

Masterarbeit

Frontend zu einer datenbankbasierten Webapplikation für den Einsatz in der digitalen Lehre

B.Sc. Osman Alić
Matrikelnummer: 184399
Studiengang: Maschinenbau

ausgegeben am:

09.09.2020

eingereicht am:

24.01.2021

Betreuer:

Dr.-Ing. Dipl.-Inf. Anne Antonia Scheidler

M. Sc. Joachim Hunker

Inhaltsverzeichnis

Abbildungsverzeichnis	I
Abkürzungsverzeichnis	III
Tabellenverzeichnis	IV
1 Einleitung	1
2 Grundlagen zur Gestaltung und Entwicklung von Weboberflächen.....	4
2.1 HTML und CSS	4
2.2 User-Experience und User-Interface.....	6
2.3 Softwaretests.....	9
2.3.1 Die sieben Grundsätze des Softwaretestens.....	10
2.3.2 Komponententest und Integrationstest.....	11
2.3.3 Systemtest und Abnahmetest	18
2.3.4 Usability- Test.....	20
3 Didaktische Theorien und Modelle in der Informatik	22
3.1 Motivation in der Psychologie	22
3.2 Kognitivistisches und konstruktivistisches Lernen.....	24
3.3 Bildungstheoretische und kritisch-konstruktive Didaktik nach Klafki.....	25
3.4 Didaktik der Informatik nach Baumann	27
4 Adaption von didaktischen Modellen in eine SQL-Lernwebapplikation.....	29
4.1 Diskussion der didaktischen Theorien und Modelle	29
4.2 Nutzung des Modells nach Baumann zur Erstellung von	32
Übungsaufgaben	32
5 Entwicklung des Frontends.....	39
5.1 Gestaltung der graphischen Benutzeroberfläche	39
5.1.1 Anmeldemaske und Arbeitsoberfläche.....	39
5.1.2 Responsives Design	49
6 Durchführung des Usability-Tests	51
6.1 Versuchsaufbau und Testpersonen	51
6.2 Auswertung der Tests.....	52

6.3	Fazit	58
7	Zusammenfassung und Ausblick.....	61
	Literaturverzeichnis	V
	Anhang I: Übungsaufgaben.....	XVI
	Anhang II: Zustandsdiagramme der Softwaremodule	XXIX
	Anhang III: Hilfe.pdf	XXXV
	Anhang IV: Fragebogen zum Usability-Test.....	XXXVIII
	Eidesstattliche Versicherung	XLVI

Abbildungsverzeichnis

Abbildung 2.1: Allgemeines V-Modell (Spillner et al., 2017).....	9
Abbildung 2.2: Schematische n -stufige Gliederung	13
Abbildung 2.3: Schematischer Aufbau des Top-Down (Liggesmeyer, 2009)	14
Abbildung 2.4: Schematischer Aufbau des Bottom-Up (Liggesmeyer, 2009).....	15
Abbildung 2.5: Schematischer Aufbau des Outside-In (Liggesmeyer, 2009)	17
Abbildung 3.1: Zusammenhang zwischen Motivatio, Motivierung und Motiv (Hubweiser, 2007)	23
Abbildung 4.1: Übersicht zu den Übungsaufgaben und deren Schwierigkeitsgrad	38
Abbildung 5.1: Anmeldemaske	40
Abbildung 5.2: zweistufige hierarchische Gliederung der Softwaremodule der Webapplikation.....	40
Abbildung 5.3: Maske zur Registrierung	41
Abbildung 5.4: Registrierungsbestätigung	42
Abbildung 5.5: Maske zum Modul „Passwort vergessen“	42
Abbildung 5.6: Versandbestätigung	43
Abbildung 5.7: Arbeitsoberfläche der Webapplikation.....	44
Abbildung 5.8: linke und rechte Seite des Banners	44
Abbildung 5.9: Navigationsleiste.....	45
Abbildung 5.10: Felder von nicht gelösten und gelösten Aufgaben.....	45
Abbildung 5.11: Aufgabenstellung in der Webapplikation	45
Abbildung 5.12: Eingabefeld mit durchgeführter Eingabe	46
Abbildung 5.13: „Ausführen“-Button	46
Abbildung 5.14: Fehlermeldungen bei syntaktischen und lexikalischen Fehlern	46
Abbildung 5.15a: Fehlermeldung bei semantischen Fehlern	47
Abbildung 5.15b: Fehlermeldung bei semantischen Fehlern	47
Abbildung 5.16: Meldung bei richtiger Eingabe.....	48
Abbildung 5.17: „Lösung“-Button	48
Abbildung 5.18: Arbeitsoberfläche der Webapplikation in der Tablet-Ansicht	49
Abbildung 5.19: Arbeitsoberfläche der Webapplikation in der Smartphone-Ansicht.....	50
Abbildung 6.1: Bewertung der intuitiven Ästhetik.....	53
Abbildung 6.2: Bewertung der Schwierigkeit der Aufgaben	54
Abbildung 6.3: Durchschnittliche Bewertung der Schwierigkeit der Aufgaben.....	55
Abbildung 6.4: Bewertung der Wiederholung der Hinweise	56

Abbildung 6.5: Bewertung der reflektiven Ästhetik und deren Auswirkung auf die Gefühlslage bei der Bedienung der Webapplikation	56
Abbildung All.I: Zustandsdiagramm zum Softwaremodul „Seite laden“	XXIX
Abbildung All.II: Zustandsdiagramm zum Softwaremodul „Verbindungsabbruch“	XXX
Abbildung All.III: Zustandsdiagramm zum Softwaremodul „Anmeldung“	XXX
Abbildung All.IV: Zustandsdiagramm zum Softwaremodul „Registrierung“	XXXI
Abbildung All.V: Zustandsdiagramm zum Softwaremodul „Passwort vergessen“ ...	XXXII
Abbildung All.VI: Zustandsdiagramm zum Softwaremodul „Seitennavigation“	XXXIII
Abbildung All.VII: Zustandsdiagramm zum Softwaremodul „Eingabepfung“	XXXIV

Abkürzungsverzeichnis

CSS	Cascading Style Sheets
GUI	Graphical User-Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ISTQB	International Software-Testing Qualification Board
JSON	JavaScript Object Notation
SMGL	Standard Generalized Markup Language
SQL	Structured Query Language
TN	Teilnehmer
UI	User-Interface
UNESCO	United Nations Educational, Scientific and Cultural Organization
URI	Uniform Resource Identifier
UX	User-Experience
WWW	World Wide Web

Tabellenverzeichnis

Tabelle 2.1: Beispiele zu HTML-Tags (Bühler et al., 2017).....	5
Tabelle 2.2: Beispiele zu CSS-Befehlen (Bühler et al., 2017).....	5
Tabelle 4.1: Parallelen von Baumann zum Kognitivismus, Konstruktivismus, und zu Klafki	31
Tabelle 4.2: Übersicht der Übungsaufgaben.....	32
Tabelle 6.1: Einwänder der Probanden während des Usability-Tests	58
Tabelle AI.I: Studierende	XV
Tabelle AI.II: Studierendenliste	XX
Tabelle AI.III: Klausurergebnisse.....	XXII
Tabelle AI.IV: Mechanikergebnisse	XXIV
Tabelle AI.V: Studierendeninformationen	XXIV
Tabelle AI.VI: Ergebnisse	XXVI
Tabelle AI.VII: Studierendeninformationen	XXVI

1 Einleitung

Im Zuge der Digitalisierung erfährt die Gesellschaft in sämtlichen Bereichen einen stetigen Wandel. Neben der alltäglichen digitalen Kommunikation, finden auch immer mehr Veränderungsprozesse im Bereich der Lehre statt (Hermann & Temath, 2019). Durch die zunehmende Digitalisierung erhalten computergestützte Lernwerkzeuge eine wachsende Anwendung. Besonders im Bereich der Informationstechnik bietet es sich an, den Lehrbetrieb durch solche Lernwerkzeuge zu unterstützen. So können den Studierenden individuelle Lernformen angeboten werden. Zu diesen Lernformen zählt das asynchrone Lernen. Das asynchrone Lernen beschreibt die zeitversetzte Kommunikation zwischen den Studierenden und der lehrenden Person. Dadurch können die Studierenden die Zeiteinteilung unabhängig von Lehrveranstaltungen gestalten und im Selbststudium unterstützt werden (Grosch, 2018). Die digitalen Werkzeuge, die den Studierenden zur Durchführung des asynchronen Lernens von Hochschule und Universitäten angeboten werden, können nicht nur den Studierenden eine Hilfe sein. Sie ermöglichen zudem den Lehrenden die durch das computergestützte Lernwerkzeug erhobenen Daten zu nutzen, um den Lernfortschritt der Studierenden auszuwerten. So kann auf die Qualität und den Nutzen des Lernwerkzeuges geschlossen und dieses gegebenenfalls weiter an die Studierenden angepasst werden.

Aktuell wird die Durchführung von Lehrveranstaltungen durch die COVID-19-Pandemie stark eingeschränkt. Seit März 2020 ist das Abhalten von Präsenzveranstaltungen weltweit nahezu unmöglich. Laut der United Nations Educational, Scientific and Cultural Organization (UNESCO) werden weltweit Millionen von Studierende durch die Corona-Krise daran gehindert an Präsenzveranstaltungen teilzunehmen (UNESCO, 2020). Dies verdeutlicht die steigende globale Notwendigkeit von digitalen Lernwerkzeugen, um den Lehrbetrieb sowohl orts- als auch zeitunabhängig zu gestalten. Während des in Deutschland verhängten Lockdowns zeigte sich zudem, dass die Menschen auch bereit dazu sind, die Digitalisierung in ihren Alltag einfließen zu lassen. So wurden bspw. Einkäufe größtenteils digital durchgeführt. Laut Statista ist der monatliche Umsatz im Versand- und Interneteinzelhandel zwischen März und Juni 2020 um 20-30% pro Monat gestiegen (Statista, 2020a), was auf eine höhere Nachfrage während des Lockdowns zurückzuführen ist. Die dazu benötigte Technologie ist bereits entwickelt worden. Nun muss diese lediglich im Bereich der Lehre adaptiert werden, sodass Studierende die Lerninhalte von Hause aus erarbeiten können, ohne dass Nachteile bei der Prüfungsvorbereitung aufkommen.

Das Ziel der vorliegenden Masterarbeit ist die Konzeption und Entwicklung eines computergestützten Lernwerkzeugs. Diese Webapplikation soll als digitales Lernwerkzeug zum Thema Structured Query Language (SQL) dienen und im Übungsbetrieb des Masterstudiums der Ingenieurwissenschaften an der Technischen

Universität Dortmund angewandt werden. Die in der Webapplikation enthaltenen Übungsaufgaben korrespondieren mit einer Auswahl an Übungsaufgaben, die vom Lehrstuhl IT in Produktion und Logistik zur Verfügung gestellt wurden. Die Entwicklung der Übungsaufgaben erfolgt dabei auf geeigneten didaktischen Gesichtspunkten. Den Studierenden soll die Eingabe von SQL-Statements ermöglicht werden, wobei der Lernprozess durch eine integrierte Benutzerunterstützung gefördert werden soll. Mit Hilfe der Mehrbenutzerfähigkeit soll jeder Studierende die Möglichkeit erhalten, Datenbanktabellen unabhängig von anderen Studierenden manipulieren und abfragen zu können.

Das Forschungsthema wird innerhalb einer Arbeitsgruppe bearbeitet. Die Arbeitsgruppe besteht aus den Teilnehmern Osman Alić (O. Alić) und Džemal Alić (D. Alić). Während sich die vorliegende Arbeit mit dem Frontend einer datenbankbasierten Webapplikation und der Gestaltung von geeigneten Übungsaufgaben auseinandersetzt, wird komplementär dazu die Konzeption und Entwicklung des Backends und die Aufbereitung des eingesetzten Datenbanksystems von D. Alić (Alić, 2021) erarbeitet.

Um das oben bereits erwähnte Ziel dieser Arbeit zu erreichen, wird diese in sieben Kapitel unterteilt. Im zweiten Kapitel werden dem Leser¹ zunächst die Grundlagen, die zur Gestaltung und Entwicklung von Weboberflächen benötigt werden, nähergebracht. Dazu werden zu Beginn die Themen HTML und CSS näher erläutert, da HTML- und CSS-Dateien im Webbrowser ausgelesen werden müssen, um die Webapplikation im Webbrowser darzustellen. Zudem erfährt der Leser inwieweit das User-Interface, das User-Experience und des Error-Handling für Gestaltung einer Weboberfläche von Bedeutung sind. Ferner werden in diesem Kapitel die Softwaretest nach dem V-Modell und zudem der Usability-Test vorgestellt. Diese sind wichtig, da nach erfolgreicher Programmierung durch verschiedene Tests Fehler gesucht werden, um sie anschließend zu beheben.

Da die zu erstellenden Übungsaufgaben didaktisch zu fundieren sind, werden im dritten Kapitel zunächst verschiedene didaktische Modelle insbesondere Modelle, die in der Informatiklehre verwendet werden, vorgestellt.

Im vierten Kapitel werden die didaktischen und psychologischen Grundlagen aus Kapitel drei miteinander verglichen, um eine didaktische Grundlage zur Erstellung von Übungsaufgaben zu schaffen. So werden auf dieser Grundlage und auf Basis der in der Arbeit von D. Alić (Alić, 2021) vorgestellten SQL-Befehle geeignete Übungsaufgaben erstellt. Dem Leser wird detailliert beschrieben warum die Aufgaben so konzipiert wurden, wie sie letztendlich in der Webapplikation vorliegen. Dabei steht der Lernerfolg der Studierenden im Vordergrund.

Nachdem die Übungsaufgaben entwickelt wurden, wird im fünften Kapitel die Entwicklung des Frontends thematisiert. Dabei werden dem Leser die Funktionen der

¹ Aus Gründen der besseren Lesbarkeit wird im Text verallgemeinert das generische Maskulinum verwendet.

Webapplikation anhand der Benutzeroberfläche und der einzelnen Module, aus denen sich die Webapplikation zusammensetzt, erörtert. Bei den Modulen handelt es sich unter anderem um die Funktionen Registrierung, Login und die Prüfung der Eingaben der Nutzer. Ferner befasst sich dieses Kapitel mit dem User-Interface und der User Experience in Bezug auf die Gestaltung der Benutzeroberfläche und klärt die Frage, warum die Benutzeroberfläche so gestaltet wurde, wie sie der Nutzer sieht.

Im Rahmen des sechsten Kapitels werden verschiedene Usability-Tests durchgeführt. Es wird geprüft inwieweit die didaktischen Grundlagen auf die Übungsaufgaben angewandt wurden und wie sich die Gestaltung der Webapplikation auf die User Experience auswirkt. Darüber hinaus werden alle Funktionen der Webapplikation von verschiedenen Versuchspersonen getestet, um herauszufinden, ob Usability-Probleme oder Programmierfehler vorliegen. Schließlich wird aus den gewonnenen Erkenntnissen ein Fazit gezogen.

Im siebten Kapitel werden die Ergebnisse zusammengefasst und es wird dem Leser ein Ausblick auf die Zukunft gegeben.

2 Grundlagen zur Gestaltung und Entwicklung von Weboberflächen

Im folgenden Kapitel wird der Leser in die Grundlagen, die zur Gestaltung und Entwicklung einer Weboberfläche notwendig sind, eingeführt. Dazu werden im ersten Abschnitt die Sprachen HTML und CSS und deren Zusammenhang beschrieben. An dieser Stelle ist anzumerken, dass zum Entwickeln einer Webapplikation die drei Elemente HTML, CSS und JavaScript notwendig sind. Während in diesem Kapitel lediglich auf HTML und CSS eingegangen wird, wird JavaScript in Abschnitt 3.2 der Arbeit von D. Alić thematisiert.

Im zweiten Abschnitt dieses Kapitels wird das User-Interface und die User-Experience eingeführt und voneinander abgegrenzt. Während mit HTML und CSS eine Webapplikation nach Wünschen des Entwicklers erstellt werden kann, wird im zweiten Abschnitt erläutert warum die Wünsche der Benutzer dieser Applikation ebenso wichtig sind und was das User-Interface und User-Experience dazu beitragen. Ferner wird beschrieben wie durch geeignete Gestaltung einer Applikation dem Nutzer nicht nur ein zielführendes, sondern auch Freude bereitendes Werkzeug zur Verfügung gestellt wird. Daher wird zum einen auf das sogenannte Error-Handling und zum anderen auf psychologischen Auswirkungen, die eine Benutzeroberfläche hat, eingegangen. Zudem sind Softwaretest, deren Vorgehensweisen und Auswirkungen Gegenstand dieses Kapitels.

2.1 HTML und CSS

Seitdem das *World Wide Web* (WWW) zu Beginn der neunziger Jahre von Tim Berners-Lee² bei CERN in Genf (Schweiz) erfunden wurde, gewann es mit fortschreitender Zeit immer mehr an Bedeutung für den Alltag eines jeden Menschen. Entscheidend für diesen Erfolg war das sogenannte HTML, welches ebenfalls von Berners-Lee erfunden wurde (Mellis et al., 2003). HTML ist die Abkürzung für *Hypertext Markup Language* – zu Deutsch: Hypertext Auszeichnungssprache. Das Kunstwort Hypertext beschreibt die Option eine Verbindung von Texten mit Hilfe von sogenannten Hyperlinks (oder auch kurz *Links*) herzustellen. So können Hypertexte die Struktur gedruckter Texte verlassen und erlauben durch Auswählen eines Links das Öffnen eines weiteren Dokuments. Dieses Prinzip trug maßgeblich dazu bei, dass sich das WWW so ausdehnen konnte wie es heute der Fall ist (Bühler et al., 2017).

² Sir Timothy John Berners-Lee, geboren am 08. Juni 1955 in London ist ein britischer Physiker und Informatiker. Heute ist er Professor am Massachusetts Institute of Technology (MIT) und Lehrstuhlinhaber an der Oxford-University (Oxford, 2016)

Auszeichnungssprachen dienen der Strukturierung elektronischer Dokumente und der Definierung einzelner Textbestandteile, wie Überschriften oder Absätze (W3C, 2020). Die Auszeichnung der Textbestandteile erfolgt durch die in ISO 8897 genormten SMGL-Elemente – auch *Tags* genannt. SMGL ist dabei die Abkürzung für *Standard Generalized Markup Language* – genormte verallgemeinerte Auszeichnungssprache (Mellis et al., 2003). In der folgenden Tabelle werden beispielhaft einige Elemente vorgestellt.

Tabelle 2.1: Beispiele zu HTML-Tags (Bühler et al., 2017)

<code><h1>...</h1></code>	markiert eine Überschrift (Headline) der Größe 1
<code>...</code>	zeichnet einen Text fett aus
<code>News</code>	definiert einen Link zu „news.htm“

Dabei wird durch die Punkte („...“) der Inhalt, auf den der Tag angewandt wird, repräsentiert. Zudem beinhaltet der schließende Teil des Tags zusätzlich einen Schrägstrich (Slash, „ /“) (Bühler et al., 2017).

Mit HTML wird also das Ziel verfolgt die Möglichkeiten zu schaffen Informationen aller Art in Dokumenten im Web weltweit plattformübergreifend zur Verfügung zu stellen. Dabei wird angestrebt, dass das Bereitstellen der Daten sowohl software- als auch hardwareunabhängig geschieht, was bei der Vielzahl an möglichen Zielplattformen eine große Herausforderung darstellt (Mellis et al., 2003).

Da HTML lediglich die Strukturierung eines Dokuments ermöglicht und nicht die Formatierung begünstigt, wird auf weitere Werkzeuge zurückgegriffen. Die visuelle Darstellung wird dabei von Gestaltungssprachen wie den *Cascading Style Sheets* (CSS) – gestufte Gestaltungsbögen – realisiert (W3C, 2020). Das Ziel von CSS ist die strikte Trennung der Darstellungsweise von Elementen und des Inhalts innerhalb eines Dokuments. Bei vollständiger Realisierung dieser Trennung, werden in HTML nur die inhaltliche Strukturierung eines Dokuments und die Bedeutung der einzelnen Bestandteile beschrieben, während bei CSS die Darstellung der Inhalte, wie z.B. das Layout, die Farben oder auch die Schriftart, definiert wird. Dazu können **Tabelle 2.2** beispielhafte Befehle zu CSS entnommen werden.

Tabelle 2.2: Beispiele zu CSS-Befehlen (Bühler et al., 2017)

<code>font-family</code>	Schriftgröße
<code>color</code>	Schriftfarbe
<code>text-align</code>	Textausrichtung

So können die Autoren eines Dokuments oder einer Webapplikation den Inhalt und das äußere Erscheinungsbild dieser individuell gestalten und beeinflussen (Flanagan, 2006). CSS-Anweisungen können in einer separaten Datei gespeichert werden und sind zudem auf sämtliche HTML-Dokumente anwendbar (Pomaska, 2005). Ein weiterer Vorteil von CSS ist, dass sich das ausgegebene Dokument bzw. die Webapplikation der

Bildschirmgröße anpassen kann. So können die nötigen Daten problemlos auch auf Smartphone oder Tablet aufgerufen werden (Haunschild, 2017).

2.2 User-Experience und User-Interface

Im vorigen Abschnitt wurde aufgezeigt, dass HTML und CSS zur Gestaltung einer Webapplikation für den Autor unentbehrlich sind. Zum einen kann der Autor die Webapplikation nach seinen Wünschen manipulieren. Jedoch geht auch mit der Beeinflussung der Webapplikation das Beeinflussen der Gefühle, die ein User bei der Benutzung dieser empfindet, einher. Um dies näher zu erläutern, werden im Folgenden die Begriffe User-Experience (UX) und User-Interface (UI) beschrieben und die Unterschiede und Gemeinsamkeiten voneinander abgegrenzt. Wie das Präfix „User“ (Deutsch: Nutzer, Benutzer) bereits verrät, steht beim Design der UX und UI der Benutzer bzw. der Endkonsument der Anwendung im Vordergrund. Das UX und das UI gehen immer zusammen einher (Herczeg, 2018).

User-Experience

Der Begriff UX wird in der Norm DIN EN ISO 9241-210 definiert als

„die Wahrnehmungen und Reaktionen einer Person, welche aus dem Gebrauch und/oder erwartetem Gebrauch eines Produkts, Systems oder Dienstes resultieren. Es schließt alle Emotionen, Überzeugungen, Präferenzen, physische und psychologische Wahrnehmungen und Reaktionen, Verhaltensweisen und Fertigkeiten ein, die vor, während oder nach dem Gebrauch auftreten.“
(Thesmann, 2016)

und wird in derselben Norm als „Nutzererlebnis“ übersetzt. Also besteht das Ziel beim Design der UX ein für den Nutzer emotional ansprechendes System zu entwickeln, welche zusätzlich Emotionen und Erwartungen des Nutzers berücksichtigt. Im Rahmen der sogenannten Software-Ergonomie ergibt sich daraus eine Vielzahl an Kriterien, die die Zufriedenstellung (vgl. DIN EN ISO 9241-11) des Nutzers durch die UX gewährleisten. Die Kriterien werden abhängig von der gestalteten Webapplikation anwendungsspezifisch ausgewählt (Herczeg, 2018). Denn es ist bspw. zu unterscheiden, ob mit der Webapplikation eine Aufgabe gelöst werden soll und diese eine hohe Usability (deutsch: Gebrauchstauglichkeit) aufweist oder, ob lediglich soziale Bedürfnisse zu befriedigen sind, wie es bei sozialen Netzwerken der Fall ist (Ilmberger et al., 2009). Ersteres unterliegt den in DIN EN ISO 9241-110 beschriebenen Grundsätzen der Dialoggestaltung. Diese beinhalten unter anderem die Kriterien Selbstbeschreibungsfähigkeit und Lernförderlichkeit (Herczeg, 2018). Jedoch umfasst die Usability auch die Art und Weise wie die Webapplikation damit umgeht, wenn der

Benutzer einen Fehler macht. Dies wird in der IT „*Error Handling*“ genannt. Die Vernachlässigung des Error Handlings während der Entwicklung einer Webapplikation kann dazu führen, dass die Applikation plötzlich beendet wird oder dem Nutzer nicht klar ist, warum die Aktion, die dieser ausführen möchte, nicht ausgeführt werden kann. Andererseits kann das Error Handling genutzt werden, um den Nutzer auf den begangenen Fehler aufmerksam zu machen und diesem eine Lösung vorzuschlagen (Moser, 2012; Springer, 2013). Im Bereich der Dialogstrategie wird untersucht wie bestimmte Informationen im Mensch–Maschine–Dialog verarbeitet werden. Dialogsysteme zwischen Mensch und Computer haben nur ein begrenztes Wissen und sind auf eine bestimmte Domäne beschränkt, sodass ein Austreten aus der Domäne oder das Anwenden von unbekanntem Wissen für einen fehlerhaften Dialog sorgen können (Denecke, 2002). Es lässt sich sagen, dass ein Fehler aus mangelnder Abdeckung auf einer der folgenden Ebenen resultiert (Gieselmann, 2007):

1. Lexikalische Ebene: Fehler auf dieser Ebene werden ausgelöst, wenn der Benutzer Begriffe verwendet, welche nicht im Vokabular der Maschine existieren
2. Syntaktische Ebene: Werden bekannte Worte oder erwartete Befehle grammatikalisch falsch konstruiert, treten Fehler auf dieser Ebene auf
3. Semantische Ebene: Es ist möglich, dass der Nutzer den Befehl syntaktisch zwar fehlerfrei angibt, jedoch dabei die eigentliche Aufgabe missachtet. Der eingegebene Code ist zwar durchführbar, erreicht jedoch nicht das angestrebte Ziel.

Neben der Usability einer Webapplikation, wurde um die Jahrtausendwende herum zunehmend der Einfluss des *Joy Of Use* (Deutsch: Freude an der Benutzung) untersucht. Dazu werden heute beim Design einer Webapplikation die Ästhetik (Hassenzahl, 2002; Ilmberger et al., 2009), beeinflusst durch Symmetrie (Seckler & Tuch, 2012), Buntheit/ Farbvielfalt und die visuelle Komplexität (Miniukovich & De Angeli, 2015; Tuch et al., 2012), berücksichtigt. Es ist jedoch nicht möglich, dass die UX allein betrachtet wird. Denn zunächst muss eine Benutzeroberfläche entwickelt werden, durch welche der Nutzer überhaupt eine Erfahrung bei der Anwendung der Webapplikation machen kann. Die Entwicklung und Gestaltung einer solchen Oberfläche wird im Rahmen des Designs eines User-Interfaces thematisiert.

User-Interface

Zum Begriff UI sind in der Literatur mehrere Übersetzungen zu finden. Neben „Benutzungsschnittstelle“ (Herczeg, 2018) und „Softwareoberfläche“ (Chlebek, 2006) wird der Begriff „Benutzerschnittstelle“ (Chlebek, 2006) am häufigsten genutzt. Im weiteren Verlauf der Arbeit wird sich auf den Terminus „Benutzerschnittstelle“ beschränkt. Das UI ist maßgeblich für die UX, da sich die UX nach der Nutzung des UI beim Nutzer einstellt. UI ist ein weitreichender Begriff, welcher allgemein als

Benutzerschnittstelle zwischen Mensch und Maschine beschrieben werden kann (Chlebek, 2006; Herczeg, 2018; Thesmann, 2016), da das UI weder zum Menschen noch zur Maschine gehört. Dabei ist die Schnittstelle nicht nur in digitaler, sondern auch in physischer Form im Alltag zu finden. So wird bei der Bedienung eines Fernsehers eine Fernbedienung verwendet. Die Fernbedienung gehört weder zum Menschen noch zur Maschine, sondern bildet lediglich die Schnittstelle, durch welche der Mensch mit der Maschine interagieren kann (Chlebek, 2011). In der IT gehört allerdings im Rahmen der computergestützten Schnittstellen ein sogenanntes *Graphical User Interface* – grafische Benutzeroberfläche (GUI) zur UI. Die meisten Computer, verwenden eine GUI³, die sich stark auf die Verwendung von grafischen Symbolen und Icons als visuelle Elemente des Bildschirms stützt. Diese Bildelemente dienen als Darstellung von Computerfunktionen auf niedriger Ebene, die von den Benutzern durch Aktionen auf höherer Ebene manipuliert werden können, ohne eine komplexe Syntax oder eine direkte Programmiersprache verwenden zu müssen (Shneidermann et al., 2017). Über die grafische Benutzeroberfläche können Benutzer komplexe Computerfunktionen durch die üblichen Aktionen wie Eingeben von Tasten, Bewegen des Mauszeigers oder durch Benutzung von berührungsempfindlichen Bildschirmen steuern (Jacko et al., 2003).

Die zuvor erwähnte Ästhetik einer Webapplikation wird somit durch eine ästhetische Gestaltung der UI realisiert (Schmidt & Wolff, 2017). In der Wissenschaft der Mensch-Maschine-Interaktion wird der Begriff Ästhetik als „*die subjektive Attraktivität und visuelle Schönheit eines User Interfaces aus Nutzersicht*“ verstanden (Tractinsky, 2004)⁴, die sogar zur Steigerung der Usability führt (Moshagen & Musch, 2009; Müller-Radtke, 2018) und die emotionale Lage des Nutzers verbessert. So kann allein durch die Ästhetik einer Software die Nutzung als „einfach“ eingestuft und vom Nutzer ein Gefühl von Vergnügen bei der Nutzung empfunden werden (Sauer & Sonderegger, 2009). Die Beurteilung der Ästhetik geschieht in zwei Stufen. Zum einen wird durch die intuitive Ästhetik die UI bereits nach 500ms Präsentationszeit vom Nutzer beurteilt. So kann der Nutzer sich einen ersten Eindruck von der UI machen. Dies ist wichtig, da der erste Eindruck ausschlaggebend dafür ist, ob der Nutzer die UI als ansprechend und schön empfindet und somit entscheidet, ob er diese Software bzw. Applikation in Zukunft weiterverwenden möchte. Die reflektive Ästhetik hingegen beschreibt die Beurteilung der Attraktivität und Kreativität durch den Nutzer nach beliebig langer Betrachtungszeit (Lavie & Tractinsky, 2003). Unter Betrachtung der obenerwähnten Einflussfaktoren der Ästhetik Symmetrie, Buntheit/ Farbvielfalt und die visuelle Komplexität zeigte die Studie von Schmidt und Wolff (Schmidt & Wolff, 2017), wie diese zu gestalten sind, um eine hohe Ästhetik zu erreichen. Zum einen wirkt sich die achsensymmetrische Anordnung der Elemente sowohl auf die intuitive als auch die reflektive Ästhetik positiv aus, wodurch auch die Studie von Tuch et al. (Tuch et al., 2010) bestätigt wurde. Zudem ist es

³ Im weiteren Verlauf der Arbeit wird zur Beschreibung der grafischen Schnittstelle bzw. der grafischen (Be-)Benutzeroberfläche lediglich der Begriff „UI“ bzw. „User-Interface“ genutzt.

⁴ übersetzt von und zitiert aus (Schmidt & Wolff, 2017)

empfehlenswert, dass der Designer der UI eine gemäßigte Anzahl an Farben verwendet und grelle und knallige Farben eher dezent einsetzt, falls es überhaupt nötig ist. Zur visuellen Komplexität lässt sich sagen, dass eine hohe visuelle Komplexität die Ästhetik negativ beeinflusst. Dabei interpretieren viele Nutzer die visuelle Komplexität als eine Vielzahl von Elementen auf einer Seite. Folglich kann die Ästhetik positiv beeinflusst werden, indem die Quantität an Elementen auf einer Seite geringgehalten wird (Schmidt & Wolff, 2017; Tuch et al., 2012; Reinecke & Gajos, 2014).

2.3 Softwaretests

Bevor das Produkt letztendlich den Endkonsumenten erreicht, werden in der Praxis sogenannte Softwaretests verwendet. Durch derartige Tests ist es möglich die Qualität der Software im Sinne der Erfüllung von definierten Anforderungen zu messen und zu bewerten (Baumgartner et al., 2009; Riedel, 2019; Spillner et al., 2017). Seit 2004 wird zu diesem Zwecke am häufigsten das allgemeine *V-Modell* (vgl. **Abbildung 2.1**) verwendet (Greching et al., 2010).

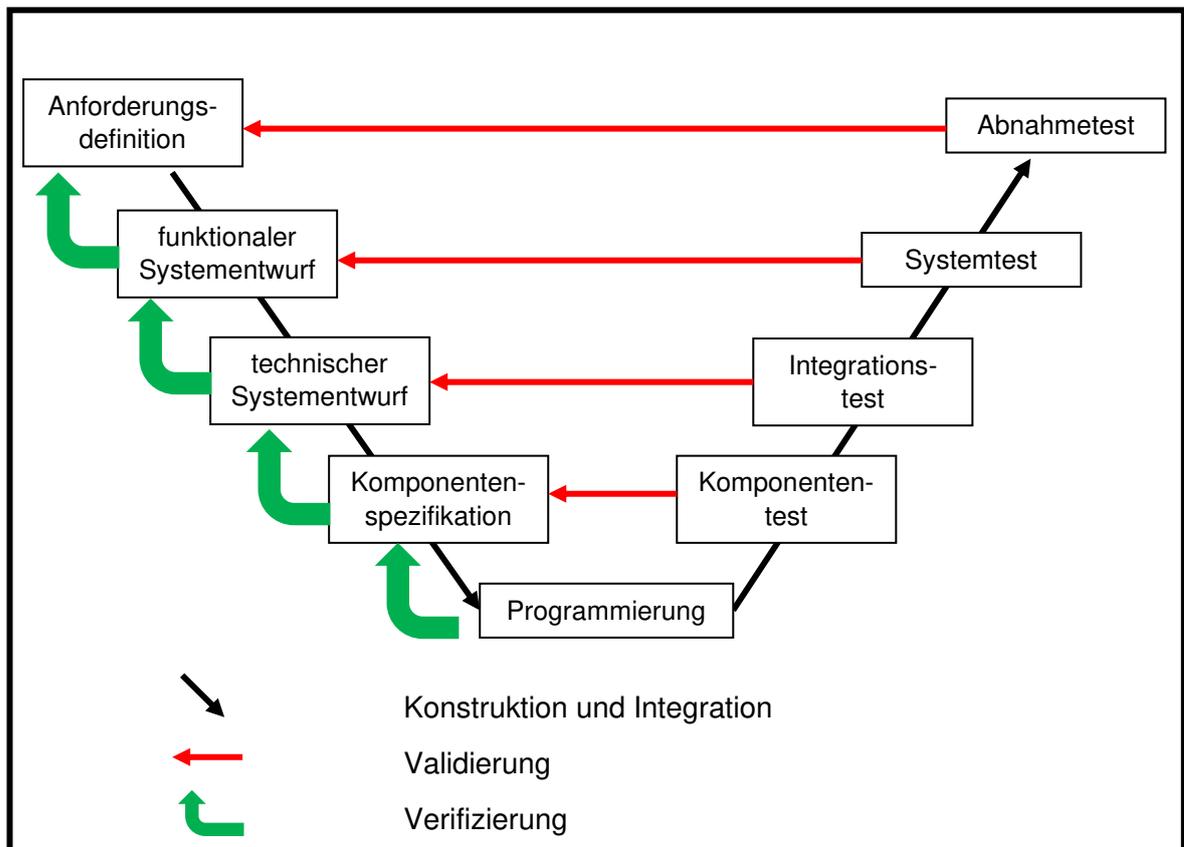


Abbildung 2.1: Allgemeines V-Modell (Spillner et al., 2017)

Abbildung 2.1 lässt sich entnehmen, dass nach dem allgemeinen V-Modell die vier Teststufen Komponententest, Integrationstest, Systemtest und Abnahmetest in dieser Reihenfolge zu durchlaufen sind.

Im Folgenden werden zunächst die sieben Grundsätze des Softwaretestens beschrieben. Anschließend werden die vier Teststufen, die dem V-Modell unterliegen, und zusätzlich ein sogenannter *Usability-Test* näher erläutert. Auch wenn der Usability-Test nicht zum V-Modell gehört, wird er an dieser Stelle herangezogen, da dieser es ermöglicht die Gebrauchstauglichkeit eines Systems zu prüfen (Brau & Sarodnick, 2016).

2.3.1 Die sieben Grundsätze des Softwaretestens

Laut dem International Software-Testing Qualification Board⁵ (ISTQB) haben sich in den letzten 40 Jahren sieben Grundsätze des Testens herauskristallisiert, die als Test-Leitfaden betrachtet werden können ((Spillner & Linz, 2019; Kribernegg, 2013).

1. Grundsatz Testen zeigt die Anwesenheit von Fehlern

Das Testen ermöglicht den Nachweis des Vorhandenseins von Fehlfunktionen. Die Wahrscheinlichkeit, dass noch unentdeckte Fehlerzustände im Objekt vorhanden sind, wird durch eine ausreichende Anzahl an Tests minimiert. Jedoch ist es wichtig zu wissen, dass mit Tests nachgewiesen werden kann, dass Fehler existieren, aber nicht, dass sie nicht existieren. Selbst wenn der Test keine Fehlfunktion aufdeckt, kann dies nicht als Nachweis für Fehlerfreiheit gesehen werden.

2. Grundsatz Vollständiges Testen ist unmöglich

Ein vollständiger Test ist nur bei sehr trivialen Testobjekten möglich. Ein Test, welcher ausnahmslos alle möglichen Eingabewerte und deren Kombinationen unter Berücksichtigung aller unterschiedlichen Vorbedingungen erfasst, ist nicht durchführbar. Da Tests lediglich Stichprobenmengen umfassen, wird der Testaufwand gegenüber den Prioritäten und Risiken abgewogen.

3. Grundsatz Mit dem Testen frühzeitig beginnen

Das Testen sollte nicht nach Fertigstellung des Programms erfolgen, sondern im System- und Softwarelebenszyklus so früh wie möglich beginnen. Das frühzeitige Beginnen der Tests sorgt dafür, dass Fehler rechtzeitig erkannt und eliminiert werden können.

4. Grundsatz Häufung von Fehlern

Wenn in einem Teil eines Testobjekt Fehler nachgewiesen werden können, dann kann davon ausgegangen werden, dass sich dort weitere Fehler befinden. Häufig liegen die meisten Fehler in nur wenigen Teilen eines Testobjektes. Eine Gleichverteilung der Fehlerzustände ist nicht gegeben.

⁵ Eine sich aus ehrenamtlich zusammenarbeitenden Expertengremien zusammengesetzte Zertifizierungsstelle für Softwaretester. Dabei ist die Erarbeitung einer standardisierten Ausbildung für professionelle Softwaretester das Ziel (ISTQB, 2020).

5. Grundsatz Zunehmende Testresistenz (pesticide paradox)

Das pesticide paradox ist auf die Restistenz von Schädlingen gegen Pflanzenschutzmittel zurückzuführen. So wie Pflanzenschutzmittel lassen auch Tests in ihrer Wirksamkeit nach. Wenn Tests nur wiederholt werden, dann werden keine neuen Fehlfunktionen mehr aufgedeckt. Es kann sein, dass sich weitere Fehlerzustände in Teilen des Programms befinden, welche nicht von diesem Test erfasst werden. Zudem berücksichtigen Tests nicht alle Konstellationen bei der Eingabe, weshalb diese Fehler ebenfalls unentdeckt bleiben können. Daher ist es empfehlenswert die bereits genutzten Tests durch neue und oder modifizierte Tests zu ergänzen, sodass immer wieder neue Fehler aufgedeckt werden können.

6. Grundsatz Testen ist abhängig vom Umfeld

Die Art und Weise wie und welche Tests durchzuführen sind, hängt vom Einsatzgebiet und Umfeld des zu prüfenden Systems ab. Zu prüfende Systeme unterscheiden sich vor allem in der Intensität des Testens und auch in weiteren Punkten wie der Definition der Ausgangskriterien, weshalb der Test stets individuell angepasst werden muss. So müssen bspw. Systeme der Sicherheitstechnik anders getestet werden als solche, die für Online-Versandhäuser entwickelt wurden.

7. Grundsatz Trugschluss: Keine Fehler bedeutet ein brauchbares System

Das Auffinden und Eliminieren von Fehlern sind keine Garantie dafür, dass das System die Ansprüche der zukünftigen Nutzer erfüllt. Daher empfiehlt es sich die späteren Nutzer so früh wie möglich in Entwicklungsprozess einzubeziehen. So können durch die Verwendung von Prototypen frühzeitige Anpassungen durchgeführt und spätere Probleme vorgebeugt werden.

2.3.2 Komponententest und Integrationstest

Der Komponententest (in weiteren Literaturen auch bekannt als „Einzeltest“ (Witte, 2016), „Modultest“ (Witte, 2016; Spillner & Linz, 2019; Riedel, 2019), „Unit-Test“ (Witte, 2016; Spillner & Linz, 2019) oder auch „Klassentest“⁶ (Spillner & Linz, 2019) prüft jeden einzelnen Baustein des Systems auf korrekte Funktionalität, also dem Ein- und Ausgabeverhalten des Testobjekts. Um diese Korrektheit zu prüfen werden die Komponenten mehreren Testfällen unterzogen, wodurch mit jedem Testfall eine weitere Ein- und Ausgabekombination auf Plausibilität untersucht wird. Durch die geringe Komplexität der Algorithmen auf Komponentenebene können die Komponenten durch wenige Testfälle umfassend getestet werden (Witte, 2016). Jedoch muss nicht nur die Funktionalität, sondern auch die Robustheit einer Komponente getestet werden. Nichtberücksichtigen von ungültigen Eingaben, bzw. der Verwendung einer Komponente

⁶ Im weiteren Verlauf dieser Arbeit wird lediglich der Begriff „Komponententest“ verwendet.

entgegen ihrer Spezifikation kann dazu führen, dass das System abstürzt. So wird durch den Komponententest mit nicht zulässigen Testeingaben – auch Negativtest genannt – getestet wie gut die Komponente auf solche Fehlersituationen reagiert (Spillner & Linz, 2019).

Die einzelnen Komponenten werden jeweils isoliert von anderen Bausteinen des Systems getestet, sodass komponentenexterne Einflüsse ausgeschlossen werden können. Dies ist wichtig, da lediglich komponenteninterne Aspekte geprüft werden sollen und nicht die Wechselwirkung zwischen mehreren Komponenten (Spillner & Linz, 2019).

Die Durchführung des Komponententests erfordert sogenannte Testtreiber (test driver), also ein Programm, welches das Testobjekt mit nötigen Testdaten versorgt und schließlich die Reaktion des Testobjekts entgegennimmt (Riedel, 2019). Ein Testtreiber ermöglicht es eine Testkomponente über seine Schnittstelle anzusprechen bzw. aufzurufen, wenn es nicht durch eine weitere Komponente aufgerufen werden kann. Daher muss dieser stets auf die verwendete Programmiersprache angepasst werden, weshalb der Komponententest in der Regel vom Entwickler selbst durchgeführt wird (Spillner & Linz, 2019).

Ein großer Vorteil des Komponententest ist zum einen, dass dieser früh in den Entwicklungsprozess eingepflegt werden kann (Witte, 2016) und zum anderen kann die Ursache eines erfolgreich aufgedeckten Fehlers eindeutig der getesteten Komponente zugeordnet werden (Spillner & Linz, 2019).

Integrationstest

Nachdem die Komponenten einzeln geprüft wurden, können nun abhängige Komponenten schrittweise zu größeren Komponenten zusammengesetzt und schließlich deren Zusammenspiel miteinander geprüft werden (Liebig et al., 2014). So werden Komponenten zu Teilsystemen und die Teilsysteme abschließend zum Gesamtsystem zusammengefügt - dies wird als Integration bezeichnet (Riedel, 2019). Dabei wird das Ziel verfolgt, Fehlerzustände in Schnittstellen und im Zusammenspiel zwischen den integrierten Komponenten zu finden und zu beseitigen (Spillner & Linz, 2019). Zur Durchführung des Integrationstests werden verschiedene Testszenarien erstellt, die die Zusammenwirkungen der getesteten Komponenten abbilden. Die Anzahl an Testszenarien und Integrationsstufen ist abhängig von der Komplexität des Systems (Witte, 2016). Der Integrationstest kann mit verschiedenen Integrationsstrategien durchgeführt werden:

TOP-DOWN

Der Top-Down-Methode (sowie der Bottom-Up-, Inside-Out- und Outside-In-Methode, werden im weiteren Verlauf näher erläutert) liegt eine n -stufige hierarchische Gliederung der einzelnen Komponenten zugrunde mit jeweils k Komponenten pro Stufe (vgl. **Abbildung 2.2**).

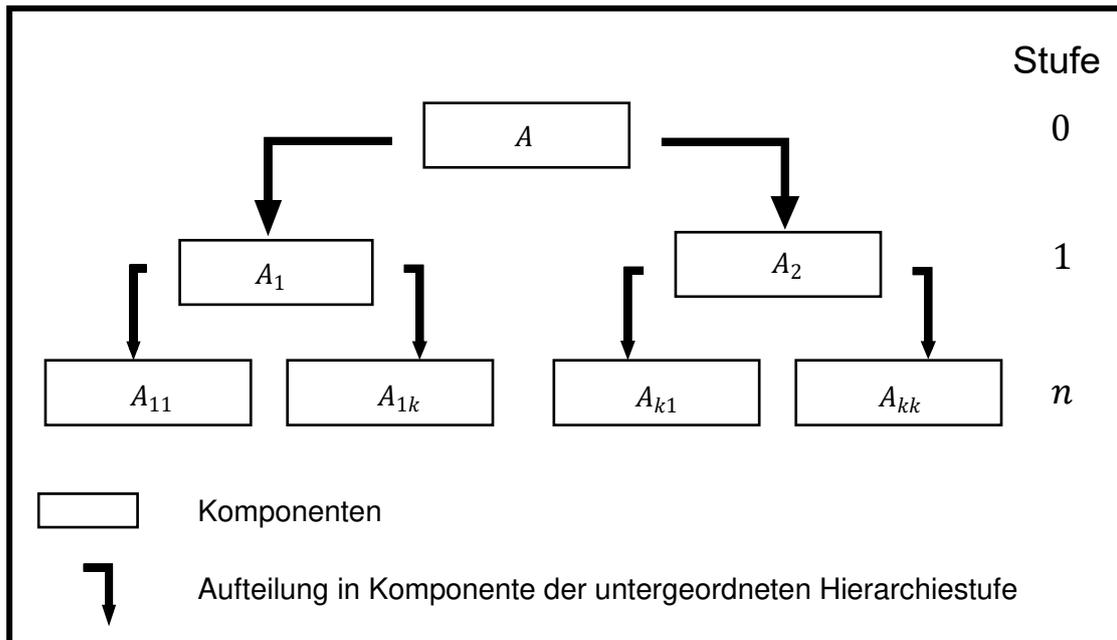


Abbildung 2.2: Schematische n -stufige hierarchische Gliederung

Die Top-Down-Integration beginnt mit der hierarchisch ranghöchsten Komponente (Riedel, 2019). Diese ruft andere Komponenten auf, wird aber selbst nicht (lediglich durch das Betriebssystem) aufgerufen. Sukzessiv werden die Komponenten der untergeordneten Hierarchiestufen hinzutintegriert. Die Stufe m dient dabei jeweils als Testtreiber der Stufe $m + 1$ (Spillner & Linz, 2019). Schematisch kann die Top-Down-Methode anhand von **Abbildung 2.3** verdeutlicht werden.

Die Komponenten einer untergeordneten Hierarchiestufe werden durch Platzhalter⁷ ersetzt, falls diese noch nicht vollständig implementiert sind und folglich nicht einsatzbereit sind. Sie simulieren das Ein- und Ausgabeverhalten des eigentlich abzurufenden Programmteils (Spillner & Linz, 2019). Es existieren drei verschiedene Formen von Platzhaltern: Zum einen ein sogenannter *stub*, welcher lediglich eine rudimentäre Funktionalität bietet. Zum anderen werden häufiger *Dummies* verwendet, da diese einen nahezu vollwertigen Ersatz für die echte Implementierung darstellen. Darüber hinaus existiert ein sogenannter *Mock*, welcher sich vom Dummy insofern unterscheidet, als dass dieser zu den Funktionalitäten eines Dummies noch zusätzliche Funktionalitäten beinhaltet (Link & Fröhlich, 2002). Das Top-Down-Verfahren hat den Vorteil, dass keine bzw. lediglich einfache Testtreiber benötigt werden (Spillner & Linz, 2019). Des Weiteren werden Fehler in der obersten Hierarchieebene, welche letztendlich die Logik der Anwendung enthält, früh entdeckt (Witte, 2016). Nachteilig jedoch ist der Einsatz von Platzhaltern, welcher zu einem hohen Aufwand führen kann (Spillner & Linz, 2019).

⁷ Platzhalter wird zudem in weiteren Literaturen als „Teststumpf“ bezeichnet, vgl. dazu (Witte, 2016)

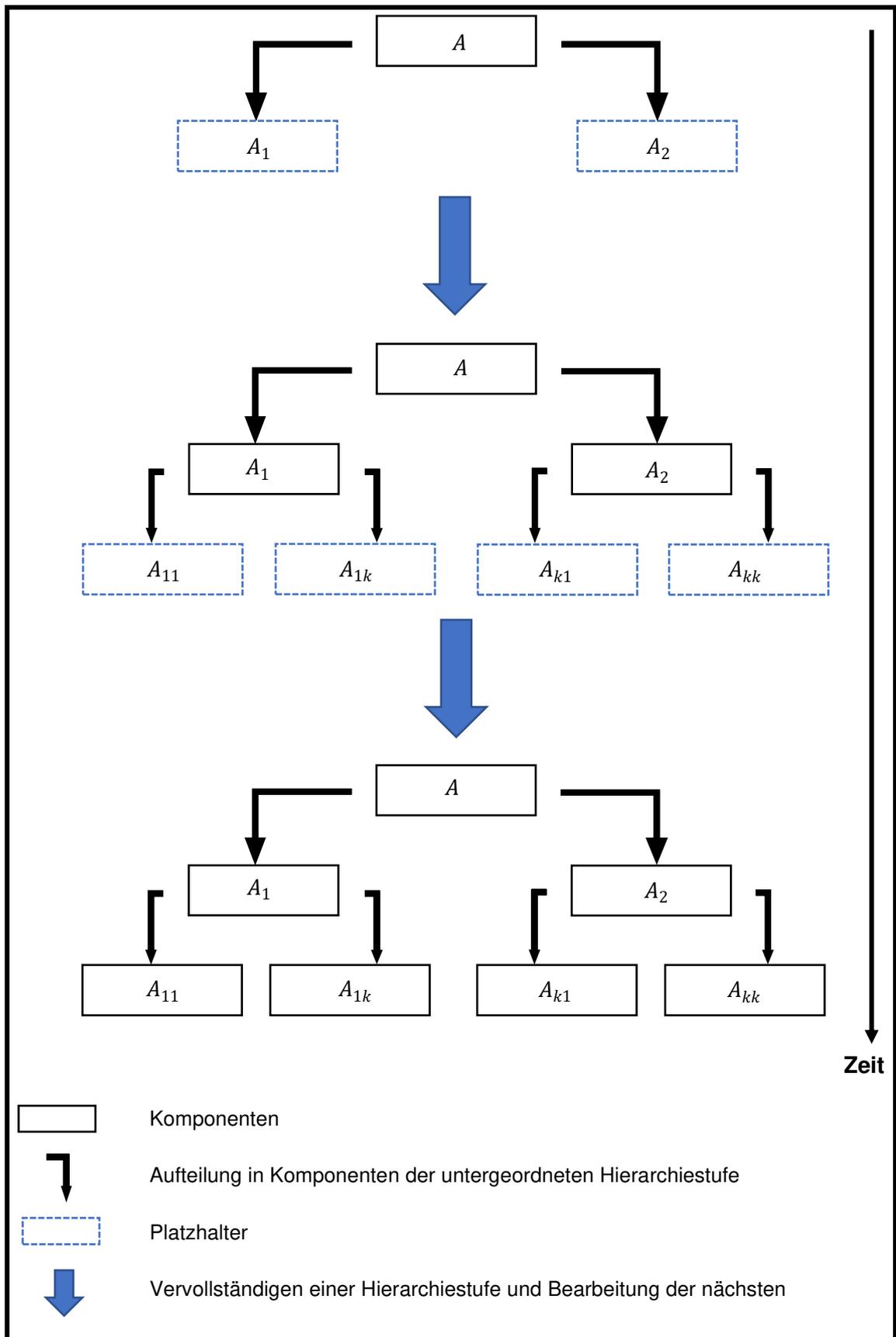


Abbildung 2.3: Schematischer Aufbau des Top-Down (Liggesmeyer, 2009)

BOTTOM-UP

Im Gegensatz zur Top-Down-Methode verläuft das Bottom-Up-Prinzip in die entgegengesetzte Richtung (vgl. **Abbildung 2.4**).

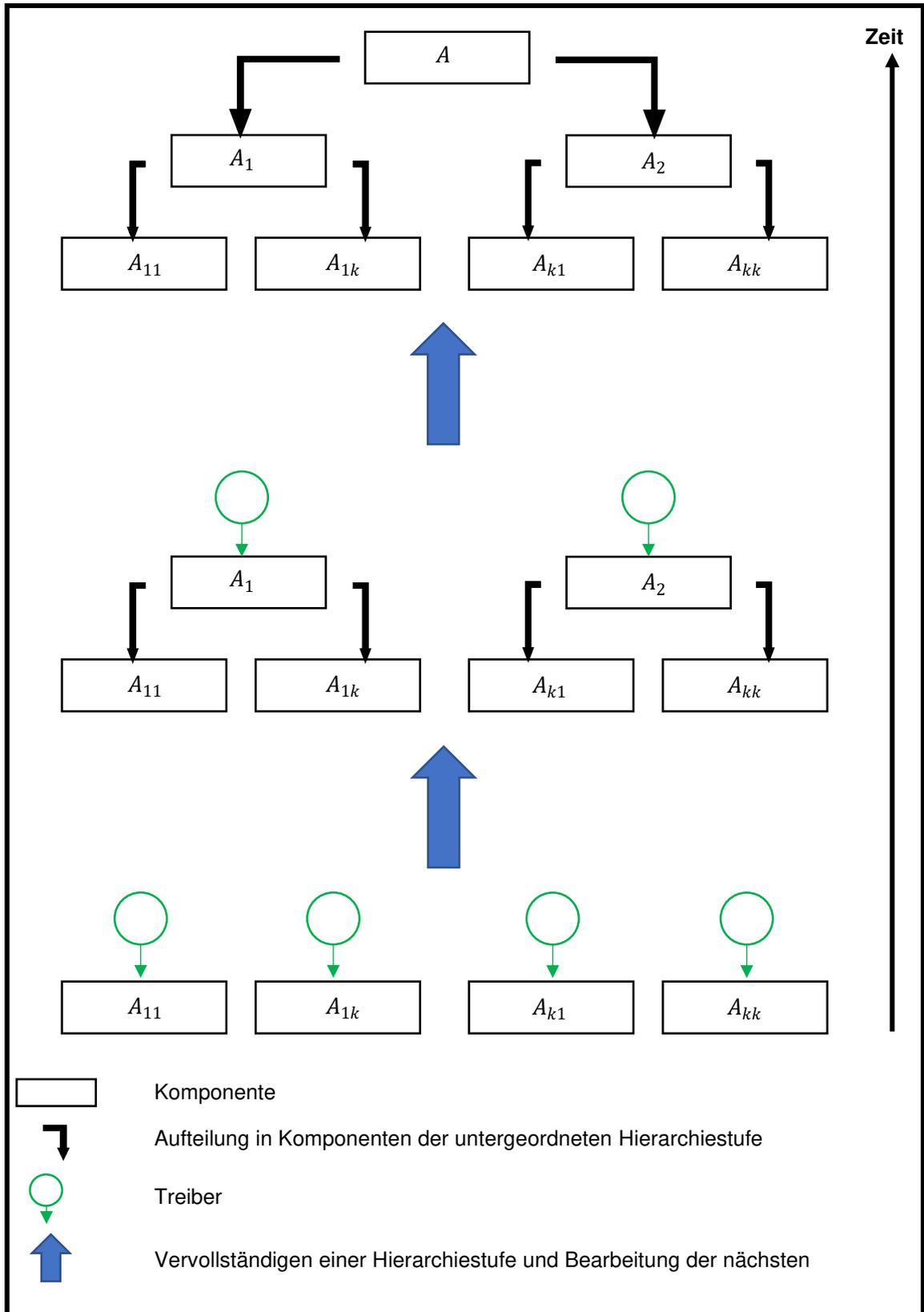


Abbildung 2.4: Schematischer Aufbau des Bottom-Up (Liggesmeyer, 2009)

Während der Ausgangspunkt beim Top-Down die höchste hierarchische Komponente ist, wird beim Bottom-Up mit der untersten Stufe begonnen (Riedel, 2019; Witte, 2016), also mit den Komponenten, die zwar aufgerufen werden, aber selbst keine weiteren Komponenten aufrufen. So werden sukzessiv größere Teilsysteme aus den getesteten Komponenten entwickelt, welche anschließend getestet werden. Vorteilhaft ist, dass keine aufwendigen Platzhalter benötigt, jedoch die übergeordneten Komponenten durch Testtreiber simuliert werden (Spillner & Linz, 2019). Das Bottom-Up-Verfahren ist in der Praxis die am häufigsten anzutreffende Integrationsart (Witte, 2016).

INSIDE-OUT und OUTSIDE-IN

Bei der Inside-Out- und der Outside-In-Methode⁸ werden jeweils die Strategien Top-Down und Bottom-Up miteinander kombiniert. **Abbildung 2.5** stellt das Outside-In-Verfahren schematisch dar. Für gewöhnlich wird die Outside-In-Methode von zwei unabhängigen Tester-Teams durchgeführt. Während das eine Team sich von der unteren Hierarchiestufe zur Mitte hinaufarbeitet, fängt das andere Team auf der obersten Hierarchiestufe an und arbeitet sich so lange runter, bis es ebenfalls die Mitte erreicht (Liggesmeyer, 2009). Während das Outside-In von *außen nach innen* arbeitet, wird beim Inside-Out *von innen nach außen* getestet, was in der Vorgehensweise das einzige Unterscheidungsmerkmal ist (Riedel, 2019). Es werden bei diesen Verfahren die Vorteile, aber auch die Nachteile von Top-Down und Bottom-Up kombiniert. Ein großer Nachteil dieses Verfahrens ist, dass sowohl Treiber als auch Platzhalter benötigt werden (Liggesmeyer, 2009).

⁸ In weiteren Literaturen auch als Schalenmodell und Zwiebelmodell (Riedel, 2019) oder auch als Sandwich-Strategie (Witte, 2016) bezeichnet

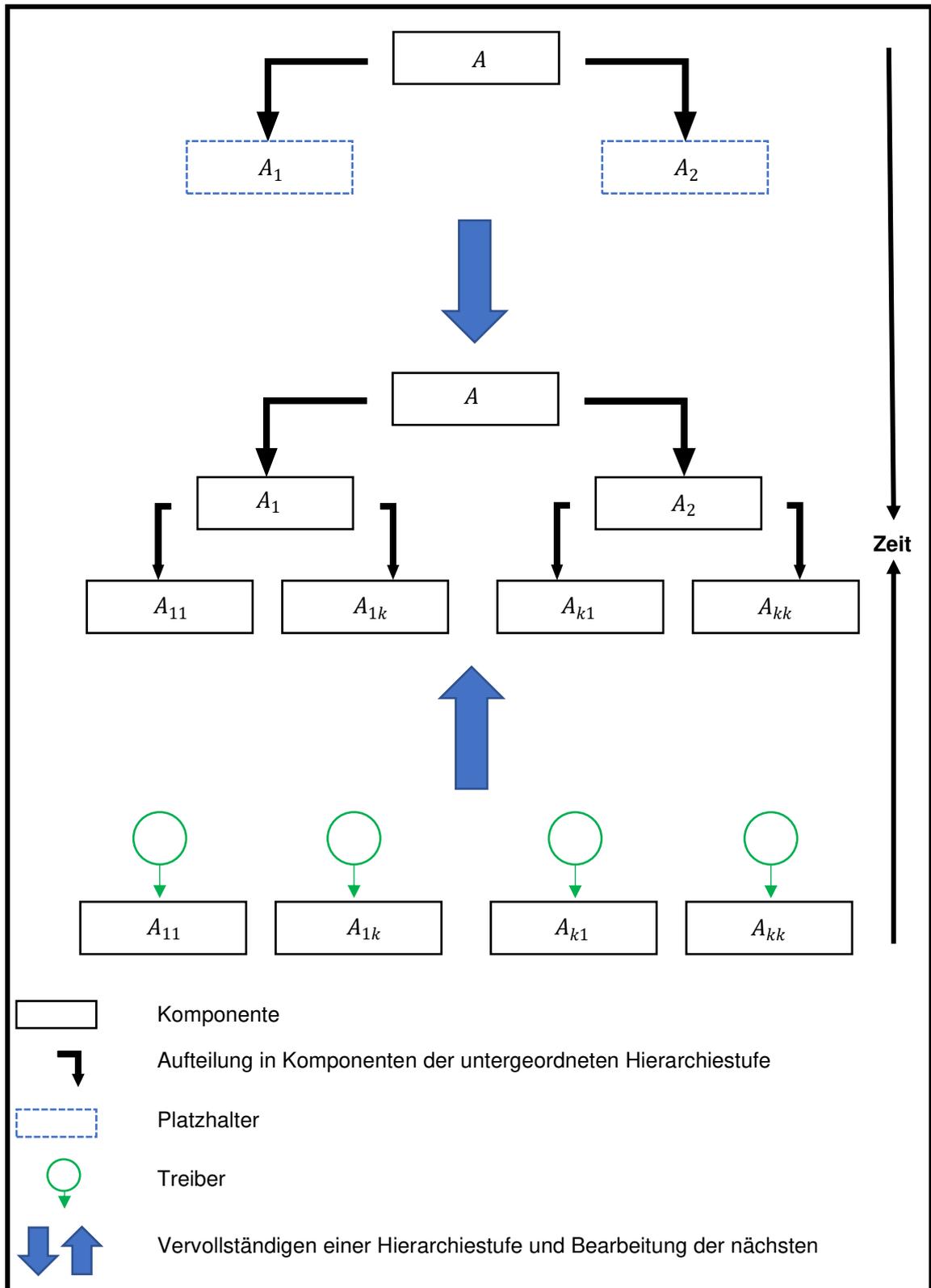


Abbildung 2.5: Schematischer Aufbau des Outside-In (Liggesmeyer, 2009)

BIG BANG

Die oben beschriebenen Methoden gehören zu den sogenannten inkrementellen Integrationstests. Eine nicht inkrementelle Methode ist das sogenannte Big Bang Verfahren (Liggesmeyer, 2009; Witte, 2016). Dies bedeutet, dass alle Komponenten auf

einmal zum Software-System zusammengefügt werden. Dazu muss gewartet werden, dass alle Softwarekomponenten bereits entwickelt wurden (Liggesmeyer, 2009; Spillner & Linz, 2019; Witte, 2016). Da bei inkrementellen Methoden Fehler frühzeitig erkannt werden können, zeigt der Big Bang einen wesentlichen Nachteil: Wenn mit dem Test gewartet wird bis alle Komponenten entwickelt werden, wird eine hohe Anzahl an Fehlern gleichzeitig erkannt. Zum einen ist die Lokalisierung und schließlich das Beseitigen dieser Fehler problematisch und zum anderen sind diese dementsprechend kosten- und zeitintensiv (Spillner & Linz, 2019). Allgemein wird aufgrund der oben beschriebenen Nachteile vom Big Bang Verfahren in der Praxis abgeraten (Spillner & Linz, 2019; Witte, 2016).

In der Realität werden Testverfahren oft nicht strikt durchgeführt, da bspw. das Top-Down oder das Bottom-Up eine klare hierarchische Gliederung voraussetzt. Da dies nicht immer möglich ist, wird der Integrationstest häufig als individuelle Mischung mehrerer Teststrategien durchgeführt (Spillner & Linz, 2019).

2.3.3 Systemtest und Abnahmetest

Nach erfolgtem Integrationstest folgt der sogenannte Systemtest. Dabei wird im Gegensatz zum Komponenten- bzw. Integrationstest nicht aus der Sicht eines Entwicklers, sondern aus der Kundenperspektive getestet. Das Ziel ist es zu überprüfen, ob und wie gut ein System die gestellten Anforderungen erfüllt. Zudem sollen fehlerhaft umgesetzte oder gar fehlende Anforderungen identifiziert und korrigiert werden (Spillner & Linz, 2019). Dabei ist zwischen funktionalen und nicht funktionalen Anforderungen zu unterscheiden. Die funktionalen Anforderungen legen fest, was das Produkt tun soll. Beim Systemtest für die funktionalen Anforderungen kommen folgende Testansätze zum Einsatz:

- Geschäftsprozessbasierter Test: die Basis zur Ableitung und Priorisierung der Testfälle wird durch einzelne Geschäftsvorfälle oder die Rangfolge der Prozesse gebildet
- Anwendungsfallbasierter Test: die häufigsten Anwendungen bzw. ihre Reihenfolge dienen als Testbasis
- Anforderungsbasierter Test: Die Testfälle werden aus den Anforderungen abgeleitet.

Mit diesen Tests werden folgende Anforderungen und Fragestellungen untersucht:

- Angemessenheit: Sind die verwendeten Funktionen für die vorgesehenen Anforderungen geeignet?
- Interoperabilität: Arbeitet das System in der Systemumgebung fehlerfrei?

- Ordnungsmäßigkeit: Werden Normen und gesetzliche Rahmenbedingungen eingehalten?
- Richtigkeit: Werden von den Funktionen die erwarteten Ergebnisse geliefert?
- Sicherheit: Sind die Daten und Anwendungen gegen unbefugtes Nutzen geschützt?

Das System wird dabei als Ganzes betrachtet und es wird mit Testdaten das Ineinandergreifen aller Systemkomponenten getestet. Dabei wird eine Testumgebung erzeugt, welche der Produktivumgebung des Kunden möglichst nahekommt. Tests in der realen Produktivumgebung sind allerdings zu vermeiden, da das Produktivsystem durch den Test beschädigt werden und somit Störungen, wie Produktionsausfälle, im betrieblichen Ablauf hervorgerufen werden können (Witte, 2016).

Gegenstand des Systemtests ist die Optimierung des Systems durch das Testen von nicht-funktionalen Anforderungen. Eine nicht-funktionale Anforderung ist in der Literatur nicht einheitlich definiert. Häufig wird der Begriff so verstanden, dass nicht-funktionale Anforderungen über funktionale Anforderungen hinausgehen. So geht es nicht darum was ein System tun soll, sondern viel mehr wie gut ein System arbeitet. Es wird daher als eine Randbedingung oder ein Qualitätsmerkmal, das das System erfüllen muss, verstanden (Eigner et al., 2012). Um die nicht-funktionalen Anforderungen zu testen, stehen unter anderem folgende Testverfahren zur Verfügung:

- Lasttest: Es wird beobachtet wie sich das System verhält, wenn das System durch Erhöhung der parallel arbeitenden Nutzer belastet wird (Shneidermann et al., 2017). Es wird immer von der maximalen Systemlast ausgegangen und der Extremwert in der Regel um 20% erhöht (Witte, 2016).
 - Stresstest: Das Systemverhalten wird bei Überbelastung beobachtet (Shneidermann et al., 2017)
 - Crashtest: Stresstests mit dem Versuch ein Absturz des Systems hervorzurufen (Witte, 2016)
- Wiederinbetriebnahmetest: Prüfung, ob ein abgeschaltetes oder abgestürztes System wieder in Betrieb genommen werden kann (Witte, 2016)
- Stabilitätstest: Test im Dauerbetrieb. Wie häufig fällt das System (pro Stunde) aus? (Shneidermann et al., 2017)
- Test auf Robustheit: Verhalten des Systems bei Fehlbedienung (Shneidermann et al., 2017)

Abnahmetest

In der letzten Teststufe – dem Abnahmetest (auch bekannt als: Verfahrenstest, Akzeptanztest, User Acceptance Test (Witte, 2016) oder auch nur Acceptance Test (Riedel, 2019) – werden im Gegensatz zum Systemtest das System unter realen Betriebsbedingungen mit realen Daten getestet. Diese Teststufe wird also nicht von den

Entwicklern, sondern von den Kunden durchgeführt. Dieser Test wird über einen längeren Zeitraum durchgeführt, sodass die Zuverlässigkeit des Systems durch den Kunden bzw. Anwender bewertet werden kann (Liebig et al., 2014). Der Abnahmetest verfolgt im Gegensatz zum Komponenten-, Integrations- und Systemtest nicht das Ziel Fehler zu identifizieren, da diese bei der Durchführung des Abnahmetests bereits in den vorigen drei Stufen behoben sein sollen. Vielmehr ist das Ziel, dass der Kunde sich mit dem System vertraut macht und das System das Vertrauen des Kunden gewinnt (Witte, 2016). Der Kunde beeinflusst den Abnahmetest nach seinen individuellen Wünschen insofern, als dass dieser eigene von ihm bestimmte Testfälle durchspielt, um so zumindest regelmäßig auftretende Fälle im Geschäftsprozess abzudecken. So kann der Kunde sich selbst von der Zuverlässigkeit des Systems überzeugen (Spillner & Linz, 2019). Im Sinne der Einhaltung von Zahlungsfristen, Wartungsvereinbarungen oder Garantiefristen ist der Abnahmetest unabdingbar, weshalb Abnahmekriterien bereits bei Vertragsschluss beider Parteien definiert werden sollten (Witte, 2016).

2.3.4 Usability- Test

In Abschnitt 2.2 wurde bereits die Usability bzw. Gebrauchstauglichkeit eingeleitet, welche maßgeblich dafür ist, dass ein System einem Nutzer die Erreichung seines Ziels erleichtert und somit dem System zu einer höheren Akzeptanz verhelfen. Um die Gebrauchstauglichkeit zu messen und zu bewerten wird in den meisten Fällen auf den sogenannten Usability-Test zurückgegriffen. Usability-Tests lassen sich in zwei Kategorien aufteilen: den induktiven und den deduktiven Tests. Die induktiven Tests dienen der formativen Evaluation, bei welcher Prototypen auf Schwachstellen und Verbesserungsmöglichkeiten hinsichtlich der Funktionalität und Gestaltung untersucht werden. Deduktive Tests dienen der summativen Evaluation. Dies bedeutet, dass mehrere Systeme miteinander verglichen werden, sodass die beste Alternative ausgewählt werden kann (Brau & Sarodnick, 2016).

Wie aus dem vorigen Abschnitt hervorgeht, ist ein System erst gebrauchstauglich, wenn der Kunde bzw. Endkonsument dieses als für ihn tauglich und hilfreich ansieht, vor allem für Webanwendungen. Dazu werden neben den im Systemtest beschriebenen Tests zusätzliche Tests benötigt, welche durch verschiedene Probanden zeigt, inwieweit die Webanwendung Funktionen für den Anwender verständlich macht, wie die Informationen angeordnet sind oder auch wie sich der Nutzer bei Bedienung der Webanwendung fühlt (Witte, 2016). Zu diesem Zwecke empfiehlt Faulkner den Einsatz von 20 oder mehr Testern, da auf diese Weise mindestens 98,4 Prozent der Usability-Probleme aufgedeckt werden können. Je mehr Testpersonen teilnehmen, desto mehr Fehler können aufgedeckt werden (Faulkner, 2003). Jedoch ist es häufig schwierig so viele Testpersonen zu finden, weshalb Nielsen die Effektivität des Usability-Tests mit lediglich fünf bis sechs Personen untersucht hat. Das Ergebnis war, dass im Durchschnitt ca. 80% der Usability-Probleme aufgedeckt werden konnten (Nielsen, 1993). Laut Sarodnick und Brau sollen die Testpersonen die Endnutzer angemessen repräsentieren.

Allerdings soll das zu testende System auch für die Testpersonen unbekannt sein, da sonst viele Probleme durch das Vorwissen über das System umgangen werden können (Brau & Sarodnick, 2016). Zur Unterstützung der Auswertung der Eingaben des Probanden können Videoaufzeichnungen, lautes Denken, vom Probanden auszufüllende Fragebögen oder auch komplexere Hilfsmittel wie Eyetracking-Software herangezogen werden. Die Videoaufzeichnungen, die Methode des lauten Denkens und das Eyetracking helfen dem Testleiter synchron oder asynchron nachzuvollziehen, warum die Testperson wann welche Aktion durchgeführt hat (Lennard & Surkau, 2011). Durch die Fragebögen kann die Testperson seine Meinung sowohl zum User-Interface (Anordnung der Symbole, Fülle an Symbolen, Farbgebung, Symmetrie, Komplexität) als auch zu seiner Empfindung bezüglich der gemachten Eingaben und erhaltenen Ausgaben kundtun (Schmidt & Wolff, 2017).

3 Didaktische Theorien und Modelle in der Informatik

Die Wurzeln des Begriffs *Didaktik* liegen etymologisch betrachtet in der griechischen Antike (ca. 600 - 200 v. Chr.). In der altgriechischen Sprache wurden die Begriffe „didáskein“ (zu Deutsch: unterrichten oder lehren), „didáskalos“ (der Lehrer, welcher zumeist ein männlicher Sklave gewesen ist) und „didaksaleíon“ (die Schule bzw. der zum Lernen und Lehren genutzte Raum) verwendet. Daraus lässt sich ableiten, dass die Didaktik als „die Praxis des Lehrens und Lernens“ oder als „Wissenschaft von der Lehre und die Kunst des Unterrichts“ verstanden werden kann. (Lehner, 2019)

Darauf aufbauend stellten Jank und Meyer folgende Definitionen des Begriffs *didaktisches Modell* auf:

1. „Ein didaktisches Modell ist ein erziehungswissenschaftliches Theoriegebäude zur Analyse und Planung didaktischen Handelns in schulischen und nichtschulischen Handlungszusammenhängen.“
2. Ein didaktisches Modell stellt den Anspruch, theoretisch umfassend und praktisch folgenreich die Voraussetzungen, Möglichkeiten und Grenzen des Lehrens und Lernens aufzuklären.
3. Ein didaktisches Modell wird in seinem Theoriekern in der Regel einer wissenschaftstheoretischen Position (manchmal auch mehreren) zugeordnet.“ (Jank & Meyer, 2019)

Da im Rahmen dieser Arbeit das Augenmerk auf die Online-Lehre bzw. dem asynchronen Online-Lernen der Abfragesprache SQL gelegt wird, ist es zunächst wichtig zu wissen, dass bei der Konzeption entsprechender Lernmedien das Irrtum weitverbreitet ist, wonach es ausreicht den Lernenden bestimmte Inhalte lediglich zugänglich zu machen. Jedoch reicht dies allein nicht aus, da für größere Lernerfolge eine systematische Aufbereitung unumgänglich ist (Kerres et al., 2009). Daher werden in den folgenden Abschnitten mehrere didaktische, insbesondere in der Informatik genutzte, Modelle vorgestellt. Anhand dieser wird eine didaktische Grundlage für die Entwicklung der in zu entwickelnden Webapplikation enthaltenen Übungsaufgaben geschaffen.

3.1 Motivation in der Psychologie

Ohne den Willen zu lernen, ist jeder Unterrichtsaufwand nutzlos. Deshalb ist eine angemessene Motivation erforderlich, um bei Lernenden einen Lernwillen zu erzeugen. Somit stellt sie das Hauptziel der Lehrtätigkeit dar. Zunächst müssen die Begriffe Motivation, Motivierung und Motiv voneinander abgegrenzt werden. In der Psychologie wird *Motivation* als ein kurzer Zustand des inneren Antriebs verstanden, während *Motivierung* die aktive Anstrengung ist, einen solchen Zustand der Motivation durch

endogene oder exogene Einflüsse zu schaffen. Die Initiations- und langfristigen Orientierungsfaktoren wie der Wunsch zu Lernen, die Freude an der Schöpfung, der Erfolg, der Ehrgeiz usw. werden *Motive* genannt. (Hubweiser, 2007) Dieser Zusammenhang kann schematisch in **Abbildung 3.1** dargestellt werden.

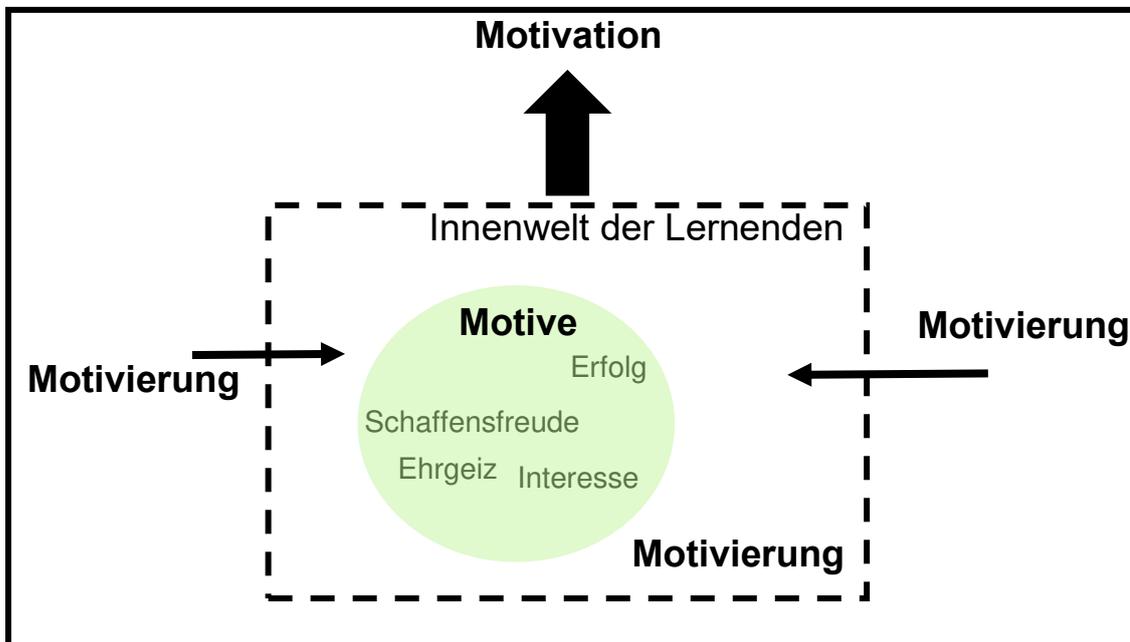


Abbildung 3.1: Zusammenhang zwischen Motivation, Motivierung und Motiv (Hubweiser, 2007)

Laut dem Modell des psychologischen Systems des Person-Umwelt-Bezug-Postulats befindet sich die Motivation in direkter Wechselwirkung mit der emotionalen, kognitiven und physiologischen Dimension (Hommel, 2019). Wiederum bilden diese den Kern der kognitivistischen und konstruktivistischen Lerntheorie (vgl. 3.2). In Bezug auf die Lernaktion muss jedoch zwischen Lernmotivation und Leistungsmotivation unterschieden werden (Krapp & Hascher, 2014). Lernmotivation steht für die Bereitschaft der Lernenden, Wissen zu erwerben und ihre Fähigkeiten zu verbessern, indem sie sich absichtlich, nachhaltig und zielorientiert auf konkrete Lerninhalte einlassen (Krapp & Hascher, 2014; Boekarts, 2015). Die Lernmotivation stellt somit die internen Motivationsquellen dar (Schlag, 2013). Es existieren zwei Situationskontexte, in denen die Lernmotivation bedeutend ist: vor dem Lernprozess und während des Lernprozesses (die Motivation möglichst lange aufrechterhalten) (Hommel, 2019). Die Leistungsmotivation beschreibt die Lernleistung als ausschlaggebende Beweggründe den Lernprozess durchzuführen, sodass der Lernende am Ende des Lernprozesses bspw. eine gute Note oder Lob erwartet. Dies gilt für den Lernenden als Vergleichsmaßstab und bestimmt die zu erfüllende Erwartung, die an der Lernenden gestellt wird (Krapp & Hascher, 2014). Somit stellt die Leistungsmotivation die externen Motivationsquellen dar (Schlag, 2013). Die Motivation und der Lernerfolg stehen in konstanter gegenseitiger Wechselwirkung zueinander. Zudem ist die Motivation sowohl Voraussetzung als auch Ergebnis von Lernprozessen. Folglich kann sich jede einzelne

Aufgabe positiv oder negativ auf die Motivation auswirken. Aus diesem Grund ist es für Lehrpersonen wichtig vor allem in Lehrveranstaltungen keine zu hohen Erwartungen zu haben. Stattdessen empfiehlt es sich die Erwartungen mit fortschreitender Zeit an die Lernentwicklung der Lernenden anzupassen und diesen zu Beginn des Lernprozesses ein positives Gefühl gegenüber den Lerninhalten zu geben (Hubweiser, 2007). Denn eine dauerhafte Beschäftigung mit dem Lerngegenstand ist nur möglich, wenn die Lernsituation insgesamt als positiv wahrgenommen wird (Hubweiser, 2007; Schnirch et al., 2020).

3.2 Kognitivistisches und konstruktivistisches Lernen

Vielen didaktischen Aspekten liegt die Kombination zweier Lerntheorien zugrunde: dem kognitivistischen Lernen und dem konstruktivistischen Lernen (Weickler, 2007). Dabei spielen die kognitiven, pragmatischen und affektiven Lernziele⁹ eine große Rolle. Die kognitive Dimension umfasst dabei das Ziel, dass der Lernende etwas weiß, was er vorher nicht wusste, wohingegen in der pragmatischen Dimension das Ziel verfolgt wird, dass der Lernende etwas kann, was er vorher nicht konnte. Zudem thematisiert das affektive oder auch emotionale Lernziel die Änderung der Haltung der Lernenden gegenüber neuen Problemen, welche jenseits der Lehrveranstaltung vorkommen (Velica, 2010).

Kognitivistisches Lernen

Im kognitivistischen Lernmodell werden Modelle der neuronalen Wissensverarbeitung aufgegriffen. Die Theorie des Kognitivismus ist mit den aktuellen Erkenntnissen der Erziehungswissenschaft und der Didaktik vereinbar, sodass diese Theorie eine gute didaktische Grundlage bietet. (Modrow & Strecker, 2016). Der Kognitivismus versteht das Lernen als einen strengen regelmäßig Prozess des Wissenserwerbs. Dieser wird stark abhängig von der individuellen Verarbeitungs- und Aufnahmefähigkeit von Informationen beeinflusst. Externe Einflüsse können zum Lernen beitragen, aber nicht vollständig gesteuert werden (Faulstich, 2013). Die Lerninhalte werden als fertiges System von den Lehrkräften vermittelt. Die Lernenden sind bei dieser Vermittlung passiv. Beim kognitivistischen Ansatz steht der Wissensinput durch den Lehrenden im Vordergrund. Das richtige Lernen findet in der Regel nicht in den Lehrveranstaltungen, sondern im Selbststudium statt. Der kognitive Input bestimmter Inhalte und die praktische Anwendung dieser liegen sehr eng beieinander. Dadurch werden zum einen die Inhalte besser und langfristiger gelernt und zum anderen können auftauchende Probleme und Unklarheiten früher erkannt und beseitigt werden (Hoblitz, 2015).

Für den IT-Unterricht ist eine strukturierte Wissensvermittlung in Form von Frontalunterricht in Vorlesungen im üblichen Massenbetrieb der ersten Semester

⁹ Neben dem Begriff „pragmatische Lernziele“ wird auch der Begriff „psychomotorische Lernziele“ verwendet (Meyer & Stocker, 2018).

unabdingbar. Um diese Kompetenzen in kürzerer Zeit einstellen zu können, müssen die Prinzipien des Kognitivismus um weitere didaktische Elemente erweitert werden. Der Kognitivismus fördert nicht explizit die Kompetenzen auf affektiver und pragmatischer Ebene. Auf diesen Ebenen wird in Rahmen der konstruktivistischen Lerntheorie eingegangen. (Weickler, 2007)

Konstruktivistisches Lernen

Im konstruktivistischen Ansatz wird Wissen als eine individuelle Konstruktion verstanden. Vor diesem Hintergrund ist das Lernen ein aktiver, konstruktiver Prozess in einem bestimmten Handlungskontext. Das Wissen wird durch die Lernenden in der Auseinandersetzung mit neuen Lerngegenständen und Lernwerkzeugen konstruiert (Vollmers, 1997). Dazu muss die Lernumgebung vor allem Situationen bieten, in denen vom Lernenden eigene Konstruktionsarbeit geleistet werden kann. Lehren wird in diesem Zusammenhang eher als Unterstützung, Anregung und Beratung verstanden. Der Lehrer nimmt eine reaktive Rolle hinter der aktiven Rolle des Lernenden ein (Weickler, 2007). In diesem theoretischen Lernmodell findet das Lernen während der Veranstaltungen statt. Dies ermöglicht es den Lernenden, neues Wissen direkt in ihrem individuellen Vorwissen zu verankern. Denn bereits die Gehirnforschung bestätigt, dass nur unter Einbeziehung des individuellen Vorwissens ein sinnvolles Lernen gewährleistet werden kann (Vester, 2018). Praxis und Wissensaufbau werden durch die eigene Prüfung miteinander verbunden. Darüber hinaus hat jeder Lernende die Möglichkeit, seinen eigenen Lernweg, angepasst an den eigenen Lerntyp, zu verfolgen. Die Lerngeschwindigkeiten der Lernenden sind in der Regel sehr unterschiedlich und die inhaltlichen Ziele können nicht im Voraus abgeschätzt werden. Durch zusätzliche Phasen des Wissensinputs ist es möglich, dass die Lernenden zumindest auf einer Wissensbasis lernen können, anstatt sich alles selbst erarbeiten zu müssen. Im Kontext der Informatik ist der Konstruktivismus trotz seiner unbestreitbaren Nachteile von großer Bedeutung, da in konstruktivistischen Lernphasen pragmatische und affektive Lernziele angestrebt werden können, die im Kognitivismus der individuellen Auseinandersetzung des Einzelnen überlassen bleiben. (Weickler, 2007)

3.3 Bildungstheoretische und kritisch-konstruktive Didaktik nach Klafki

Die bildungstheoretische Didaktik wurde bereits 1958 von Wolfgang Klafki begründet (Klafki, 1958). Mit diesem Ansatz stellte Klafki die vom Lehrenden durchzuführende Auswahl der Unterrichtsinhalte in den Vordergrund. Dazu wird von den Lehrenden verlangt, dass sie sich in der Vorbereitung des Unterrichts fünf Leitfragen beantworten. 1980 wurden zwei weitere Fragen hinzugefügt, welche seit 40 Jahren aktiv in der

Didaktik genutzt werden.¹⁰ Die Fragen lassen sich zudem in vier Sinnesabschnitte unterteilen (Klafki, 1980):

Begründungszusammenhang:

1. Gegenwartsbedeutung:

Welche Bedeutung haben dieser Inhalt oder die zu diesem Thema zu erwerbenden Erfahrungen, Kenntnisse, Fähigkeiten oder Fertigkeiten bereits im Leben der Kinder in der Klasse? / Welche Bedeutung sollte der Inhalt aus pädagogischer Sicht haben?

2. Zukunftsbedeutung:

Wo lässt sich das an diesem Thema zu Gewinnende als Ganzes oder in einzelnen Elementen – Einsichten, Vorstellungen, Wertbegriffen, Arbeitsmethoden, Techniken – später als Moment fruchtbar machen?

3. Exemplarische Bedeutung:

Welche exemplarische Bedeutung hat das Thema? Welche allgemeinen Sachverhalte bzw. welche allgemeinen Probleme lassen sich erschließen?

Thematische Strukturierung

4. Thematische Struktur:

Wie ist das zu vermittelnde Thema strukturiert? In Bezug auf das E-Learning ist es wichtig zunächst festzulegen welche Lehrform für die thematische Strukturierung verwendet wird (PELe, 2006).

5. Erweisbarkeit und Überprüfbarkeit:

Anhand welcher erworbenen Fähigkeiten, Kenntnissen, Handlungen kann gezeigt werden, dass der Lernprozess erfolgreich war?

Bestimmung von Zugangs- und Darstellungsmöglichkeiten

6. Zugänglichkeit bzw. Darstellbarkeit:

Ist den Lehrenden Personen der Zugang zu den Lerninhalten (durch Medien) möglich? Welche Schwierigkeiten können entstehen, die den Zugang erschweren?

Methodische Strukturierung

7. Lehr-Lern-Prozessstruktur:

Welche Lehr-Lern-Prozessstruktur kann aus den vorangegangenen Fragen abgeleitet werden? Wie sieht die Abfolge der Unterrichts- und Interaktionsformen aus? Die Lehr-Lernprozessstruktur wird verstanden als variables Konzept notwendiger oder möglicher Organisations- und Vollzugsformen des Lernens einschließlich entsprechender Lehrhilfen.

¹⁰ So werden Klafkis Theorien noch heute u.a. an der Uni Trier (Hoffmann, 2009), TU Kaiserslautern (ZFL, 2020) oder der Fachhochschule in Wien (Lehner, 2019) gelehrt und zudem u.a. von (Jank & Meyer, 2019) und (Hubweiser, 2007) zitiert.

3.4 Didaktik der Informatik nach Baumann

Der gesamte Abschnitt bezieht sich auf das Werk von Rüdiger Baumann (Baumann, 1996), dessen Theorien noch in heutigen Werken oft zitiert und zudem an verschiedensten Universitäten gelehrt wird.¹¹

Baumann beschreibt in seinem Werk insgesamt zwölf didaktische Prinzipien, welche auf die Informatiklehre angewandt werden können. Die Prinzipien könne allerdings nicht vollständig unabhängig voneinander betrachtet. Diese werden im Folgenden näher erläutert.

Prinzip des aktiven Lernens

Durch das Prinzip des aktiven Lernens wird der zu lernende Inhalt durch aktives Tun vermittelt. Lernende sollen sich selbständig mit dem zu erlernenden Stoff auseinandersetzen, indem sie diesen entdecken und auch anwenden. Dadurch ergibt sich der Vorteil, dass sich der Lernende aus eigenem Antrieb heraus mit den Lerninhalten befassen und ein höheres Interesse für die Inhalte hervorgerufen wird. Dafür muss sich der Lernende allerdings in einer Lernumgebung befinden, in welcher er sich wohlfühlt.

Integrationsprinzip

Das Integrationsprinzip verfolgt das Ziel Beziehungsnetze und Sinnzusammenhänge zu bereits erlerntem Wissen auszubauen. So kann der Lernende umso erfolgreicher mit seiner Umwelt in Wechselwirkung treten je vollständiger seine gewonnenen Erkenntnisse in das Beziehungsnetz eingebaut sind.

Prinzip der Veranschaulichung

Neue Inhalte sollen über Lernsituationen vermittelt werden, bei denen nur einzelne Elemente oder Aspekte neu sind, die aber möglichst viele Ansatzpunkte für eine Anwendung bekannter kognitiver Schemata bieten. So soll die Vermittlung in einer für die Lernenden bereits bekannten Situation erfolgen.

Prinzip der Stabilisierung

Neu gelernte Inhalte sollen in regelmäßigen Abständen wiederholt und in neue Inhalte aufgenommen werden. Dadurch wird verhindert, dass der Inhalt vorzeitig vergessen wird, und hilft, den Lerninhalt zu verinnerlichen. Zum Zwecke der Stabilisierung bauen neue Inhalte auf Inhalten aus der Vergangenheit auf. Daher ist das Lehren, in dem die Inhalte in unzusammenhängende Gebiete zerfallen und nicht wiederholt werden, nicht zielführend.

¹¹ So werden seine Theorien noch heute u.a. an der Carl von Ossietzky Universität Oldenburg (Stoffers, 2015), der Universität Potsdam (Potsdam, 2020) oder auch der Technischen Universität Dortmund (Pasternak & Vahrenhold, 2009) gelehrt und u.a. in (Modrow & Strecker, 2016; Schubert & Schwill, 2011; Hubweiser, 2007) zitiert

Operatives Prinzip

Die Schüler sollten ermutigt werden, mental mit den Lernobjekten zu arbeiten.

Mentale Operationen spielen eine entscheidende Rolle beim Erwerb von Wissen und der kognitiven Entwicklung. Daher werden im Kurs konkrete Materialien, grafische Darstellungen und Texte verwendet, in denen die Lernenden real oder mental arbeiten und selbst forschen können. Medien, die nicht bearbeitet, sondern nur angesehen werden können, sind wertlos.

Prinzip der Stufenmäßigkeit

Der zu vermittelnde Lerninhalt muss an den geistigen Entwicklungsstand angepasst werden. Daher ist es sinnvoll den Schwierigkeitsgrad von Aufgabe zu Aufgabe zu erhöhen, sodass die nächste Aufgabe auf den aktuellen Entwicklungsstand aufbaut. So werden die Fähigkeiten stufenmäßig immer weiter ausgebaut. Dies ist ausschlaggebend dafür, dass die Lernenden nicht über- oder unterfordert werden und somit nicht die Motivation verlieren (vgl. Abschnitt 3.1).

Spiralprinzip

Die Behandlung eines komplexen Wissensgebiets soll bereits in früheren Lernstufen in einfacher Form eingeleitet werden, sodass dies mit Fortschreiten des Lernprozesses immer weiter aufgegriffen und vertieft werden kann.

Genetisches Prinzip

Der Lerninhalt soll so organisiert werden, dass er entsprechend der zeitgeschichtlichen Entwicklung vermittelt wird.

Prinzip der Lebensnähe und Aktualität

Die Lerninhalte müssen so aufbereitet werden, dass sie Bezüge zur aktuellen Lebenswelt der Lernenden aufweisen.

Prinzip des sachstrukturellen Aufbaus

Lerninhalte sind in logischer fachsystematischer Reihenfolge zu vermitteln

Prinzip der Zielvorstellung

Dem Lernenden wird das Ziel der Lehrveranstaltung mitgeteilt

Prinzip des individuellen Lerntempos

Lehrveranstaltungen sind gemäß des individuellen Lerntempos der Lernenden zu differenzieren.

4 Adaption von didaktischen Modellen in eine SQL- Lernwebapplikation

Die vorliegende Arbeit verfolgt das Ziel eine didaktisch fundierte Webapplikation zu entwickeln, mit welcher Studierende sich im Selbststudium asynchron grundlegende Kenntnisse zur Abfragesprache SQL aneignen können. Die Recherche nach geeigneter Literatur, die sich damit beschäftigt wie SQL am besten zu lernen ist, zeigte das Problem auf, dass keine eindeutige Reihenfolge und Gliederung der einzelnen Befehle zu erkennen ist. Vielmehr werden die einzelnen Befehle lediglich vorgestellt und ein Beispiel dazu gegeben. So werden die Befehle als einzelne isolierte Elemente behandelt und es wird kein erkennbarer themenübergreifender Zusammenhang deutlich. Dies führt dazu, dass die Reihenfolge der vorgestellten Befehle willkürlich und unstrukturiert erscheint (Beispiele dafür sind (Ebner, 1999; Fuchs, 2018)). So wird bereits die in Abschnitt 3 erwähnte Erkenntnis, dass der alleinige Zugang zu Lerninhalten nicht ausreicht, um größere Lernerfolge zu erzielen, missachtet. Für einen größeren Lernerfolg ist es unumgänglich die Lerninhalte systematisch aufzubereiten.

Um dieser Problematik entgegenzuwirken, werden in den folgenden Abschnitten zunächst die didaktischen Theorien und Modelle miteinander verglichen, um für den späteren Verlauf eine geeignete Grundlage zur Erstellung von strukturierten SQL-Lernaufgaben zu schaffen. Darauffolgend werden die in der Arbeit von D. Alić (Alić, 2021) vorgestellten SQL-Befehle zunächst nach Ihrer Wirkung in Gruppen gegliedert. Anschließend werden die SQL-Lernaufgaben erstellt sowie die nach jeder Aufgabe neu erworbenen Kompetenzen erläutert.

4.1 Diskussion der didaktischen Theorien und Modelle

Das kognitivistische Lernmodell beschreibt, dass der eigentliche Lernprozess während des Selbststudiums stattfindet. Der kognitive Input bestimmter Inhalte und ihre praktische Anwendung liegen eng beieinander, wodurch die Lerninhalte besser und langfristiger gelernt werden können. So können Studierende im Fall, der in dieser Arbeit behandelt wird, nach der Vorstellung von SQL in den Vorlesungsveranstaltungen die Webapplikation nutzen, um sich schließlich eigenständig der praktischen Anwendung anhand verschiedener Beispiele mit SQL zu befassen.

Der konstruktivistische Ansatz hingegen beschreibt den Lernprozess als konstruktiv, sodass der Lernende Wissen mit den zur Verfügung stehenden Lernwerkzeugen konstruiert. Allerdings muss der Studierende sich in einer Umgebung befinden, in welcher das Konstruieren ihm ermöglicht wird. Zudem unterscheidet sich der konstruktivistische Ansatz vom kognitivistischen Ansatz insofern, als dass der Konstruktivismus den Lernprozess während der Lernveranstaltung sieht. Denn so

können Lehrende direkt neues Wissen an ihr Vorwissen anknüpfen. Die Hirnforschung bestätigt, dass das Einbeziehen von Vorwissen unabdingbar für den Erwerb von neuem Wissen ist.

Die bildungstheoretische und kritisch-konstruktive Didaktik nach Klafki beschreibt sieben Fragen, die die Auswahl von durchzuführenden Unterrichtsinhalten von Lehrenden in den Vordergrund stellt und den Lehrpersonen als Leitfaden dienen kann. Jedoch stellt Klafkis Theorie ein Problem im Kontext dieser Arbeit dar: drei der sieben Leitfragen (vgl. Begründungszusammenhang) beziehen sich auf die Unterrichtsinhalte, die eine Lehrperson für die Lernenden bereitstellt und geht damit auf das Zusammenstellen der Inhalte eines gesamten Unterrichtsfaches bzw. Moduls ein. Jedoch ist die Webapplikation nur ein kleiner Teil der gesamten Modulinhalte. Zudem bezieht sich eine Frage auf die Lehr-Lern-Prozessstruktur, was ein variables Konzept, welches je nach Bedarf auf verschiedene Lehrhilfen zurückgreift. Die Webapplikation ist auf die eine Teilmenge der bereits bestehenden Lehrinhalte angepasst und bietet damit lediglich eine Lehrhilfe zum asynchronen Lernen. Somit ist die bloße Nutzung der Webapplikation und nicht die Strukturierung seiner Inhalte die Antwort auf die Frage nach der Lehr-Lern-Prozessstruktur. Die Frage nach der Zugänglichkeit bietet ebenfalls keinen Mehrwert, da lediglich ein Link und eine Internetverbindung nötig ist. Laut Statista nutzen 94% der Menschen in Deutschland das Internet (Statista, 2020b) und 100% der Deutschen zwischen 14 und 49 Jahren (Statista, 2020c). Weiterhin verzeichnet Statista, dass 219 Studierende in Deutschland zwischen 14 und 39 Jahre alt waren (Statista, 2019). Daraus kann abgeleitet werden, dass die Zugänglichkeit zur Webapplikation kein Problem darstellt. Somit bleiben noch die Fragen nach der thematischen Struktur und nach der Überprüfbarkeit, welche auf den Inhalt der Webapplikation angewandt werden können.

Die Didaktik der Informatik nach Baumann greift viele Punkte auf, die den beiden fundamentalen Lerntheorien des kognitivistischen und konstruktivistischen Lernens zu Grunde liegt. Zudem beschäftigt sich Baumann ebenfalls mit den beiden Fragen Klafkis, die für den weiteren Verlauf dieser Arbeit in Frage kommen. Baumann definiert zwölf Prinzipien, von denen allerdings drei nicht in auf die Erstellung der Inhalte der Webapplikation angewandt werden können. Zum einen das genetische Prinzip und das Prinzip der Lebensnähe und Aktualität, welche beschreiben, dass die Lerninhalte so organisiert werden sollen, dass sie entsprechend der zeitgeschichtlichen Entwicklung vermittelt werden und zudem Bezüge zur aktuellen Lebenswelt der Studierenden aufweisen. Und zum anderen das Prinzip der Zielvorstellung, wodurch den Lernenden das Ziel der Lehrveranstaltung mitgeteilt wird. Da diese Prinzipien die Aufgabe derjenigen, die die Lehrinhalte des Moduls bestimmen, ist, haben diese Prinzipien keine Bedeutung für die Erstellung der Übungsaufgaben oder der Applikation und werden daher nicht weiter betrachtet.

In **Tabelle 4.1** wird ein Überblick über die Parallelen, die Baumann mit der kognitivistischen und konstruktivistischen Lerntheorie und mit Klafkis Theorie, aufweist,

dargestellt. Zudem zeigt die Tabelle auf, ob die jeweiligen Prinzipien in der Erstellung der Aufgaben oder in der bloßen Entwicklung der Lernumgebung innerhalb der Applikation berücksichtigt werden können.

Tabelle 4.1: Parallelen von Baumann zum Kognitivismus, Konstruktivismus und zu Klafki

Prinzipien nach Baumann	weist Parallelen auf zu:	wird angewandt auf
Prinzip des aktiven Lernens	Kognitivismus	Applikation
Integrationsprinzip	Kognitivismus Konstruktivismus Frage 4 von Klafki	Aufgaben
Prinzip der Veranschaulichung	Kognitivismus	Aufgaben
Prinzip der Stabilisierung	Konstruktivismus Frage 5 von Klafki	Aufgaben
Operatives Prinzip	Konstruktivismus	Applikation
Prinzip der Stufenmäßigkeit	Konstruktivismus Frage 4 von Klafki	Aufgaben
Spiralprinzip	Kognitivismus Konstruktivismus	Aufgaben
Genetisches Prinzip	Nicht relevant	Nicht relevant
Prinzip der Lebensnähe und Aktualität	Nicht relevant	Nicht relevant
Prinzip des sachstrukturellen Aufbaus	Kognitivismus Konstruktivismus	Aufgaben
Prinzip der Zielvorstellung	Nicht relevant	Nicht relevant
Prinzip des individuellen Lerntempos	Konstruktivismus	Applikation

Somit kann zusammenfassend gesagt werden, dass das Modell von Baumann, alle didaktischen Erkenntnisse, die für die Entwicklung der Lernform innerhalb der Webapplikation samt Übungsaufgaben wichtig sind, berücksichtigt. Somit werden im folgenden Kapitel die Übungsaufgaben auf Basis von Baumanns Prinzipien erstellt. Die Erstellung wird zudem unter Berücksichtigung der psychologischen Definition von Motivation (vgl. Abschnitt 3.1) durchgeführt, da diese beim Prinzip der Stufenmäßigkeit eine entscheidende Rolle spielt.

4.2 Nutzung des Modells nach Baumann zur Erstellung von Übungsaufgaben

Die verschiedenen SQL-Befehle können in sechs Gruppen eingeteilt werden, sodass auch insgesamt sechs Aufgaben mit weiteren Unteraufgaben erstellt werden können. Die Überbegriffe und die Aufteilung der Aufgaben können **Tabelle 4.2** entnommen werden. Zudem sind sämtliche Aufgaben mit der entsprechenden Lösung detailliert im Anhang I einsehbar.

Tabelle 4.2: Übersicht der Übungsaufgaben

Aufgabe	Unteraufgabe	Überbegriff	zugehörige Tabellen
1	a), b), c)	SELECT (DISTINCT) FROM WHERE	Tabelle AI.I
2	a), b), c)	Datenmanipulation	Tabelle AI.I
3	a), b)	Boolsche Operatoren	Tabelle AI.I
4	a), b), c)	LIKE	Tabelle AI.II
5	a), b), c)	Sortieren, Rechnen und Gruppieren	Tabelle AI.III
6	a), b), c),	Abfragen über mehrere Tabellen/ JOIN	Tabelle AI.IV und A.V
	d), e)		Tabelle AI.VI und A.VII

Die erste Aufgabe kann in einem Block mit dem Titel *SELECT (DISTINCT) FROM WHERE* zusammengefasst werden. Zunächst werden die Übungen mit einer üblichen *SELECT - FROM –* Aufgabe eingeleitet. Diese sind fundamental, da fast alle SQL-Abfragen mit *SELECT* beginnen und in den meisten Fällen unmittelbar von *FROM* gefolgt werden. Damit lautet die Abfrage zur ersten Übungsaufgabe (Aufgabe 1a))

```
SELECT      Nachname, Vorname
FROM        Studierende .
```

Im Anschluss werden die gewonnenen Erkenntnisse sofort genutzt, um die erste Aufgabe um dem *WHERE –* Befehl zu erweitern. Daher lautet der hier einzugebende Befehl zu Aufgabe 1b)

```
SELECT      Nachname, Vorname
FROM        Studierende
WHERE       Universität = 'TU Dortmund' .
```

So kann der Lernende einen sehr leichten Einstieg erfahren. Dies ist wichtig, da der Lernende durch das Lösen von sehr einfachen Aufgaben die Lernsituation als positiv wahrnimmt und folglich ein positives Gefühl gegenüber dem Lernwerkzeug bekommt. Dadurch wird bereits zu Beginn dafür gesorgt, dass der Lernende eine Steigerung der Motivation erlebt. Diese Motivation gilt es nun möglichst lange aufrecht zu erhalten, wodurch dem Prinzip der Stufenmäßigkeit Folge geleistet wird. Zudem können die Lernenden die Erkenntnisse sofort nutzen, um die nächste Aufgabe zu lösen, sodass bereits zu Beginn das Integrationsprinzip angewandt wird.

Abgerundet wird die erste Aufgabe durch die Einführung einer DISTINCT-Aufgabe, in der möglichst einfachen Form mit lediglich einem weiteren FROM- Befehl. So können die erworbenen Kompetenzen nach dieser Aufgabe zusammengefasst werden zu:

- Einzelne Abfragen von Spalten und einfache Eingrenzungen der Suchkriterien mit „=“
- Duplikate bei der Abfrage einer Spalte entfernen.

Die zweite Aufgabe, welche mit dem Überbegriff *Datenmanipulation* beschrieben werden kann, behandelt sehr einfache Befehle der INSERT-, UPDATE- und DELETE-Funktionen. Diese Aufgabe scheint zunächst von der ersten Aufgabe isoliert zu sein und greift nur die kennengelernte WHERE- Funktion wieder auf.

Die Lernenden erwerben durch diese Aufgabe folgenden Kompetenzen:

- Zeilen löschen, hinzufügen oder einzelne Einträge verändern
- Zudem Eingrenzung durch „=“ und „>“
- Unterschied beim Eingrenzen von INTEGER/ NUMERIC/ FLOAT und CHAR

In der dritten Aufgabe werden die Booleschen Operatoren thematisiert. Dazu wird zunächst in Aufgabe 3a) auf das Wissen der ersten Aufgabe zurückgegriffen, da ohne die Kenntnisse aus Aufgabe 1 die Bearbeitung der Aufgabe 3 nicht möglich ist. Da nur der Operator OR als neu angesehen werden kann, kommt neben dem Integrationsprinzip an dieser Stelle das Prinzip der Veranschaulichung zu tragen. So kann der Lernende versuchen das neue Element in das bereits kennengelernte Schema einer SQL-Abfrage logisch einzuordnen. Dies bietet ihm die Möglichkeit selbstständig fremde Aufgaben durch logisches Denken zu lösen. Das entsprechende SQL-Statement dazu lautet

```
SELECT      Nachname
FROM        Studierende
WHERE       Universität = 'TU München' OR Matrikelnummer > 170000
```

Aufgabe 3b) verknüpft nun alle zuvor gewonnenen Erkenntnisse und kombiniert die Datenmanipulation mit der Eingrenzung und den Booleschen Operatoren gemäß der folgenden Abfrage

```
DELETE FROM      Studierende
WHERE            Universität = 'TU Dortmund' OR Studienbeginn = 2015
```

Die erworbenen Kompetenzen nach Aufgabe 3 sind somit:

- Anwendung der Booleschen Operatoren AND und OR
- In einem Befehl die Datenmanipulation mit den Booleschen Operatoren kombinieren

Aufgabe 4 kann als erweiterte Eingrenzung der Abfrage mit Hilfe der LIKE- Funktion betrachtet werden. Einleitend wird in Aufgabe 4a) eine einfache Möglichkeit aufgezeigt wie der LIKE- Befehl zu nutzen ist.

```
SELECT          Matrikelnummer
FROM            Studierendenliste
WHERE           Name LIKE '%Müller'
```

So kann sich der Lernende in Aufgabe 4b) gemäß dem das Prinzip der Veranschaulichung selbst erschließen wie LIKE mit den Booleschen Operatoren genutzt werden kann und wie die LIKE- Funktion in dieser Aufgabe aussieht. Denn die Art und Weise wie LIKE zu nutzen ist, unterscheidet sich leicht von dem in Aufgabe 4a) vorgestellten Fall. Die vom Lernenden erwartete Eingabe lautet wie folgt:

```
SELECT *
FROM            Studierendenliste
WHERE           Name LIKE 'M%n' AND PLZ LIKE '%2%'
```

Die Aufgabe 4 wird damit abgeschlossen, dass innerhalb von Aufgabe 4c) sowohl das Integrationsprinzip, das Prinzip der Stufenmäßigkeit und zudem das Prinzip der Stabilisierung beachtet werden. Diese Aufgabe umfasst die Datenmanipulation, die Booleschen Operatoren und die erweiterte Eingrenzung mit der LIKE- Funktion. Gemäß dem Prinzip der Stabilisierung werden die in den vorigen Aufgaben erworbenen Kenntnisse wiederholt und somit das Wissen stabilisiert und fester im Gedächtnis verankert. Die gesuchte SQL-Eingabe lautet:

```
UPDATE          Studierendenliste
SET             Studiengang = 'Informatik'
WHERE           Name LIKE '%n' AND Universität LIKE 'TU%'
```

Die erworbenen Kompetenzen nach Aufgabe 4 sind somit:

- Anwendung des LIKE- Befehls und des Symbols „%“ in verschiedenen Varianten
- Kombination von LIKE und den Booleschen Operatoren
- Kombination von LIKE, Datenmanipulation und den Booleschen Operatoren

Aufgabe 5 – Sortieren, Rechnen und Gruppieren – beginnt mit einer leichten einleitenden Aufgabe (Aufgabe 5a)), in der die Zähl- und Gruppierungsfunktion behandelt wird. Dort wird dem Lernenden ein kleiner, aber dennoch wichtiger Unterschied nähergebracht, welcher auf in den darauffolgenden Aufgaben zu beachten ist. Nämlich, dass wenn ein mathematischer Operator wie COUNT, SUM, AVG etc. genutzt wird, zuvor ein Komma gesetzt werden muss. Dies wird am untenstehenden Befehl zur Lösung von Aufgabe 5a) deutlich.

```
SELECT          Modul,  
COUNT(*)  
FROM            Klausurergebnisse  
GROUP BY       Modul .
```

In Aufgabe 5b) wird die Aufgabenstellung aus Aufgabe 5a) durch die Sortierfunktion erweitert, sodass ein zusätzlicher Befehl in die SQL- Eingabe eingefügt werden muss. Dadurch ergibt sich der Befehl

```
SELECT          Modul,  
SUM(Teilnehmerzahl)  
FROM            Klausurergebnisse  
GROUP BY       Modul  
ORDER BY       Modul DESC .
```

Abschließend wird analog zur Aufgabe 4c) die Aufgabe 5c) genutzt, um alle vorigen Befehle zu wiederholen und in einen gemeinsamen Befehl einzusetzen. Dadurch lautet die erforderliche Eingabe in Aufgabe 5c)

```
SELECT          Modul,  
AVG(durchgefallen)  
FROM            Klausurergebnisse  
WHERE           Modul LIKE '%c%'  
GROUP BY       Modul  
ORDER BY       Modul ASC
```

Die erworbenen Kompetenzen nach Aufgabe 5 sind:

- Zählen, Summe und Durchschnitt können bestimmt werden
- Spalten können gruppiert und in auf- bzw. absteigender Reihenfolge sortiert werden
- Kombination von Booleschen Operatoren, LIKE, Zählen, Summieren, Sortieren und Gruppieren

Die sechste und somit letzte Aufgabe unterscheidet sich von den vorigen Aufgaben insofern, als dass hier die SQL- Abfrage über zwei Tabellen statt nur einer behandelt

wird. An dieser Stelle kommt das Spiralprinzip zu tragen: das Arbeiten mit zwei Tabellen ist allgemein komplexer als mit nur einer Tabelle. Die Befehle, die beim Arbeiten mit zwei Tabellen notwendig sind, wurden allerdings bereits in den 5 vorigen Aufgaben eingeleitet. Somit wurde das in diesem komplexen Gebiet benötigte Wissen bereits zuvor intuitiv genutzt und die Lernenden sind bereits in der Lage die Komplexität des Arbeitens mit zwei Tabellen auf die Arbeit mit einer Tabelle zu abstrahieren. So zeigt das einleitende Beispiel in Aufgabe 6a), dass sie dem SQL-Befehl von Aufgabe 1a) und 1b) sehr ähnlich.

```
SELECT          Studierendeninformationen.Nachname,  
Studierendeninformationen.Vorname, Mechanikergebnisse.Note  
FROM            Mechanikergebnisse, Studierendeninformationen  
WHERE           Mechanikergebnisse.Matrikelnummer =  
Studierendeninformation.Matrikelnummer
```

In der darauffolgenden Aufgabe 6b) wird die Aufgabe aus 6a) nur um den Booleschen Operator erweitert. So dienen diese beiden Aufgaben dazu den Lernenden zu demonstrieren, dass ihnen das Arbeiten mit zwei Tabellen bereits in anderer Form bekannt ist. Dadurch sollen die Lernenden erneut ein positives Gefühl vermittelt bekommen, welche die Motivation steigert. Mit dieser Lernmotivation müssen sich die Lernenden nun Aufgabe 6c) stellen. Analog zu 4c) dient diese dazu das neue Wissen mit bereits erworbenem Wissen in Zusammenhang zu bringen und die zuvor kennengelernten Befehle wiederholen. Die Befehle aus 6a) und 6b) werden folgendermaßen erweitert:

```
SELECT          Studierendeninformationen.Nachname,  
Studierendeninformationen.Vorname  
FROM            Mechanikergebnisse, Studierendeninformationen  
WHERE           Mechanikergebnisse.Matrikelnummer =  
Studierendeninformationen.Matrikelnummer AND Modul = 'Mechanik A' AND Wohnort  
LIKE '%n%'  
ORDER BY Nachname DESC
```

Aufgabe 6d) greift erneut das Arbeiten mit zwei Tabellen auf. Jedoch ist diese Aufgabe ein einfaches Einführungsbeispiel der JOIN- Funktion. Diese Aufgabe weißt eine geringe Komplexität auf und dient dazu dem Lehrenden zu ermöglichen sich mit den elementaren Aufgaben zum JOIN- Befehl vertraut zu machen. Es werden lediglich die Kenntnisse aus Aufgabe 1a) und die grundlegenden Kenntnisse zu JOIN benötigt, um den folgenden Befehl zu erstellen:

```
SELECT          Studierendeninformation.Nachname, Ergebnisse.Note  
FROM            Studierendeninformation  
INNER JOIN      Ergebnisse ON Ergebnisse.Matrikelnummer =  
Studierendeninformation.Matrikelnummer
```

Im Anschluss wird die gesamte Web-Übung mit einer komplexen Aufgabe beendet. Innerhalb der Aufgabe 6e) wird der JOIN- Befehl mit möglichst vielen zuvor erlernten Befehlen kombiniert. Da der Lernende bereits alle Aufgaben erledigt hat kann daraus geschlossen werden, dass eine dauerhafte Beschäftigung und somit eine insgesamt als positiv wahrgenommene Lernsituation geschaffen wurde. Somit sollten der Lernerfolg und die Motivation den nötigen Antrieb für den Lernenden schaffen, um der letzten Aufgabe gegenüber zu treten. Die stetige Wiederholung der SQL- Befehle sorgt dafür, dass der Lernende eine hohe Entwicklungsstufe erreicht und viele Zusammenhänge innerhalb verschiedener SQL-Befehle kennt. Anhand des Befehls

```
SELECT      Ergebnisse. Matrikelnummer,  
Studierendeninformation.Nachname, Ergebnisse.Note  
FROM        Studierendeninformation  
INNER JOIN  Ergebnisse  
ON          Ergebnisse.Matrikelnummer =  
Studierendeninformation.Matrikelnummer  
WHERE       Note < 5 AND Wohnort LIKE 'D%'  
ORDER BY   Nachname ASC
```

ist erkennbar, dass, bis auf wenige Ausnahmen, alle vorigen Befehle zur Anwendung kommen.

Die erworbenen Kompetenzen nach Aufgabe 6 sind:

- Einfache JOIN- Befehle
- Abfragung über mehrere Tabellen
- Verknüpfung von JOIN mit Booleschen Operatoren, LIKE und Sortierung

Insgesamt zeigen die Aufgaben und die befolgten Prinzipien nach Baumann, dass das Prinzip des sachstrukturellen beachtet wurde. Dadurch werden die Lerninhalte in logischer fachsystematischer Reihenfolge vermittelt und bieten eine sehr gute Möglichkeit für Studierende sich die Grundlagen von SQL anzueignen. Das Prinzip des aktiven Lernens, das operative Prinzip und das Prinzip des individuellen Lerntempos sind Gegenstand des fünften Kapitels.

Des Weiteren bietet **Abbildung 4.1** eine Übersicht über den Umfang und den Schwierigkeitsgrad aller Aufgaben und Unteraufgaben. Diese Abbildung skizziert außerdem die Anwendung des Integrationsprinzips, des Prinzips der Stufenmäßigkeit und des Prinzips der Stabilisierung schematisch. Alle Aufgabennummern, werden von mindestens einem Rahmen umschlossen. Dabei hat ein Rahmen zwei Bedeutungen: zum einen gibt die Größe des Rahmens qualitativ den Schwierigkeitsgrad einer Aufgabe wieder. Zum anderen kann ein Rahmen mehrere Aufgabennummern umschließen. Dies bedeutet, dass für die Aufgabe, deren Rahmen weitere Aufgaben innehat, das Wissen der vorigen Aufgaben notwendig ist und somit mehrere Kompetenzen im Zusammenhang gebracht werden müssen.

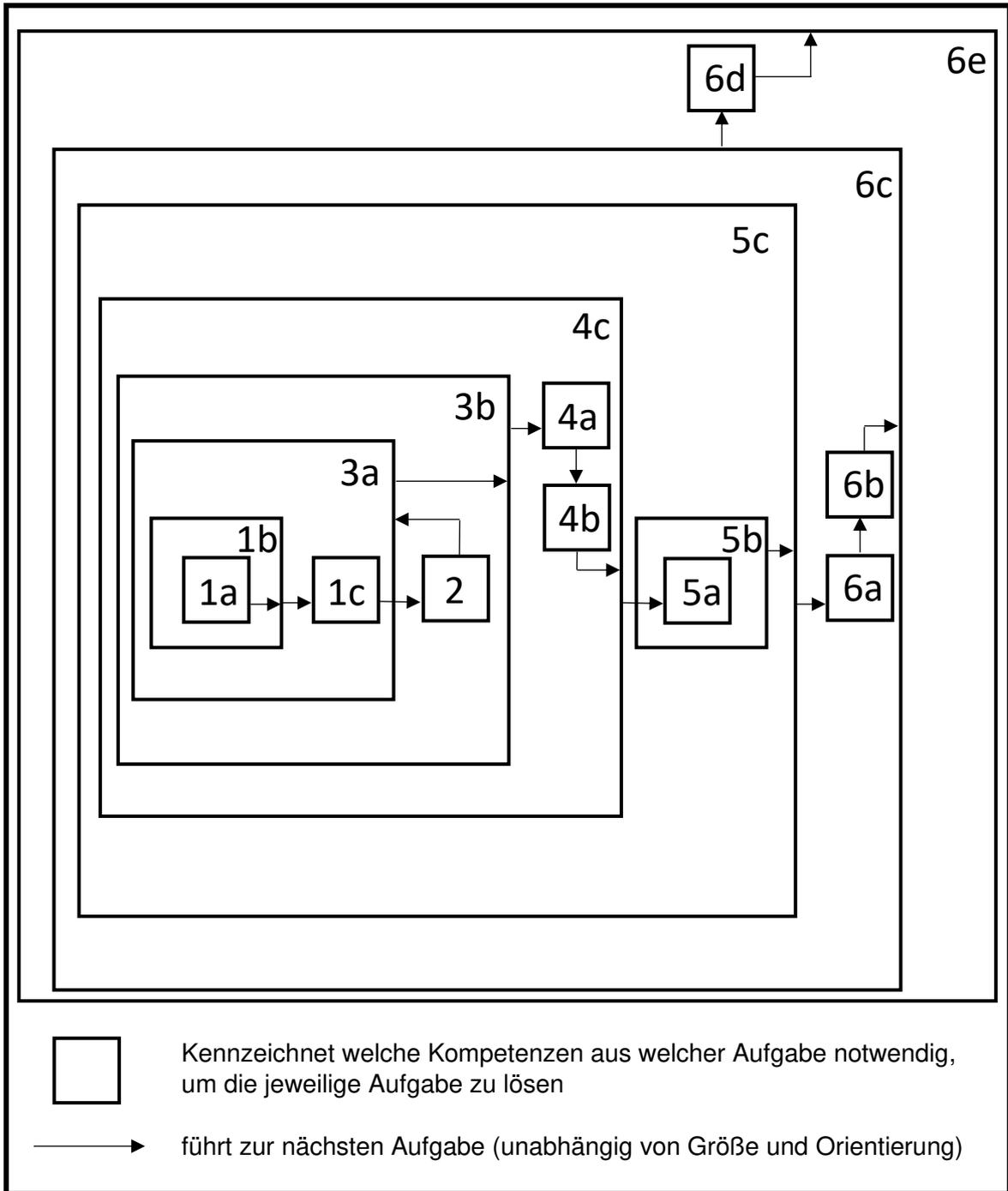


Abbildung 4.1: Übersicht zu den Übungsaufgaben und deren Schwierigkeitsgrad

5 Entwicklung des Frontends

Nachdem die Übungsaufgaben erstellt wurden, wird im folgenden Kapitel auf die Entwicklung des Frontends eingegangen. Dabei wird in Abschnitt 5.1 zunächst der Aufbau der graphischen Benutzeroberfläche beschrieben. Zusätzlich erfährt der Leser anhand der Abschnitte 2.2 und 4.1 warum die diese so gestaltet wurde. Zudem werden die einzelnen Softwaremodule, die innerhalb der Webapplikation entwickelt wurden, anhand der graphischen Benutzeroberfläche beschrieben.

5.1 Gestaltung der graphischen Benutzeroberfläche

Im folgenden Abschnitt wird der Aufbau der graphischen Oberfläche erläutert. Dabei wird auf die Einflussfaktoren der Ästhetik einer Benutzeroberfläche und auf die drei didaktischen Prinzipien *Prinzip des aktiven Lernens*, *operatives Prinzip* und *Prinzip des individuellen Lerntempos* (vgl. Abschnitt 3.4) eingegangen. Zunächst wird das Anmeldefenster und die eigentliche Arbeitsoberfläche beschrieben und dem Leser der technische Hintergrund anhand der einzelnen Softwaremodule näher gebracht. In jedem Modul werden sowohl Vorgänge im Front- als auch im Backend definiert. Eine genaue Beschreibung der einzelnen Module, jeweils gestützt durch Zustandsdiagramme, kann in Abschnitt 3.2 der Arbeit von D. Alić (Alić, 2021) und im Anhang II dieser Arbeit eingesehen werden. Das Verhalten der Module im Rahmen des Backends und die Elemente, die für die Schnittstellen zwischen Front- und Backend verwendet werden, werden ebenfalls ausführlicher in derselben Arbeit von D. Alić diskutiert. Dieses Kapitel konzentriert sich auf die Geschehnisse im Frontend. Ferner ist die Gestaltung des responsiven Designs Gegenstand dieses Abschnittes.

5.1.1 Anmeldemaske und Arbeitsoberfläche

Sobald der Nutzer die Webapplikation öffnet, erscheint eine Maske, über die sich der Nutzer anmelden kann (siehe **Abbildung 5.1**). Die Anmeldung bildet zwar ein einzelnes Modul, jedoch kommt zunächst das Softwaremodul „Seite laden“ zum Einsatz. Wie die einzelnen Module miteinander verbunden sind, kann anhand der in **Abbildung 5.2** dargestellten hierarchischen Struktur der Softwaremodule entnommen werden (eine nähere Beschreibung der hierarchischen Gliederung kann in Abschnitt 3.2 der Arbeit von D. Alić (Alić, 2021) eingesehen werden). Sobald der Nutzer den Uniform Resource Identifier (URI) der Webapplikation in einem Webbrowser ausführt, wird der Startzustand dieses Moduls eingestellt. Dadurch wird eine Hypertext Transfer Protocol (HTTP) GET-Anfrage vom Webbrowser an den Webserver gesendet, wodurch die Datei `index.html` vom Backend angefordert wird. Nachdem das Backend diese Datei zur Verfügung gestellt hat, wird geprüft, ob eine UserID im localstorage vorhanden ist. Falls diese nicht

vorhanden ist, wird das Modul „Anmeldung“ aufgerufen, wodurch die in **Abbildung 5.1** gezeigte Anmeldemaske erscheint. Ist eine UserID im localStorage hinterlegt, so hat der Nutzer sich bereits zuvor von diesem Webserver aus angemeldet. Es wird eine Verbindung zum WebSocket geöffnet und der Verbindungsaufbau vom Backend bewerkstelligt.

Abbildung 5.1: Anmeldemaske

Anschließend ruft der Webbrowser den URI für die nächste noch zu lösende Aufgabe auf. Zudem wird eine weitere HTTP GET-Anfrage an das Backend gesendet, um die Aufgabenstellung und die dazugehörigen Aufgabentabellen im JSON-Format bereitgestellt zu bekommen. So werden die Daten aus den JSON-Objekten ausgelesen und schließlich wird die zu lösende Aufgabe für den Nutzer sichtbar. Damit wird das Modul „Seitennavigation“ aufgerufen.

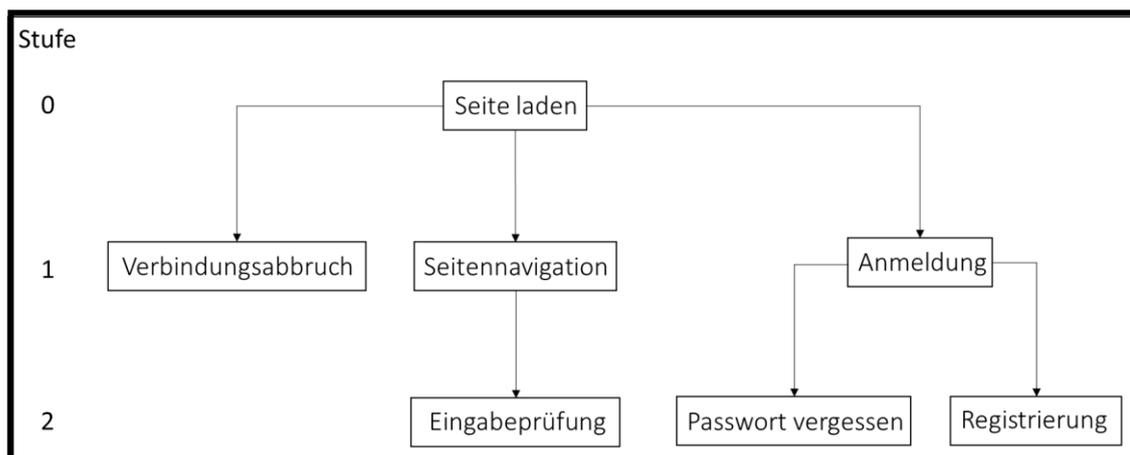


Abbildung 5.2: zweistufige hierarchische Gliederung der Softwaremodule der Webapplikation

Nun werden dem Nutzer drei Handlungsmöglichkeiten geboten: Hat der Nutzer noch kein Profil angelegt, so kann er dies mit der Funktion „noch nicht registriert?“ tun. Dadurch wird das Modul „Registrierung“ aufgerufen. Liegen Anmeldedaten vor, so kann der Nutzer diese eingeben. Nach der Benutzereingabe wird zum einen geprüft, ob die

Eingabefelder überhaupt gefüllt sind und ob die UniMail die korrekte Form einer TU-Dortmund-UniMail besitzt. Wird eine der beiden Kriterien missachtet, wird eine Fehlermeldung generiert, woraufhin der Nutzer seine Eingabe erneut tätigen kann. Sind beide Kriterien erfüllt wird eine HTTP POST-Anfrage an das Backend gesendet, wo die Anmeldedaten geprüft werden. Ist die UniMail nicht registriert oder ist das Passwort falsch, so erhält der Nutzer erneut eine Fehlermeldung. Ist die Eingabe richtig, wird vom Webserver aus die UserID an den Webbrowser gesendet. Im Anschluss wird die UserID in die localStorage geschrieben und dort hinterlegt. Abschließend wird das Modul „Seite laden“ aufgerufen. Wenn Anmeldedaten vorliegen, aber der Nutzer das Passwort vergessen hat, so kann dieser die „Passwort vergessen?“-Funktion nutzen, wodurch das Modul „Passwort vergessen“ aufgerufen wird und das Passwort erfragt werden kann. Hat der Nutzer noch kein Profil angelegt, so kann er dies mit dem Button „Noch nicht registriert“. Mit der Betätigung des „Noch nicht registriert“-Buttons erscheint die in **Abbildung 5.3** abgebildete Maske. Dadurch wird das Modul „Registrierung“ aktiv.



Abbildung 5.3: Maske zur Registrierung

An dieser Stelle muss der Nutzer seine TU-UniMail und schließlich ein von ihm gewähltes Passwort, welches im Anschluss wiederholt werden muss, eingeben. Die Registrierung wird mit der Betätigung des Buttons „Registrieren!“ abgeschlossen. Nach der Eingabe der UniMail und des von Nutzer gewählten Passwortes werden die Eingaben geprüft. Es wird geprüft, ob alle Felder gefüllt sind, ob die UniMail die korrekte Form einer TU-Dortmund-UniMail besitzt und ob das gewählte Passwort korrekt wiederholt wurde. Dies ist wichtig, damit sichergestellt wird, dass der Nutzer sich nicht versehentlich ein Passwort zuweist, das er nicht möchte. Wird mindestens eine der obigen Kriterien verletzt, erhält der Nutzer eine Fehlermeldung, sodass die Eingaben anschließend wiederholt werden können. Sind alle Kriterien erfüllt wird eine HTTP POST-Anfrage an den Webserver gesendet, welche die Eingabe erneut prüft. Dabei wird geprüft, ob die UniMail bereits registriert ist oder nicht. Ein Nutzer kann sich nicht mit einer UniMail registrieren, mit welcher bereits einmal eine Registrierung abgeschlossen wurde. Ist dies der Fall, dann wird diese Information an den Webbrowser gesendet und eine entsprechende Fehlermeldung generiert. Die Eingabe kann schließlich wiederholt

werden. Ist die UniMail nicht registriert, so werden die Daten vom Webserver verarbeitet und eine Nachricht zur erfolgreichen Registrierung an das Frontend gesendet. Es erscheint im Webbrowser ein Fenster mit der Registrierungsbestätigung aus **Abbildung 5.4**. Zusätzlich erhält der Nutzer die Registrierungsbestätigung per E-Mail. Wird der Button „Zur Anmeldung!“ angeklickt, wird erneut das Modul „Anmeldung“ aufgerufen.



Abbildung 5.4: Registrierungsbestätigung

Die Anmeldung erfolgt nach Eingabe der Anmeldedaten schließlich durch den Button „Anmelden!“. Hat der Nutzer sein Passwort vergessen, kann er dieses mit dem Button „Passwort vergessen?“ erfragen. Nach der Betätigung dieses Buttons wird das Modul „Passwort vergessen“ aufgerufen. Es erscheint eine weitere Maske (vgl. **Abbildung 5.5**) und der Nutzer wird aufgefordert seine UniMail einzugeben.



Abbildung 5.5: Maske zum Modul „Passwort vergessen“

Schließlich wird nach der Nutzung des Buttons „Passwort anfordern!“ dafür gesorgt, dass die Benutzereingaben zunächst geprüft werden. Dabei wird geprüft, ob die Eingabefelder gefüllt sind und ob die UniMail die korrekte Form einer TU-Dortmund-UniMail besitzt. Ist eine der beiden Kriterien nicht erfüllt, so wird eine Fehlermeldung generiert und dem Nutzer über den Webbrowser angezeigt. Sind beide Prüfungen positiv, so wird eine HTTP POST-Anfrage an den Webserver, wo geprüft wird, ob die UniMail bereits registriert ist oder nicht. Ist diese nicht registriert, wird auch hier dem

Nutzer eine Fehlermeldung angezeigt. Denn es kann kein Passwort zu einer Mailadresse erfragt werden, wenn dieser Adresse noch kein Passwort zugewiesen wurde. Ist die UniMail registriert, so werden die Daten im Webserver verarbeitet und im Anschluss im Webbrowser ein Fenster mit der Versandbestätigung der E-Mail mit dem Passwort angezeigt (vgl. **Abbildung 5.6**).



Abbildung 5.6: Versandbestätigung

Nun kann mit dem Button „Zur Anmeldung!“ das Modul „Anmeldung“ erneut aufgerufen werden. Nach der Anmeldung wird der Nutzer zur Arbeitsoberfläche geleitet, wo die Bearbeitung der Übungsaufgaben durchgeführt werden kann. Dadurch wird das Modul „Seitennavigation“ aufgerufen. Die Arbeitsoberfläche wird in **Abbildung 5.7** dargestellt und im Weiteren näher erläutert. Wie in Abschnitt 2.2 erwähnt, ist es beim Design der Benutzeroberfläche im Sinne der Ästhetik empfehlenswert eine gemäßigte Anzahl an Farben zu verwenden und grelle Farben eher dezent einzusetzen oder gar zu vermeiden. Daher wurde hier überwiegend die Farbe Grün ausgewählt, um das Design der Oberfläche an die Farben der TU Dortmund anzulehnen. Dadurch soll zum einen erreicht werden, dass die Buntheit so gering wie möglich gehalten wird und zum anderen, dass sich die Studierenden durch die Farbassoziation in einer für sie bekannten Lernsituation wiederfinden. Dazu wurde zudem die Kopfzeile mit den Logos der Technischen Universität Dortmund des Fachgebiets IT in Produktion und Logistik versehen. Zudem ist die Anzahl an Elementen, welche als die visuelle Komplexität verstanden wird, so gering wie möglich gehalten worden. Die größte Anzahl an Elementen befindet sich in der Navigationsleiste, deren Auswirkung auf die Ästhetik im Usability-Test noch beurteilt werden muss. Auf die Navigationsleiste wird im weiteren Verlauf näher eingegangen.

Im Softwaremodul „Seitennavigation“ hat der Nutzer mehrere Handlungsmöglichkeiten: Button Klick auf „Lösung“, auf „Hilfe“, auf „Abmelden“ oder eine andere Aufgabe in der Navigationszeile auswählen. Auf diese Aktionen wird im weiteren Verlauf schrittweise eingegangen.

tu Technische Universität Dortmund

SQL Web-Übungen

itpi IT-Produktion und Support

Angemeldet als Osman Alic

Gelöste Aufgaben: 19

Hilfe Abmelden

Aufgabe 1: a, b, c

Aufgabe 2: a, b, c

Aufgabe 3: a, b

Aufgabe 4: a, b, c

Aufgabe 5: a, b, c

Aufgabe 6: a, b, c, d, e

Aufgabe 1 a)

Erzeugen Sie eine SQL-Abfrage, die Ihnen lediglich die Nachnamen und Vornamen der Studierenden ausgibt.

Tabelle: Studierende

Matrikelnummer	Nachname	Vorname	Studiengang	Studienbeginn	Universität
123456	Klaus	Felix	Maschinenbau	2015	TU München
145789	Said	Abdel	Logistik	2017	TU Dortmund
165472	Weiss	Werner	BWL	2019	HRU Düsseldorf
127945	Kaye	Hassen	Logistik	2015	TU München
154978	Durr	Wolfgang	Mathematik	2018	HRU Düsseldorf
164722	Jackson	Michael	Musik	2019	TU Dortmund
155520	Kätzger	Shindhi	Maschinenbau	2016	TU Dortmund
179011	Mersov	Dimitri	Logistik	2020	TU Dortmund

Ihre Eingabe:

Lösung

Ausführen

Abbildung 5.7: Arbeitsoberfläche der Webapplikation

Unter der Kopfzeile befindet sich ein grünes Banner, welches diverse Funktionen hat. Zum einen werden auf der linken Seite der Vor- und Nachname des Nutzers und die Anzahl der bereits gelösten Aufgaben angezeigt. Auf der rechten Seite befinden sich zwei Buttons: der Button „Hilfe“ und der Button „Abmelden“ (siehe **Abbildung 5.8**).



Abbildung 5.8: linke und rechte Seite des Banners

Bedient der Nutzer die „Hilfe“-Funktion so wird vom Webbrowser eine HTTP GET-Anforderung an den Webserver gesendet, um dem Nutzer die Datei „Hilfe.pdf“ (siehe Anhang III) in einem neuen Fenster anzuzeigen. In diesem Dokument werden dem Nutzer die Funktionen der Applikation ausführlich erklärt. Klickt der Nutzer auf den Button „Abmelden“ wird das Modul „Seite laden“ erneut aufgerufen und der Nutzer verlässt die Arbeitsoberfläche woraufhin wieder die Anmeldemaske aus **Abbildung 5.1** erscheint. Bei erneuter Nutzung der Webapplikation muss erneut eine Anmeldung durchgeführt werden. Wird die Webapplikation geschlossen ohne dass der Nutzer sich zuvor abgemeldet hat, so gibt es bei erneuter Nutzung der Applikation zwei Möglichkeiten: hat der Nutzer sein Passwort gespeichert, muss er sich nicht mehr anmelden. Hat dieser sein Passwort nicht gespeichert, muss er die Anmeldedaten bei jeder Nutzung erneut eingeben. Der Nutzer kann zu jedem Zeitpunkt die Webapplikation schließen und dadurch das Modul „Verbindungsabbruch“ aktivieren. Dadurch wird die Verbindung zum WebSocket-Kanal geschlossen und der Webserver löscht die temporär erstellten Datenbanken des Benutzers (siehe unten).

Unter dem Banner befinden sich die zu lösenden Aufgaben. Zum einen hat der Nutzer durch die Navigationsleiste (**Abbildung 5.9**) die Möglichkeit sich zu der gewünschten Aufgabe zu navigieren. Sobald eine Aufgabe ausgewählt wird, wird das Modul „Seite laden“ mit der Ziel-URI zur ausgewählten Aufgabe aufgerufen.



Abbildung 5.9: Navigationsleiste

Ist eine Aufgabe nicht gelöst so wird das entsprechende Feld in der Navigationsleiste grau hinterlegt (Aufgabe 6d) und 6e)), wie **Abbildung 5.10** zu entnehmen ist.

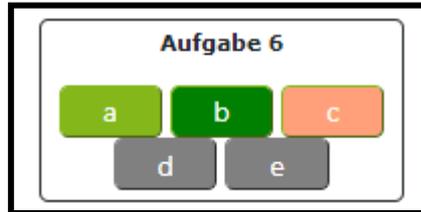


Abbildung 5.10: Felder von nicht gelösten und gelösten Aufgaben

Ist eine Aufgabe gelöst worden so wird das entsprechende Feld in der Navigationsleiste grün hinterlegt (Aufgabe 6a) und 6b)). Befindet sich der Nutzer bei einer Aufgabe, die bereits gelöst wurde, so wird dieses Feld mit einer dunkelgrünen Farbe hinterlegt (Aufgabe 6b)). Ist die Bearbeitung einer Aufgabe freigeschaltet und der Nutzer eine andere Aufgabe ausgewählt hat, so wird dieses Feld rot markiert (Aufgabe 6c)).

Unterhalb der Navigationszeile befindet sich die Aufgabenstellung mit den entsprechenden Tabellen. In **Abbildung 5.11** wird beispielhaft die Aufgabenstellung zu Aufgabe 1a) gezeigt. Es werden für jeden User separate temporäre Aufgabentabellen erstellt. Dadurch kann der Nutzer mit den Tabellen experimentieren und diese beliebig modifizieren ohne dass Aufgabentabellen anderer Nutzer beeinflusst werden. Wie zuvor erwähnt werden diese Tabellen bei Schließung der Applikation vom Webserver gelöscht.

Aufgabe 1 a)		Tabelle: Studierende					
Erzeugen Sie eine SQL-Abfrage, die Ihnen lediglich die Nachnamen und Vornamen der Studierenden ausgibt.		Matrikelnummer	Nachname	Vorname	Studiengang	Studienbeginn	Universität
		123456	Klaus	Felix	Maschinenbau	2015	TU München
		145794	Said	Abdel	Logistik	2017	TU Dortmund
		165472	Weiss	Werner	BWL	2019	HHU Düsseldorf
		127945	Kaya	Hassan	Logistik	2015	TU München
		154878	Dürr	Mirjam	Mathematik	2018	HHU Düsseldorf
		164722	Jackson	Michael	Musik	2019	TU Dortmund
		155520	Kataga	Shinichi	Maschinenbau	2018	TU Dortmund
		176001	Morozov	Dmitri	Logistik	2020	TU Dortmund

Abbildung 5.11: Aufgabenstellung in der Webapplikation

Die rote gestrichelte Linie in **Abbildung 5.7** kennzeichnet die Symmetrieachse. Je symmetrischer die Oberfläche ist, desto besser wirkt sich dies auf die Ästhetik dieser aus. Hier ist die Symmetrie so zu verstehen, dass sich in der linken Bildschirmhälfte die Aufgabenstellung und in der rechten Bildschirmhälfte die dazugehörigen Tabellen befinden. Wie der obigen Abbildung zu entnehmen ist, überquert die Tabelle die Symmetrieachse. Dies liegt daran, dass ein weiteres Verrücken der Tabelle hin zum rechten Rand dazu führt, dass bei Bildschirmen mit üblicher Größe (etwa 14-15 Zoll) innerhalb der Tabelle ein Zeilenumbruch entstehen würde. Wie die Tabellen auf kleineren oder größeren Bildschirmen dargestellt werden, wird in Abschnitt 5.1.2 - responsives Design - diskutiert. Da viele Studierende Laptops mit den etwaigen

Bildschirmgrößen benutzen, wurde hier bewusst auf eine vollständige Symmetrie verzichtet.

Im letzten unteren Viertel der Arbeitsoberfläche befindet sich das Eingabefeld. Diese dient dem Nutzer dafür seine SQL-Statements einzugeben, um die Aufgabe zu lösen. Zunächst ist das Feld leer. Bei Eingabe der SQL-Abfrage erscheint das Statement in schwarzer Schrift bis auf die bekannten SQL-Schlagwörter wie `SELECT`, `FROM`, `WHERE` etc. Diese färben sich, wie bei üblichen SQL-Abfragesystemen, nach der Eingabe und ändern die Schriftart. Hier wurde die Farbe Grün und die Schriftart kursiv gewählt, wie **Abbildung 5.12** zeigt.

Abbildung 5.12: Eingabefeld mit durchgeführter Eingabe

Nach erfolgter Eingabe kann der Nutzer seine Eingabe prüfen, indem er auf den Button „Ausführen“ (**Abbildung 5.13**) klickt.



Abbildung 5.13: „Ausführen“-Button

Wenn der „Ausführen“-Button betätigt wird, wird das Modul „Eingabeprüfung“ aufgerufen. Dabei wird eine HTTP POST-Anfrage an den Webserver gesendet. Dort werden die Eingabedaten empfangen und auf ihre Richtigkeit überprüft. Ist die Eingabe falsch, so werden dem Webbrowser u.a. die in **Abbildung 5.14** gezeigten Fehlermeldungen aufgezeigt. Der Fehler *Relation >>temp_2_studierenden<< existiert nicht* ist ein Fehler auf lexikalischer Ebene. Dieser wird ausgelöst, wenn der Benutzer Begriffe verwendet, die nicht im Vokabular der Maschine existieren.

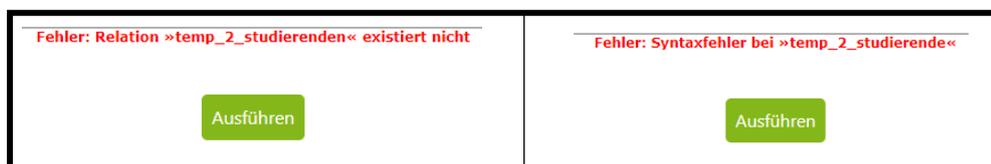


Abbildung 5.14: Fehlermeldungen bei syntaktischen und lexikalischen Fehlern

Statt den Tabellennamen „Studierende“ zu schreiben, wurde hier „Studierenden“ eingegeben. Der Fehler *Syntaxfehler bei >>temp_2_studierende<<* ist ein Fehler auf syntaktischer Ebene. Dabei wurde eine SQL-Funktion falsch geschrieben. Zum Beispiel wurde statt `FROM` nur `FRO` geschrieben. Liegen sowohl lexikalische als auch syntaktische Fehler vor, so wird der Nutzer zunächst auf die Syntaxfehler hingewiesen. Jedoch

berücksichtigt die Webapplikation auch Fehler auf semantischer Ebene. Es ist möglich, dass der Nutzer den Befehl syntaktisch und lexikalisch zwar fehlerfrei angibt, jedoch dabei die eigentliche Aufgabe missachtet. Der eingegebene Code ist zwar durchführbar, erreicht jedoch nicht das angestrebte Ziel. Semantische Fehler werden von der Webapplikation auf zweierlei Art berücksichtigt: Resultiert die getätigte Eingabe in einer leeren Ausgabe, so erscheint der in **Abbildung 5.15a** gezeigte Hinweis.

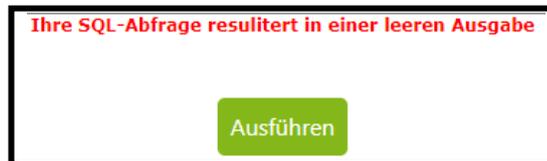


Abbildung 5.15a: Fehlermeldung bei semantischen Fehlern

Resultiert die Eingabe nicht in einer leeren Ausgabe, aber ist dennoch semantisch falsch, so erscheint der Hinweis aus **Abbildung 5.15b**. Dem Nutzer wird mitgeteilt, dass die Antwort falsch ist und zudem die Ausgabe seines Befehls und die erwartete Ausgabe gezeigt, anhand welcher sich dieser beim weiteren Versuch die Aufgabe zu lösen orientieren kann.



Abbildung 5.15b: Fehlermeldungen bei semantischen Fehlern

Mit dem Button „Zurück“ gelangt der Nutzer wieder zur Bearbeitung der Aufgabe.

Ist die Eingabe richtig, so erscheint das Fenster, welches in **Abbildung 5.16** gezeigt wird.

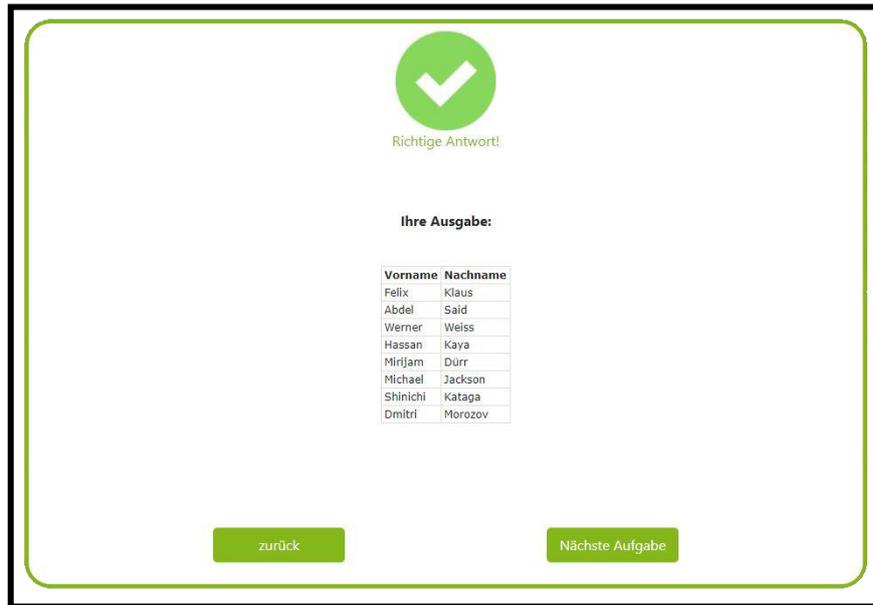


Abbildung 5.16: Meldung bei richtiger Eingabe

Mit der Nutzung des Buttons „zurück“ kann der Nutzer dieselbe Aufgabe erneut bearbeiten. Mit der Betätigung des Buttons „Nächste Aufgabe“ wird das Modul „Seite laden“ erneut aufgerufen und der Nutzer wird zur Bearbeitung der nächsten Aufgabe geleitet.

Ist es dem Nutzer nicht möglich eine Aufgabe zu lösen, so kann er sich mit dem „Lösung“-Button das gesuchte SQL-Statement anzeigen lassen (**Abbildung 5.17**).

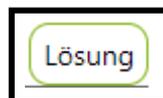


Abbildung 5.17: „Lösung“-Button

Die Lösung erscheint dabei in dem Eingabefeld. Das Modul „Eingabeprüfung“ wird unabhängig davon, ob der Nutzer seine Eingabe selbstständig getätigt hat oder ob er auf die „Lösung“-Funktion zurückgegriffen hat, bei Betätigung des „Ausführen“-Button aufgerufen.

Die Nutzung dieser Webapplikation ermöglicht es die eingangs genannten Prinzipien nach Baumann (vgl. Abschnitt 3.4) zu befolgen. Zum einen greift das Prinzip des aktiven Lernens, da sich die Studierenden mit dieser Webapplikation selbstständig mit dem Thema SQL befassen können. Da es keine zeitliche Limitierung bei der Lösung einer Aufgabe gibt, kann der Studierende in seinem eigenen Lerntempo die Inhalte, die die Webapplikation bietet, lernen. Damit ist das Prinzip des individuellen Lerntempos berücksichtigt worden. Zudem bietet die Fehlermeldung, die in **Abbildung 5.15b** aufgezeigt wurde, dem Studierenden einen großen Vorteil, welcher im operativen Prinzip nach Baumann formuliert wurde. Denn der Studierende kann losgelöst von der Aufgabe mit den verschiedenen Tabellen der Aufgaben experimentieren und SQL selbst „erforschen“. Auch wenn semantische Fehlermeldungen auftreten, wird dem

Studierenden dennoch die Ausgabe seiner SQL-Eingabe angezeigt. Da der Studierende eine unbegrenzte Anzahl an Versuchen für jede Aufgabe hat, kann dieser auch theoretisch unendlich oft verschiedene Eingaben testen und so auch ohne Beachtung der Aufgaben die Sprache SQL lernen.

5.1.2 Responsives Design

Um den Nutzern ein angenehmes Arbeiten auf allen Geräten zu ermöglichen, ist es wichtig ein sogenanntes responsives Design in die Webapplikation einzupflegen. So wurde sich auf drei verschieden große Bildschirmgrößen festgelegt. Diese repräsentieren die Desktopansicht, die Ansicht auf einem Tablet bzw. Laptop mit kleinem Bildschirm und die Ansicht auf einem Smartphone. Um dieses Design zu realisieren, wurden zwei Regeln formuliert: Ist die Pixelanzahl der Breite des Bildschirms größer als 1.150 Pixel, so nimmt die Weboberfläche die bereits aus **Abbildung 5.7** bekannte Form an. Die Schrift ist in der Schriftart Verdana und in der Schriftgröße 12 dargestellt. Wird die Anzahl der Pixel von 1.150 unterschritten, so wechselt die Weboberfläche zur Tablet-Ansicht (vgl. **Abbildung 5.18**).

Angemeldet als Osman Alic

Gelöste Aufgaben: 19 Hilfe Abmelden

Aufgabe 1

a b c

Aufgabe 4

a b c

Aufgabe 2

a b c

Aufgabe 5

a b c

Aufgabe 3

a b

Aufgabe 6

a b c

d e

Aufgabe 6 e)

Ordnen Sie den Noten den entsprechenden Nachnamen und die entsprechende Matrikelnummer zu. In der ersten Spalte soll die Matrikelnummer, in der zweiten der Nachname und in der dritte die Note stehen. Achten Sie zudem darauf, dass nur Studierende gesucht werden, die die Klausur bestanden haben (die Klausur zählt als bestanden, wenn die Note 4 oder eine bessere erreicht wird) und aus einer Stadt kommen, die mit dem Buchstaben D anfängt. Die Liste soll in aufsteigender alphabetischer Reihenfolge nach dem Nachnamen sortiert werden.

Tabelle: Ergebnisse

Matrikelnummer	Modul	Punktzahl	Note
123456	Mechanik A	100	1
145784	Mechanik D	41	4
165472	Mechanik A	51	3.3
127945	Mechanik A	59	3
154878	Mechanik C	87	1.7
164722	Mechanik B	10	5
155520	Mechanik C	1	5
146558	Mechanik B	91	1.7

Tabelle: Studierendeninformation

Matrikelnummer	Nachname	Vorname	Wohnort
123456	Klaus	Felix	Dortmund

Abbildung 5.18: Arbeitsoberfläche der Webapplikation in der Tablet-Ansicht

Es ist zu erkennen, dass sich die Navigationsleiste in diesem Fall über zwei Zeilen erstreckt und die Aufgabenstellung nicht mehr neben, sondern über der Aufgabentabellen steht. Zudem verschwinden aus Platzgründen die Logos der TU

Dortmund und des Fachgebiets IT in Produktion und Logistik. Um die SQL-Eingabe durchzuführen, muss der Nutzer nach unten scrollen, um das Eingabefeld zu erreichen.

Eine weitere Regel, die im Rahmen des responsiven Designs umgesetzt wurde, ist die Umwandlung der Weboberfläche bei einer Breite von 500 Pixel oder geringer. Die Darstellung der Weboberfläche ist in **Abbildung 5.19** zu sehen und repräsentiert die Smartphone-Ansicht.

SQL Web-Übungen

Angemeldet als Dzermal Alic

Gelöste Aufgaben: 19 Hilfe Abmelden

Aufgabe 1

a b c

Aufgabe 2

a b c

Aufgabe 3

a b

Aufgabe 4

a b c

Aufgabe 5

a b c

Aufgabe 6

a b c

d e

Aufgabe 6 e)

Ordnen Sie den Noten den entsprechenden Nachnamen und die entsprechende Matrikelnummer zu. In der ersten Spalte soll die Matrikelnummer, in der zweiten der Nachname und in der dritte die Note stehen. Achten Sie zudem darauf, dass nur Studierende gesucht werden, die die Klausur bestanden haben (die Klausur zählt als bestanden, wenn die Note 4 oder eine bessere erreicht wird) und aus einer Stadt kommen, die mit dem Buchstaben D anfängt. Die Liste soll in aufsteigender alphabetischer Reihenfolge nach dem Nachnamen sortiert werden.

Tabelle: Ergebnisse

Matrikelnummer	Modul	Punktzahl	Note
123456	Mechanik A	100	1
145784	Mechanik D	41	4
165472	Mechanik A	51	3,3
127945	Mechanik A	59	3
154878	Mechanik C	87	1,7
164722	Mechanik B	10	5
155520	Mechanik C	1	5
146558	Mechanik B	91	1,7

Tabelle: Studierendeninformation

Matrikelnummer	Nachname	Vorname	Wohnort
123456	Klaus	Felix	Dortmund
145784	Said	Abdel	Dortmund
165472	Weiss	Werner	Hagen
127945	Kaya	Hassan	Duisburg
164722	Jackson	Michael	Hagen
155520	Kataga	Shinichi	Düsseldorf
176001	Morozov	Dmitri	Duisburg
146558	Halilovic	Edin	Hagen

Abbildung 5.19: Arbeitsoberfläche der Webapplikation in der Smartphone-Ansicht

Im Gegensatz zur Tablet-Ansicht wurde lediglich die Schriftgröße von 12 auf 10 geändert, um die Oberfläche ideal auf die Bildschirmgröße eines handelsüblichen Smartphones anzupassen.

6 Durchführung des Usability-Tests

In diesem Kapitel werden verschiedene Usability-Tests durchgeführt. Wie in Abschnitt 2 bereits beschrieben, können zum einen Softwaretest nach dem V-Modell und zum anderen Usability-Tests durchgeführt werden, um eine Webapplikation zu testen und auf Fehler zu untersuchen. Die Tests Komponententest, Integrationstest und Systemtest werden in der Arbeit von D. Alić (Alić, 2021) durchgeführt. Dieses Kapitel beschränkt sich auf den Usability-Test. Hierbei handelt es sich um einen sogenannten induktiven Test, da dieser der formativen Evaluation, bei welcher Prototypen auf Schwachstellen und Verbesserungsmöglichkeiten hinsichtlich der Funktionalität und Gestaltung untersucht werden, dient. Dazu werden im Folgenden zunächst der Versuchsaufbau und die Bedingungen, unter denen der Test durchgeführt wird, beschrieben. Mit diesem Test wird die in Abschnitt 2.2 erwähnte Ästhetik einer Webapplikation bewertet. Des Weiteren wird untersucht inwieweit die nach didaktischen und psychologischen Gesichtspunkten erstellten Übungsaufgaben in der Praxis ihre Zweckmäßigkeit erfüllen. Zudem wird analysiert, ob den Testpersonen Programmierfehler oder Unstimmigkeiten innerhalb der Webapplikation auffallen. Der Usability-Test wird von einem Fragebogen, begleitet, welcher im Anschluss ausgewertet wird (eine Vorlage des verwendeten Fragebogens kann im Anhang IV eingesehen werden). Zudem werden die Aktionen der Teilnehmer per Bildschirmaufnahme verfolgt, um den Gedankengang der Teilnehmer im Nachhinein besser nachvollziehen zu können. Abschließen wird aus den gewonnenen Erkenntnissen ein Fazit gezogen.

6.1 Versuchsaufbau und Testpersonen

Wie bereits bekannt ist werden mehr Fehler aufgedeckt je mehr Testpersonen am Usability-Test teilnehmen. Da es nicht ohne weiteres möglich war viele Testpersonen zu finden, wurde dieser Test auf Empfehlung von Nielsen (vgl. Abschnitt 2.3.4) mit fünf Testpersonen durchgeführt. Dabei wurden Testpersonen gewählt, die unterschiedlich gute Kenntnisse in SQL haben und denen zudem die Webapplikation bisher unbekannt war. Aufgrund der aktuellen COVID-19-Pandemie der Versuch unter Einhaltung der Corona-Hygiene- und Abstandsregelungen durchgeführt. Zudem wird der Versuch von einem Fragebogen begleitet und die Aktionen der Teilnehmer per Bildschirmaufnahme festgehalten.

Der Test besteht aus drei Teilen. Bevor der Test beginnt, geben die Teilnehmer zunächst auf einer Skala von 0 bis 6 an, wie sie ihre SQL-Kenntnisse selber einschätzen. Im ersten Test wird gemäß Abschnitt 2.2 die intuitive Ästhetik bewertet. Dazu werden den Teilnehmern zu Beginn die Definition der Ästhetik erläutert. Dadurch kann jeder Teilnehmer mit dem gleichen Verständnis von Ästhetik diese bewerten. Den Versuchspersonen wird ein zehnstündiges Video abgespielt, welches lediglich ein

schwarzes Bild zeigt. In einem unbekanntem Zeitpunkt innerhalb der zehn Sekunden wird für 500ms die graphische Oberfläche der Webapplikation (vgl. **Abbildung 5.5**) eingeblendet. Bevor das Video abgespielt wird, müssen die Teilnehmer zunächst die Frage beantworten, ob sie den Begriff „Symmetrie“ mit der Punkt- oder Achsensymmetrie assoziieren. Daraufhin wird das Video abgespielt. Nachdem das Video beendet wurde und die Teilnehmer einen ersten Eindruck der graphischen Oberfläche erhalten haben, werden sie gebeten die Ästhetik zu bewerten. Dazu werden die drei Einflussfaktoren Symmetrie, Buntheit/ Farbgebung und die visuelle Komplexität und deren Auswirkung auf die Ästhetik separat von den Nutzern bewertet.

Im zweiten Teil des Tests werden die Teilnehmer gebeten die Übungsaufgaben innerhalb der Webapplikation selbstständig zu bearbeiten und den Schwierigkeitsgrad der Aufgaben zu bewerten. Dazu wird vom Versuchsteilnehmer festgehalten, ob die Teilnehmer auf die „Lösung“-Funktion zurückgreifen müssen. Schließlich wird erfragt, ob aus der Sicht der Teilnehmer die Befehle aus vorigen Aufgaben regelmäßig wiederholt wurden und in logische Zusammenhänge verknüpft werden konnten. Zudem bewerten die Teilnehmer inwiefern die Hinweise, die die Webapplikation ihnen bietet, bei der Lösungsfindung der Aufgaben geholfen haben. Im Anschluss wird die reflektive Ästhetik (Abschnitt 2.2) bewertet. Dazu werden die Teilnehmer gebeten die Fragen aus der intuitiven Ästhetik erneut zu beantworten, nachdem sie sich nun einen umfassenden Eindruck von der Webapplikation machen konnten. Darüber hinaus werden die Nutzer gefragt, inwiefern die Ästhetik bei der Bedienung der Webapplikation und zum emotionalen Empfinden während der Bedienung beigetragen hat.

Im dritten Teil des Tests werden haben die Teilnehmer die Möglichkeit sämtliche Funktionen der Webapplikation nach ihrem Belieben zu testen. Dazu wird ihnen lediglich der Hinweis gegeben, dass es erwünscht ist absichtlich falsche Eingaben zu tätigen. Auf diese Weise können sie testen wie gut die Applikation funktioniert und zudem Fehler aufdecken. Sämtliche Einwände können die Teilnehmer auf dem Fragebogen formulieren. Alle Teilnehmer führten den Versuch mit demselben Rechner durch, welcher einer Bildschirmdiagonale von 15,6“ besitzt. Dadurch wurden für alle Probanden die gleichen Rahmenbedingungen geschaffen und die Bewertung der Ästhetik kann einheitlich durchgeführt werden.

6.2 Auswertung der Tests

Bevor der Versuch begonnen hat, wurden die Teilnehmer darum gebeten auf einer Skala von 0 bis 6 einzutragen wie sie ihre eigenen SQL-Kenntnisse einschätzen. Dies hat den Hintergrund, dass die Kenntnisse ausschlaggebend dafür sind, ob ein Nutzer gewisse SQL-Aufgaben als leicht oder als schwer empfindet. Dies wird bei der Auswertung des zweiten Versuchsteil helfen den Gedankengang der Teilnehmer besser nachvollziehen zu können.

Im ersten Test wurde die intuitive Ästhetik der Benutzeroberfläche untersucht. Dazu wurde den Probanden die Benutzeroberfläche für lediglich 500ms gezeigt. Doch bevor, die Testpersonen die Benutzeroberfläche sehen konnten, wurden sie gefragt, ob sie den Begriff „Symmetrie“ mit der achsen- oder punktsymmetrischen Darstellung von Objekten assoziieren. Dabei gaben alle Teilnehmer an, dass sie Symmetrie als Achsensymmetrie auffassen. Demnach bezogen sich die Angaben zur Bewertung der Symmetrie auf die Achsensymmetrie.

Auf Grundlage des ersten Eindrucks sollten die Symmetrie, die Buntheit bzw. Farbgebung und die visuelle Komplexität anhand einer 7-Punkte-Skala von 0 bis 6 beurteilt werden. Bei der Symmetrie bedeutet 0, dass die Benutzeroberfläche sehr asymmetrisch ist und 6, dass die Benutzeroberfläche eine hohe Symmetrie aufweist. Bei der Buntheit ist die Bewertungsskala so zu verstehen, dass 0 bedeutet, dass zu wenig Farben und 6, dass zu viele Farben genutzt wurden. Somit stellt 3 die Mitte und damit den idealen Grad zwischen zu vielen und zu wenigen Farben dar. Da die visuelle Komplexität als eine Vielzahl von Elementen auf einer Seite zu verstehen ist, bedeutet 0 in diesem Zusammenhang, dass zu wenig und 6, dass zu viele Elemente auf der Nutzeroberfläche zu sehen sind. Der Einfluss dieser drei Einflussfaktoren wird ebenfalls von 0 (negativ) bis 6 (positiv) bewertet. Da fünf Testpersonen an dieser Studie teilgenommen haben, liegen folglich fünf Bewertungen vor. Aus den jeweiligen Bewertungen wurde schließlich der arithmetische Mittelwert bestimmt und graphisch abgebildet (vgl. **Abbildung 6.1**).

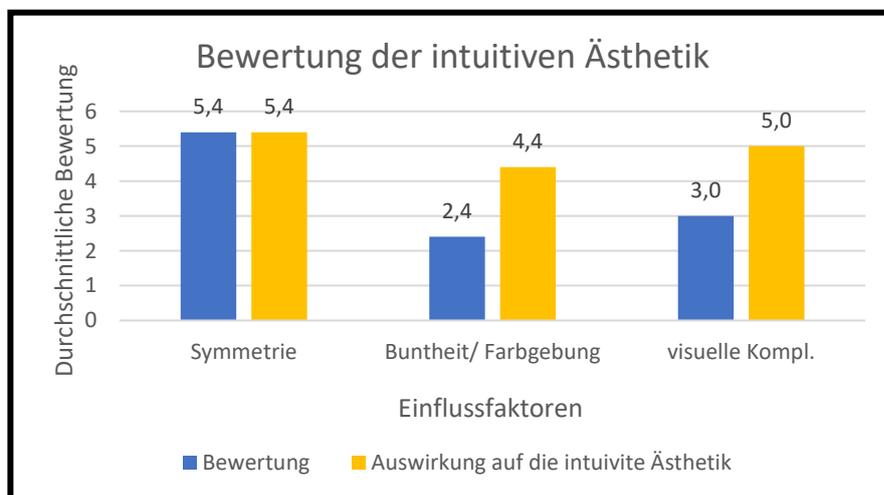


Abbildung 6.1: Bewertung der intuitiven Ästhetik

Es ist zu erkennen, dass eine hohe Symmetrie mit einer positiven Auswirkung der intuitiven Ästhetik einhergeht. Diese wurden jeweils mit 5,4 bewertet. Die Buntheit/ Farbgebung wurde mit 2,4 bewertet. Die Probanden empfanden die Benutzeroberfläche somit als zu farblos. Die visuelle Komplexität wurde im Durchschnitt mit einer 3,0 bewertet. Zudem ist zu erkennen, dass sich sowohl eine gemäßigte Buntheit als auch eine nicht zu hohe visuelle Komplexität positiv auf die intuitive Ästhetik auswirkt. Eine Bewertung der Buntheit von 2,4 führt im Durchschnitt auf eine Bewertung der intuitiven

Ästhetik von 4,4. Die visuelle Komplexität von 3,0 geht mit einer Bewertung von 5,0 in Bezug auf die Auswirkung auf die Ästhetik einher.

Im zweiten Teil des Versuchs wurde untersucht wie leicht oder schwer es den Teilnehmern fiel die Übungsaufgaben zu lösen. Die Schwierigkeit jeder Aufgabe konnte auf einer Skala von 1 bis 8 aufgetragen werden. Dabei wurde bereits im Vorhinein erwähnt, dass die Aufgabe 1a) mit 1 bewertet wird. Diese Aufgabe sollten die Probanden als Referenz bei der Bewertung der restlichen Aufgaben nehmen. Da die Teilnehmer die Möglichkeit hatten sich die Lösung anzeigen zu lassen, wird dies ebenfalls in die Analyse mit einbezogen. So kann besser nachvollzogen werden warum welche Aufgabe als schwierig empfunden wurde. Die Ergebnisse sind sehr unterschiedlich wie **Abbildung 6.2** zeigt.¹²

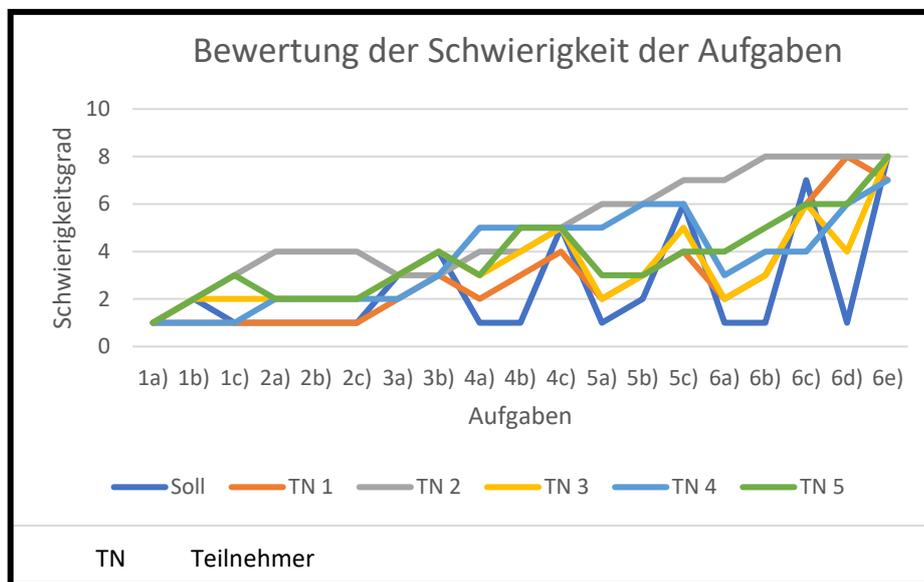


Abbildung 6.2: Bewertung der Schwierigkeit der Aufgaben

Der blaue Graph zeigt den „Soll“. Darunter ist zu verstehen, dass die Schwierigkeitsgrade der Übungsaufgaben unter dem Aspekt der in **Abbildung 4.1** dargestellten Schwierigkeitsgrade quantitativ abgebildet wurden. Jedoch sind die Unterschiede so zu erklären, dass die Teilnehmer unterschiedlich gute SQL-Kenntnisse hatten und somit einigen Teilnehmern gewisse Befehle wie der JOIN-Befehl oder das Arbeiten mit zwei Tabellen unbekannt waren. Dies spiegelt sich bei Aufgabe 6a) und 6d) wider. Während diese Aufgaben einfache Einführungsbeispiele zum Arbeiten mit zwei Tabellen und zum JOIN-Befehl darstellten, empfanden die Teilnehmer TN 1 und TN 2 diese Aufgaben als besonders schwierig, da sie diese nicht eigenständig lösen konnten. Da die Übungsaufgaben nach didaktischen Gesichtspunkten konzipiert wurden, wurde ein Augenmerk daraufgelegt, dass die Befehle aus vorigen Aufgaben regelmäßig

¹² In Abbildung 6.2 sind offensichtlich diskrete Punkte graphisch aufgetragen, wobei eine Interpolation zwischen diesen Punkten nicht zulässig ist. Jedoch dient die Interpolation in diesem, wie auch in den zukünftigen Fällen, lediglich dem Zweck eine bessere Lesbarkeit zu gewährleisten.

wiederholt und logisch in neuen Befehlen verknüpft werden konnten. Dies dient dem Zweck, dass sich das Gelernte besser im Gedächtnis verankert (vgl. Abschnitt 3.2 und 3.4). Zudem wurde bei der Erstellung der Aufgaben darauf geachtet, dass der Schwierigkeitsgrad auf die Entwicklungsstufe der Studierenden angepasst wird. Um zu prüfen inwieweit diese didaktischen Prinzipien greifen konnten, wurden zur besseren Übersicht die Graphen von TN 1 bis TN 5 zu einer Kurve zusammengefasst, indem der arithmetische Mittelwert der angegebenen Schwierigkeitsgrade bestimmt wurde (siehe **Abbildung 6.3**). Der Graph dieser Abbildung zeigt, dass der Schwierigkeitsgrad stetig zunimmt und Aufgabe 6e) im Durchschnitt als die schwierigste Aufgabe empfunden wird. Somit kann gesagt werden, dass das Prinzip der Stufenmäßigkeit (vgl. Abschnitt 3.4) im Rahmen dieser Übungsaufgaben greift.

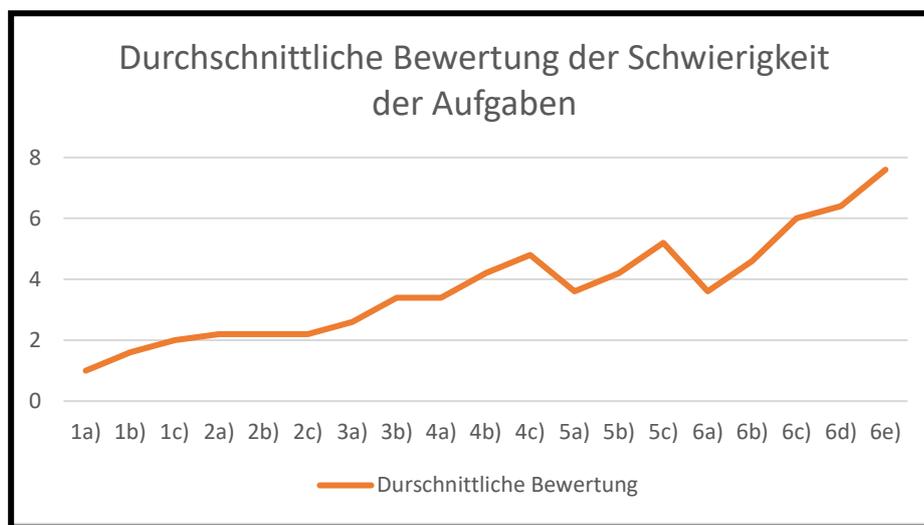


Abbildung 6.3: Durchschnittliche Bewertung der Schwierigkeit der Aufgaben

Darüber hinaus beurteilten die Probanden an, ob eine regelmäßige Wiederholung der Befehle aus vorigen Aufgaben stattgefunden hat und ob diese mit neuen Befehlen logisch verknüpft werden konnten. Die Bewertung wurde ebenfalls auf einer Skala von 0 (stimmt nicht) bis 6 (stimmt) durchgeführt. Die Antworten der Testpersonen sind **Abbildung 6.4** abgebildet.

Anhand dieser Grafik lässt sich entnehmen, dass die regelmäßige Wiederholung und logische Verknüpfung mit einer 6 und die Nützlichkeit der Hinweise mit 5,8 bewertet wurde. Somit lässt sich sagen, dass das Integrationsprinzip, das Prinzip der Stabilisierung (Abschnitt 3.2) und die Prinzipien des konstruktivistischen Lernens (Abschnitt 3.4) mit hohem Erfolg in die Übungsaufgaben eingepflegt wurden.

Nachdem sich die Probanden ausgiebig mit der Webapplikation beschäftigen konnten, wurde im zweiten Teil des Versuchs die reflektive Ästhetik bewertet. Dazu erhielten die Testpersonen die gleichen Fragen, die sie bei der intuitiven Ästhetik beantwortet haben.

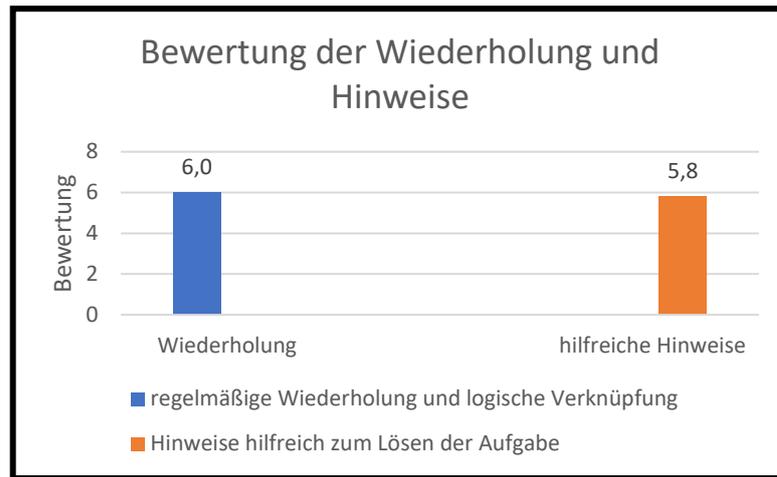


Abbildung 6.4: Bewertung der Wiederholung und Hinweise

Darüber hinaus bewerteten sie inwieweit die Ästhetik dazu beigetragen hat, dass die Nutzer Vergnügen bei der Nutzung der Webapplikation empfunden haben und inwiefern die Ästhetik die Bedienung der Webapplikation erleichtert hat. Die Bewertung der letzten beiden Fragestellungen wurde anhand einer Skala von 0 (stimmt nicht) bis 6 (stimmt) durchgeführt. Die Ergebnisse werden in **Abbildung 6.5** festgehalten.

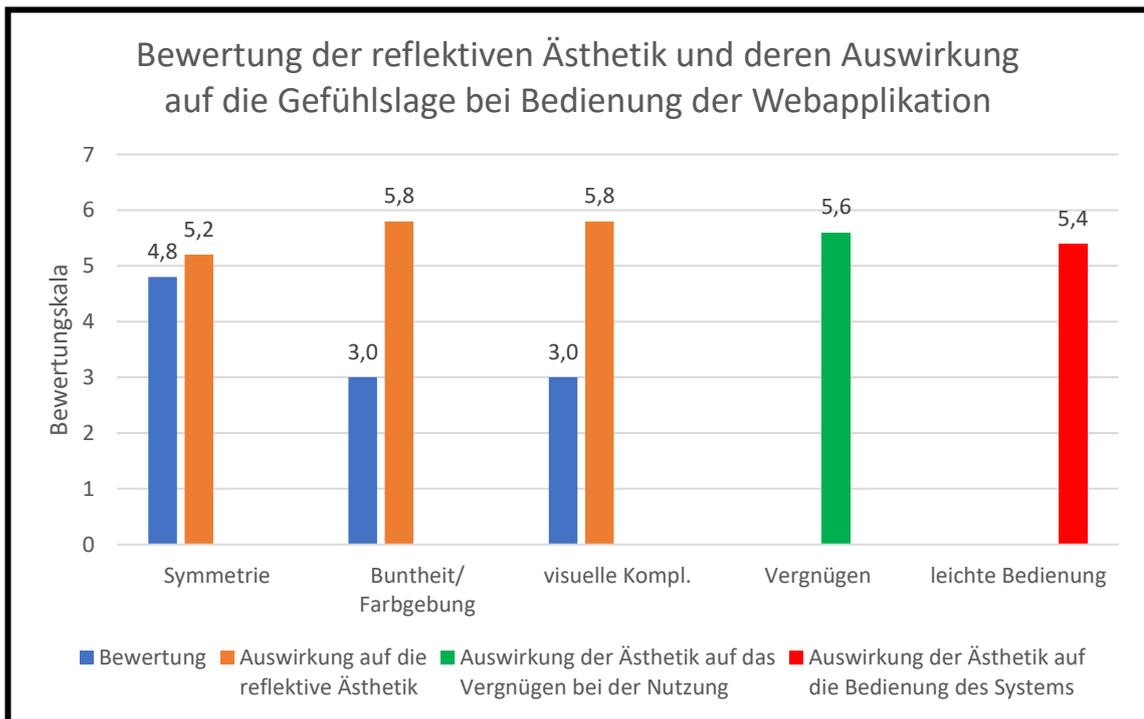


Abbildung 6.5: Bewertung der reflektiven Ästhetik und deren Auswirkung auf die Gefühlslage bei der Bedienung der Webapplikation

Dazu wurde erneut der arithmetische Mittelwert der von den Testpersonen angegebenen Bewertung errechnet und in die Grafik eingepflegt.

Wird die Bewertung der intuitiven Ästhetik mit der der reflektiven Ästhetik verglichen, sind Unterschiede zu erkennen. Zur Symmetrie lässt sich sagen, dass bei

der reflektiven Ästhetik die Symmetrie insgesamt schlechter bewertet wurde und somit die Ästhetik abnimmt. Dies lässt sich mit dem in Abschnitt 5.1.1 erwähnten Gestaltungsproblem erklären, da dieses Problem erst deutlich sichtbar wird, wenn der Nutzer die Webapplikation länger in Augenschein nimmt.

Die Buntheit/ Farbgebung wurde nach längerer Nutzung der Webapplikation deutlich besser bewertet. Während die visuelle Komplexität statt mit 2,4 mit 3,0 bewertet wurde, stieg die Auswirkung auf die intuitive Ästhetik von 4,4 auf 5,8. Bei der Untersuchung der intuitiven Ästhetik gaben die Probanden an, dass zu wenig Farben auf der Benutzeroberfläche zu sehen waren. Beschäftigen sie sich länger mit der Webapplikation, so erscheint an diversen Stellen die Farbe Rot, welche Fehlermeldungen signalisieren. Diese konnte bei der Bewertung der intuitiven Ästhetik nicht gesehen werden. Somit wurde die Anforderung nach einer höheren Buntheit im Rahmen der reflektiven Ästhetik erfüllt, wodurch diese folglich besser bewertet wurde.

Die visuelle Komplexität wurde zwar, wie bei der intuitiven Ästhetik, mit 3,0 bewertet, jedoch stieg die Bewertung der Auswirkung auf die Ästhetik von 5,0 auf 5,8. Daher muss untersucht werden, weshalb die visuelle Komplexität gleich, jedoch die Auswirkung auf die Ästhetik unterschiedlich bewertet wurde. Da bei der Auswertung der Ergebnisse mit dem arithmetischen Mittelwert gearbeitet wurde, gingen wichtige Informationen verloren. Die visuelle Komplexität bei der Untersuchung der reflektiven Ästhetik eindeutig mit 3,0 bewertet wurde, liegt eine Standardabweichung von 0,0 vor. Bei der intuitiven Ästhetik wurde die visuelle Komplexität zwar im Durchschnitt, aber nicht eindeutig von jedem mit 3,0 bewertet. Die Standardabweichung liegt hierbei bei 0,7. Deshalb wirken die Aussagen der Probanden zunächst widersprüchlich, sind allerdings statistisch zu erklären.

Darüber hinaus wurde die Auswirkung der Ästhetik auf das Vergnügen und auf die Erleichterung der Bedienung mit 5,6 respektive mit 5,4 bewertet. Da 6,0 die höchstmögliche Bewertung ist, lässt sich daraus schließen, dass sich die Gestaltung der Webapplikation in höchstem Grad positiv auf die UX auswirkt.

In der letzten Phase des Usability-Tests konnten die Probanden losgelöst von den Aufgabenstellungen die Funktionen der Webapplikation beliebig lange testen und sämtliche Auffälligkeiten dokumentieren. Daraufhin wurde geprüft, ob die formulierten Einwände gerechtfertigt sind und auf Basis dessen die Webapplikation überarbeitet. Die Kritiken wurden zusammengetragen, woraus sich ein Ranking ergeben hat. Aufgrund der Vielzahl der formulierten Einwände, werden im Folgenden die drei häufigsten vorgestellt und näher erläutert. Eine Übersicht über die Beanstandungen der Testpersonen und deren absolute Häufigkeit bietet **Tabelle 6.1**.

Der Einwand „Passwort falsch eingegeben → keine Fehlermeldung“ stellt ein Error-Handling-Problem dar (vgl. Abschnitt 2.1). Hat der Nutzer bei der Anmeldung sein Passwort fehlerhaft eingegeben und anschließend versucht sich anzumelden, so erscheint keine Fehlermeldung, die ihm mitteilt, dass das Passwort falsch ist. Lediglich ist nach versuchter Anmeldung das Anmeldefenster zu sehen. Nun hat der Nutzer das

Problem, dass er nicht versteht, dass das Passwort falsch eingegeben wurde und ist gezwungen selbst herauszufinden, warum die Anmeldung nicht durchgeführt wird.

Ein weiterer häufiger Einwand ist „Aufgabe 5c) AVG wird nicht grün hinterlegt“. Dies ist ein Usability-Problem, welches aufgrund von nicht einheitlicher Programmierung der SQL-Funktionen zustande gekommen ist. Der Nutzer stellt sich bereits früh darauf ein, dass die SQL-Funktionen kursiv und in grüner Schrift geschrieben werden, was auch als Kontrolle genutzt werden kann, um syntaktische Fehler zu umgehen. Jedoch wird die Funktion AVG nicht grün hinterlegt, sodass der Nutzer vermuten kann diesen Befehl falsch eingegeben zu haben, obwohl dies nicht der Fall ist.

Tabelle 6.1: Einwände der Probanden während des Usability-Tests

Einwand	absolute Häufigkeit
Passwort falsch eingegeben → keine Fehlermeldung	3
Aufgabe 5c) AVG wird nicht grün hinterlegt	3
Aufgabe 6c) Sortierung ORDER BY falschherum	3
Aufgabe 5b) Sortierung ORDER BY falschherum	2
JOIN in Aufgabe 6e) erwähnen. Aufgabe kann bei aktueller Formulierung auch ohne JOIN gelöst werden.	2
Registrierung mit „. @tu-dortmund.de“ funktioniert	1
Bei semantischen Fehlern, die nicht in einer leeren Ausgabe enden, verschwindet die getätigte Eingabe → Aufwand wird erhöht	1
Aufgabe 6: Namen der Tabellen zu lang → unnötiger Aufwand	1
Aufgabe 6c): Reihenfolge Vorname, Nachname vertauscht	1
Schrift könnte größer sein	1
Aufgabe 5c) Aufgabenstellung kann wegen „c“ und „C“ irreführend sein	1

Der letzte Einwand, den drei von fünf Teilnehmern vorgebracht haben, ist „Aufgabe 6c) Sortierung ORDER BY falschherum“, welcher in der Formulierung der Aufgabenstellung begründet ist. Die Aufgabe verlangt lediglich, dass die Ausgabetablelle in alphabetischer Reihenfolge nach dem Nachnamen sortiert werden soll. In der Lösung ist die Sortierung jedoch in absteigender alphabetischer Reihenfolge nach dem Nachnamen hinterlegt. So war es für den Anwender nicht möglich die Aufgabe zu lösen, auch wenn in der SQL-Eingabe jeder Aspekt der Aufgabenstellung beachtet wurde.

Alle Einwände, welche in **Tabelle 6.1** aufgelistet wurden, wurden berücksichtigt und die Webapplikation dementsprechend korrigiert.

6.3 Fazit

In diesem Abschnitt werden die Ergebnisse dieser Arbeit zusammengeführt. Dabei wird geprüft, ob die Ziele dieser Arbeit erfüllt worden sind. Das Oberziel dieser Arbeit ist die

Konzeption und Entwicklung einer datenbankbasierten Webapplikation gewesen. Diese soll als digitales Lernwerkzeug zum Thema SQL dienen und im Übungsbetrieb des Masterstudiums am Lehrstuhl IT in Produktion und Logistik der Technischen Universität Dortmund eingesetzt werden. Dieses Lernwerkzeug soll zudem unter der Berücksichtigung didaktisch fundierter Gesichtspunkte konzipiert werden. Die Webapplikation sollte dabei mehrbenutzerfähig sein, sodass die Studierenden eigene SQL-Befehle eingeben und ausführen können ohne dabei die Tabellen und Aufgaben der anderen simultan arbeitenden Nutzern zu manipulieren. Das Forschungsthema ist in einer Arbeitsgruppe behandelt worden. Dabei ist die Konzeption und Entwicklung des Frontends der Webapplikation Gegenstand dieser Arbeit. Die Konzeption und Entwicklung des Backends, ist in der Arbeit von D. Alić thematisiert worden (Alić, 2021). Zur Erfüllung des Oberziels mussten zwei Unterziele erfüllt werden. Das erste Unterziel war die Erstellung von geeigneten Übungsaufgaben, die die Studierenden in Zukunft nutzen können, um im Rahmen des asynchronen Lernens die Sprache SQL zu erlernen. Dazu mussten zunächst verschiedene didaktische und psychologische Modelle miteinander verglichen werden, um die ideale didaktisch fundierte Basis zur Erstellung der Übungsaufgaben zu schaffen. Es wurden die didaktischen Prinzipien nach Baumann unter Berücksichtigung der psychologischen Definition von „Motivation“ und der Grundlagen des kognitivistischen bzw. konstruktivistischen Lernen herangezogen.

Das zweite Unterziel war die Gestaltung der Oberfläche der Webapplikation. Um eine ideale Darstellung der Weboberfläche zu gewährleisten, wurden zum einen ebenfalls auf didaktische Grundlagen, aber auch auf psychologische Grundlagen zur sogenannten Ästhetik und deren Einflussfaktoren zurückgegriffen. Die Ästhetik ist maßgebend für das UI und UX der Webapplikation.

Zur Entwicklung des Frontends der Webapplikation mussten verschiedene Module entwickelt und die Kommunikation mit dem Backend gewährleistet werden. Bei den Modulen handelt es sich unter anderem um die Funktionen Registrierung, Login und die Prüfung der Eingabe.

Die Ergebnisse aus beiden Unterzielen wurden zusammengefügt, um die Oberfläche der Webapplikation zu entwickeln. Inwieweit die berücksichtigten didaktischen Modelle auf die Übungsaufgaben angepasst wurden und wie ästhetisch der Nutzer die Webapplikation empfindet, wurde im sogenannten Usability-Test untersucht. Dieser bestand aus drei Teilen, in welcher die intuitive Ästhetik, die reflektive Ästhetik, der Schwierigkeitsgrad der Aufgaben, die Gefühlslage der Nutzer während der Benutzung und zudem die Existenz von Fehlern untersucht wurden.

Der Test der intuitiven Ästhetik hat gezeigt, dass die Nutzer die Benutzeroberfläche nach bereits 500ms unter dem Aspekt der Symmetrie und der visuellen Komplexität als sehr ansprechend wahrnehmen. Auch wenn die Ästhetik der Symmetrie bei längerer Nutzung der Webapplikation negativer bewertet wurde als beim ersten Test, wirken sich die Buntheit und die visuelle Komplexität auf die Ästhetik positiver aus. Diese Ästhetik ermöglichte es die Nutzer großes Vergnügen bei der Nutzung der Webapplikation empfinden zu lassen und erleichterte diesen zudem die

Bedienung des Systems sehr. So lässt sich insgesamt sagen, dass die Benutzeroberfläche als sehr attraktiv und schön von den Anwendern empfunden wird.

Des Weiteren wurde getestet, ob und inwiefern die didaktischen Prinzipien, nach denen die Übungsaufgaben erstellt wurden, in die Webapplikation eingepflegt wurden. Der Usability-Test ergab, dass eine Aufgabe von mehreren Teilnehmern als unterschiedlich schwer eingestuft werden. Dies hat den Hintergrund, dass die Probanden unterschiedlich gute SQL-Kenntnisse haben. Eine Aufgabe, die für einen Probanden als leicht empfunden wurde, war gleichzeitig für einen anderen Probanden unlösbar. Jedoch zeigt sich, dass alle Nutzer die Steigerung des Schwierigkeitsgrades der Aufgaben gleichermaßen wahrgenommen haben. Dadurch wurde dem Prinzip der Stufenmäßigkeit (vgl. Abschnitt 3.4) nach Baumann Folge geleistet. Zudem waren sich alle Teilnehmer einig, dass eine regelmäßige Wiederholung zuvor gelernter Befehle, die logisch mit neuen Aufgaben verknüpft werden konnten, vorhanden ist. Somit wurde ebenfalls das Prinzip der Stabilisierung, das Integrationsprinzip (vgl. Abschnitt 3.4) und zudem die Grundlagen des konstruktivistischen Lernens (vgl. Abschnitt 3.2) erfolgreich in die Webapplikation eingepflegt. Darüber hinaus gaben die Teilnehmer mit einer nahezu hundertprozentigen Quote an, dass die Hinweise, die die Applikation ihnen bietet, als sehr hilfreich angesehen werden. Dadurch konnte bestätigt werden, dass die didaktischen Grundlagen während der Bearbeitung der Aufgaben hochgradig gegriffen haben.

Darüber hinaus wurden von den Testpersonen insgesamt elf Usability-Probleme bzw. Systemfehler entdeckt (vgl. **Tabelle 6.1**). Laut Nielsen (Nielsen, 1993) werden bei einem Test mit fünf Testpersonen im Durchschnitt 80% der Usability-Probleme aufgedeckt (vgl. Abschnitt 2.3.4). Da in diesem Fall elf Probleme aufgedeckt wurde, kann davon ausgegangen werden, ca. zwei bis drei weitere Probleme noch unentdeckt geblieben sind.

7 Zusammenfassung und Ausblick

Im Fokus dieser Masterarbeit steht die Entwicklung des Frontends einer datenbankbasierten Webapplikation. Diese Applikation soll als digitales Werkzeug zum asynchronen Lernen der Abfragesprache SQL dienen und im Übungsbetrieb des Masterstudiums am Fachgebiet IT in Produktion und Logistik der Technischen Universität Dortmund zum Einsatz kommen. Dabei wurde das Backend der Webapplikation von D. Alić (Alić, 2021) konzipiert und entwickelt.

Hierzu wurden zunächst im zweiten Kapitel die Grundlagen, welche zur Gestaltung und Entwicklung von Weboberflächen vorgestellt. Dazu zählen Grundlagen zu HTML, CSS und zur UI und UX insbesondere in Bezug auf die Ästhetik einer Weboberfläche. Ferner wurden in diesem Kapitel die Softwaretest nach dem V-Modell und zudem der Usability-Test vorgestellt, welche genutzt wurden, um durch verschiedene Tests Fehler und Probleme aufzudecken und um sie anschließend zu beheben. Da die SQL-Übungsaufgaben innerhalb der Webapplikation nach didaktisch fundierten Modellen und Prinzipien zu erstellen waren, wurden im dritten Kapitel verschiedene didaktische Theorien und Modelle vorgestellt. Darunter der Kognitivismus, der Konstruktivismus, die didaktischen Theorien nach Klafki und den Prinzipien der Didaktik in der Informatik nach Baumann. Die einzelnen SQL-Befehle werden in Abschnitt 2.2.3 der Arbeit von D. Alić vorgestellt. Daraufhin fand im vierten Kapitel ein Vergleich dieser Modelle statt, um herauszufinden, welche Grundsätze der Didaktik und Psychologie am besten auf die vorliegende Aufgabenstellung angewandt werden können. Als Grundlage wurden die Prinzipien nach Baumann unter Berücksichtigung der psychologischen Definition von Motivation und der Grundsätze des Kognitivismus und des Konstruktivismus herangezogen. Auf Basis dessen wurden anschließend geeignete Übungsaufgaben erstellt, um diese in die Webapplikation einzupflegen. Schließlich wurde die Entwicklung des Frontends im fünften Kapitel thematisiert. Dem Leser wurden die Funktionen der Webapplikation anhand der Benutzeroberfläche und der einzelnen Module, aus denen sich die Webapplikation zusammensetzt, erörtert. Bei den Modulen handelt es sich unter anderem um die Funktionen Registrierung, Login und die Prüfung der Eingabe. Neben dem technischen Hintergrund der Webapplikation wird dem Leser beschrieben welche Aspekte des User-Interfaces und der User-Experience bei der Gestaltung der Benutzeroberfläche berücksichtigt wurden.

Im Rahmen des sechsten Kapitels wurde der sogenannte Usability-Test mit fünf Probanden, die den künftigen Nutzerkreis möglichst gut repräsentieren, durchgeführt. Der ebenfalls in Kapitel zwei vorgestellte Komponententest, Integrationstest und Systemtest wurden in Abschnitt 4 der Arbeit von D. Alić (Alić, 2021) durchgeführt. Der Usability-Test verfolgte zum einen das Ziel zu bewerten inwieweit sich die Gestaltung der Benutzeroberfläche der Webapplikation auf die Gefühlslage der Nutzer auswirkt. Zudem wurde untersucht, wie gut die didaktischen Prinzipien, auf denen die

Übungsaufgaben basieren, in die Webapplikation eingepflegt werden konnten. Abschließend galt der Usability-Test der Auffindung von Fehlern und von Usability-Problemen. Der Test der intuitiven Ästhetik ergab, dass die Nutzer die Webapplikation auf den ersten Blick als ansprechend empfunden haben. Die reflektive Ästhetik wurde nach längerer Nutzung deutlich besser bewertet als die intuitive Ästhetik. Dies sorgte dafür, dass die Anwender unabhängig von den Aufgabenstellungen in großem Maße Vergnügen bei der Nutzung empfunden haben und ihnen durch Ästhetik die Bedienung der Applikation erleichtert wurde.

Die Studie bestätigte zudem, dass die didaktischen Prinzipien insofern greifen konnten, als dass den Benutzern bei der Bearbeitung der Aufgaben hilfreiche Hinweise zur Verfügung gestellt wurden. Darüber hinaus wurden laut den Aussagen der Probanden die Übungsaufgaben so gestaltet, dass eine regelmäßige Wiederholung der zuvor gelernten SQL-Befehle vorliegt und die gelernten Befehle logisch mit weiteren Aufgaben verknüpft werden konnten. Dies spricht dafür, dass die berücksichtigten didaktischen Modelle sehr gut auf die Übungsaufgaben angewandt wurden und somit von einem großen Lernerfolg der Studierenden ausgegangen werden kann.

Des Weiteren war das Auffinden von Fehlern und von Usability-Problemen Gegenstand dieses Tests. Es wurden insgesamt elf gerechtfertigte Einwände von den Testpersonen formuliert, welche auf Fehler bzw. Probleme hinweisen. Diese wurden anschließend innerhalb der Webapplikation behoben. Da laut Nielsen fünf Personen im Durchschnitt 80% der Usability-Probleme aufdecken (Nielsen, 1993), kann davon ausgegangen werden, dass zwei bis drei Probleme noch unentdeckt geblieben sind. So empfiehlt es sich laut Faulkner künftig 20 oder mehr Testpersonen in den Usability-Test mit einzubeziehen, da auf diese Weise mindestens 98,4 Prozent der Usability-Probleme aufgedeckt werden können. Je mehr Testpersonen teilnehmen, desto mehr Fehler können aufgedeckt werden (Faulkner, 2003).

Die von D. Alić durchgeführten Tests deckten durch 212 Testfälle diverse Fehler auf, sodass der Programmcode dementsprechend überarbeitet wurde. Des Weiteren zeigte D. Alić in seiner Arbeit, dass die Webapplikation bei 200 gleichzeitig arbeitenden Nutzern problemlos arbeitet. Erst bei 5.000 gleichzeitig agierenden Nutzern ist die Belastungsgrenze erreicht. Somit sind die Anforderungen im Hinblick auf die Nutzung im Masterstudium des Fachgebiets IT in Produktion und Logistik in jeder Hinsicht erfüllt. Unter Berücksichtigung der Tatsache, dass seit März 2020 an sämtlichen Universitäten die Durchführung von Präsenzveranstaltungen durch die COVID-19-Pandemie nahezu unmöglich ist, ist es umso wichtiger auf computergestützte Lernwerkzeuge zurückzugreifen. Diese Arbeit leistete in Zusammenarbeit mit D. Alić einen Beitrag zum digitalen asynchronen Lernen der Abfragesprache SQL, um der Technischen Universität die Anpassung an die aktuellen pandemiebedingten Maßnahmen zu erleichtern. Durch die im Rahmen dieser Arbeit entwickelte Webapplikation wird jedoch lediglich ein Themengebiet abgedeckt. Dennoch können große Teile des Programmcodes für die Entwicklung weiterer Webapplikationen verschiedener Themengebiete genutzt werden. So können die Darstellung, das Benutzermanagement und das Aufgabenmanagement

der Webapplikation übernommen und erweitert werden. Eine sinnvolle Erweiterung wäre zum Beispiel eine dynamische Aufgabenverwaltung, welche es Dozenten ermöglicht, neue Aufgabenstellungen zu definieren, ohne den Programmcode anzupassen. Eine weitere denkbare Erweiterung der Webapplikation wäre ein Prüfungsmodus. Solch ein Prüfungsmodus könnte die Prüfungsbewertung in kurzer Zeit automatisiert durchführen.

Neben den aufgeführten Erweiterungsmöglichkeiten existiert noch eine Vielzahl an Möglichkeiten für den Einsatz von computergestützten Lernwerkzeugen.

Literaturverzeichnis

- (Alić, 2021) Alić, D. (2021).: *Backend zu einer datenbankbasierten Webapplikation für den Einsatz in der digitalen Lehre*, Masterarbeit am Lehrstuhl für IT in Produktion und Logistik, Technische Universität Dortmund (2021)
- (Baumann, 1996) Baumann, R. (1996).: *Didaktik der Informatik*. Klett-Schulbuchverlag 1990.
- (Baumgartner et al., 2009) Baumgartner, M., Sneed, H. & Seidl, R. (2009).: *Der Systemtest – von den Anforderungen zum Qualitätsnachweis* (2. Aktualisierte und erweiterte Auflage). Hanser Verlag München 2009
- (Boekarts, 2015) Boekarts, M. (2015).: *Die entscheidende Rolle von Motivation und Emotion im Unterricht*. Aus: *The Nature Of Learning – Forschungsergebnisse für die Praxis* (S. 101-120). Beltz: Weinheim/ Basel
- (Bühler et al., 2017) Bühler, P., Schlaich, P. & Sinner, D. (2017).: *HTML5 und CSS3 – Semantik, Design, Responsive Layouts*. Springer-Verlag GmbH Deutschland 2017
- (Brau & Sarodnick, 2016) Brau, H. & Sarodnick, F. (2016).: *Methoden der Usability Evaluation – wissenschaftliche Grundlagen und praktische Anwendung* (3., unveränderte Auflage). Hogrefe Verlag Bern 2016
- (Chlebek, 2006) Chlebek, P. (2006).: *User Interface-orientierte Softwarearchitektur – Bauentwurfslehre für interaktive Softwareoberfläche, Kompass für die Entwicklung dialogintensiver Anwendungen, Leitfaden für erlebbare User Interfaces*. Friedr. Vieweg & Sohn Verlag, GWV Fachverlage GmbH, Wiesbaden 2006

- (Chlebek, 2011) Chlebeck, P. (2011).: *Praxis der User Interface-Entwicklung – Informations-strukturen, Designpatterns, Vorgehensmuster*. Vieweg+Teubner Verlag, Springer Fachmedien Wiesbaden GmbH, 2011
- (Denecke, 2002) Denecke, M. (2002).: *Generische Interaktionsmuster für aufgabenorientierte Dialogsysteme*. Dissertation der Fakultät für Informatik an der Universität Karlsruhe (Technische Hochschule), 2002
- (Ebner, 1999) Ebner, M. (1999).: *SQL lernen* (1. Aufl.). Adisson-Wesler-Longman, Bonn
- (Eigner et al., 2012) Eigner, M., Gerhardt, F., Gilz, T. & Mogo Nem, F. (2012).: *Informationstechnologie für Ingenieure*. Springer-Verlag Berlin Heidelberg 2012
- (Faulkner, 2003) Faulkner, L. (2003).: *Beyond The Five-User Assumption: Benefits Of Inceades Sample Sizes In Usability Testing*. Aus: *Behavior Research Method, Instrument & Computers* 2003, 35(3), S. 379-383
- (Faulstich, 2013) Faulstich, P. (2013).: *Menschliches Lernen – Eine kritisch-pragmatistische Lerntheorie* (2016). transcript Verlag, Bielefeld.
- (Flanagan, 2006) Flanagan, D. (2006).: *JavaScript – The Definitive Guide* (5. Aufl.). O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA. August 2006
- (Fuchs, 2018) Fuchs, E. (2018).: *SQL – Grundlagen und Datenbankdesign*. Technische Universität Dortmund in Kooperation mit dem HERDT-Verlag

- (Gieselmann, 2007) Gieselmann, P. (2007).: *Fehlerbehandlung in Mensch-Maschine-Dialogen*. Dissertation am Institut für maschinelle Sprachverarbeitung, Universität Stuttgart, 2007
- (Greching et al., 2010) Grechenig, T., Bernhart, M. & Breitender, R. (2010).: *Softwaretechnik – Mit Fallbeispielen aus realen Entwicklungsprozessen*. Pearson Studium München 2012
- (Grosch, 2018) Grosch, M. (2018).: *Digitale Lernszenarien – Konzept, tutorielle Betreuung und gemeinsame Lernprozesse lassen digitales Lernen lebendig werden*. Aus PADUA (2018), 13 (2), S. 115-117
- (Haunschild, 2017) Haunschild, M. (2017).: *CSS – Cascading Style Sheets – Level 3, Grundlagen* (3. Ausgabe). HERDT-Verlag in Kooperation mit der Technischen Universität Dortmund, Dezember 2017
- (Hassenzahl, 2002) Hassenzahl, M. (2002).: *The effect of perceived hedonic quality on product appealingness*. Aus: *Journal of Human-Computer Interaction*, 13(4), S. 481-499. Lawrence Erlbaum Associates Inc. 2002
- (Herczeg, 2018) Herczeg, M. (2018).: *Software-Ergonomie – Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme* (4.Aufl.). Walter de Gruyter GmbH, Berlin/Boston, 2018
- (Hermann & Temath, 2019) Hermann, K. & Temath, C. (2019).: *Digitalisierung in der wissenschaftlichen Weiterbildung*. Projekt „QUP – Qualifizierung – Unterstützung – Professionalisierung zur Gestaltung des demografischen Wandels“ der Universität Potsdam. In Kooperation mit dem Bundesministerium für Bildung und Forschung

- (Hoblitz, 2015) Hoblitz, A. (2015).: *Spielend Lernen im Flow – Die motivationale Wirkung von Serious Games im Schulunterricht*. Dissertation Universität Paderborn. Springer Fachmedien Wiesbaden 2015
- (Hoffmann, 2009) Hoffmann, B. (2009).: *Bildungstheoretische Didaktik – Wolfgang Klafki und die Didaktische Analyse*. Universität Trier 2008
<https://www.uni-trier.de/fileadmin/fb1/prof/PAD/BW2/Hoffmann/Bildungstheoret.Didaktik.pdf>
zuletzt aufgerufen: 30.11.2020
- (Hommel, 2019) Hommel, M. (2019).: *Lernhandeln mit integrierter Unternehmenssoftware*. Habilitationsschrift – TU Dresden.
- (Hubweiser, 2007) Hubweiser, P. (2007).: *Didaktik der Informatik – Grundlagen, Konzepte, Beispiele* (3., überarbeitete und erweiterte Auflage). Springer- Verlag Berlin Heidelberg.
- (Ilmberger et al., 2009) Ilmberger, H., Held, T. & Schrepp, M. (2009).: *Was macht studiVZ attraktiv?* Aus: Wandke, H., Kain, S. & Struve, D.: *Mensch & Computer, 9. Fachübergreifende Konferenz für interaktive und kooperative Medien – Grenzenlos frei?* Oldenburger Verlag München, 2009
- (ISTQB, 2020) ISTQB (2020).: <https://www.istqb.org/>
- (Jacko et al., 2003) Jacko, J., Scott, I., Sainfort, F., Moloney, K., Kongnakorn, T., Zorich, B. & Emery, V. (2003).: *Effects of Multimodal Feedback on the Performance of Older Adults with Normal and Impaired Vision*. Aus: *Universal Acces – Theoretical Perspectives, Practice and Experience. 7th ERCIM International Workshop on User Interfaces for All. Paris, France, October 2002*. Springer-Verlag Berlin Heidelberg 2003
- (Jank & Meyer, 2019) Jank, W. & Meyer, H. (2019).: *Didaktische Modelle*. Cornelsen Berlin 2019

- (Kerres et al., 2009) Kerres, M., Ojstersek, N. & Stratmann, J. (2009).: *Didaktische Konzeption von Angeboten des Online-Lernens*. Aus: Issing, L.J. & Klimsa, P. (Hrsg.). (2009).: *Online-Lernen – Handbuch für das lernen mit Internet*. Oldenburg Verlag
- (Klafki, 1958) Klafki, W. (1958).: *Fünfte Studie: Didaktische Analyse als Kern der Unterrichtsvorbereitung*. Aus: *Studien zur Bildungstheorie und Didaktik* in der Zeitschrift *Die Deutsche Schule* (1958), H.10. S. 450-471
- (Klafki, 1980) Klafki, W. (1980).: *Die bildungstheoretische Didaktik im Rahmen kritisch- konstruktiver Erziehungswissenschaft zur Neufassung der didaktischen Analyse*. Aus: *Westermanns pädagogische Beiträge*, Jg. 32 (1980), Heft 1, S. 32-37
- (Krapp & Hascher, 2014) Krapp, A. & Hascher, T. (2014).: *Theorien der Lern- und Leistungsmotivation*. Springer Verlag: Berlin/ Heidelberg
- (Kribernegg, 2013) Kribernegg, A. (2013).: *Software-Test IT Professionally*. Cyber Publishing E-Verlag GmbH, Wien 2013
- (Lavie & Tractinsky, 2003) Lavie, T. & Tractinsky, N. (2003).: *Assessing dimensions of perceived visual aesthetics of web sites*. Aus: *International Journal of Human-Computer Studies* 60 (2004), S. 269-298
- (Lehner, 2019) Lehner, M. (2019).: *Didaktik*. Haupt Verlag Bern (2019)
- (Lennard & Surkau, 2011) Lennard, H. & Surkau, M. (2011).: *Benutzerevaluation und Usability-Test zur neuen Suchoberfläche Primo (Ex Libris)*. Masterarbeit im Rahmen des weiterbildenden Fernstudiums Master of Arts (Library an Information Science). Humboldt-Universität Berlin, 01.06.2011
- (Liebig et al., 2014) Liebig, H., Flik, T., Rechenberg, P., Reinefeld, A. & Mössenböck, H. (2014).: *Das Ingenieurwissen – Technische Informatik*. Springer Verlag Berlin Heidelberg 2014

- (Liggesmeyer, 2009) Liggesmeyer, P. (2009).: *Software-Qualität – Testen, Analysieren und Verifizieren von Software* (2. Auflage). Spektrum Akademischer Verlag Heidelberg 2009.
- (Link & Fröhlich, 2002) Link, J. & Fröhlich, P. (2002).: *Unit Test mit Java: Der Test-First-Ansatz*. dpunkt.verlag Heidelberg 2002
- (Mellis et al., 2003) Avci, O., Trittmann, R. & Mellis, W. (Hrsg.) (2003).: *Webprogrammierung – Software mit Internettechnologien, Grundlagen, Auswahl, Einsatz, XHTML, HTML, CSS, XML, JavaScript, VBScript, PHP, ASP, Java*. Friedr. Vieweg & Sohn Verlag / GWV Fachverlage GmbH, Wiesbaden. Dezember 2003.
- (Meyer & Stocker, 2018) Meyer, R. & Stocker, F. (2018).: *Lehren Kompakt I: Von der Fachperson zur Lehrperson* (5. Aufl.). hep verlag
- (Miniukovich & De Angeli, 2015) Miniukovich, A. & De Angeli, A. (2015).: *Computation of Interface Aesthetics*, Conference: the 33rd Annual ACM Conference, April 2015
- (Modrow & Strecker, 2016) Modrow, E. & Strecker, K (2016).: *Didaktik der Informatik*. Walter de Gruyter GmbH, Berlin/ Boston (2016).
- (Moshagen & Musch, 2009) Moshagen, M. & Musch, J. (2009).: *A blessing, not a curse: experimental evidence for beneficial effects of visual aesthetics on performance*. Aus: *Ergonomics* 52(10), S. 1311-1320, Oktober 2009
- (Moser, 2012) Moser, C. (2012).: *User Experience Design – Mit erlebniszentrierter Softwareentwicklung zu Produkten, die begeistern*. Springer-Verlag Berlin Heidelberg 2012
- (Müller-Radtke, 2018) Müller-Radtke, J. (2018).: *Emotionale Gestaltung multimedialer Lernumgebungen – der Einfluss visueller Ästhetik und Usability auf affektives Erleben, Motivation und Lernerfolg*. Berlin, wvb – Wissenschaftlicher Verlag Berlin, 2018

- (Nielsen, 1993) Nielsen, J. (1993).: *Usability Engineering*. Academic Press, Harcourt Place, 32 Jamestown Roa, London, NW1 7BY, UK
- (Oxford, 2016) University of Oxford: *Sir Tim Berners-Lee joins Oxford's Department of Computer Science*.
<https://www.ox.ac.uk/news/2016-10-27-sir-tim-berners-lee-joins-oxfords-department-computer-science>
Veröffentlicht: 27. Oktober 2016
Zuletzt aufgerufen: 05. November 2020
- (Pasternak & Vahrenhold, 2009) Pasternak, A. & Vahrenhold, J. (2009).: *Rote Fäden und Kontextorientierung im Informatik Unterricht*. Technische Universität Dortmund, Fakultät für Informatik
- (PELe, 2006) Redaktionsteam PELe (2006).: *Didaktische Modelle*. Stand: 19.07.2006
- (Pomaska, 2005) Pomaska, G. (2005).: *Grundkurs Web-Programmierung – Interaktion, Grafik und Dynamik, mit XHTML und CSS, XML, JavaScript, Applets, SVG, PHP*. Friedr. Vieweg & Sohn Verlag / GWV Fachverlage GmbH, Wiesbaden. Januar 2005.
- (Potsdam, 2020) Universität Potsdam, Didaktik der Informatik. Übersicht zur Lehrveranstaltung „Didaktik der Informatik I“ aus dem Sommersemester 2020.
Zuletzt aufgerufen: 15.12.2020
<http://ddi.cs.uni-potsdam.de/Lehre/ddi1>
- (Reinecke & Gajos, 2014) Reinecke, K. & Gajos, K. (2014).: *Quantifying Visual Preferences Around the World*. Aus: *Proceeding oft the SIGCHI Conference on Human Factors in Computing Systems*, Toronto, Canada, 26. April 2014 – 01. Mai 2014, S. 11-20
- (Riedel, 2019) Riedel, R. (2019).: *Management von Informatik-Projekten – Digitale Transformation erfolgreich gestalten (2., vollständig überarbeitete Auflage)*. Walter De Gruyter GmbH, Berlin/ Boston 2019

- (Sauer & Sonderegger, 2009) Sauer, J. & Sonderegger, A. (2009).: *The influence of prototype fidelity and aesthetics of design in usability tests: Effects on user behaviour, subjective evaluation and emotion*. Aus: *Applied Ergonomics* 40 (2009), S. 670-677
- (Schlag, 2013) Schlag, B. (2013).: *Lern- und Leistungsmotivation* (4. überarbeitete und aktualisierte Aufl.). Springer Fachmedien Wiesbaden
- (Schmidt & Wolff, 2017) Schmidt, T., & Wolff, C. (2017).: *Der Einfluss von User Interface-Attributen auf die Ästhetik*. Veröffentlicht durch die Gesellschaft für Informatik e. V. 2017 in Burghardt, M., Wimmer, R. & Wolff, C. & Womer-Hacker, C (Hrsg.): *Mensch und Computer 2017 – Tagungsband, 10.-13. September 2017, Regensburg*
- (Schnirch et al., 2020) Schnirch, A., Ridinger, N. & Weschenfelder, F. (2020).: *Raspberry Pi im Informatik und Technikunterricht – Konzeption eines handlungs- und problemorientierten Unterrichts mit der MicroBerry-Lernumgebung*. Springer Fachmedien Wiesbaden GmbH
- (Schubert & Schwill, 2011) Schubert, S. & Schwill, A. (2011).: *Didaktik der Informatik* (2. Aufl.). Spektrum Akademischer Verlag Heidelberg
- (Seckler & Tuch, 2012) Seckler, M. & Tuch, A. (2012).: *Linking objektive web-design factors to facets of subjective aesthetic perception*. Aus: *NordiCHI '12: Proceeding off he 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Desing*, S. 809-810, Oktober 2012
- (Shneidermann et al., 2017) Shneidermann, B., Plaisant, C., Cohen, M., Jacobs, S. & Elmqvist, N. (2017).: *Designing the User Interface: Strategies for Effective Human-Computer Interaction, Global Edition*. Pearson Higher Education Verlag, London 2017
- (Spillner & Linz, 2019) Spillner, A. & Linz, T. (2019).: *Basiswissen-Softwaretest* (6. überarbeitete Auflage).dpunkt.verlag GmbH Heidelberg 2019

- (Springer, 2013) Springer, S. (2013).: *Error Handling in JavaScript*
<https://entwickler.de/online/error-handling-in-javascript-137091.html>
Veröffentlicht am: 30. September 2013
Zuletzt aufgerufen am: 11. November 2020
- (Statista, 2019) Statista (2019): *Studenten in Deutschland nach Altersgruppen im Vergleich mit der Bevölkerung im Jahr 2019*
<https://de.statista.com/statistik/daten/studie/860069/umfrage/umfrage-in-deutschland-zum-alter-der-studenten/>
Veröffentlicht am: November 2019
Zuletzt aufgerufen am: 24. Dezember 2020
- (Statista, 2020a) Statista (2020).: *Monatliche Umsatzentwicklung im Versand- und Internet-Einzelhandel in Deutschland von Januar 2018 bis Oktober 2020*
<https://de.statista.com/statistik/daten/studie/579708/umfrage/monatliche-umsatzentwicklung-im-versand-und-internet-einzelhandel/>
Veröffentlicht am: Dezember 2020
Zuletzt aufgerufen am: 24. Dezember 2020
- (Statista, 2020b) Statista (2020).: *Anteil der Internetnutzer in Deutschland in den Jahren 1997 bis 2020*
<https://de.statista.com/statistik/daten/studie/36009/umfrage/anteil-der-internetnutzer-in-deutschland-seit-1997/>
Veröffentlicht am: Oktober 2020
Zuletzt aufgerufen am: 24. Dezember 2020

- (Statista, 2020c) Statista (2020).: *Anteil der Internetnutzer nach Altersgruppen in Deutschland in den Jahren 1997 bis 2020*
<https://de.statista.com/statistik/daten/studie/36149/umfrage/anteil-der-internetnutzer-in-deutschland-nach-altersgruppen-seit-1997/>
Veröffentlicht am: Oktober 2020
Zuletzt aufgerufen am: 24. Dezember 2020
- (Stoffers, 2015) Stoffers, A. (2015).: *Subjektive Theorien von Informatiklehrkräften zur fachdidaktischen Strukturierung ihres Unterrichts*. Fakultät II – Department für Informatik, Carl von Ossietzky Universität Oldenburg 2015.
- (Thesmann, 2016) Thesmann, S. (2016).: *Interface Design – Usability, User Experience und Accessibility im Web gestalten* (2. Aufl.). Springer Fachmedien Wiesbaden 2016
- (Tractinsky, 2004) Tractinsky, N. (2004).: *Towards the study of aesthetics in information technology*. Aus: *ICIS 2004 Proceedings*, paper 62
- (Tuch et al., 2010) Tuch A., Bargas-Avila, J., & Opwis, K. (2010).: *Symmetry and aesthetics in website design: It's a man's business*. Aus: *Computers in Human Behavior 26* (2010), S. 1831-1837. Elsevier Ltd.
- (Tuch et al., 2012) Tuch, A., Presslauer, E., Ströcklin, M., Opwis, K. & Bargas-Avila, J. (2012).: *The role of visual complexity and prototypicality regarding first impressions of websites: Working towards understanding aesthetic judgements*. Aus: *International Journal of Human-Computer Studies 70* (2012), S. 794-811. Elsevier Ltd.
- (UNESCO, 2020) UNESCO (2020).: *COVID-19 – Education: From disruption to recovery*
<https://en.unesco.org/covid19/educationresponse>
zuletzt aufgerufen am: 28. Dezember 2020

- (Velica, 2010) Velica, I. (2010).: *Lernziele und deren Bedeutung im Unterricht*. Aus: Neue Didaktik (2012) 2, S. 10-24. Deutsches Institut für Internationale Pädagogische Forschung.
- (Vester, 2018) Vester, F. (2018).: *Denken, Lernen, Vergessen – Was geht in unserem Kopf vor, wie lernt das Gehirn, und wann lässt es uns im Stich?* (38. Auflage). Dtv Verlagsgesellschaft mbH & Co. KG, München
- (Vollmers, 1997) Vollmers, B. (1997).: *Learning by doing – Piagets konstruktivistische Lerntheorie und ihre Konsequenzen für die pädagogische Praxis*. Aus: *International Review of Education – Internationale Zeitschrift für Erziehungswissenschaft – Revue Internationale de l'Education* 43(1): S. 73-85
- (W3C, 2020) World Wide Web Consortium: HTML.
<https://html.spec.whatwg.org/print.pdf>
Letztes Update: 22. Oktober 2020
Zuletzt aufgerufen: 01. November 2020
- (Weickler, 2007) Weickler, N. (2007).: *Zielorientierte Didaktik der Informatik – Kompetenzvermittlung bei engen Zeitvorgaben*. Aus: Schubert, S. (2007).: *Didaktik der Informatik in Theorie und Praxis*. 12. GI-Fachtagung „Informatik und Schule – INFOS 2007“. 19-21. September 2007 an der Universität Siegen.
- (Witte, 2016) Witte, F. (2016).: *Testmanagement und Softwaretest – Theoretische Grundlagen und praktische Umsetzung*. Springer Fachmedien Wiesbaden 2016
- (ZFL, 2020) ZFL TU Kaiserslautern.
<https://service.zfl.uni-kl.de/wp/glossar/bildungstheoretische-didaktik>
zuletzt aufgerufen: 30.11.2020

Anhang I: Übungsaufgaben

Aufgabe 1 a)

Erzeugen Sie eine SQL-Abfrage, die Ihnen lediglich die Nachnamen und Vornamen der Studierenden ausgibt. (siehe **Tabelle AI.I**: Studierende)

Tabelle AI.I: Studierende

Matrikelnummer	Nachname	Vorname	Studiengang	Studienbeginn	Universität
123456	Klaus	Felix	Maschinenbau	2015	TU München
145794	Said	Abdel	Logistik	2017	TU Dortmund
165472	Weiss	Werner	BWL	2019	HHU Düsseldorf
127945	Kaya	Hassan	Logistik	2015	TU München
154878	Dürr	Mirijam	Mathematik	2018	HHU Düsseldorf
164722	Jackson	Michael	Musik	2019	TU Dortmund
155520	Kataga	Shinichi	Maschinenbau	2018	TU Dortmund
176001	Morozov	Dmitri	Logistik	2020	TU Dortmund

Lösung:

```
SELECT      Nachname, Vorname
FROM        Studierende
```

Ausgabe:

Nachname	Vorname
Klaus	Felix
Said	Abdel
Weiss	Werner
Kaya	Hassan
Dürr	Mirijam
Jackson	Michael
Kataga	Shinichi
Morozov	Dmitri

Aufgabe 1 b)

Erzeugen Sie eine SQL-Abfrage, die Ihnen lediglich die Nachnamen und Vornamen der Studierenden der TU Dortmund ausgibt. (siehe **Tabelle AI.I**: Studierende)

Lösung:

```
SELECT      Nachname, Vorname
FROM        Studierende
WHERE       Universität = 'TU Dortmund'
```

Nachname	Vorname
Said	Abdel
Jackson	Michael
Kataga	Shinichi
Morozov	Dmitri

Aufgabe 1 c)

Erzeugen Sie eine SQL- Abfrage, die Ihnen die Namen der Universitäten lediglich einfach ausgibt. (siehe **Tabelle A1.1**: Studierende)

Lösung:

```
SELECT DISTINCT   Universität
FROM               Studierende
```

Ausgabe:

Universität
TU München
TU Dortmund
HHU Düsseldorf

Aufgabe 2 a)

Fügen Sie den Studierenden Alexander Müller, mit der Matrikelnummer 166014, welcher im Jahr 2019 sein BWL-Studium an der HRW Mülheim begonnen hat, hinzu.

(siehe **Tabelle A1.1**: Studierende)

Lösung:

```
INSERT INTO      Studierende
VALUES          (166014, 'Müller', 'Alexander', 'BWL', 2019, 'HRW Mülheim')
```

Ausgabe:

Matrikelnummer	Nachname	Vorname	Studiengang	Studienbeginn	Universität
123456	Klaus	Felix	Maschinenbau	2015	TU München
145794	Said	Abdel	Logistik	2017	TU Dortmund
165472	Weiss	Werner	BWL	2019	HHU Düsseldorf
127945	Kaya	Hassan	Logistik	2015	TU München
154878	Dürr	Mirijam	Mathematik	2018	HHU Düsseldorf
164722	Jackson	Michael	Musik	2019	TU Dortmund
155520	Kataga	Shinichi	Maschinenbau	2018	TU Dortmund
176001	Morozov	Dmitri	Logistik	2020	TU Dortmund
166014	Müller	Alexander	BWL	2019	HRW Mülheim

Aufgabe 2 b)

Ändern Sie den Studiengang vom Studierenden mit der Matrikelnummer 127945 von Logistik zu Maschinenbau. (siehe **Tabelle A1.1**: Studierende)

Lösung:

```
UPDATE      Studierende
SET         Studiengang = 'Maschinenbau'
WHERE      Matrikelnummer = 127945
```

Ausgabe:

Matrikelnummer	Nachname	Vorname	Studiengang	Studienbeginn	Universität
123456	Klaus	Felix	Maschinenbau	2015	TU München
145794	Said	Abdel	Logistik	2017	TU Dortmund
165472	Weiss	Werner	BWL	2019	HHU Düsseldorf
127945	Kaya	Hassan	Maschinenbau	2015	TU München
154878	Dürr	Mirijam	Mathematik	2018	HHU Düsseldorf
164722	Jackson	Michael	Musik	2019	TU Dortmund
155520	Kataga	Shinichi	Maschinenbau	2018	TU Dortmund
176001	Morozov	Dmitri	Logistik	2020	TU Dortmund

Aufgabe 2 c)

Löschen Sie alle Studierenden aus der Tabelle, die eine Matrikelnummer haben, die kleiner ist als 150000. (siehe **Tabelle A1.1**: Studierende)

Lösung:

```
DELETE FROM      Studierende
WHERE           Matrikelnummer < 150000
```

Ausgabe:

Matrikelnummer	Nachname	Vorname	Studiengang	Studienbeginn	Universität
165472	Weiss	Werner	BWL	2019	HHU Düsseldorf
154878	Dürr	Mirijam	Mathematik	2018	HHU Düsseldorf
164722	Jackson	Michael	Musik	2019	TU Dortmund
155520	Kataga	Shinichi	Maschinenbau	2018	TU Dortmund
176001	Morozov	Dmitri	Logistik	2020	TU Dortmund

Aufgabe 3 a)

Ermitteln Sie den Nachnamen aller Studierenden der TU München oder der Studierenden, die eine Matrikelnummer besitzen, die größer ist als 170000.

(siehe **Tabelle A1.1**: Studierende)

Lösung:

```
SELECT      Nachname
FROM        Studierende
WHERE       Universität = 'TU München' OR Matrikelnummer > 170000
```

Ausgabe:

Nachname
Klaus
Kaya
Morozov

Aufgabe 3 b)

Löschen Sie alle Einträge von Studierenden, die an der TU Dortmund studieren oder ihr Studium im Jahr 2015 begonnen haben. (siehe **Tabelle A1.1**: Studierende)

Lösung:

```
DELETE FROM Studierende
WHERE     Universität = 'TU Dortmund' OR Studienbeginn = 2015
```

Ausgabe:

Matrikelnummer	Nachname	Vorname	Studiengang	Studienbeginn	Universität
165472	Weiss	Werner	BWL	2019	HHU Düsseldorf
154878	Dürr	Mirijam	Mathematik	2018	HHU Düsseldorf

Aufgabe 4 a)

Bei der Datenerfassung wurden Vor- und Nachnamen der Studierenden nicht getrennt. Erzeugen Sie eine SQL-Abfrage, die Ihnen die Matrikelnummern der Studierenden mit Nachnamen Müller ausgibt. (siehe **Tabelle AI.II**: Studierendenliste)

Tabelle AI.II: Studierendenliste

Matrikelnummer	Name	Studiengang	Studienbeginn	Universität	PLZ
123456	Klaus Müller	Maschinenbau	2015	TU München	80333
145794	Mirjam Möller	Logistik	2017	TU Dortmund	44227
165472	Lena Weiss	BWL	2019	HHU Düsseldorf	40225
127945	Thomas Müller	Logistik	2015	TU München	80333
154878	Steffan Schmidt	Mathematik	2018	HHU Düsseldorf	40225
164722	Mark Müller	Musik	2019	TU Dortmund	44227
155520	Gustav Behrend	Maschinenbau	2018	TU Dortmund	44227
176001	Sophie Müller	Logistik	2020	TU Dortmund	44227

Lösung:

```
SELECT      Matrikelnummer
FROM        Studierendenliste
WHERE       Name LIKE '%Müller'
```

Ausgabe:

Matrikelnummer
123456
127945
164722
176001

Aufgabe 4 b)

Erzeugen Sie eine SQL-Abfrage, die alle Informationen der Studierenden ausgibt, deren Vorname mit "M" beginnt und deren Nachname mit "r" endet. Zudem sollen diese Personen Studierende einer Universität sein, deren PLZ eine 2 beinhaltet.

(siehe **Tabelle AI.II**: Studierendenliste)

Lösung:

```
SELECT *
FROM        Studierendenliste
WHERE       Name LIKE 'M%r' AND PLZ LIKE '%2%'
```

Ausgabe:

Matrikelnummer	Name	Studiengang	Studienbeginn	Universität	PLZ
145794	Mirijam Möller	Logistik	2017	TU Dortmund	44227
164722	Mark Müller	Musik	2019	TU Dortmund	44227

Aufgabe 4 c)

Ändern Sie den Studiengang aller Studierenden, deren Nachname mit "r" endet und die an einer Technischen Universität studieren zu Informatik.

(siehe **Tabelle AI.II**: Studierendenliste)

Lösung:

```
UPDATE      Studierendenliste
SET         Studiengang = 'Informatik'
WHERE      Name LIKE '%r' AND Universität LIKE 'TU%'
```

Ausgabe:

Matrikelnummer	Name	Studiengang	Studienbeginn	Universität	PLZ
123456	Klaus Müller	Informatik	2015	TU München	80333
145794	Mirijam Möller	Informatik	2017	TU Dortmund	44227
165472	Lena Weiss	BWL	2019	HHU Düsseldorf	40225
127945	Thomas Müller	Informatik	2015	TU München	80333
154878	Steffan Schmidt	Mathematik	2018	HHU Düsseldorf	40225
164722	Mark Müller	Informatik	2019	TU Dortmund	44227
155520	Gustav Behrend	Maschinenbau	2018	TU Dortmund	44227
176001	Sophie Müller	Informatik	2020	TU Dortmund	44227

Aufgabe 5 a)

Erzeugen Sie eine SQL-Abfrage, die Ihnen ausgibt wie oft welches Modul in den letzten Jahren abgelegt wurde. (siehe **Tabelle AI.III**: Klausurergebnisse)

Tabelle AI.III: Klausurergebnisse

Modul	Jahr	Teilnehmerzahl	bestanden	durchgefallen
Thermodynamik	2012	254	28	226
Strömungsmechanik	2013	371	211	160
Höhere Mathematik III	2014	213	97	116
Thermodynamik	2015	410	342	68
Thermodynamik	2016	99	55	44
Höhere Mathematik III	2017	180	130	50
Mechanik C	2018	257	111	146
Höhere Mathematik III	2019	300	151	149
Strömungsmechanik	2020	197	76	121

Lösung:

```
SELECT      Modul,
COUNT(*)
FROM        Klausurergebnisse
GROUP BY    Modul
```

Ausgabe:

Modul	COUNT(*)
Thermodynamik	3
Strömungsmechanik	2
Höhere Mathematik III	3
Mechanik C	1

Aufgabe 5 b)

Erzeugen Sie eine SQL-Abfrage, die Ihnen ausgibt, wie viele Teilnehmer die jeweiligen Module abgelegt haben und ordnen Sie die Module in absteigender alphabetischer Reihenfolge.

(siehe **Tabelle AI.III**: Klausurergebnisse)

Lösung:

```
SELECT      Modul,
SUM(Teilnehmerzahl)
FROM        Klausurergebnisse
GROUP BY    Modul
ORDER BY    Modul DESC
```

Ausgabe:

Modul	SUM(Teilnehmerzahl)
Thermodynamik	763
Strömungsmechanik	568
Mechanik C	257
Höhere Mathematik III	693

Aufgabe 5 c)

Erzeugen Sie eine SQL-Abfrage, die Ihnen die durchschnittliche Anzahl an Studierenden ausgibt, die ein Modul, welches ein 'c' im Modulnamen beinhaltet, nicht bestanden haben. Ordnen Sie die Module aufsteigend in alphabetischer Reihenfolge.

(siehe **Tabelle AI.III**: Klausurergebnisse)

Lösung:

```
SELECT      Modul,
AVG(durchgefallen)
FROM        Klausurergebnisse
WHERE       Modul LIKE '%c%'
GROUP BY   Modul
ORDER BY   Modul ASC
```

Ausgabe:

Modul	AVG(durchgefallen)
Mechanik C	146
Strömungsmechanik	140,5

Aufgabe 6 a)

Erzeugen Sie eine SQL-Abfrage, die Ihnen den Vornamen, Nachnamen und die Note der jeweiligen Studierenden ausgibt.

(siehe **Tabelle AI.IV**: Mechanikergebnisse und **Tabelle AI.V**: Studierendeninformationen)

Tabelle AI.IV: Mechanikergebnisse

Matrikelnummer	Modul	Punktzahl	Note
123456	Mechanik A	100	1
145784	Mechanik D	41	4
165472	Mechanik A	54	3.3
127945	Mechanik A	59	3
154878	Mechanik C	57	1.7
164722	Mechanik B	10	5
155520	Mechanik C	1	5
176001	Mechanik A	66	2.7
146558	Mechanik B	91	1.7

Tabelle AI.V: Studierendeninformationen

Matrikelnummer	Nachname	Vorname	Wohnort
123456	Klaus	Felix	Dortmund
145784	Said	Abdel	Dortmund
165472	Weiss	Werner	Hagen
127945	Kaya	Hassan	Duisburg
154878	Dürr	Mirijam	Minden
164722	Jackson	Michael	Hagen
155520	Kataga	Shinichi	Düsseldorf
176001	Morozov	Dmitri	Duisburg
146558	Halilovic	Edin	Hagen

Lösung:

```
SELECT      Studierendeninformationen.Nachname,
Studierendeninformationen.Vorname, Mechanikergebnisse.Note
FROM        Mechanikergebnisse, Studierendeninformationen
WHERE      Mechanikergebnisse.Matrikelnummer =
Studierendeninformation.Matrikelnummer
```

Ausgabe:

Nachname	Vorname	Note
Klaus	Felix	1
Said	Abdel	4
Weiss	Werner	3.3
Kaya	Hassan	3
Dürr	Mirijam	1.7
Jackson	Michael	5
Kataga	Shinichi	5
Morozov	Dmitri	2.7
Halilovic	Edin	1.7

Aufgabe 6 b)

Erzeugen Sie eine SQL-Abfrage, die Ihnen den Vor- und Nachnamen der Studierenden ausgibt, die in Dortmund wohnen und eine bessere Note als 3.0 erhalten haben.

(siehe **Tabelle AI.IV**: Mechanikergebnisse und **Tabelle AI.V**: Studierendeninformationen)

Lösung:

```
SELECT      Studierendeninformationen.Nachname,
Studierendeninformationen.Vorname
FROM        Mechanikergebnisse, Studierendeninformationen
WHERE       Mechanikergebnisse.Matrikelnummer =
Studierendeninformationen.Matrikelnummer AND Note < 3.0 AND Wohnort =
'Dortmund'
```

Ausgabe:

Nachname	Vorname
Klaus	Felix

Aufgabe 6c)

Erzeugen Sie eine SQL-Abfrage, die Ihnen den Vor- und Nachnamen der Studierenden ausgibt, die Mechanik A geschrieben haben und in einer Stadt wohnen, die ein "N" beinhaltet. Sortieren Sie nach dem Nachnamen in absteigender alphabetischer Reihenfolge.

(siehe **Tabelle AI.IV**: Mechanikergebnisse und **Tabelle AI.V**: Studierendeninformationen)

Lösung:

```
SELECT      Studierendeninformationen.Nachname,
Studierendeninformationen.Vorname
FROM        Mechanikergebnisse, Studierendeninformationen
WHERE       Mechanikergebnisse.Matrikelnummer =
Studierendeninformationen.Matrikelnummer AND Modul = 'Mechanik A' AND Wohnort
LIKE '%n%'
ORDER BY Nachname DESC
```

Ausgabe:

Nachname	Vorname
Klaus	Felix
Weiss	Werner

Aufgabe 6 d)

Erzeugen Sie mit Hilfe des JOIN-Befehls eine Tabelle, in welcher der Nachname und die Note aufgelistet sind. Die neue Tabelle soll keine leeren Felder beinhalten. (siehe **Tabelle AI.VI**: Ergebnisse und **Tabelle AI.VII**: Studierendeninformation)

Tabelle AI.VI: Ergebnisse

Matrikelnummer	Modul	Punktzahl	Note
123456	Mechanik A	100	1
145784	Mechanik D	41	4
165472	Mechanik A	51	3.3
127945	Mechanik A	59	3
154878	Mechanik C	87	1.7
164722	Mechanik B	10	5
155520	Mechanik C	1	5
146558	Mechanik B	91	1.7

Tabelle AI.VII: Studierendeninformation

Matrikelnummer	Nachname	Vorname	Wohnort
123456	Klaus	Felix	Dortmund
145784	Said	Abdel	Dortmund
165472	Weiss	Werner	Hagen
127945	Kaya	Hassan	Duisburg
164722	Jackson	Michael	Hagen
155520	Kataga	Shinichi	Düsseldorf
176001	Morozov	Dmitri	Duisburg
146558	Halilovic	Edin	Hagen

Lösung:

```
SELECT      Studierendeninformation.Nachname, Ergebnisse.Note
FROM        Studierendeninformation
INNER JOIN  Ergebnisse ON Ergebnisse.Matrikelnummer =
Studierendeninformation.Matrikelnummer
```

Ausgabe:

Nachname	Note
Klaus	1
Said	4
Weiss	3
Kaya	1,7
Jackson	5
Kataga	5
Halilovic	1.7

Aufgabe 6e)

Ordnen Sie den Noten den entsprechenden Nachnamen und die entsprechende Matrikelnummer zu. In der ersten Spalte soll die Matrikelnummer, in der zweiten der Nachname und in der dritte die Note stehen. Achten Sie zudem darauf, dass nur Studierende gesucht werden, die die Klausur bestanden haben (die Klausur zählt als bestanden, wenn die Note 4 oder eine bessere erreicht wird) und aus einer Stadt kommen, die mit dem Buchstaben D anfängt. Die Liste soll in aufsteigender alphabetischer Reihenfolge nach dem Nachnamen sortiert werden. (siehe **Tabelle AI.VI**: Ergebnisse und **Tabelle AI.VII**: Studierendeninformation)

Lösung:

```
SELECT      Ergebnisse.Matrikelnummer,
Studierendeninformation.Nachname, Ergebnisse.Note
FROM        Studierendeninformation
INNER JOIN  Ergebnisse
ON         Ergebnisse.Matrikelnummer =
Studierendeninformation.Matrikelnummer
WHERE      Note < 5 AND Wohnort LIKE 'D%'
ORDER BY   Nachname ASC
```

Ausgabe:

Matrikelnummer	Nachname	Note
127945	Kaya	3
123456	Klaus	1
145784	Said	4

Anhang II: Zustandsdiagramme der Softwaremodule

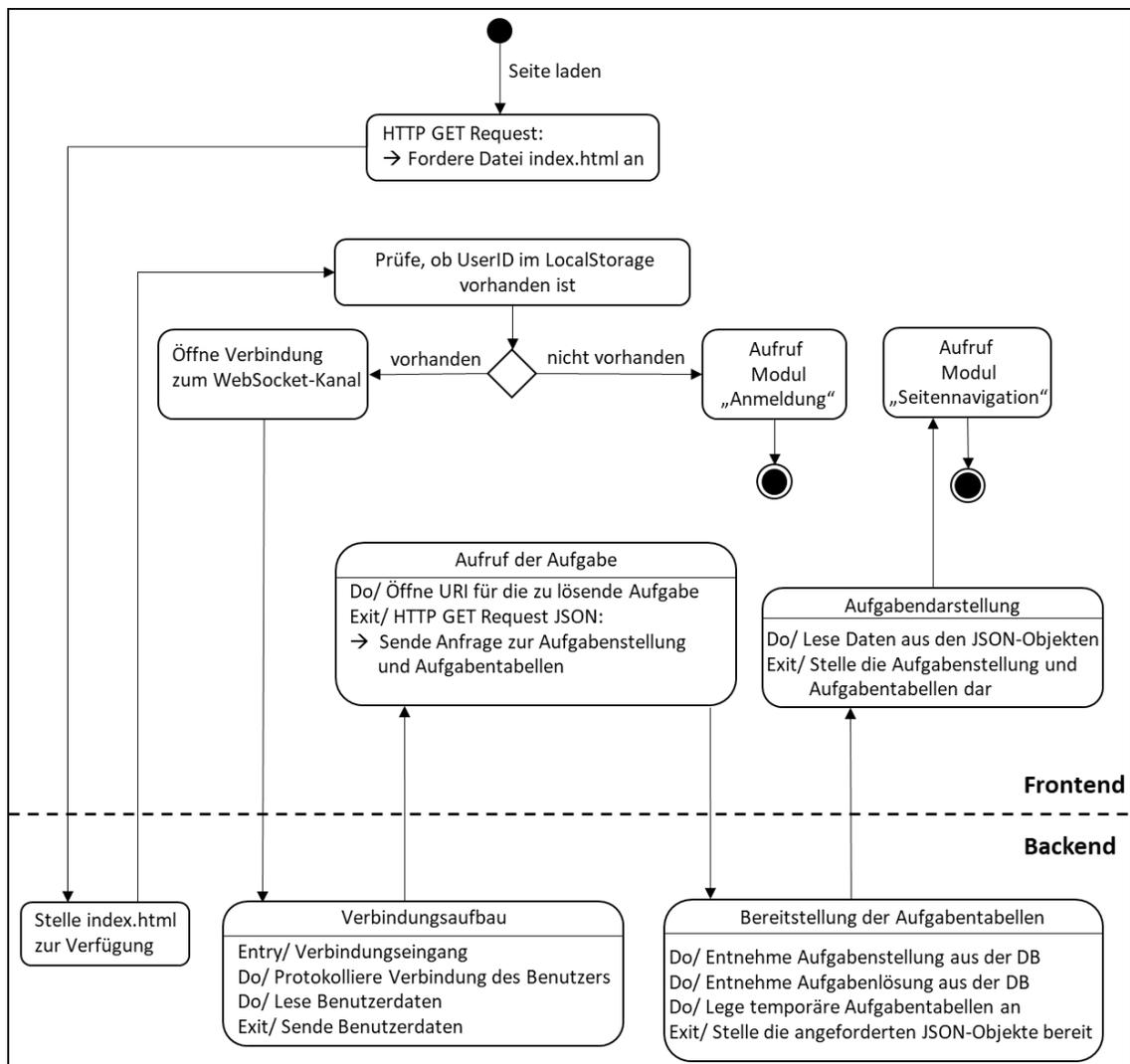


Abbildung All.I: Zustandsdiagramm zum Softwaremodul „Seite laden“

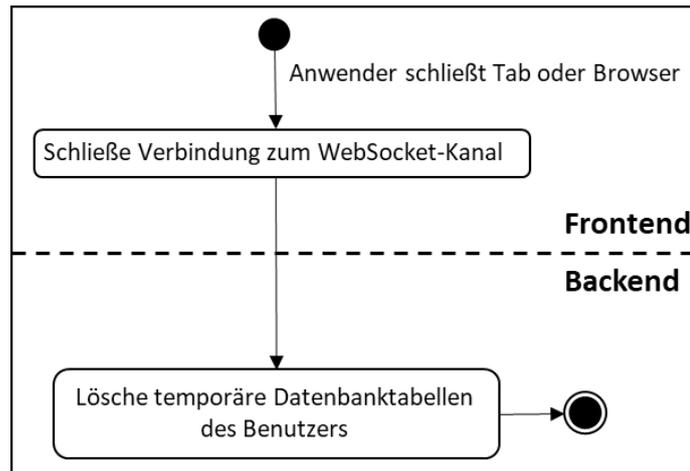


Abbildung AII.II: Zustandsdiagramm zum Softwaremodul „Verbindungsabbruch“

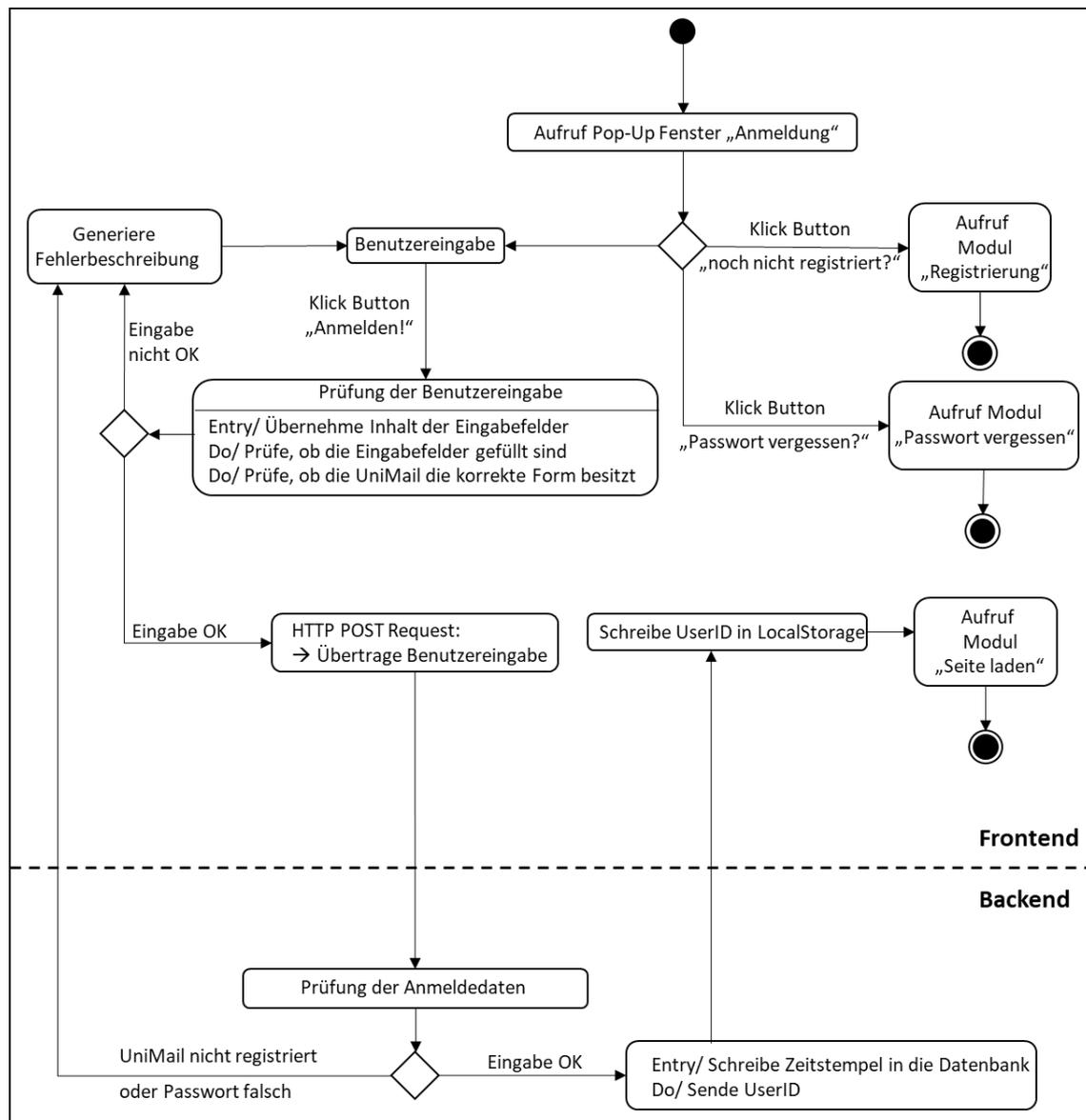


Abbildung AII.III: Zustandsdiagramm zum Softwaremodul „Anmeldung“

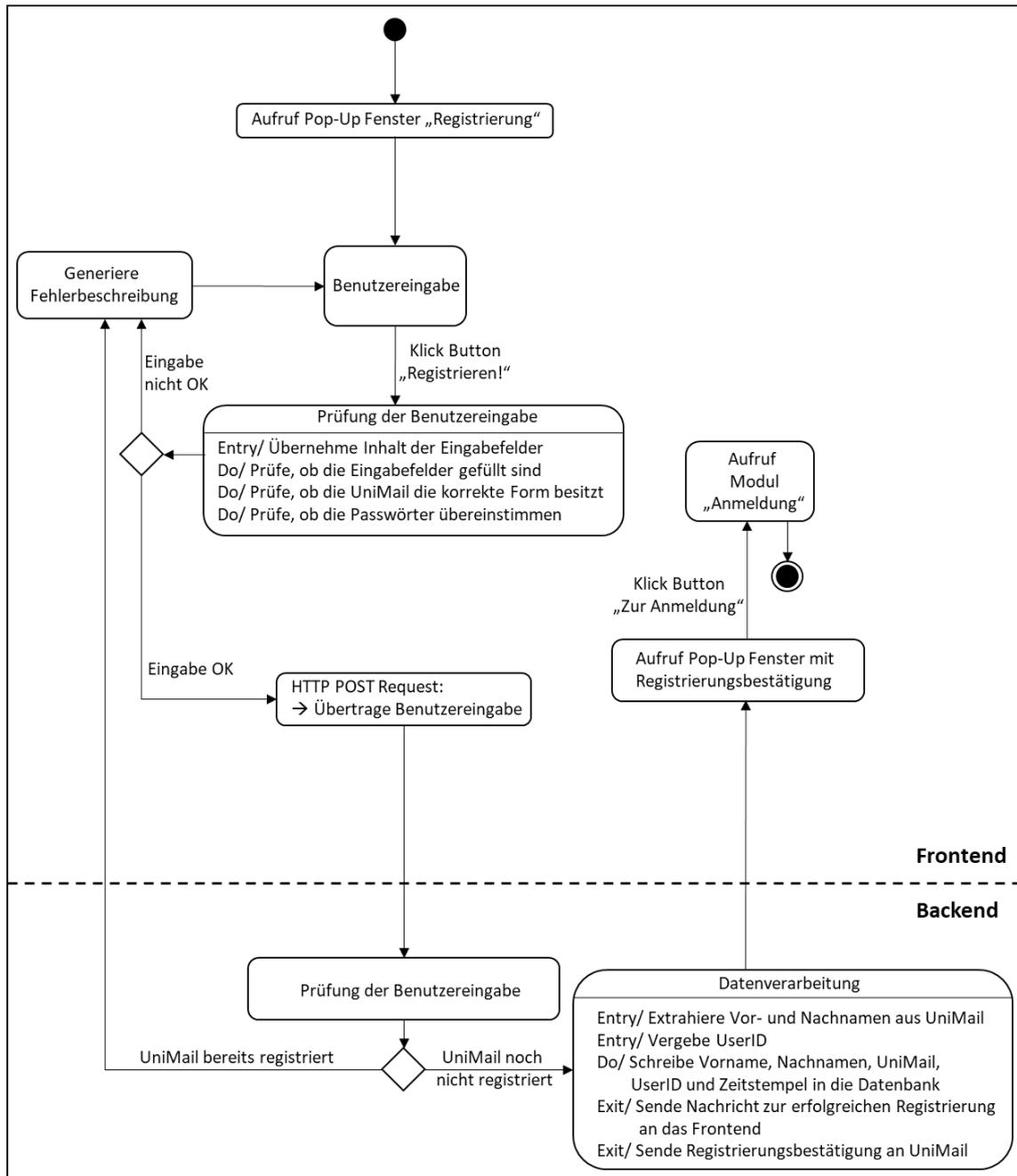


Abbildung All.IV: Zustandsdiagramm zum Softwaremodul „Registrierung“

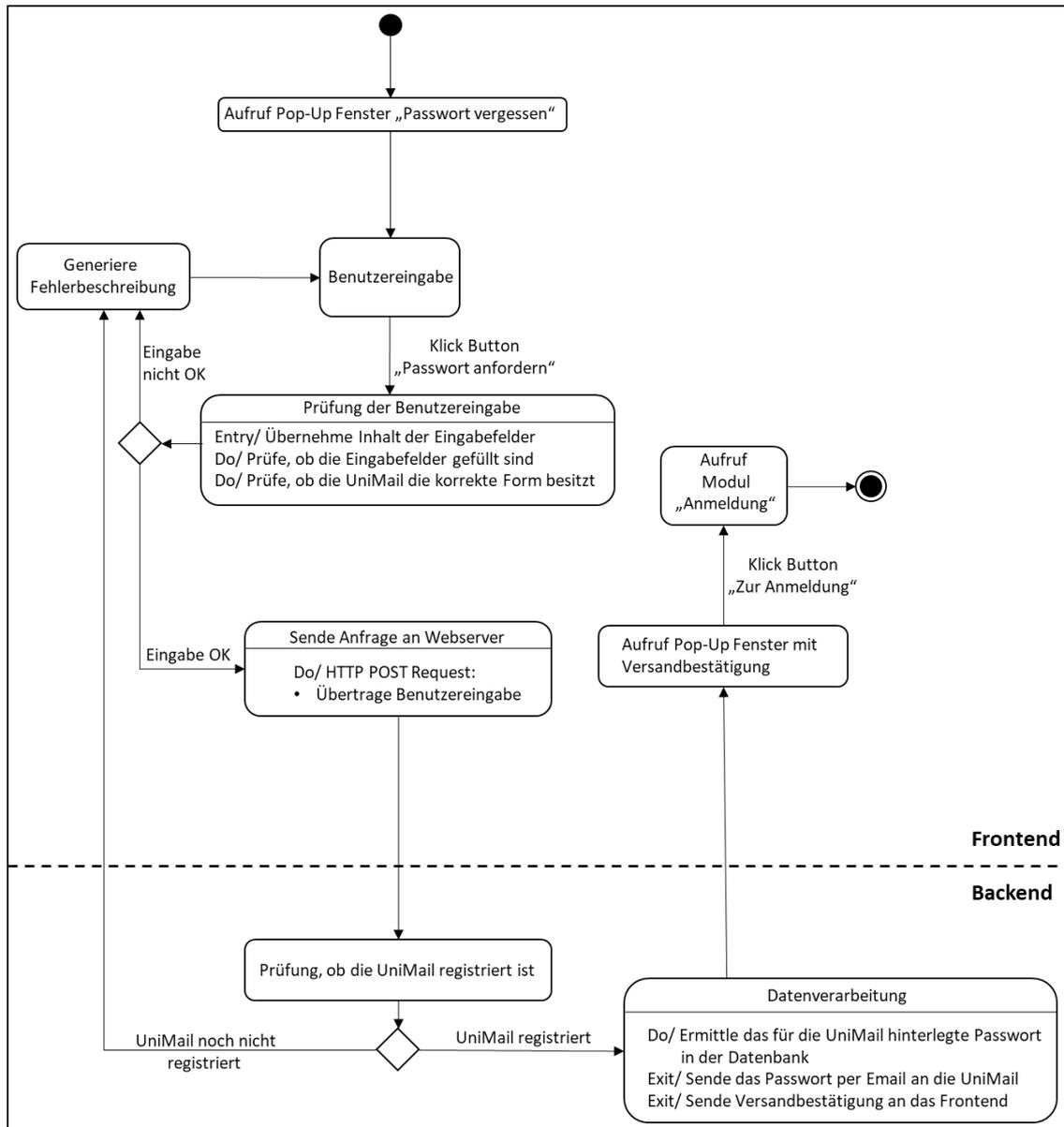


Abbildung AII.V: Zustandsdiagramm zum Softwaremodul „Passwort vergessen“

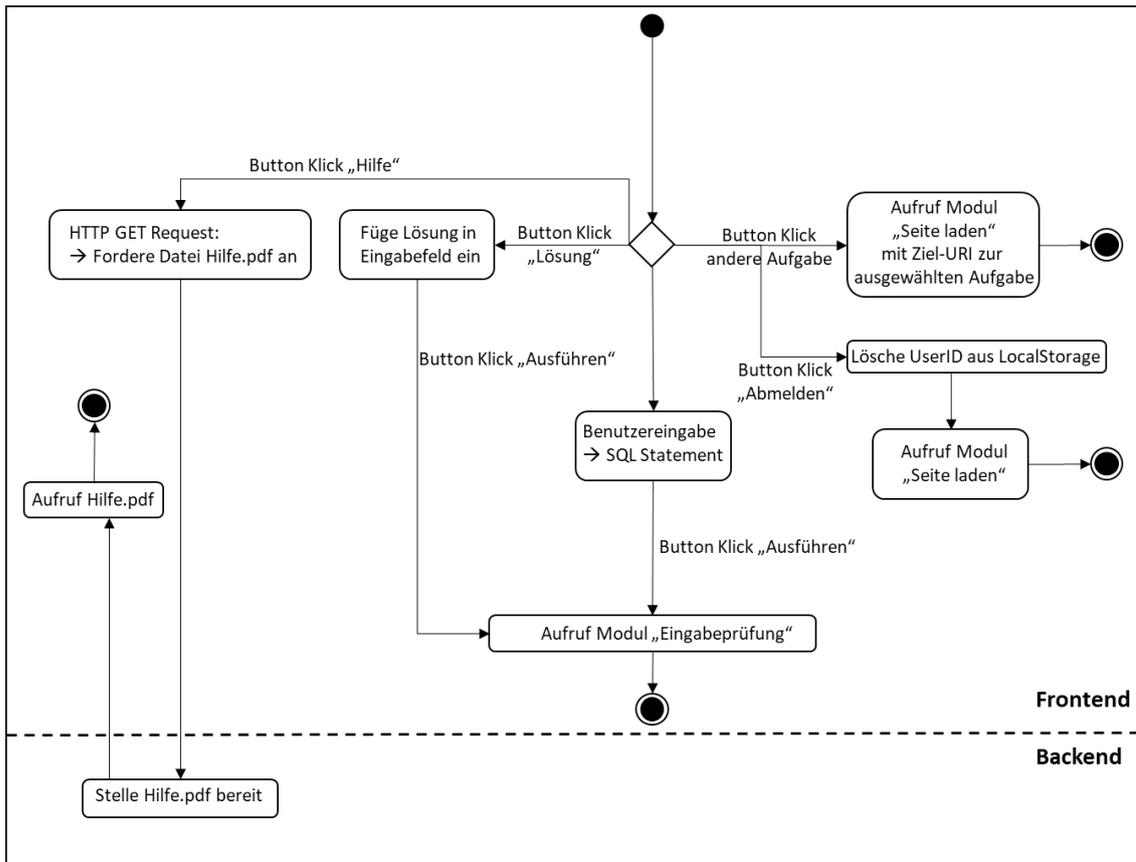


Abbildung AII.VI: Zustandsdiagramm zum Softwaremodul „Seitennavigation“

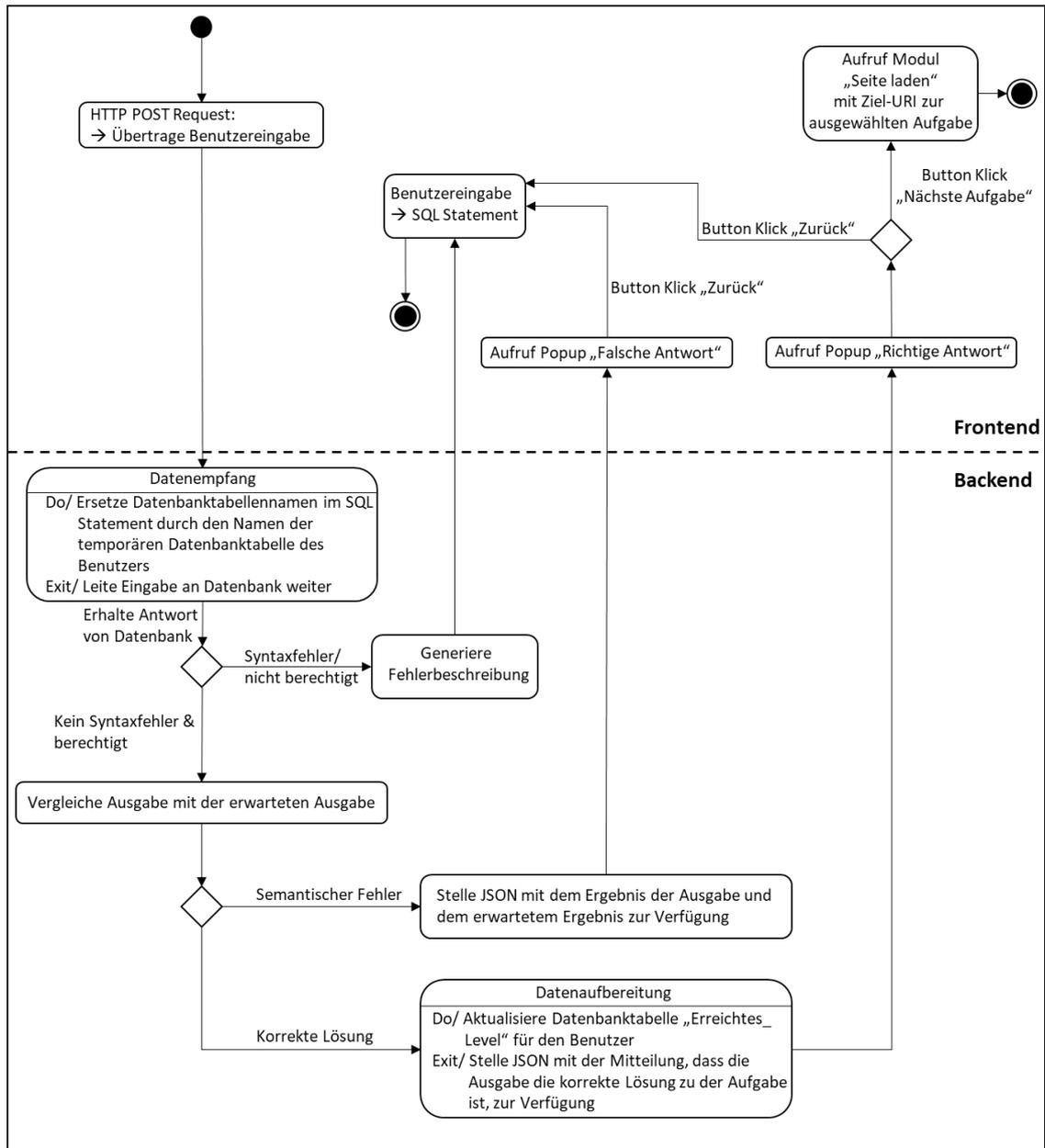


Abbildung AII.VII: Zustandsdiagramm zum Softwaremodul „Eingabeprüfung“

Anhang III: Hilfe.pdf



itpl

Anmeldung

Unimail:

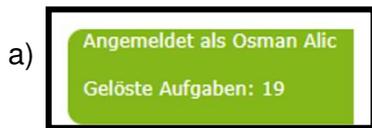
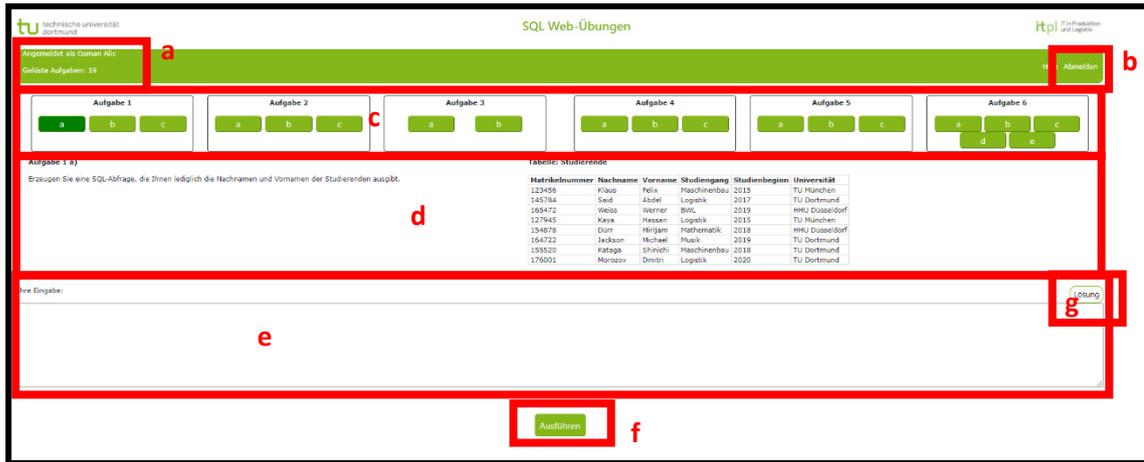
Passwort:

Passwort vergessen? **A**

Anmelden!

Noch nicht registriert?

Ist das Anmeldefenster geöffnet, können Sie sich entweder anmelden oder mit einer TU-Mail-Adresse registrieren. Zudem haben Sie die Möglichkeit ihr Passwort per E-Mail anzufordern, sofern sie dieses vergessen haben. Da klicken Sie auf das Feld **A** und füllen Sie die erscheinende Maske mit Ihren Daten aus.



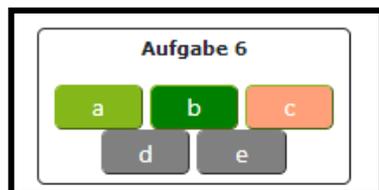
An dieser Stelle erscheint ihr Name. Zudem wird angezeigt wie viele Aufgaben Sie bereits erfolgreich bearbeitet hat.



Mit Hilfe dieser Funktion können Sie sich abmelden. Der geleistete Fortschritt wird automatisch gespeichert.



An dieser Stelle wird die Navigationsleiste angezeigt. Somit können Sie die Aufgabe auswählen, die Sie Lösen möchte. Eine nicht gelöste Aufgabe wird grau hinterlegt (Aufgabe 6d) und 6e)). Ist eine Aufgabe gelöst worden so wird das entsprechende Feld in der Navigationsleiste grün hinterlegt (Aufgabe 6a)).



Befinden Sie sich bei einer Aufgabe, die bereits gelöst wurde, so wird dieses Feld mit einer dunkelgrünen Farbe hinterlegt (Aufgabe 6d)). Ist die Bearbeitung einer Aufgabe freigeschaltet und der Nutzer eine andere Aufgabe ausgewählt hat, so wird dieses Feld rot markiert (Aufgabe 6c)). Es nicht möglich eine Aufgabe zu bearbeiten, bevor nicht alle vorigen Aufgaben gelöst wurden.

d) **Aufgabe 1 a)**
Erzeugen Sie eine SQL-Abfrage, die Ihnen lediglich die Nachnamen und Vornamen der Studierenden ausgibt.

Tabelle: Studierende

Matrikelnummer	Nachname	Vorname	Studiengang	Studienbeginn	Universität
123456	Klaus	Felix	Maschinenbau	2015	TU München
145794	Said	Abdel	Logistik	2017	TU Dortmund
165472	Weiss	Werner	BWL	2019	HHU Düsseldorf
127945	Kaya	Hassan	Logistik	2015	TU München
154878	Dürr	Mirjam	Mathematik	2018	HHU Düsseldorf
164722	Jackson	Michael	Musik	2019	TU Dortmund
155520	Kataga	Shinichi	Maschinenbau	2018	TU Dortmund
176001	Morozov	Dmitri	Logistik	2020	TU Dortmund

In diesem Bereich wird die Aufgabe angezeigt. Die Aufgabenstellung befindet sich in der linken und die dazugehörige Tabelle in der rechten Hälfte.

e) Ihre Eingabe: Lösung

Hier führt der Nutzer seine SQL- Eingabe durch.

f) Ausführen

Mit diesem Feld können Sie Ihre SQL- Eingabe durchführen. Bei falscher Eingabe erscheint ein Hinweis unmittelbar über dem „Ausführen“-Feld.

Fehler: Relation »studierend« existiert nicht

Ausführen

g) Lösung

Ist es Ihnen nicht möglich die Aufgabe zu lösen, so können Sie sich durch Anklicken des Feldes „Lösung“ die gesuchte Lösung der Aufgabe ausgeben lassen. Die Lösung erscheint im in e) beschriebenen Eingabefeld.

Anhang IV: Fragebogen zum Usability-Test

Fragebogen zum Usability-Test

Daten der Testperson

Vorname													
Nachname													
SQL-Kenntnisse	<table border="1"><tr><td>Keine Kenntnisse</td><td>↔</td><td>Grundkenntnisse</td><td>↔</td><td>Experte</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	Keine Kenntnisse	↔	Grundkenntnisse	↔	Experte	0	1	2	3	4	5	6
Keine Kenntnisse	↔	Grundkenntnisse	↔	Experte									
0	1	2	3	4	5	6							

Ich bin damit einverstanden, dass alle Daten, die durch diesen Fragebogen und durch die Videoaufzeichnung erhoben werden und im Zusammenhang mit der Durchführung der Studie stehen, in vollem Umfang für die obengenannte Masterarbeit verwendet werden dürfen. Dieser Fragebogen wird vollständig ausgefüllt im Anhang der Masterarbeit einsehbar sein.

(Name, Vorname)

(Ort, Datum)

(Unterschrift)

Alle Anmerkungen, die über den Inhalt der Fragen auf diesem Bogen hinausgehen, können auf der letzten Seite gemacht werden.

Teil 1: intuitive Ästhetik

Der Begriff Ästhetik wird verstanden als „die subjektive Attraktivität und visuelle Schönheit eines User Interfaces aus Nutzersicht“.

Ihnen wird gleich ein kurzes Video gezeigt. Dort wird Ihnen an unbekannter Stelle die graphische Oberfläche der Webapplikation für 500ms eingeblendet. Bevor das Video startet, beantworten Sie bitte die folgende Frage:

Verbinden Sie den Begriff „Symmetrie“ eher mit der achsensymmetrischen oder punktsymmetrischen Darstellung von Objekten?

	achsensymmetrische Darstellung
	punktsymmetrische Darstellung

Auszufüllen nachdem das Video beendet wurde.

Beantworten Sie bitte folgende Fragen:

Symmetrie: wie bewerten Sie die Symmetrie der Nutzeroberfläche?

asymmetrisch		←—————→				symmetrisch	
0	1	2	3	4	5	6	

Wie wirkt sich Ihre Wahl auf die Ästhetik aus?

negativ		←—————→				positiv	
0	1	2	3	4	5	6	

Buntheit/ Farbgebung: wie bewerten Sie die Buntheit/ Farbgebung der Nutzeroberfläche?

zu wenig Farben		←—————→				zu viele Farben	
0	1	2	3	4	5	6	

Wie wirkt sich Ihre Wahl auf die Ästhetik aus?

negativ		←—————→				positiv	
0	1	2	3	4	5	6	

Visuelle Komplexität: wie bewerten Sie die visuelle Komplexität auf der Nutzeroberfläche?

zu wenig Elemente auf der Seite		←—————→				zu viele Elemente auf der Seite	
0	1	2	3	4	5	6	

Wie wirkt sich Ihre Wahl auf die Ästhetik aus?

negativ		←—————→				positiv	
0	1	2	3	4	5	6	

Teil 2: Übungsaufgaben und reflektive Ästhetik

Im Rahmen des zweiten Teils dieses Versuchs werden Sie darum gebeten die Übungsaufgaben selbstständig zu lösen. Im Anschluss können Sie sowohl den Schwierigkeitsgrad der Aufgaben als auch die reflektive Ästhetik bewerten. Wenn Sie eine Aufgabe nicht lösen können, können Sie auf den Button „Lösung“ drücken und zur nächsten Aufgabe übergehen. Durch eine Bildschirmaufnahme wird der Versuchsleiter nach dem Versuch Ihre Schritte nachvollziehen und diese ggf. in die Analyse miteinbeziehen. Bearbeiten Sie zunächst in Ruhe die Aufgaben und füllen Sie anschließend die untere Tabelle aus.

Die Aufgabe 1a) wird hat die Schwierigkeitsstufe 1. Nehmen Sie die Aufgabe 1a) als Referenz und beurteilen Sie anhand dessen den Schwierigkeitsgrad der restlichen Aufgaben. Sie dürfen dazu die Aufgaben beliebig oft bearbeiten.

Aufgabe	von der Testperson auszufüllen								vom Versuchsleiter auszufüllen		
	Schwierigkeitsgrad								selbstständig gelöst?		
1a)	1	2	3	4	5	6	7	8	ja	nein	
1b)	1	2	3	4	5	6	7	8	ja	nein	
1c)	1	2	3	4	5	6	7	8	ja	nein	
2a)	1	2	3	4	5	6	7	8	ja	nein	
2b)	1	2	3	4	5	6	7	8	ja	nein	
2c)	1	2	3	4	5	6	7	8	ja	nein	
3a)	1	2	3	4	5	6	7	8	ja	nein	
3b)	1	2	3	4	5	6	7	8	ja	nein	
4a)	1	2	3	4	5	6	7	8	ja	nein	
4b)	1	2	3	4	5	6	7	8	ja	nein	
4c)	1	2	3	4	5	6	7	8	ja	nein	
5a)	1	2	3	4	5	6	7	8	ja	nein	
5b)	1	2	3	4	5	6	7	8	ja	nein	
5c)	1	2	3	4	5	6	7	8	ja	nein	
6a)	1	2	3	4	5	6	7	8	ja	nein	
6b)	1	2	3	4	5	6	7	8	ja	nein	
6c)	1	2	3	4	5	6	7	8	ja	nein	
6d)	1	2	3	4	5	6	7	8	ja	nein	
6e)	1	2	3	4	5	6	7	8	ja	nein	

Beantworten Sie bitte folgende Fragen:

Die Befehle aus vorigen Aufgaben wurden regelmäßig wiederholt und konnten logisch mit neuen Befehlen verknüpft werden. Dadurch wurde das Gelernte stärker im Gedächtnis verankert.

stimmt nicht		←—————→				stimmt	
0	1	2	3	4	5	6	

Die Hinweise, die mir die Webapplikation angezeigt hat, haben mir geholfen die Aufgaben zu lösen.

stimmt nicht		←—————→				stimmt	
0	1	2	3	4	5	6	

reflektive Ästhetik

Beantworten Sie nun die Fragen, die Sie bereits aus a) kennen. Sie dürfen sich die Nutzeroberfläche beliebig lange anschauen.

Symmetrie: wie bewerten Sie die Symmetrie der Nutzeroberfläche?

asymmetrisch		←—————→				symmetrisch	
0	1	2	3	4	5	6	

Wie wirkt sich Ihre Wahl auf die Ästhetik aus?

negativ		←—————→				positiv	
0	1	2	3	4	5	6	

Buntheit/ Farbgebung: wie bewerten Sie die Buntheit/ Farbgebung der Nutzeroberfläche?

zu wenig Farben		←—————→				zu viele Farben	
0	1	2	3	4	5	6	

Wie wirkt sich Ihre Wahl auf die Ästhetik aus?

negativ		←—————→				positiv	
0	1	2	3	4	5	6	

Visuelle Komplexität: wie bewerten Sie die visuelle Komplexität auf der Nutzeroberfläche?

zu wenig Elemente auf der Seite		←—————→				zu viele Elemente auf der Seite	
0	1	2	3	4	5	6	

Wie wirkt sich Ihre Wahl auf die Ästhetik aus?

negativ		←—————→				positiv	
0	1	2	3	4	5	6	

Die Ästhetik hat dafür gesorgt, dass ich Vergnügen bei der Nutzung der Webapplikation empfunden habe.

stimmt nicht		←—————→				stimmt	
0	1	2	3	4	5	6	

Die Ästhetik hat dafür gesorgt, dass mir die Bedienung der Webapplikation erleichtert wurde.

stimmt nicht		←—————→				stimmt	
0	1	2	3	4	5	6	

