits per month Utilise the API in any of the following ways: <ul style="list-style-type: none"> ✓ Up to 5 concurrent generations ✓ Alchemy, Prompt Magic v3, PhotoReal and Motion ✓ Train models with API credits ✓ Purchase additional API credits to continue using the API ✓ API credits do not expire ✗ No discount on API credits top-ups ✓ Automatic API credits top-ups 	25.000 API credits per month Utilise the API in any of the following ways: <ul style="list-style-type: none"> ✓ Up to 5 concurrent generations ✓ Alchemy, Prompt Magic v3, PhotoReal and Motion ✓ Train models with API credits ✓ Purchase additional API credits to continue using the API ✓ API credits do not expire ✓ 20% discount on API credits top-ups ✓ Automatic API credits top-ups 	200.000 API credits per month Utilise the API in any of the following ways: <ul style="list-style-type: none"> ✓ Up to 5 concurrent generations ✓ Alchemy, Prompt Magic v3, PhotoReal and Motion ✓ Train models with API credits ✓ Purchase additional API credits to continue using the API ✓ API credits do not expire ✓ 40% discount on API credits top-ups ✓ Automatic API credits top-ups 	Custom API credits amount Utilise the API in any of the following ways: <ul style="list-style-type: none"> ✓ Up to ∞ concurrent generations ✓ Alchemy, Prompt Magic v3, PhotoReal and Motion ✓ Train models with API credits ✓ Purchase additional API credits to continue using the API ✓ API credits do not expire ✓ Custom discount on API credits top-ups ✓ Automatic API credits top-ups

Abbildung 4: Kostenübersicht der Abonnenten-Modelle. Quelle: Leonardo.ai.

¹²⁴ <https://docs.midjourney.com/docs/plans>. Nutzung nur kostenlos mit dem Discord-Bot (Konferenz-Plattform).

¹²⁵ <https://openai.com/pricing>.

¹²⁶ <https://app.leonardo.ai/buy-api>.

5 Ergebnisse

In dem Kapitel werden die Ordnerpfade der Dokumentation und der Demonstrationsvideos des Proof-of-Concept verlinkt. Danach folgt die Erklärung der Templates ([Kapitel 5.3](#)). Zuletzt werden ausgewählte Teile der Implementierung dargestellt ([Kapitel 5.4](#)).

5.1 Dokumentation

Der Programmcode wurde ausführlich kommentiert. Des Weiteren wurde die Dokumentation mit Hilfe des Dokumentationswerkzeugs Compodoc, das auf Angular-Webapplikationen spezialisiert ist, resultierend aus den Doc-Comments und der Programmstruktur erstellt.¹²⁷

Die Dokumentation liegt innerhalb des Ordnerpfads „proof-of-concept/documentation“. (<https://gitlab.mi.hdm-stuttgart.de/cf061/kurzgeschichten-thesis>)

5.2 Demonstrationsvideos

Die Demonstrationsvideos des Proof-of-Concept sollen zum Einen die Nutzung der Webapplikation als Werkzeug für das kreative Schreiben von Kurzgeschichten demonstrieren (primär für die Zielgruppen [2.1.1 Weiterführende Schulen](#) sowie [2.1.2 Grundschulkinder](#)).

Dabei konnte zusätzlich das vorgeschlagene Feature des Text-to-Speech-Supports getestet und demonstriert werden. Die Stimmen wurden kreiert mit Hilfe von <https://www.narakeet.com/>.

Darüber hinaus stellt das zweite Demonstrationsvideo die Nutzung der Webapplikation als Werkzeug zum Vorlesen von illustrierten Kurzgeschichten dar (primär für die Zielgruppe [2.1.3 Vorschulkinder: 3-6 Jahre alt](#)). Für das zweite Demonstrationsvideo wurde zusätzlich ein kurzer Soundtrack komponiert und aufgenommen, um die Gamification der entsprechenden Teile der Kurzgeschichte (Die Legende von Kelserra) zu verstärken. Um das vorgeschlagene Feature des Text-zu-Musik-Generators ([Kapitel 6.2.2.3.3](#)) zu demonstrieren und zu testen, wurde die Datei des Sound-Recordings in kleine Teile passend zu den Textstellen der Kurzgeschichte (z.B. die Textstellen mit der Ankunft der Piraten, der Wut des Drachen, der Trauer der Dorfbewohner) geschnitten. Danach wurde jeweils ein Textabschnitt zusammen mit dem dazu passenden Musikabschnitt mit Hilfe des Open-Source-Projektes Music Gen, das von Meta (früher Facebook) ins Leben gerufen wurde, verarbeitet und verbessert.¹²⁸

Die Demonstrationsvideos liegen auf dem Cloud-Server mit der Adresse:

<https://drive.google.com/drive/folders/1bnhNYNmstiXmFQdcxtpXt4wAdrznNJZf?usp=sharing>.

¹²⁷ <https://compodoc.app/>. Datum des Abrufs: 13.02.2024.

¹²⁸ <https://huggingface.co/spaces/facebook/MusicGen>. Datum des Abrufs: 13.02.2024.

5.3 Templates (angefangene Kurzgeschichten)

Für die Thesis wurden sechs Templates (Deutsch: Vorlagen) verfasst, die in der Methode des kreativen Schreibens als Schreibimpuls für die Zielperson dienen sollen ([Kapitel 2.2.1](#)).

Das Template mit dem Namen „Die Legende von Kelserra“ ist allerdings abgeschlossen und nicht editierbar, da diese Kurzgeschichte die Einführung und Hintergrundgeschichte für die Webapplikation bildet. Die sechs Kurzgeschichten bzw. Templates wurden von dem Autor der Thesis verfasst, mit der Ausnahme der Kurzgeschichte „Neue Stadt neues Glück“. Für diese Kurzgeschichte wurde das in [Kapitel 6.2.2.1](#) vorgeschlagene Feature getestet, nämlich die „Reglementierte Einbindung intelligenter Chatbots“. Dafür wurde Chat GPT genutzt. Mit diesem Chatbot wurde die genannte Kurzgeschichte in „Teamarbeit“ verfasst.

Die Kurzgeschichten sind inhaltlich nach dem Prinzip der Heldenreise verfasst, mit dem Fokus auf die ersten beiden Abschnitte, in denen der Hauptcharakter der Kurzgeschichte mitsamt seiner gewohnten Lebenswelt kurz vorgestellt wird und dann mit einem Problem, mit einer Herausforderung oder ähnlichen Handlungsaufrufen konfrontiert wird. Dies kann den Nutzer der Webapplikation dazu anregen, die Kurzgeschichte weiterzuschreiben und textuell Lösungen für den Hauptcharakter zu finden oder gar neue Herausforderungen zu kreieren.¹²⁹

Die Kurzgeschichten können während der Nutzung der Webapplikation nachgelesen werden sowie in dem Repository in den Dateien des Pfads „proof-of-concept/src/assets/Templates“.

5.4 Implementierung

Die Implementierung der in den Software-Anforderungen definierten Arbeitspakete zur Programmierung des Prototypen der Web-Applikation kann unter dem folgenden Link nachvollzogen werden: <https://gitlab.mi.hdm-stuttgart.de/cf061/kurzgeschichten-thesis>.

Zum besseren Verständnis werden nachfolgend ausgewählte Teile des Programmcodes, die entwickelt wurden, um die dazugehörigen Benutzeranforderungen umzusetzen, kurz erklärt.

Aspekt 1: Div-Struktur und @media-flags (Responsiveness)

Die Responsiveness wird sichergestellt, indem die HTML-Layouts der Webapplikation eine klare Struktur von div-Elementen aufweisen. Dabei hat sich in der Text-Editor-Komponente die Nutzung einer Metapher bzw. eines Design-Patterns angeboten, nämlich die des Bücherregals. Das gesamte Layout ist dabei in einem großen Div-Element enthalten mit der Klasse „book-rack“ (deutsch: Bücherregal). Darin findet sich dann das Div-Element der

¹²⁹ Campbell (1949). The Hero with a Thousand Faces. Kapitel 1: The Call to Adventure. S.33-46.

Klasse „book-rack-top“, das den Namen der aktuell bearbeiteten Kurzgeschichte und den Button „Zurück zum Hauptmenü“ enthält.

Die nächste Reihe bildet das Div-Element der Klasse „book-rack-middle“, in der z.B. das Textfeld sowie das dazu generierte Bild liegen. Schließlich bildet das Div-Element der Klasse „book-rack-bottom“ mit dem Button „Regal-Reihe hinzufügen.“ den Abschluss.

Damit sich das Layout in der Größe der einzelnen visuellen Elemente an die Bildschirmgröße des Benutzers bzw. an die Größe des Browser-Fensters anpasst, wird das globale CSS-Stylesheet von Angular genutzt (Dateiname: „styles.css“). Darin wird mit Hilfe der @media-Flagge definiert, welche Größe die Überschriften (englisch: Headings) und Paragraphen für verschiedene definierte Bildschirmgrößen annehmen sollen.

Zur Veranschaulichung wird auf der nächsten Seite die grafische Darstellung des DOM-Trees der Text-Editor-Komponente gezeigt. DOM steht dabei für Document Object Model. Dabei handelt es sich um eine API ([Glossar](#)) für valide HTML- sowie für XML-Dokumente.

Das heißt konkret, dass das DOM die Struktur dieser Dokumente definiert und die Schnittstellen festlegt, mit Hilfe derer der Inhalt der Dokumente verändert und abgefragt werden kann. Letztlich handelt es sich bei dem Inhalt dieser Dokumente um Daten.¹³⁰

¹³⁰ Le Hégaret et al. (2004). S. 13.

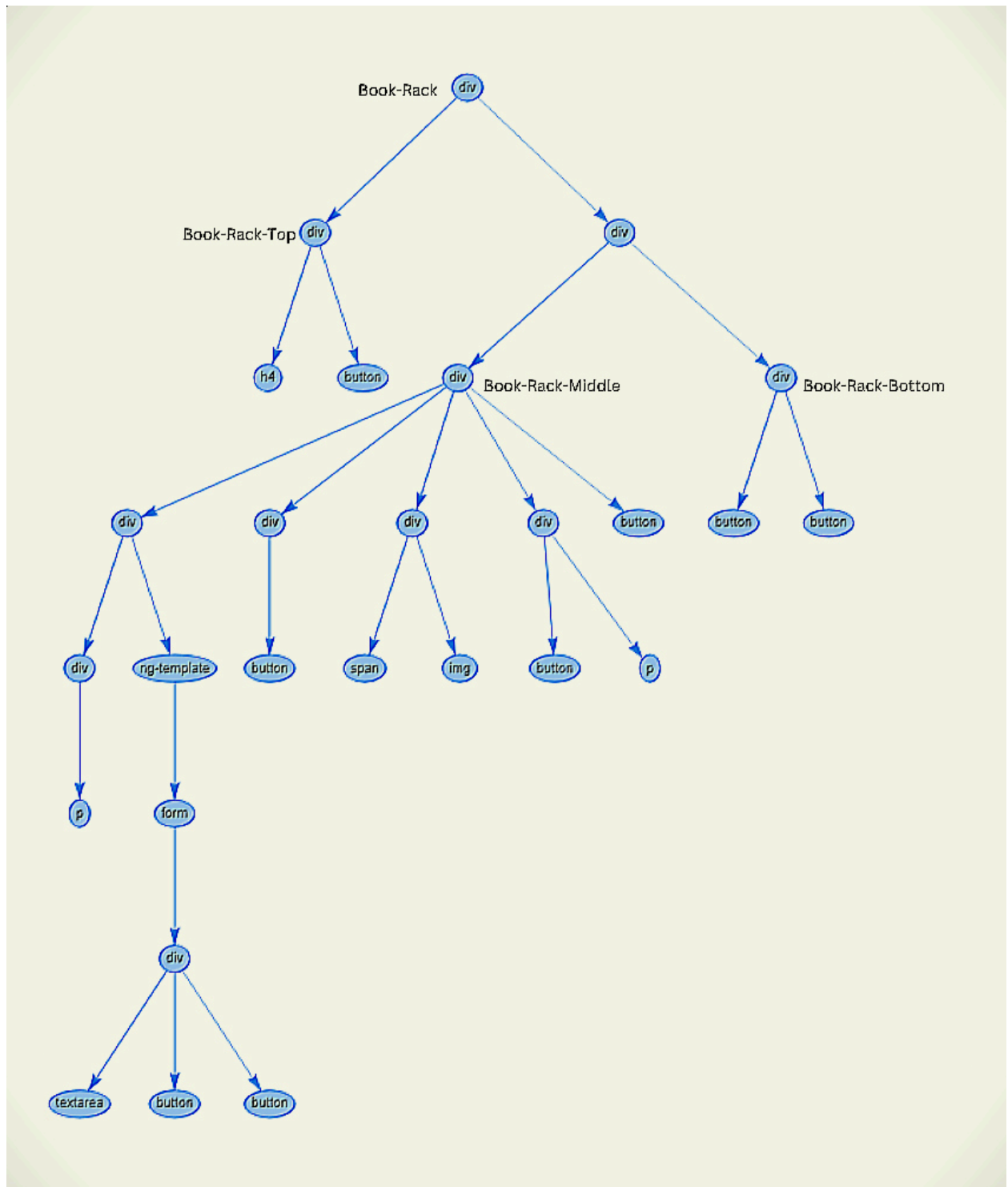


Abbildung 5: Text-Editor-Component DOM-Tree. Quelle: Compodoc (Dokumentationswerkzeug).

Aspekt 2: Grafisches Design

Das grafische Design nutzt genau wie der Aspekt der Responsiveness die Metapher und die Div-Struktur des Bücherregals, indem die einzelnen Div-Elemente des Bücherregals passende Farben nutzen (Holz). Dabei ist das Textfeld der Regalreihen auf der linken Seite platziert, während der Bilderrahmen, der das generierte Bild enthält, rechts davon positioniert ist.

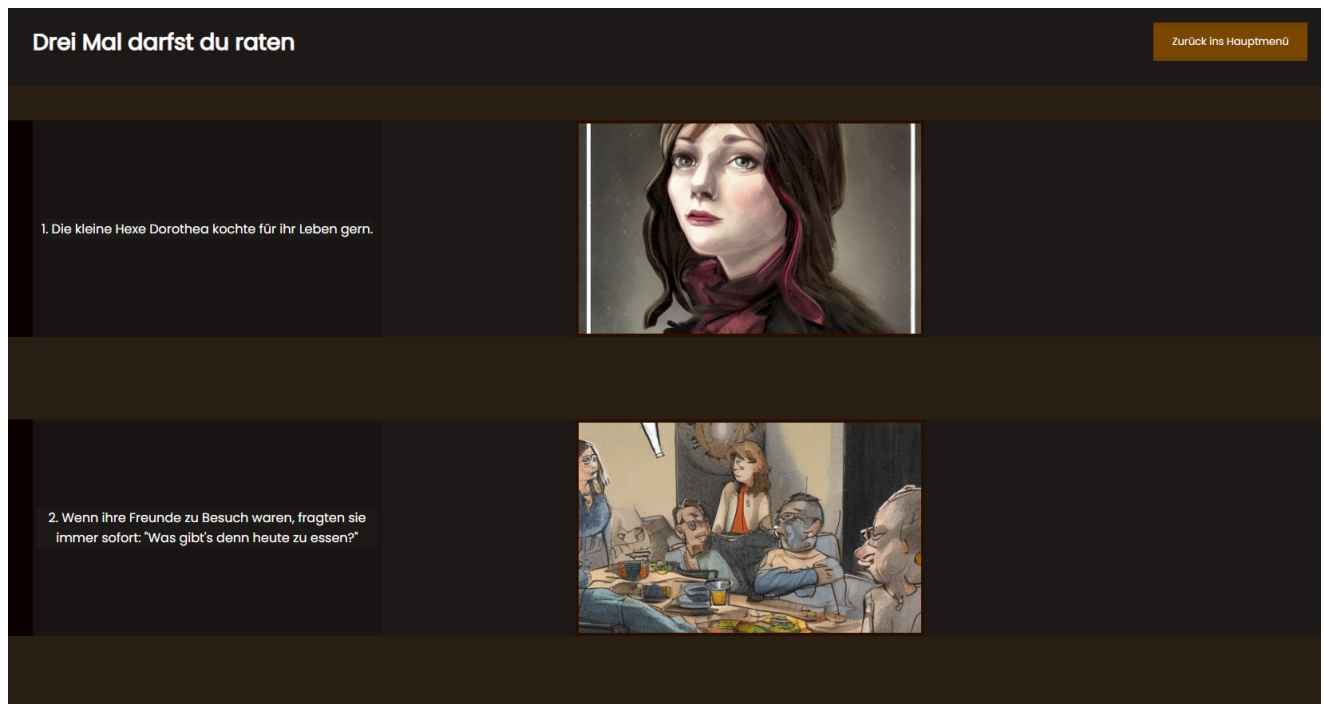


Abbildung 6: Screenshot der Webapplikation. Bilder generiert mit Leonardo.AI.

Aspekt 3: HTML-Alt-Attribut (Barrierefreiheit: visuelles Handicap)

Damit Menschen mit beeinträchtigter Sehleistung die Webapplikation navigieren und nutzen können, wird für die wichtigen Elemente des HTML-Layouts das Alt-Attribut eingesetzt. Diese Abkürzung steht für „alternative text“ und wird dafür eingesetzt, HTML-Elemente mit einer textuellen Beschreibung zu versehen. Dadurch können Menschen mit visuellem Handicap sich diesen Text mit Hilfe von eingebauten Text-zu-Sprache-Erweiterungen (englisch: Text-to-Speech-Plugins) des Webbrowsers vorlesen lassen.

Aspekt 4: Model-View-Controller-Prinzip (Dynamisches Benutzerinterface)

Die nicht-funktionale Anforderung des Dynamischen Benutzerinterfaces ([Kapitel 2.3.2.3](#)) wird durch die Anwendung des in Angular implementierten Model-View-Controllers erfüllt. Dafür werden z.B. `*ngIf`-Abfragen definiert. Dabei handelt es sich um eine Direktive, die basierend auf einer logischen Aussage, die letztlich in einem Boolean-Wert mündet (wahr oder falsch), HTML-Elemente in das Template einfügt (wahr) bzw. ignoriert (falsch).¹³¹

Um ein Beispiel anzuführen: Lässt sich der Nutzer am Ende der Kurzgeschichte, die er gerade bearbeitet, ein neues Textfeld erstellen, ist dieses zunächst nicht im Bearbeitungsmodus. Dementsprechend wird für diese Regalreihe ein festes Textfeld dargestellt. Das wird dadurch gewährleistet, dass das Div-Element, in dem sich das Textfeld befindet, ein `*ngIf`-Attribut mit dem Wert `„!field.editMode; else editModeTemplate”` aufweist. Das Ausrufezeichen steht für die Aussage „wenn folgendes nicht zutrifft”. Damit wird das feste Textfeld nur angezeigt, wenn sich das Textfeld nicht im Bearbeitungsmodus befindet. Die TypeScript-Datei der zugehörigen Komponente (Text-Editor-Komponente) hat derweil für jedes existierende Textfeld einen Boolean mit dem Namen `„editMode”` gespeichert. Diesem ist nach der Erstellung des Textfeldes zunächst der Wert `„falsch”` zugewiesen, das heißt, das Textfeld befindet sich nicht im editierbaren Modus. Wird nun der Button mit dem Stift-Icon geklickt, der sich neben den editierbaren Textfeldern befindet, wird diesem Boolean der Wert wahr zugewiesen und statt dem festen Textfeld erscheint nun das editierbare Textfeld. Innerhalb des Programmcodes findet sich eine Vielzahl solcher logischer Abfragen und Zusammenhänge.

Neben den `*ngIf`-Abfragen werden darüber hinaus Property-Bindings eingesetzt, z.B. zum Anzeigen der Bilder. Das HTML-Source-Attribut wird dafür in eckige Klammern gesetzt und kann so an eine Variable in der dazugehörigen TypeScript-Datei der Text-Editor-Komponente gebunden werden: `[src]="field.pictureURLs[field.pictureSelector]"`. Damit wird für genau diesen Textabschnitt eines der vier dafür generierten Bilder angezeigt. Wird der Wert der Variablen `pictureSelector` auf Wunsch des Nutzers durch den Klick auf den Pfeil-Button nun geändert, wird stattdessen das nächste Bild bzw. das Bild mit dem Index (Wert der Variablen `pictureSelector`), der dann dem Application State entspricht, angezeigt ([Kapitel 4.1.1.2](#)). Damit werden die Prinzipien des reaktiven Programmierparadigmas für die Aktualisierung des Benutzerinterfaces in Abhängigkeit des Nutzer-Inputs angewandt ([Kapitel 4.1.1.3](#)).

¹³¹ <https://angular.io/api/common/NgIf>. Datum des Abrufs: 19.02.2024.

6 Fazit & Ausblick

6.1 Fazit

Den Ausgangspunkt der Bachelor-Thesis bildete die Definition und Beschreibung des Lesens und des Schreibens als grundlegende Kulturtechniken. Des Weiteren konnten Belege gefunden werden, die dafür sprechen, dass das Lernen und Beherrschen dieser Kulturtechniken eine Notwendigkeit für schulischen und beruflichen Erfolg darstellt und dass ihre Anwendung zunehmend digital erfolgt. Dabei konnte herausgearbeitet werden, dass bei deutschen Schülern, durchschnittlich betrachtet, ein Negativ-Trend festzustellen ist, was die genannten Kompetenzen angeht ([Kapitel 1.1.1](#)).

Hinsichtlich des zweiten großen Einführungs-Themas der Bachelor-Thesis, der Digitalisierung innerhalb der deutschen Schullandschaft, wurde festgestellt, dass der qualitative und quantitative Ausbau der schulischen IT-Infrastruktur und des IT-Supports nur schleppend voranschreitet, während die Digitalisierung jedoch von der Bundesregierung u.v.m. als notwendige Voraussetzung angesehen wird, den Kindern und Jugendlichen einen effizienten und lösungsorientierten Umgang mit digitalen Medien beizubringen, zu dem die genannten Grundfertigkeiten des Lesens und Schreibens gehören. Dies soll, langfristig betrachtet, in einem vernünftigen, zielgerichteten und motivierenden didaktischen Rahmen, der Lehrer- und Schülerinteressen berücksichtigt, erfolgen. Mit Hilfe von Forschungsergebnissen konnte gezeigt werden, dass die deutsche Schullandschaft diesbezüglich im internationalen Vergleich Nachholbedarf aufweist ([Kapitel 1.1.2](#)).

Die angeführten Forschungsbefunde motivierten, einen kleinen Beitrag zu leisten zur Optimierung der Lernprozesse von Kindern im Lesen und Schreiben: Ein Lesen und Schreiben, das Spaß macht und gleichzeitig die Kompetenzen und Fertigkeiten der Kinder hinsichtlich der deutschen Schriftsprache, aber auch hinsichtlich der Anwendung digitaler Werkzeuge spielerisch erweitert. Es ergab sich die Fragestellung, wie das Lernen und die Kompetenzen, was die Kulturtechniken des Lesens und Schreibens betrifft, in den Schulen gefördert werden und wie gleichzeitig das Lernen, was eine produktive und problemlösungsorientierte Anwendung von technischen Endgeräten und Technologien in Schulen betrifft, weiterentwickelt werden können.

Zum Abschluss der vorliegenden Bachelor-Thesis kann die Frage, ob eine digitale Anwendung für das kreative Schreiben und ähnliche Anwendungen mit den eingesetzten Technologien realisiert werden können, mit Ja beantwortet werden. Die Schüler können mit

dem entwickelten Prototyp an ihren Endgeräten auf kreative Weise eigene Kurzgeschichten verfassen und schreiben. Die weitergehende Frage, ob die Web-Applikation die Motivation und Leistungen der Schüler hinsichtlich des Lesens und Schreibens wirklich steigern kann, könnte jedoch in zukünftigen Nutzertests ([Kapitel 6.2](#)) geprüft werden.

Eine Differenzierung der Zielgruppen der zu konzipierenden digitalen Anwendung wurde bereits vorgenommen, um bedarfsgerecht die altersspezifischen Kenntnisse und Interessen zu berücksichtigen ([Kapitel 2.1](#)). Die daraufhin erarbeiteten Bedürfnisse ([Kapitel 2.2](#)) wurden dann mit dem Fokus auf die Zielgruppe der Grundschulkinder (6-10 Jahre alt) in funktionale und nicht-funktionale Benutzer-Anforderungen übertragen ([Kapitel 2.3](#)).

Die Literaturlauswertung, deren Ergebnisse in den Teilkapitel der Bedarfsanalyse angeführt werden, zeigte, dass die didaktischen Varianten und Methoden des „Kreativen Schreibens“ ([Kapitel 2.2.1](#)), des „Digitalen Schreibens“ ([Kapitel 2.2.2](#)) und der „Gamification“ ([Kapitel 2.2.3](#)) eine sehr große Affinität zu den Bedürfnissen, Phantasien, Interessen und Alltagserfahrungen der Zielgruppen aufweisen. Um einen Gamification-Mechanismus für das Schreiben von Texten bzw. Kurzgeschichten zu erzielen, wurde die Entscheidung getroffen, einen Text-zu-Bild-Generator in Webapplikation einzubinden.

Alle drei genannten didaktisch-methodischen Konzepte werden teilweise in den Schulen schon praktiziert, allerdings meist isoliert und nicht im Zusammenspiel. Ergänzend zu dem übrigen Curriculum könnte die Kombination der drei Konzepte insbesondere im Fach Deutsch, aber auch im Sachunterricht, das schulische Lernen sehr gut unterstützen. Voraussetzung dafür ist jedoch eine gute technische Infrastruktur, ein verlässliches und adäquates Hosting und das Beachten von Datenschutz und Datensicherheit, weshalb diese Themenfelder in den Kapiteln [2.2.4](#) bis [2.2.6](#) untersucht wurden.

Dabei hat sich gezeigt, dass Deutschland, was das Thema der technischen Infrastruktur in der Schullandschaft angeht, im Vergleich zu anderen Industrieländern wie Südkorea, den USA, und auch im Vergleich zu Dänemark und Finnland hinterherhinkt. Es konnte gezeigt werden, dass sich die deutsche Bundesregierung dieser Thematik schon länger bewusst ist und deswegen z.B. den sogenannten „DigitalPakt Schule“ aufgesetzt hat, mit dem eine hohe monetäre Investitionssumme für die Digitalisierung der deutschen Schulen bereitgestellt wird.

Die Gegenüberstellung der gemachten Erfahrungen des lokalen und des Cloud-Hostings dort, wo in den Schulen bereits digital unterrichtet wird, spricht für das Cloud-Hosting. Insbesondere erleichtert diese Technik den Einsatz landeseinheitlicher digitaler Bildungsplattformen und das Nutzen kommunaler Rechenzentren. Außerdem ermöglicht das Cloud-Hosting eine bessere Skalierbarkeit und das Arbeiten mit webbasierten Anwendungen.

Jedoch sind bei all diesen Vorhaben der Datenschutz und die Datensicherheit nicht zu vernachlässigende Aspekte. Es gilt, die europäischen Datenschutzvorgaben der DSGVO (Datenschutz-Grundverordnung) einzuhalten. Die personenbezogenen Daten von Schülern und Lehrern müssen geschützt werden. Personenbezogene Daten dürfen nur für bestimmte Zwecke und nur mit der Einwilligung der betroffenen Person erhoben und verarbeitet werden.

Dabei sind für Web-Programmierer viele Vorgaben zu beachten, da einige Ministerien, darunter auch das Kultusministerium Baden-Württemberg, Empfehlungen erlassen haben, was die Datenerhebung, die Datenverarbeitung und die Nutzung von Cloud-Servern angeht. Diese Empfehlungen dienen dazu, die Schulleitungen zu entlasten, da diese letztlich die Verantwortung dafür tragen, dass webbasierte Anwendungen sicher genutzt werden können. Hierzu gehört auch, dass aufgrund eines Urteils der Europäischen Gerichtshofs (EuGH) aus dem Jahr 2020 empfohlen wird, ausschließlich Tools auf Servern mit europäischen Standorten zu nutzen und die vom Datennetz der wissenschaftlichen Einrichtungen des Landes Baden-Württemberg gehostete Software „Moodle“ als Lern-, Informations- und Kommunikationsplattform zu verwenden.

Abgeleitet von der Bedarfsanalyse waren nun die funktionalen und nicht-funktionalen Benutzeranforderungen zunächst anhand von „Use Cases“ zu beschreiben, darunter die Listenanzeige, die die Referenzen auf die angefangenen Kurzgeschichten (Templates) mit den bereits verfassten Starttexten und dazu passenden Bildern als Schreibimpuls enthalten sollte, um der Methode des kreativen Schreibens gerecht zu werden ([Kapitel 2.3.1](#), [Kapitel 2.2.1](#)).

Als nicht-funktionale Anforderungen, d.h. qualitative Anwendungseigenschaften, wurden eine leicht verständliche Sprache mit erklärenden Anwendungshinweisen, eine übersichtliche Darstellung auf dem jeweiligen Endgerät (Responsiveness) und die alternativen Texte der HTML-Elemente als ein Faktor der Barrierefreiheit implementiert. Weitere Features, die der Barrierefreiheit dienen können, werden in [Kapitel 6.2.2.4](#) dargelegt.

Die angeführte nicht-funktionale Anforderung der Nutzerzahl bezog sich derweil auf die Robustheit bzw. Leistungsfähigkeit des Backends mitsamt Webserver sowie Datenbank und spielte bei der Implementierung des frontend-basierten Prototypen keine Rolle. Sie wurde aber angeführt, da das Backend für schlüsselfertige Webapplikationen unabdingbar ist und die potentielle Nutzerzahl berücksichtigt werden sollte.

Die Anpassbarkeit des Prozess-Workflows der Webapplikation zur Laufzeit in Abhängigkeit des Alters des Nutzers ([Kapitel 2.3.2.5](#)) ist mit der Wahl von Angular als Web-Development-Plattform und seiner dargestellten mannigfaltigen Möglichkeiten der Implementierung von Business-Logik definitiv gegeben ([Kapitel 4.1](#)). Für die Entwicklung des Prototypen wurde aus zeitlichen Gründen allerdings nur der in dem Arbeitspaket mit der

Nummer GUI.006 ([Kapitel 3.4.4](#)) dargelegte Workflow für die Zielgruppe der Grundschulkinder (6-10 Jahre alt) ([Kapitel 2.1.2](#)) implementiert.

Genauso positiv verhält es sich mit der Anpassbarkeit und Erweiterungsmöglichkeit der Features. Diese sind durch die Wahl von Angular und dessen auf Skalierbarkeit und Veränderbarkeit ausgelegte Modul-Struktur definitiv gegeben ([Kapitel 4.1](#)).¹³²

Die für das Programmieren des Prototyps notwendigen Anwendungsfälle und Prozesse wurden mit Hilfe von UML-Diagrammen modelliert und in technische Programmierschritte aufgeschlüsselt ([Kapitel 3.4](#)). Alle definierten Arbeitsschritte konnten mit den gewählten Technologien implementiert werden, darunter das kreative Weiterschreiben der Kurzgeschichten, das in Textfeldern mit Zeichenbegrenzung erfolgt, deren Inhalt einzeln mit Hilfe einer automatischen Rechtschreibprüfung korrigiert werden kann und mit Hilfe der API-Anbindung von Leonardo.AI ([Kapitel 4.2](#)) in vier Bilder transformiert werden kann.

Zusammengefasst kann festgehalten werden, dass die Webapplikation im Rahmen der technischen Konzeption ([Kapitel 3](#)) und der technischen Grundlagen ([Kapitel 4](#)) entwickelt und als den Anforderungen entsprechender, tauglicher Prototyp fertiggestellt werden konnte.

Verbesserungen in Design, Benutzerfreundlichkeit sowie neue Features ([Kapitel 6.2.2](#)) können auf Grundlage des Prototyps hinzugefügt werden.

¹³² <https://angular.io/guide/ngmodules>. Datum des Abrufs: 19.02.2024.

6.2 Ausblick

6.2.1 Vorschläge für Nutzertests

Sollte die professionelle Weiterentwicklung der konzipierten Webapplikation in Erwägung gezogen werden, könnten Nutzertests des in dieser Thesis entwickelten Prototyps weitere Klarheit darüber geben, zu welchem Qualitätsgrad die an die Webapplikation gerichteten Benutzeranforderungen ([Kapitel 2.3](#)) erfüllt sind und ob sich die Webapplikation eignet, innerhalb des Deutschunterrichts eingesetzt zu werden. Die folgenden Punkte stellen Vorschläge für Fragen und Aussagen in einem Nutzertest dar, die mit Hilfe von verschiedenen Antwortmöglichkeiten beantwortet bzw. bewertet werden könnten.

Weitere Fragen sowie Aussagemöglichkeiten dieser Form können sinnvoll sein. Die Fragen bzw. Aussagen sind so verständlich wie möglich und unter Einsatz von kindgerechten Wörtern konzipiert, allerdings nicht in übertriebener Weise. Damit der Leser der Thesis einen besseren Überblick hat, wird jeweils eine Erklärung der einzelnen Punkte vorangestellt.

Die Punkte 1-3 zielen darauf ab, zu überprüfen, ob die Webapplikation in der aus Benutzersicht geplanten Weise genutzt werden kann, d.h. ob die Anwendungsfälle ([Kapitel 2.3.1](#)) durchgeführt werden können, und wie positiv bzw. negativ die Webapplikation nach dem Testen insgesamt bewertet wird.

1. Ich habe eine kleine Kurzgeschichte geschrieben und viele schöne Bilder generiert. („Ja“, „Geht so“, „Nein“).

2. Die generierten Bilder haben gut zu meiner Geschichte gepasst. („Ja“, „Geht so“, „Nein“).

3. Das Schreiben von Kurzgeschichten mit Kelserra (Name der Webapplikation) fand ich... („Langweilig“, „Geht so“, „Super“).

Punkt 4 dient als Blaupause für Fragen in Richtung der Benutzerfreundlichkeit, der Bedienbarkeit und des Designs, d.h. nicht-funktionaler Aspekte der Webapplikation.

4. Wenn ich die Kurzgeschichte nicht gut fand, konnte ich sie leicht schließen und eine neue Kurzgeschichte starten. („Ja“, „Nein“, „Weiß nicht“).

Punkt 5 zielt wie Punkt 4 auf eine nicht-funktionale Anforderung ab, wird aber in einem extra Punkt erwähnt, da es um die technische Lauffähigkeit der Webapplikation geht.

5. Gab es einen Punkt, an dem das Programm abgestürzt ist oder gar nicht mehr funktioniert hat? („Ja, und zwar: <Antwort des Nutzers>“, „Nein, alles lief gut.“)

Punkt 6 ist ein Beispiel für Fragen, die den Nutzer um eine inhaltliche Bewertung des Konzepts der Webapplikation bitten.

6. Fandest du es gut, dass die Kurzgeschichten schon einen Anfang hatten? („Ja“, „Geht so“, „Nein, ich will lieber mit der eigenen Idee starten.“)

Das nächste Kapitel zeigt ergänzende Vorschläge zum Konzept der Webapplikation auf.

6.2.2 Vorschläge für weitere Features

6.2.2.1 Reglementierte Einbindung intelligenter Chatbots

Statt dass die Templates für angefangene Kurzgeschichten von den Entwicklern der Webapplikation als darin verpackte Dateien für den Nutzer bereitgestellt werden, könnten sie anhand von eingegebenen Schlüsselwörtern des Nutzers von einer Textgenerierungs-KI wie z.B. Chat GPT geschrieben werden. Das heißt, dass der Nutzer, wenn er eine neue Geschichte startet, gefragt werden könnte: „Von welchen Themen soll deine Geschichte handeln? Bitte nenne mir ein paar Wörter!“ Durch die Einbindung der Chat-GPT-API könnten dann mit Hilfe einer passenden Texteingabe (englisch: Prompt), die der Entwickler definiert (z.B. „Mit den folgenden Schlüsselwörtern soll der Anfang einer Kurzgeschichte generiert werden. Die Schlüsselwörter sind: <Eingabe des Nutzers> Bitte generiere diesen Anfang, aber nur maximal 6 Sätze.“), kann dann die Antwort von Chat GPT, d.h. die angefangene Kurzgeschichte, erbeten und schließlich an den Nutzer weitergegeben werden.

Weiterhin wäre es möglich, dass der Nutzer auch während des Schreibprozesses in einer begrenzten Form, z.B. maximal dreimal pro Kurzgeschichte, die Chance hat, sich einen kleinen Textabschnitt generieren zu lassen, falls der Nutzer zu diesem Zeitpunkt keine Idee hat, wie er die Geschichte weiterschreiben könnte.

Anmerkung: Die gelisteten Vorschläge sind als Zusatz zur jetzigen Webapplikation zu sehen, d.h. der Anwendungsfall, dass der Nutzer von Anfang an nur mit dem eigenen Text startet und weiter verfährt, d.h. ganz ohne von den Entwicklern vorgefertigte oder KI-generierte Templates und Textabschnitte, muss nicht gestrichen werden.

6.2.2.2 PDF-Exportfunktion

Damit die verfassten Kurzgeschichten mitsamt der generierten Bilder in einem geeigneten Dateiformat geteilt und an die Eltern versandt werden kann, sei hier der Vorschlag vermerkt, der Web-Applikation eine Exportfunktion, z.B. in das Dateiformat „PDF“, hinzuzufügen.

6.2.2.3 Gamification (Verstärkung)

6.2.2.3.1 Kooperative Mini-Spiele

Für die Weiterentwicklung des Prototyps kann die Einbindung von kooperativen Mini-Spielen, die für den Einsatz an Schulen geeignet sind, sinnvoll sein. An dieser Stelle soll nur eine mögliche Idee vorgeschlagen werden, zu der das Spiel „Stille Post“ als Vorbild dient. Bei der klassischen Version der „Stillen Post“ setzen sich die Schüler mit oder ohne den Lehrer in einen Stuhlkreis und man entscheidet sich zunächst für eine Richtung, z.B. den Uhrzeigersinn. Daraufhin startet eine Person (Jim) das Spiel, indem sie der links neben ihr sitzenden Person einen Satz ins Ohr flüstert, sodass die anderen Personen den Satz nicht hören können. Auf diese Weise setzt sich die Schlange fort und die Botschaft wird immer nur paarweise an den Sitznachbarn weitergegeben.

Meist versteht der jeweilige Zuhörer den Satz nicht komplett und korrekt oder kann sich nicht alles merken und so wird die Botschaft bei jeder Weitergabe etwas verändert. Wird der Kreis geschlossen und die „Stille Post“ erreicht die letzte Person des Kreises (sitzend auf dem Platz zur Rechten neben der Start-Person Jim), spricht diese letzte Person den empfangenen Satz laut aus, woraufhin dieser mit dem Startsatz verglichen wird. Der Unterschied zwischen den beiden Sätzen kann sehr groß sein, was zu allgemeiner Heiterkeit führen kann. Diesen Effekt zu erzielen, macht den Reiz des Spiels aus, deswegen sollte er nicht unerwähnt bleiben.

Dieses Spielprinzip könnte in den in der vorliegenden Arbeit entwickelten Prototyp eingebaut werden, indem unter Berücksichtigung von [Kapitel 6.2.2.1](#) zunächst die erste Person, z.B. der Lehrer (Jim), das Thema der Geschichte mit Hilfe von Schlüsselwörtern vorgibt und sich den Start der Geschichte in Textform durch den eingebauten Chatbot generieren lässt. Daraus werden mit der Text-zu-Bild-KI ([Kapitel 4.2](#)) Bilder generiert.

Die zweite Person (Richard) in der Reihe erhält daraufhin ausschließlich diese Bilder (keinen Text), und hat die Aufgabe, die Bilder in Textform zu beschreiben. Mit dem neuen Text der zweiten Person (Richard) werden dann neue Bilder generiert. Auf diese Weise setzt sich die Reihe fort, und wird der Lehrer (Jim) wieder erreicht, kann auch hier überprüft werden, ob der Text des letzten Schülers (vor Jim) in der Reihe noch eine inhaltliche Ähnlichkeit zu dem Starttext aufweist oder sogar Wörter des Start-Textes enthält.

6.2.2.3.2 Kreieren von Sammelkarten

Der Weg von illustrierten Kurzgeschichten zu Sammelkarten ist nicht weit, wie man anhand einer Sammelkarte des Sammelkartenspiels Magic the Gathering zeigen kann:



Abbildung 7: Sammelkarte: Bogenschütze der Elfen. Quelle: Wizards of the Coast (1995). Bild: Maddocks.

Die Abbildung zeigt die Sammelkarte mit dem Namen „Bogenschütze der Elfen“. Diese Karte weist die folgenden Attribute auf, die in drei Kategorien eingeteilt werden können:

1. Deklarative Attribute:

- Name der Sammelkarte,
- Name des Künstlers,
- Name des Unternehmens (Herausgeber),
- Jahr der Veröffentlichung.

2. Künstlerische Attribute:

- grafische Darstellung,
- Farbe bzw. Sorte der Karte (grün)
- Das zum Namen der Karte passende Zitat aus der literarischen Welt von „Magic the Gathering“.

3. Spielerische Attribute:

- Fähigkeiten
- Kampf- und Verteidigungskraft (2/1)
- Klassifizierung (Elf)
- Kosten an Spiel-Ressourcen, die Karte ins Spiel zu bringen (oben rechts).

Die Web-Applikation könnte um ein Feature erweitert werden, das die Erstellung von eigenen Sammelkarten durch den Nutzer nach dem künstlerischen Vorbild der gezeigten Sammelkarte möglich macht. Dafür könnte der Einfachheit halber auf die „spielerischen Attribute“ verzichtet werden und stattdessen der Fokus auf die Kernattribute gelegt werden, die während des Verfassens der Kurzgeschichten mit dem in der Thesis entwickelten Prototypen ohnehin zum Einsatz kommen ([Kapitel 3.4](#)):

- Das Zitat bzw. ein durch den Nutzer der Web-Applikation verfasster Textabschnitt,
- Das dazugehörige generierte Bild,
- Der Name der Karte (Benutzereingabe),
- Die Farbe des Layout-Randes (Benutzereingabe),
- Der Name des Autors (Nutzer der Webapplikation) der Sammelkarte.

Damit wären die deklarativen sowie die künstlerischen Attribute der Sammelkarten zu einem Großteil ausgefüllt. Dieser vorgeschlagene Prozess des Kreierens von eigenen Sammelkarten durch die Nutzer bzw. Schüler kann vermutlich angesichts der ungebrochenen Popularität von Sammelkartenspielen einen zusätzlichen Gamficiation-Mechanismus ([Kapitel 2.2.3](#)) für die Web-Applikation darstellen und damit die Motivation der Schüler, Kurzgeschichten zu verfassen (und das Lesen und Schreiben sehr gut zu lernen, damit es qualitativ hochwertige Karten werden), weiter erhöhen.

Die angesprochene Popularität von Sammelkartenspielen gilt dabei sowohl für analoge als auch digitale Sammelspiele. Wizards of the Coast LLC, das Unternehmen hinter der Marke „Magic the Gathering“, bietet sein Kartenspiel inzwischen sowohl digital als auch analog an und erzielte in den Geschäftsjahren 2021 und 2022 zusammengerechnet einen Umsatz von mehr als 2,4 Milliarden US-Dollar. Das Unternehmen schätzt die Spielerzahl dabei auf circa 30 Millionen.¹³³ Das Sammelkartenspiel Hearthstone von Blizzard-Activision, das dagegen nur als digitale Applikation verfügbar ist, verzeichnet circa 5 Millionen aktive Spieler.¹³⁴

¹³³ <https://de.statista.com/infografik/30661/umsatz-von-wizards-of-the-coast-und-anteil-am-hasbro-umsatz>.

Datum des Abrufs: 06.01.2024.

¹³⁴ <https://activeplayer.io/hearthstone/>. Datum des Abrufs: 06.01.2024.

Das Spiel Pokemon Go, das zwar nicht mit Sammelkarten arbeitet, bei dem jedoch wie bei den anderen genannten Spielen das Sammeln von fantastischen Lebewesen mit jeweils einzigartigen, von den Game-Designern kreierten Fertigkeiten sowie mit der künstlerischen, grafischen Darstellung im Mittelpunkt steht, verzeichnet knapp 90 Millionen aktive Spieler.¹³⁵

6.2.2.3.3 Text-zu-Musik-Generator (Künstliche Intelligenz)

In Kombination mit der Text-zu-Bild-Verarbeitung könnte der Text, den der Nutzer verfasst, zusätzlich mit Hilfe von entsprechenden Algorithmen in Musik verwandelt werden, um weitere Anreize zu schaffen, die eigenen Texte zu erweitern, zu korrigieren und zu verbessern. Dabei bietet sich z.B. die Nutzung des Open-Source-Projekts Music Gen des Unternehmens Meta (früher Facebook) an, um das Feature in den Prototypen der Thesis zu integrieren. Dieses Werkzeug wurde für die Demonstrationsvideos genutzt ([Kapitel 5.2](#)).

6.2.2.4 Barrierefreiheit (Verbesserung)

6.2.2.4.1 Text-to-Speech-Support

Des Weiteren könnte die Web-Applikation über einen Text-to-Speech-Mechanismus verfügen, mit dem die Kurzgeschichten durch den Computer automatisch vorgelesen werden. Die Erzählstimme sollte dabei menschlich, d.h. nicht zu mechanisch klingen, damit das Nutzererlebnis positiv und das gesprochene Wort gut verständlich ist. Dieses Feature des Text-to-Speech-Support kann sowohl Kindern mit einem visuellen Handicap helfen, die Starttexte der Templates ([Kapitel 5.3](#)) sowie die Kurzgeschichten der Mitschüler zu hören und mitzuerleben (in der Schule sowie von zu Hause aus etc.), als auch das Erlebnis der anderen Kinder verbessern und komplettieren, falls diese sich eine Kurzgeschichte anhören wollen.

6.2.2.4.2 Speech-to-Text-Support

Nicht vernachlässigt werden sollte in diesem Zusammenhang der Speech-to-Text-Support, der über das Mikrofon des Endgeräts das gesprochene Wort aufnehmen kann und es innerhalb der Web-Applikation in die Textform verwandelt, damit Kinder mit körperlichen Handicaps ihre Kurzgeschichten verfassen und die dazugehörigen Bilder generieren lassen können. Dieses Feature würde zusätzlich für alle Kinder den Vorteil inhabitieren, neben dem Tippen an der Tastatur auch das deutliche Sprechen zu verbessern.

¹³⁵ <https://activeplayer.io/pokemon-go/>. Datum des Abrufs: 06.01.2024.

Literaturverzeichnis

Abraham et al. (2014). Kapitel: „Kreatives“ und „poetisches“ Schreiben.

In: Deutschunterricht in Theorie und Praxis. (DTP). Handbuch zur Didaktik der deutschen Sprache und Literatur in elf Bänden. 4. Schriftlicher Sprachgebrauch - Texte verfassen. Herausgeber: Feilke und Pohl. Verlag: Schneider Hohengehren.

Andresen et al. (2006). Freies Schreiben von Anfang an - wichtig oder schädlich? Grundschule aktuell: Zeitschrift des Grundschulverbandes, Nr. 96.

Angular Dokumentation.

Datum des Abrufs: 3.12.2023 von <https://angular.io/guide/what-is-angular>.

Angular Dokumentation.

Datum des Abrufs: 14.12.2023 von <https://angular.io/guide/component-overview>.

Autorengruppe Bildungsberichterstattung (2020). Bildung in Deutschland 2020.

Ein indikatorengestützter Bericht mit einer Analyse zur Bildung in einer digitalisierten Welt. Verlag: wbv, Bielefeld.

Bainomugisha et al. (2012). A survey on reactive programming. ACM Computing Surveys.

<https://nru.uncst.go.ug/bitstream/handle/123456789/7259/A%20Survey%20on%20Reactive%20Programming.pdf?sequence=1&isAllowed=y>, Datum des Aufrufs: 18.12.2023.

Balzert (2009). Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. 3. Auflage. Heidelberg: Spektrum Akademischer Verlag.

Becker-Mrotzek et al. (2015). Schreibkompetenz entwickeln und beurteilen. 6. Auflage. Verlag: Cornelsen. Berlin.

Becker-Mrotzek (2007). Im Deutschunterricht (neue) Medien sinnvoll nutzen.

Bos et al. (2003). Erste Ergebnisse aus IGLU. Schülerleistungen am Ende der vierten Jahrgangsstufe im internationalen Vergleich. Zusammenfassung ausgewählter Ergebnisse. Verlag: Waxmann.

Breiter et al. (2015).

Szenarien lernförderlicher IT-Infrastrukturen in Schulen. Betriebskonzepte, Ressourcenbedarf und Handlungsempfehlungen.

<https://www.bertelsmann-stiftung.de/de/publikationen/publikation/did/szenarien-lernfoerderlicher-it-infrastrukturen-in-schulen/>. Datum des Abrufs: 13.01.2024.

Brügelmann, Brinkmann (2012): Freies Schreiben im Anfangsunterricht? Eine kritische Übersicht über Befunde der Forschung. Leseforum.ch 2/2012.

https://www.leseforum.ch/myUploadData/files/2012_2_Bruegelmann.pdf.

Datum des Abrufs: 13.12.2023.

Bulut (2019). Handschrift in der digitalisierten Welt.

Mercator-Institut für Sprachförderung und Deutsch als Zweitsprache. Köln.

Bundesministerium für Bildung und Forschung. „DigitalPakt Schule 2019 bis 2024“.

https://www.digitalpaktschule.de/files/VV_DigitalPaktSchule_Web.pdf.

Datum des Abrufs: 28.12.2023.

Bundesministerium für Bildung und Forschung. Die Finanzen im „DigitalPakt Schule“.

<https://www.digitalpaktschule.de/de/die-finanzen-im-digitalpakt-schule-1763.html>.

Datum des Abrufs: 28.12.2023.

Campbell (1949). The Hero with a Thousand Faces. Kapitel 1: The Call to Adventure.

Chhetri (2016). A Comparative Analysis of Node.js (Server-Side JavaScript).

Saint Cloud State University.

Coenen (2021). Aus: Vitako aktuell. 01.2021.

Zeitschrift der Bundes-Arbeitsgemeinschaft der Kommunalen IT-Dienstleister.

Conallen (1999). Modeling Web Application Architectures with UML.

Wissenschaftliches Magazin: Communications of the Association for Computing Machinery.

Dahlmann (2021). Aus: Vitako aktuell. 01.2021.

Zeitschrift der Bundes-Arbeitsgemeinschaft der Kommunalen IT-Dienstleister.

Dehn et al. (2015). Schreiben: Von der Idee zum Text. Sammelband Grundschule.

Eickelmann et al. (2019). ICILS 2018. Deutschland auf einen Blick.

Computer- und informationsbezogene Kompetenzen von Schülerinnen und Schülern im zweiten internationalen Vergleich und Kompetenzen im Bereich Computational Thinking.

Eriksson et al. (2004). UML 2 Toolkit.

Geisler (2014). Datenbanken. Grundlagen und Design. 5. Auflage.

Grabowski et al. (2007).

Kapitel: Welche Schreibkompetenz? Handschrift und Tastatur in der Hauptschule.

In: Texte schreiben. Herausgeber: Becker-Mrotzek und Schindler. Verlag: Gilles und Francke.

Ergebnisse des 16. Symposions Deutschdidaktik 2006 in Weingarten.

Hagel. Vor-Ort- vs. Cloud-Speicher.

<https://www.hagel-it.de/cloud/vor-ort-vs-cloud-speicher-was-ist-besser-ein-vergleich.html>.

Datum des Abrufs: 30.01.2024.

Held et al. (2000). Server Management. Verlag: Auerbach, London.

Höller (2019). Angular - Das umfassende Handbuch. 2. Auflage. Rheinwerk Verlag Bonn.

Huettig et al. (2018). The culturally co-opted brain: how literacy affects the human mind. Language, Cognition and Neuroscience, Vol. 33, No. 3.

Hwang et al. (2010). E-Learning in the Republic of Korea.

Jacobson (2011). Use Case 2.0 - The Guide to Succeeding with Use Cases.

Jambor-Fahlen (2018). Lesen und Schreiben lernen in der Grundschule.

Mercator-Institut für Sprachförderung und Deutsch als Zweitsprache, Köln.

Kapp (2012). The gamification of learning and instruction: Game-based methods and strategies for training and education. Verlag: Pfeiffer. San Francisco.

Kleinbub (2018). Unterricht. Schlaglichter auf ein komplexes Forschungsfeld.

In: Empirische Forschung in der Deutschdidaktik. Band 3. Forschungsfelder der Deutschdidaktik.

Koch (2021). Aus: Vitako aktuell. 01.2021.

Zeitschrift der Bundes-Arbeitsgemeinschaft der Kommunalen IT-Dienstleister.

Kosar et al. (2008). A preliminary study on various implementation approaches of domain-specific language. Information and Software Technology 50.

Kuhn (2021). Deutsches Schulportal der Robert-Bosch-Stiftung.

<https://deutsches-schulportal.de/unterricht/rechtschreibung-an-schulen-rueckt-in-den-fokus/>.

Datum des Abrufs: 30.12.2023.

Kultusministerium Baden-Württemberg.

Leitfaden für den Betrieb von Lern-, Informations- und Kommunikationsplattformen. (März 2018).

Küspert (1998). Phonologische Bewusstheit und Schriftspracherwerb. Zu den Effekten der vorschulischen Förderung der phonologischen Bewusstheit auf den Erwerb des Lesens und Rechtschreibens. Herausgeber: Peter Lang, Frankfurt am Main.

Landesbildungsserver Baden-Württemberg.

Lese- und Schreibunterricht in der Grundschule: 4-Säulen-Modell.

https://www.schule-bw.de/themen-und-impulse/ideenpool-lesen/grundschule/grundschule34/methoden_konzepte_projekte/didaktik-und-methodik-des-lesens/modell.

Datum des Abrufs: 10.12.2023.

Laurie et al. (2002). Apache: The Definitive Guide, 3rd Edition. Kapitel 1.

Lei et al. (2014). Performance Comparison and Evaluation of Web Development Technologies in PHP, Python, and Node.js. IEEE 17th International Conference on Computational Science and Engineering, Chengdu, China.

Meiers (2008). Mehrperspektivischer Unterricht.

In: Geschichte und historische Konzeptionen des Sachunterrichts.

Herausgeber: Kaiser und Pech. 2. Aufl. Verlag: Schneider Hohengehren.

Le Hégaret et al. (2004). Document Object Model (DOM).

Level 3 Core Specification Version 1.0 W3C Recommendation.

Lewalter et al. (2023). PISA 2022. Analyse der Bildungsergebnisse in Deutschland. Technische Universität München. School of Social Sciences and Technology. Zentrum für Internationale Vergleichsstudien (ZIB). Verlag: Waxmann.

https://www.pisa.tum.de/fileadmin/w00bgi/www/Berichtsbaende_und_Zusammenfassungen/PISA-2022-zusammenfassung.pdf, Datum des Abrufs: 30.12.2023.

Mußmann et al. (2021). Digitalisierung im Schulsystem 2021.

Arbeitszeit, Arbeitsbedingungen, Rahmenbedingungen und Perspektiven von Lehrkräften in Deutschland.

Kooperationsstelle Hochschulen und Gewerkschaften der Georg-August-Universität Göttingen. <https://publications.goettingen-research-online.de/handle/2/89741>.

Prasad (2022). Research and Analysis of the Frontend Development Frameworks and Libraries with Voice Recognition in Remote Areas for Developing Ecommerce Business for People of Remote Areas where People have less or no Knowledge of Technical Devices. International Journal of Engineering Applied Sciences and Technology. Vol. 6. Issue 11.

Von Werder (2016). Lehrbuch des kreativen Schreibens. 2. Auflage. Verlag: Marix.

Reble et al. (2020). Kapitel: Am Computer oder handschriftlich Schreiben?

In: Bildung, Schule, Digitalisierung. Herausgeber: Kaspar, Becker-Mrotzek, Hofhues, König und Schmeinck. Verlag: Waxmann, Münster.

Resch et al. (2021). Aus: Vitako aktuell. 01.2021.

Zeitschrift der Bundes-Arbeitsgemeinschaft der Kommunalen IT-Dienstleister.

Rogers (1961) Kapitel: Personality Change in Psychotherapy.

In: On becoming a Person. A Therapist's View on Psychotherapy. New York.

Rose et al. (2011). Garbage In, Garbage Out: Having Useful Data Is Everything.

Rust (2024) Digitale Tools & Apps für den Unterricht. Schulflix, Klett EduLabs.

<https://www.schulflix.com/dossier/digitale-tools-fuer-den-unterricht/>.

Datum des Abrufs: 30.01.2024.

Prasanna et al. (2021). PoC Design: A Methodology for Proof-of-Concept (PoC) Development on Internet of Things Connected Dynamic Environments.

Salvaneschi et al. (2014). REScala: Bridging between object-oriented and functional style in reactive applications.

Von: https://programming-group.com/assets/pdf/papers/2014_REScala-Bridging-The-Gap-Between-Object-Oriented-And-Functional-Style-In-Reactive-Applications.pdf.

Datum des Abrufs: 18.12.2023.

Salvaneschi et al. (2016). Debugging for Reactive Programming.

Von:

https://programming-group.com/assets/pdf/papers/2016_Debugging-for-Reactive-Programming.pdf, Datum des Abrufs: 18.12.2023.

Schmid et al. (2017). Monitor Digitale Bildung. Die Schulen im digitalen Zeitalter. Gütersloh. Bertelsmann-Stiftung.

Spinner (2015). Kreatives Schreiben. Kreativer Deutschunterricht. Identität - Imagination - Kognition. 3. Aufl. Verlag: Seelze Velber.

Stampfl (2012). Die verspielte Gesellschaft. 1. Auflage. Heise Zeitschriften Verlag. Hannover.

Stangl et al. (2021). Aus: Vitako aktuell. 01.2021.

Zeitschrift der Bundes-Arbeitsgemeinschaft der Kommunalen IT-Dienstleister.

Stanat et al. (2022). IQB-Bildungstrend 2021. Kompetenzen in den Fächern Deutsch und Mathematik am Ende der 4. Jahrgangsstufe im dritten Ländervergleich. Verlag: Waxmann.

Suits (1978). The Grasshopper: Games, Life, and Utopia.

Dritte Auflage 2014. Verlag: Broadview Press.

Watt et al. (1990). Programming Language Concepts and Paradigms. Prentice Hall.

Wawrzyniak (2020). Datenschutz in der Schule – (k)eine Herausforderung! Bundeszentrale für politische Bildung.

<https://www.bpb.de/lernen/digitale-bildung/werkstatt/318529/datenschutz-in-der-schule-k-eine-herausforderung/>. Datum des Abrufs: 13.02.2024.

Wilken (2018). Angular in Action. Manning Publications Co.

Yang et al. (2023). MMA-Diffusion: MultiModal Attack on Diffusion Models.

Anhang

A Glossar

Application Programming Interface (API):

API heißt übersetzt: Programmierschnittstelle für (digitale) Anwendungen. APIs werden genutzt, um die Kommunikation von zwei oder mehr Programmen und/oder Komponenten möglich zu machen und so das Zusammenwirken verschiedener Systeme zu begünstigen.

Browser: Anwendung für den Computer zum Anzeigen von Webseiten.

Web-Development: steht für den Prozess des Konzipierens und Programmierens von Softwareprodukten, die über das Internet bzw. einen Browser erreichbar sind.

Framework: Programmiergerüst, das das Programmieren und Strukturieren von objektorientierten und komponentenbasierten Anwendungen erleichtern kann.

Angular: Der Name des für das Programmieren des Prototypen genutzten Web-Development-Frameworks ([Kapitel 4.1.1](#)). Das englische Wort für “eckig”.

Engine: Die Nutzung des Begriffs Engine bezieht sich in der Thesis auf die Software Engine. Dieser Begriff ist eine Metapher für den Automotor (englisch: car engine) und wird genutzt, um ein komplexes Teil-System eines Software-Systems zu bezeichnen.

Feature: - fasst eine Funktion des Softwareprodukts in meist einem Wort zusammen (z.B. Feature 1: Kurzgeschichten-Editor, Feature 2: Bild-Generator-Anbindung).

Anwendungsfall (englisch Use Case): Dieser Begriff wird verwendet, wenn einzelne Anwendungsfälle innerhalb eines Features aus Nutzersicht beschrieben werden (z.B. Fall 1: Der Nutzer kann die Kurzgeschichte verfassen, Fall 2: Der Nutzer fordert ein Bild an).

OECD: The Organisation for Economic Co-operation and Development.

PISA: Programme for International Student Assessment.

Grafisches Benutzer-Interface, alternativ: Grafische Benutzeroberfläche, englisch: Graphical User Interface (GUI). Das Fenster, das der Nutzer eines Software-Systems im Bildschirm sieht und in dem er mit Hilfe von grafischen Elementen mit dem System interagiert.

B Abbildungsverzeichnis

Abbildung 1: Entwicklung der Mittelbindung und des Mittelabflusses aus dem Sondervermögen des Bundes für den „DigitalPakt Schule 2019-2024“.

Quelle: Bundesministerium für Bildung und Forschung.

<https://www.digitalpaktschule.de/de/die-finanzen-im-digitalpakt-schule-1763.html>.

Datum des Abrufs: 28.12.2023.

Abbildung 2: Anwendungsfalldiagramm (englisch: Use-Case-Diagram).

Darstellung: Autor der Thesis.

Abbildung 3: Aktivitätsdiagramm (englisch: Activity-Diagram).

Darstellung: Autor der Thesis.

Abbildung 4: Kostenübersicht der Abonnenten-Modelle. Quelle: Leonardo.ai.

<https://app.leonardo.ai/buy-api>.

Abbildung 5: Text-Editor-Component DOM-Tree.

Darstellung von: Compodoc (Dokumentationswerkzeug).

Abbildung 6: Screenshot der Webapplikation.

Quelle: Autor der Thesis.

Bilder generiert mit Leonardo.AI.

Abbildung 7: Sammelkarte: Bogenschütze der Elfen.

Quelle: Wizards of the Coast (1995). Bild: Maddocks.