

MENSCHZENTRIERTE GESTALTUNG EINER TERMINVEREINBARUNGS- SOFTWARE

BACHELORARBEIT

Studierender: Johannes Schnirring

Matrikelnummer: 33422833

Studiengang: Informatik (PO 2018)

Erstprüferin: Prof. Dr. Claude Draude
Gender/Diversity in Informatiksystemen

Zweitprüfer: Prof. Dr. Albert Zündorf
Software Engineering

Semester: Wintersemester 2022

Datum: 18. Dezember 2022

Universität Kassel

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Kassel, am
.....
(Unterschrift)

Inhaltsverzeichnis

1 Einleitung	4
1.1 Zusammenfassung	4
1.2 Motivation	5
1.3 Aufbau	6
2 Methoden	8
2.1 Human Centered Design	8
2.2 Interview im Kontext	11
3 Nutzungskontext	13
3.1 Studienberatung der Universität Kassel	13
3.2 Aktuelle Softwarelösung: Stubegru	14
3.3 Historische Entwicklung der Software Stubegru	16
4 Nutzungsanforderungen	18
4.1 Interview im Kontext	18
4.2 Auswertung des Interviews	25
4.3 Gestaltungslösungen entwickeln	29
5 Gestaltungslösungen implementieren	36
5.1 Kundenwünsche und technische Machbarkeit	36
5.2 Technische Grundlagen Stubegru	37
5.3 Nutzungsanforderungen formalisieren	38
5.4 Details der Implementierung	43
6 Gestaltungslösungen evaluieren	52
6.1 Durchführung des Usertests	52

7 Reflexion und Fazit	60
7.1 Zusammenfassung der Ergebnisse	61
7.2 Reflexion der eingesetzten Methoden	66
7.3 Ausblick	69
8 Glossar	72
9 Literaturverzeichnis	77
10 Abbildungsverzeichnis	81
11 Anhang	84
11.1 Software Stubegru	84
11.2 Sequenzdiagramme	85

Kapitel 1

Einleitung

1.1 Zusammenfassung

Diese Arbeit beschäftigt sich mit dem Entwicklungsprozess einer Software zur Terminvergabe von Beratungsterminen. Die Software soll in der Studienberatung an der Universität Kassel eingesetzt werden. Der Entwicklungsprozess wird mit Methoden des *Human Centered Design* begleitet und umgesetzt. Die Arbeit erklärt die verwendeten Methoden sowie die dahinterliegende Theorie. Im Rahmen der Arbeit wird die Software zur Terminvereinbarung vollständig implementiert. Dabei werden alle Schritte des Gestaltungsprozesses beschrieben, dokumentiert und ausgewertet.

Die allgemeine Studienberatung der Universität Kassel setzt die Software Stubegru bereits seit sechs Jahren zum Management ihrer Beratungstermine, Abwesenheiten der Mitarbeitenden und als Wissensdatenbank ein. Das Modul zur Vergabe der Beratungstermine soll nun grundlegend neu implementiert werden und noch besser an die Bedürfnisse der Mitarbeitenden angepasst werden. In Zusammenarbeit mit einem Ansprechpartner aus der Abteilung Studium und Lehre wird dieses neue Modul entwickelt, designt und implementiert.

Die Bachelorarbeit begleitet den praktischen Prozess, indem theoretische Grundlagen erläutert und alle Entwicklungsschritte beschrieben werden. Die Theorie des Human Centered Design gibt vor, wie ein Gestaltungsprozess von Software aussehen kann. Neben grundlegenden Herangehensweisen bringt diese Theorie einige Methoden mit, die im Laufe des Gestaltungsprozesses angewandt werden. Hierzu gehört beispielsweise das *Interview im Kontext*,

das in dieser Arbeit ebenfalls erklärt und dessen praktische Umsetzung dokumentiert, ausgewertet und kritisch hinterfragt wird.

1.2 Motivation

Historische Entwicklung

Wir leben in einem Zeitalter, in dem immer mehr Prozesse des täglichen Lebens durch technische Systeme wie Software begleitet werden. In dieser Situation ist es von hoher Relevanz, dass alle Menschen unserer Gesellschaft die Möglichkeit erhalten, diese Systeme gleichermaßen zu nutzen und zu verstehen. In den Anfängen der Softwareentwicklung wurde Software in der Regel aus Sicht der verarbeitenden Maschine gedacht: *Wie kann die Maschine die gegebenen Aufgaben möglichst schnell, fehlerfrei und effizient abarbeiten?* Die Nutzenden dieser Maschinen gehörten meist einem eingeschränkten, speziell dafür ausgebildeten Personenkreis an. Diese Situation hat sich gänzlich verändert. Fast jeder Mensch agiert heute viele Male am Tag mit technischen, softwaregetriebenen Systemen: Bankautomaten, Smartphones, Supermarktkassen oder Computer. Außerdem wird der Umgang mit solchen Systemen durch neue Entwicklungen wie Sprachsteuerung oder Touchscreens immer interaktiver und unmittelbarer. Somit ist es enorm wichtig, dass alle diese Systeme eine klare und nutzungsfreundliche Schnittstelle für die Nutzenden bieten[1].

Rolle des Human Centered Design

Der Ansatz des Human Centered Design setzt genau an dieser Stelle an, indem er Grundkonzepte und Methoden vermittelt, die das Design von Benutzerschnittstellen thematisieren[2]. Das Kernkonzept des Human Centered Design ist es, den Blickwinkel aus der Perspektive der Menschen zu wählen, die mit den technischen Systemen täglich interagieren müssen. Die zu entwickelnden Systeme sollen Schnittstellen anbieten, die intuitiv und einfach zu verstehen sind. Prozessabläufe sollen aus Sicht des Nutzenden abgebildet werden und möglichst wenig durch technische Hintergründe eingeschränkt sein[3].

Praktische Anwendung

Der Gestaltungsprozess nach den Methoden des Human Centered Design wird in dieser Bachelorarbeit genauer untersucht. Zum einen, indem die theoretischen Hintergründe genauer erläutert und Methoden präzise erklärt werden. Den wesentlichen Teil dieser Arbeit stellt aber die Dokumentation der praktischen Anwendung dieser Methoden und Konzepte dar. Dies wird am Beispiel einer Software zur Terminvergabe in der Studienberatung an der Universität Kassel durchgeführt. Das entsprechende Modul für die Software wird im Rahmen dieser Arbeit geplant, gestaltet, implementiert und getestet. All diese Schritte werden schriftlich aufgearbeitet, festgehalten und hinterfragt. Das Ziel dieser Arbeit ist es, den Designzyklus des Human Centered Design am praktischen Beispiel einer Terminvergabe-Software anzuwenden. Die bestehenden wissenschaftlichen Erkenntnisse im Bereich der menschzentrierten Gestaltung sollen somit erarbeitet, vertieft und auf Praxistauglichkeit geprüft werden. Ein weiterer wichtiger Bestandteil ist es schließlich, die theoretischen und methodischen Grundlagen im Hinblick auf die gewonnenen praktischen Erfahrungen zu hinterfragen und zu diskutieren.

1.3 Aufbau

Zu Beginn der Arbeit werden in Kapitel 2 die verwendeten Grundlagen und Methoden des Human Centered Design erläutert und klar definiert. Exemplarisch für eine Methode wird hier das Konzept des Interviews im Kontext genauer betrachtet und eingeführt. Die Kapitel 3 bis 6 beschäftigen sich mit der praktischen Entwicklung des Softwaremoduls zur Terminvereinbarung anhand des vorgestellten Designzyklus. Um dem Lesenden einen Überblick über den praktischen Einsatz der betrachteten Software an der Universität zu geben, wird in Kapitel 3 zunächst die Situation in der allgemeinen Studienberatung erläutert. Anschließend wird die eingesetzte Software genauer vorgestellt und deren Entwicklungsgeschichte aufgegriffen. Im weiteren Verlauf wird thematisiert, aus welchen Gründen eine Überarbeitung dieser Software notwendig ist und welches Ziel die allgemeine Studienberatung damit beabsichtigt. Im Kapitel 4 wird ein Interview im Kontext mit einem Mitarbeitenden der Studienberatung durchgeführt, dokumentiert und ausgewertet. Die aus den Wünschen und Bedürfnissen der Nutzenden entstehenden Anforderungen werden ebenfalls zusammengefasst und dargestellt. Als nächster

Schritt wird in Kapitel 5 die praktische Umsetzung der technischen Anforderungen an die Software skizziert. Implementierungsdetails, Screenshots und Codeschnipsel werden in diesem Teil die Dokumentation der Umsetzung veranschaulichen. Als letzter Aspekt der praktischen Durchführung wird in Kapitel 6 das Feedback der Nutzenden bei ersten *Usertests* festgehalten und aufbereitet. In Kapitel 7 wird schließlich das Ergebnis des durchgeföhrten Entwicklungsprozesses dargestellt. Die eingesetzten Methoden werden nochmals aufgegriffen und kritisch hinterfragt: An welchen Stellen können die theoretischen Grundlagen des Human Centered Design den Entwicklungsprozess in der Praxis tatsächlich sinnvoll unterstützen? Gab es gegebenenfalls auch Methoden, die in der praktischen Umsetzung problematisch waren oder noch optimiert werden könnten? Der letzte Teil dieser Arbeit gewährt einen Ausblick auf weitere Forschungsfelder des Human Centered Design und die weitere Entwicklung der Software für die Studienberatung der Universität Kassel.

Kapitel 2

Methoden

2.1 Human Centered Design

Human Centered Design ist eine Methode, die den Entwicklungsprozess von interaktiven Systemen und Software beschreibt. Die Kernthese ist hierbei, dass die Nutzenden der Systeme und ihr Umfeld im Mittelpunkt aller Entwicklungs- und Designprozesse stehen. Inhaltlich lässt sich die Theorie des Human Centered Design in das Themenfeld der Mensch-Maschine-Interaktion einordnen[3].

Die Entwicklung von Software war in den Anfängen geprägt von den technisch sehr eingeschränkte Möglichkeiten des Computers. Software wurde größtenteils so entwickelt, dass sie möglichst problemlos und effizient auf dem entsprechenden Computer ausgeführt werden kann. Alle Entwicklungsprozesse wurden aus Perspektive der ausführenden Maschine gedacht. Der Ansatz des Human Centered Design setzt dieser maschinenzentrierten Vorgehensweise eine klare Alternative entgegen. Martin Ludwig Hofmann beschreibt den Grundgedanken in *Human Centered Design* so: „[Es geht] nicht darum, mit Geräten zu denken, sondern mit Menschen und ihren Weltanschauungen“[2]. Wie in Kapitel 1 erläutert, gewinnt dieser Ansatz in der heutigen Zeit immer größere Bedeutung. Alan Dix erklärt in *Human Computer Interaction*, dass sich die Art und Weise, wie Menschen und Computer interagieren, in den letzten Jahrzehnten stark verändert hat. Während Computer früher meist nur mathematische Berechnungen durchführten, sind Softwaresysteme heute hochgradig interaktiv und sollen von allen Teilen der Gesellschaft ungehindert genutzt werden können[4].

Um die Methodik des Human Centered Design weiter zu spezifizieren, ist zunächst festzuhalten, dass dem Designprozess an sich eine wichtige Rolle zugeschrieben wird. Es reicht nicht aus, dass eine Software korrekt funktioniert, sich also alle Berechnungen und Prozesse fehlerfrei ausführen lassen. Im Human Centered Design liegt der Fokus auf der Interaktion zwischen dem Menschen und der Maschine. In Abgrenzung zum *User Centered Design* werden beim Human Centered Design nicht nur die unmittelbaren Nutzenden mit einbezogen. Alle Personen, die von der Gestaltung der Systeme betroffen sind, sollen während des Designprozesses beachtet werden[5]. Man betrachtet, wie die Menschen mit dem Computer interagieren. Wie stoßen sie Prozesse an? Welche Reaktionen erwarten sie von dem System? Die Betroffenen sollen mit den Systemen möglichst intuitiv interagieren können. Die Benutzeroberfläche der Software soll selbst erklären, welche Funktionen auf welchem Weg erreicht werden können. Ein essentieller Bestandteil hierfür ist zu verstehen, wie die Systeme genutzt werden. Relevante Fragestellung sind: In welchem Kontext werden die Systeme eingesetzt? Welche Menschen arbeiten mit den Systemen? Welche Informationen müssen die Nutzenden schnell erfassen können? Um diese Fragen sinnvoll beantworten zu können, müssen die Betroffenen während der Designphase kontinuierlich in den Entwicklungsprozess einbezogen werden. Ihre gesamte Situation sowie ihre Bedürfnisse müssen im Designprozess untersucht und direkt in das Produktdesign integriert werden[4].

Masaaki Kurosu hält fest, dass ein weiterer wichtiger Aspekt des Human Centered Design das Durchlaufen mehrerer Iterationen ist. Ein Produkt ist nach einem ersten Entwicklungszyklus selten schon perfekt. Oftmals ist es wichtig, den Betroffenen erste Ergebnisse zu präsentieren oder ihnen Prototypen zu zeigen. Das Feedback, das Nutzende hierbei geben, muss dokumentiert und für weitere Iterationen der Entwicklung aufgearbeitet werden. Wie Kurosu weiter ausführt, steht der zyklische Ablauf des Human Centered Design auch in der entsprechende ISO-Norm im Fokus[6].



Abbildung 2.1: Iteratives Vorgehen im Human Centered Design nach ISO 9241 [7].

Abbildung 2.1 verdeutlicht den Ablauf eines Designprozesses in mehreren Phasen. Nachdem geplant ist, welches Projekt umgesetzt werden soll, beginnt der zyklische Kreislauf des Entwicklungsprozesses. Im ersten Schritt liegt der Fokus auf der Analyse der Situation, in der das zu entwickelnde System später eingesetzt werden soll. Nur wer den Kontext eines interaktiven Systems kennt, kann die Schnittstelle zwischen Maschine und Mensch adäquat gestalten. In Schritt zwei werden aus den geplanten Erwartungen an das System und der Analyse des Nutzungskontextes konkrete Anforderungen formuliert. Dabei steht noch nicht im Fokus, wie diese Anforderungen technisch umgesetzt werden könnten. Vielmehr geht es darum, welche Anforderungen das fertige System überhaupt erfüllen soll. Im dritten Schritt geht es darum, diese Anforderungen tatsächlich umzusetzen und technische Lösungen zu implementieren. Gibt es erste Ergebnisse, wird der vierte Schritt relevant: Die umgesetzten Lösungen müssen im tatsächlichen Nutzungskontext evaluiert werden. Das setzt wieder eine intensiven Austausch mit den Betroffenen des Systems voraus. Mit dem so gewonnen Feedback kann der Prozess wieder von vorne beginnen. Fehlende Features können nachgebessert, missverständliche Schnittstellen klarer gestaltet und Probleme im praktischen Einsatz minimiert werden[8].

G.A. Boy berichtet in *The Handbook of Human-Machine Interaction*, wie wichtig es ist, dem späteren Nutzenden der Systeme erste Prototypen und

Ideen zu präsentieren. Oftmals wissen die Nutzenden gar nicht, welche technischen Möglichkeiten es gibt oder welche alternativen Bedienkonzepte in ihrem Kontext besonders gut funktionieren könnten. Im Mittelpunkt steht also das Zuhören und Eingehen auf die Nutzenden und ihr Umfeld. Allerdings kann auch das Einbringen neuer, für die Nutzenden unbekannter Lösungsansätze von hoher Bedeutung sein[3].

2.2 Interview im Kontext

Ein wesentlicher Bestandteil im Entwicklungsprozess nach den Methoden des Human Centered Design ist der intensive Austausch mit den Betroffenen der Systeme. Dafür ist es wichtig, dass Softwareentwickler und Designer von Benutzerschnittstellen in den direkten Austausch mit den Nutzenden der Systeme gehen. Um diesen Prozess strukturiert und zielführend zu durchlaufen, kann die Methode des *Interviews im Kontext* benutzt werden.

Beim Interview im Kontext geht es darum, die Nutzenden der Systeme in dem Umfeld zu beobachten, in dem die Software tatsächlich auch im praktischen Alltag eingesetzt wird. Der Fokus liegt also auf dem Umfeld der Interaktion mit technischen Systemen. In herkömmlichen Interviews begegnen sich Interviewender und Interviewter oftmals in einer neutralen Umgebung, wie beispielsweise einem Konferenzraum und sprechen über die Prozesse, für die sich der Interviewende interessiert. Beim Interview im Kontext steht nicht das Sprechen über, sondern das direkte Miterleben des Prozesses im Mittelpunkt[9]. In *Manual on Human-Computer Interaction* legen die Autoren den Fokus auf die genaue Beobachtung der Nutzenden in der Interaktion mit den Systemen. Mit dieser Methode könnte man noch viel mehr praxisnahe Details erfassen, als wenn man mit den Nutzenden nur über das System spreche. Spricht man mit Nutzenden beispielsweise darüber, wie sie eine Software bedienen, gibt es möglicherweise viele Dinge, die sie nicht erwähnen, weil sie vergessen wurden, nicht relevant erscheinen oder Nutzende nicht wissen, wie sie genau darüber sprechen sollen[10].

Dies kann in der Praxis bedeuten, dass man sich mit den Nutzenden der Systeme in ihren Büroräumen trifft und für mehrere Stunden mit dabei ist, wenn sie ihrer Arbeit nachgehen und mit den technischen Systemen interagieren. Die Rolle des Interviewenden ist dabei aus einer beobachtenden Perspektive zu verstehen. Der Interviewte soll die Richtung bestimmen, in die sich das Interview entwickelt. Der Interviewende hört hauptsächlich zu und beobach-

tet, wie die Personen in ihrer Umgebung, ihrem Kontext agieren[10]. Dazu kann der Interviewende Notizen machen, um eine spätere Auswertung des Interviews zu erleichtern. An passenden Stellen kann der Interviewende gezielte Nachfragen stellen, um beispielsweise Hürden bei der Interaktion mit den Systemen zu provozieren. Eine weitere Intention für Nachfragen kann aber auch sein, die Gedanken und Emotionen des Nutzenden zu erfragen, welche er beim Benutzen des Systems empfindet.



Abbildung 2.2: Während eines Interviews im Kontext können viele Aspekte beobachtet werden, die über das benutzte System selbst hinausgehen[11].

Das Ergebnis eines Interviews im Kontext ist also eine ausführliche Beobachtung der Interaktion von Nutzenden mit den entsprechenden Systemen im alltäglichen Kontext. Diese Beobachtungen können schriftlich festgehalten werden oder auch durch Video- und Tonaufzeichnungen ergänzt werden. Im Nachgang des Interviews müssen die Beobachtungen ausgewertet und analysiert werden. Ziel ist es, aus den Beobachtungen des Interviews konkrete Nutzungsanforderung an das zu entwickelnde System zu formulieren[3]. Hier können beispielsweise Featurelisten, Sequenzdiagramme oder User-Scenarios zum Einsatz kommen[5].

Kapitel 3

Nutzungskontext

Um den Bedarf und Entstehungsprozess der Software besser einordnen zu können, werden nun sowohl der Kontext als auch die Prozessabläufe der Nutzergruppe kurz skizziert. Die Beschreibung der Situation im Team hilft zu erkennen, wie sich die aktuelle Softwarelösung in den Arbeitsalltag eingliedert, welche Prozessabläufe bereits gut durch Software begleitet werden, und an welchen Stellen noch Optimierungsbedarf besteht.

3.1 Studienberatung der Universität Kassel

Als Nutzergruppe in dieser Bachelorarbeit werden die Mitarbeitenden der Abteilung *Studium und Lehre* der Hochschulverwaltung an der Universität Kassel dienen. Zu deren alltäglichen Aufgaben gehört es, alle erdenklichen Organisationen zu übernehmen, die Studierenden und Lehrenden ein erfolgreiches Zusammenarbeiten an der Universität ermöglichen. Hierzu gehören beispielsweise das Einschreiben und Exmatrikulieren von Studierenden, die Durchführung des Bewerbungsverfahrens, das Betreiben der Information Studium und die allgemeine Studienberatung. In dieser Arbeit wird der Fokus auf die Mitarbeitenden der Studienberatung als Teilgruppe der Abteilung Studium und Lehre gesetzt.

Die allgemeine Studienberatung der Universität Kassel berät Studierende zu allen Fragen rund um das Studium. Insbesondere bei persönlichen Problemen mit der Fertigstellung des eigenen Studiums hilft die Studienberatung mit einem persönlichen Lösungsgespräch und kann an weitere fachspezifische Beratungsstellen weiter vermitteln. Des Weiteren bietet die allgemeine

Studienberatung verschiedene Workshops und Seminare an. Hierbei können sich Studierende mit Fokus auf bestimmten Fragestellungen austauschen und Qualifikationen im Umgang mit herausfordernden Studiensituationen erlangen. Auch Schnupperkurse für Schüler werden von der allgemeinen Studienberatung angeboten um den Studieninteressierten einen möglichst unmittelbaren Einblick in den Studienalltag zu gewähren[12].

Eines der zentralen Themen im Alltag der Studienberatung sind Beratungstermine. Studienberatende müssen Termine mit den Ratsuchenden vereinbaren und abstimmen. Beratungstermine können über verschiedene Kontaktkanäle stattfinden: Es ist eine telefonische Beratung oder auch eine Beratung über eine Videokonferenz möglich. Ebenso ist es auch möglich einen persönlichen Beratungskontakt vor Ort zu vereinbaren. Über all diese Termine muss jeder Studienberatende den Überblick behalten und gleichzeitig neue Terminanfragen schnell beantworten können. Um diesen Prozess zu erleichtern und mögliche Fehler, wie beispielsweise Terminüberschneidungen, zu minimieren, wird hierfür die Software **Stubegru** eingesetzt.

3.2 Aktuelle Softwarelösung: Stubegru

Stubegru ist ein umfangreiches Softwarepaket für akademische Beratungsstellen. Die webbasierte Groupware begleitet viele Arbeitsabläufe im Alltag einer Beratungsstelle an einer Hochschule. In einem Softwaresystem vereint Stubegru eine Wissensdatenbank in Form eines Wikis, sowie ein Dashboard mit vielen Modulen für spezifische Workflows. So können über Stubegru an der Universität Kassel beispielsweise Abwesenheiten der Abteilung, Telefonnotizen und Beratungskontakte verwaltet werden. Jeder Mitarbeitende der Abteilung hat über einen eigenen Account Zugriff auf die Software, die er im Browser aufrufen kann. Die Software hilft dabei, tagesaktuelle Informationen schnell und übersichtlich allen Mitarbeitenden zur Verfügung zu stellen und bei Bedarf langfristige und ausführliche Informationen mit wenigen Klicks zur Verfügung zu stellen[13].

Das wichtigste Modul der eingesetzten Software für die Studienberatung ist der Kalender zur Terminvereinbarung von Beratungsterminen. Über dieses Modul können in einem zweistufigen Prozess Termine für Ratsuchende freigegeben und an die entsprechenden Studierenden und Studieninteressierten vermittelt werden.

Zeitslots erstellen

Im ersten Schritt können die Studienberatenden freie Zeitslots für ihre Beratungstermine anlegen. Diese Zeitslots zeigen an, dass der entsprechende Beratende in der eingestellten Zeitspanne potenziell Zeit für ein Beratungsgespräch hat. Bei der Erstellung der Zeitslots können weitere Attribute wie der Beratungskanal (Online Meeting, Telefongespräch oder Präsenztermin) konfiguriert werden. Außerdem können Mail Templates verknüpft werden, die im Falle einer Terminvergabe den Ratsuchenden per Email über alle wichtigen Informationen zum Termin informieren.

Terminvergabe durch Erstinformation

Im zweiten Schritt werden die eingestellten Zeitslots durch Hilfskräfte der Erstinformation an Ratsuchende vergeben. Die Erstinformation der Universität Kassel berät Studierende und Studieninteressierte zu allen Fragen rund ums Studium übers Telefon, Email und an einer Servicetheke vor Ort. Bei tiefgehenden Fragen und spezifischen Anliegen verweisen die Mitarbeitenden an die entsprechenden Sachbearbeitenden oder Beratungsstellen. Die Erstinformation ist auch für das Vereinbaren von Beratungsterminen mit der allgemeinen Studienberatung verantwortlich. Sind die Mitarbeitenden der Erstinformation in Kontakt mit einem Kunden, der einen Termin in Anspruch nehmen möchte, können sie in der Software alle freien Zeitslots der Beratenden einsehen und einen passenden Termin mit dem Kunden vereinbaren. Wenn ein freier Zeitslot vergeben wird und fest mit einem Kunden verknüpft ist, wird eine Email an den Beratenden versendet, die über alle Details wie Adresse, Kontaktinformationen und Anliegen der Ratsuchenden informiert. Des Weiteren wird eine Mail an den Kunden versendet, die auf dem zuvor verknüpften Mailtemplate aufbaut und dynamisch terminrelevante Informationen einsetzt, wie beispielsweise Datum und Uhrzeit des Termins, oder eine Wegbeschreibung zum Beratungsraum.

Auskunft bei Terminabsage

Eine weitere Verwendung des Kalendermoduls tritt ein, wenn Kunden der Erstinformationen Fragen zu einem bereits vereinbarten Termin haben oder diesen absagen möchten. In diesem Fall können die Hilfskräfte der Erstinformation über eine Suchfunktion gezielt nach den vereinbarten Terminen des

Kunden suchen und weitere Auskünfte geben.

Datenschutz

Da Datenschutz in Beratungsszenarien eine wichtige Rolle einnimmt, kann lediglich der verantwortliche Beratende das Anliegen der ratsuchenden Person einsehen. Zu Auskunftszecken können aber alle Mitarbeitenden der Abteilung sehen, wann ein Beratungstermin mit welchem Beratenden vereinbart wurde. Datensätze zu vergangenen Beratungsterminen werden täglich gelöscht, sodass möglichst wenig personenbezogenen Daten in der Datenbank gespeichert werden müssen. Über ein differenziertes Berechtigungssystem der Software Stubegru kann genau gesteuert werden, welche Nutzungsgruppen Beratungstermine anlegen und vergeben dürfen.

3.3 Historische Entwicklung der Software St**ubegru**

Die Software Stubegru wurde ursprünglich von mir als Hilfskraft der Abteilung Studium und Lehre an der Universität Kassel erstellt und betreut. Da die Software nun langfristig an der Uni Kassel und auch an anderen deutschen Hochschulen eingesetzt werden soll, wurde ein Prozess gestartet, um eine Professionalisierung und nachhaltige Betreuung der Software zu gewährleisten. In diesem Rahmen wurde die Software auch für andere Hochschulen zur Verfügung gestellt und unter einer OpenSource Lizenz veröffentlicht. Im Zuge dieser Veröffentlichung wurde in Zusammenarbeit mit der Hochschule Bremen eine grundlegend überarbeitete Variante der Software Stubegru erstellt, die im Vergleich zu der bisherigen Version deutlich flexibler ist und mehr Anpassungsmöglichkeiten bietet. Wie Erich Gamma in *Elemente wieder verwendbarer objektorientierter Software* betont, bieten wieder verwendbar gestaltete Softwarestrukturen und abstrakte Implementierungsansätze die Option, Softwaremodule in verschiedenen Kontexten zu verwenden. Dies erfordert im Softwareentwurf allerdings eine vorausschauende Planung und einen hohen Abstraktionsgrad[14]. Somit ist der Einsatz der Software St**ubegru** nun an verschiedenen Hochschulen mit verschiedenen Arbeitsabläufen realisierbar. Um von dieser neuen, überarbeiteten Softwareversion auch an der Universität Kassel zu profitieren, ist wiederum eine grundlegende Überarbeitung des Moduls zur Terminvergabe der Beratungstermine für die all-

gemeinen Studienberatung notwendig. Dieser Prozess soll im Rahmen dieser Bachelorarbeit begleitet, wissenschaftlich untermauert und evaluiert werden.

Fehlende Features

In der bisherigen Softwareversion gibt es einige Features, die noch nicht vollständig funktionieren oder nicht optimal auf den tatsächlichen Arbeitsalltag zugeschnitten sind. An diesen Stellen soll das neue Modul zur Terminvereinbarung verbessert und noch weiter an die Bedürfnisse der Nutzenden angepasst werden. Im Rahmen der Softwareüberarbeitung mit der Hochschule Bremen wurde für die neue Version von Stubegru bereits ein Kalendermodul zur Vergabe von Beratungsterminen entwickelt. Dieses Modul weist allerdings noch einige Probleme auf, um reibungslos im Arbeitsalltag der allgemeinen Studienberatung an der Universität Kassel eingesetzt zu werden. In der *Bremer Version* läuft das Erstellen und Vergeben eines Beratungstermins in einem einzigen Schritt ab. Ein zentraler Punkt, um die überarbeitete Software auch in der allgemeinen Studienberatung der Universität Kassel nutzen zu können, ist der zweistufige Prozess der Terminvergabe. Hier müssen Beratende die Möglichkeit haben, zuerst freie Terminslots freizugeben, die dann in einem getrennten zweiten Schritt durch Mitarbeitende der Erstinformation an Ratsuchende vergeben werden können.

Welche Anpassungen im Detail notwendig sind, um die Software optimal in der Studienberatung einsetzen zu können, soll in den folgenden Kapiteln methodisch herausgearbeitet werden und durch Dokumentation von praktisch durchgeföhrten Nutzerstudien und Gesprächen mit den verantwortlichen Personen ergänzt werden. Diese Bachelorarbeit soll insbesondere den Designprozess strukturiert begleiten und wissenschaftliche Methoden aufzeigen, um Softwareentwicklern und Nutzenden eine möglichst gute Zusammenarbeit zu ermöglichen.

Kapitel 4

Nutzungsanforderungen

4.1 Interview im Kontext

Das Modul zur Terminvereinbarung der Software Stubegru soll in Bezug auf den Arbeitsalltag der allgemeinen Studienberatung der Universität Kassel überarbeitet werden. Dies soll mit Methoden des Human Centered Design umgesetzt werden. Ein zentraler Bestandteil des Human Centered Design ist der enge und stetige Austausch mit den Betroffenen des Softwaresystems[4]. Um den Änderungsbedarf eines bestehenden Softwaresystems einschätzen zu können, wird im Human Centered Design häufig die Methode des *Interviews im Kontext* gewählt[9]. Diese Methode eignet sich besonders zu Beginn des Entwicklungsprozesses, da wenig Vorkenntnisse über die eingesetzte Software und das Umfeld, in dem diese eingesetzt wird, bekannt sein muss. Die Softwareentwickler können so einen guten Einstieg finden, um einen Überblick zu gewinnen, welche Funktionen die fertige Software am Ende unterstützen muss. Auch lässt sich durch ein genaues Beobachten beim Interview herausarbeiten, in welchem Kontext die Software im tatsächlichen Arbeitsalltag genutzt wird und welche weiteren Faktoren die Betroffenen der Systeme beeinflussen.

Rahmenbedingungen des Interviews im Kontext

Als erster Schritt wurde ein Termin für ein Interview im Kontext mit Herr Maier¹ vereinbart. Herr Maier ist einer von drei Mitarbeitenden der allgemei-

¹Aufgrund des Datenschutzes wurde der Name anonymisiert

nen Studienberatung der Universität Kassel. Zu seinen Aufgaben gehört die Betreuung der Software Stubegru und deren Einsatz in der Abteilung Studium und Lehre. Seit über sechs Jahren arbeitet Herr Maier bereits gemeinsam mit Hilfskräften an dem Aufbau und der Optimierung der Software Stubegru, um den täglichen Arbeitsalltag seines Teams optimal zu unterstützen. Ich habe mich persönlich mit Herr Maier in seinem Büro im Campus Center der Universität getroffen. Dort hat er mir an seinem Schreibtisch gezeigt, wie er mit der alten Version der Software Beratungstermine erstellen und vergeben kann. Herr Maier saß vor mir und hatte Maus und Tastatur in der Hand. Ich saß hinter ihm auf einem Stuhl und habe auf einem iPad Notizen mitgeschrieben. Für die Dauer von einer Stunde hat Herr Maier mir gezeigt, wie er die Software aktuell nutzt, welche Features für ihn sehr wichtig sind und an welchen Stellen noch Verbesserungspotenzial besteht.

Detaillierter Ablauf des Interviews

Am Anfang habe ich Herr Maier gebeten, mir einmal zu zeigen, wie er einen Beratungstermin in der Software anlegen und vergeben kann. Dies ist der Workflow, der im Arbeitsalltag am häufigsten vorkommt und daher eine hohe Priorität im Designprozess hat. Herr Maier klickte sich durch die verschiedenen Eingabefelder, um einen freien Zeitslot für einen Beratungstermin anzulegen. Hierbei erwähnte er, dass es ganz wichtig ist, dass Datum und Uhrzeit des Beratungstermins mit wenigen Klicks über einen **Timepicker** mit der Maus eingeben werden können. Eine Datumseingabe über die Tastatur würde er nicht bevorzugen.

The screenshot shows a user interface for creating a new appointment. It features a title field ('Titel der Veranstaltung') containing 'Beratungstermin'. A date picker is set to '10.10.2022'. Time inputs show '11:00' and '12:00'. A dropdown for 'Kanal' is set to 'Alle', and another for 'Eigentümer' is set to 'Johannes Schnirring'. Below the date picker is a dropdown menu with the text 'variablen test'. At the bottom right are buttons for 'Raum bearbeiten', 'Neuen Raum anlegen', 'Vorlage bearbeiten', and 'Neue Vorlage anlegen'. The bottom navigation bar includes 'Abbrechen' and 'Termin Speichern'.

Abbildung 4.1: Datepicker im Formular zur Erstellung eines Zeitslots.

Beim Eintragen mehrerer Termine wäre es auch besonders praktisch, dass das zuvor eingegebene Datum stehen bleibt und direkt ein weiterer Zeitslot für den gleichen Tag angelegt werden kann, ohne dass er nochmal extra das Datum auswählen muss. Die meisten weiteren Felder sind Dropdown Menüs, mit wenigen Elementen. Die Auswahl der richtigen Werte kann Herr Maier schnell vornehmen. Bei der Auswahl der verknüpften Räume werden beispielsweise die Räume, die mit seinem Nutzeraccount verknüpft sind, ganz oben in der Auswahlliste angezeigt. Da eine Beratung in der Regel in den eigenen Räumen stattfindet, ist hier eine schnelle Auswahl für den Normalfall möglich. In einer Spezialsituation, in der ein größerer Beratungstermin beispielsweise in einem gemeinsamen Gruppenraum stattfindet, ist aber auch solch eine Auswahl möglich.

The screenshot shows a dropdown menu titled 'Raum'. It lists four options: '1603 - Johannes Raum', '1603 - Johannes Raum', '123 - Marias Raum', and '1428 - Peters Raum'. The first two options are highlighted with a blue background, indicating they are the same room. The third and fourth options are in a standard grey font.

Abbildung 4.2: Dropdown zur Auswahl des Beratungsraums. Der eigene Raum wird immer als oberstes angezeigt.

Nachdem der Zeitslot für den Termin angelegt ist, wird der entsprechende

Tag in der Kalenderübersicht nun grün hinterlegt. Dies ist ein Zeichen für die Hilfskräfte der Erstinformation, dass an diesem Tag noch freie Zeitslots verfügbar sind.

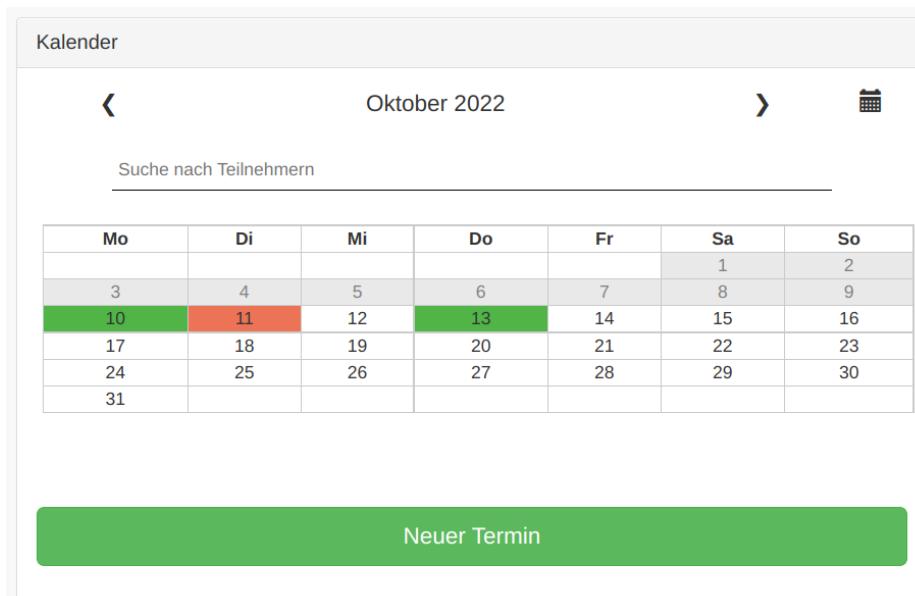


Abbildung 4.3: Kalenderübersicht. Grüne gefärbte Tage zeigen noch freie Zeitslots an. Rot gefärbte Tage weisen auf vergebenen Zeitslots hin.

Wenn Herr Maier den Mauszeiger über den entsprechenden Tag in der Monatsübersicht bewegt, kann man die genauen Termine mit Informationen über die Uhrzeit, den zuständigen Beratenden und die Anzahl der freien Plätze sehen. Herr Maier erklärte mir, dass die kompakte Monatsansicht mit den farblich hervorgehobenen Terminslots bereits eine sehr gute Lösung ist, damit die Hilfskräfte auf einen Blick erfassen können, an welchen Tagen sie den Kunden noch Beratungsgespräche anbieten können. Sobald alle Plätze der Beratungstermine an einem Tag vergeben sind, wird dieser im Kalender rot markiert. „So sehen Hilfskräfte mit einem Blick sofort, dass sie hier keinen Termin mehr vergeben werden können“, erklärt Herr Maier [15].

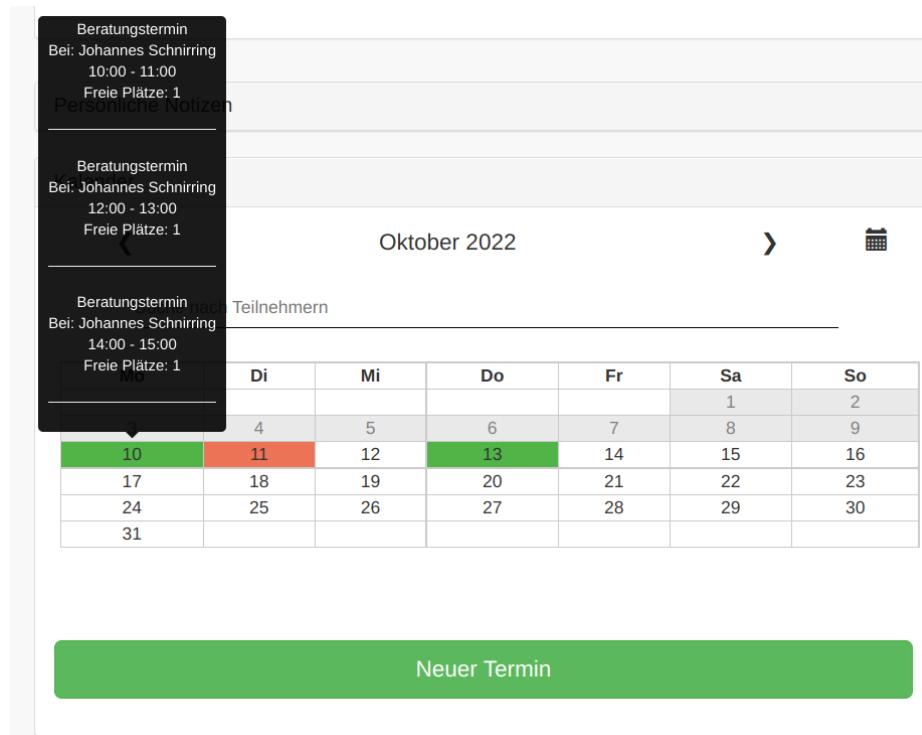


Abbildung 4.4: Bewegt man den Mauszeiger über einen Tag, erscheinen weiteren Informationen zu den Zeitslots an diesem Tag.

Soll nun ein Zeitslot tatsächlich vergeben werden, klickt man auf den entsprechenden Tag in der Monatsansicht und es öffnet sich ein **Modal**. Dies ist ein Fenster, welches sich über den anderen Bildschirminhalt legt und dem Nutzer somit deutlich anzeigt, dass hier eine Aktion im neu geöffneten Fenster notwendig ist. Herr Maier zeigt mir, wie die Mitarbeitenden der Erstinformation in diesem Detail- **View** die freien Zeitslots an die ratsuchenden Personen vergeben können. In einer Liste werden, nach Uhrzeit sortiert, alle Termine untereinander angezeigt. Neben jedem freien Termin steht ein Button zum Vergeben dieses Zeitslots zur Verfügung.

The screenshot shows a list of three available appointment slots for 'Beratungstermin' on '10.10.2022'. Each slot is associated with 'Johannes Schnirring' and includes a time range and a count of 'Noch frei: 1'. A 'Vergeben' button is present next to each slot, and a red 'X' icon is to its right. At the bottom right of the list area is a 'Zurück' button.

Abbildung 4.5: Der Detail-View: Eine Liste mit drei freien Zeitslots.

Herr Maier zeigt mir, wie eine Hilfskraft der Erstinformation nun einen solchen Zeitslot vergeben könnte. Nach Klick auf den *Vergabe-Button* klappt ein Formular auf, in dem Name, Kontaktdaten und Anliegen der Ratsuchenden erfasst werden können.

This screenshot displays a detailed booking form. It starts with two appointment slots from the previous screen. Below them is a large gray input area containing several fields:
Vor- und Nachname: Theodor Test
E-Mail: test@example.com
Telefon: 1234
Fragebogen: Nein
Kanal: Persönlich (dropdown)
Anliegen: Theodor Test hat Hunger und möchte wissen, was es in der Mensa gibt :-)
At the bottom of this area is a red footer bar with the text 'Unbedingt Bestätigen oder Abbrechen' and two buttons: 'Abbrechen' on the left and 'Bestätigen' on the right.

Abbildung 4.6: Formular zum Vergeben eines Zeitslots an eine ratsuchende Person.

Nachdem alle personenbezogenen Daten korrekt erfasst wurden, kann der Termin nun endgültig gebucht werden. Hierzu klicken die Hilfskräfte auf den But-

ton *Bestätigen*. Herr Maier erklärt mir, dass dies ein sehr wichtiger Schritt ist: Solange ein Mitarbeitender der Erstinformation das Formular zum Erfassen der persönlichen Daten des Ratsuchenden geöffnet hat, wird dieser Zeitslot mit einer Sperre versehen. So wird verhindert, dass dieser Zeitslot von einem Kollegen vergeben werden kann, während man selbst gerade mit dem Ratsuchenden am Telefon die persönlichen Daten und das Anliegen bespricht. Sollte nach dem Aufklappen des Formulars der entsprechende Zeitslot doch nicht vergeben werden, ist es deshalb notwendig, dass die terminvergebende Person auf *Abbrechen* klickt, um die Sperre dieses Zeitslots aufzuheben und ihn so mit für die Kollegen wieder freizugeben. Herr Maier betont, dass dieser Schritt manchmal nicht ganz intuitiv ist und für die Hilfskräfte daher in Einführungsschulungen immer besonders hervorgehoben wird. Es wäre allerdings deutlich schlimmer, einen Termin doppelt zu vergeben und somit mindestens einem Kunden wieder absagen zu müssen, als einen Zeitslot versehentlich zu sperren.

10.10.2022		
Beratungstermin Bei: Johannes Schnirring	10:00 - 11:00 Noch frei: 1	Vergeben
Beratungstermin Bei: Johannes Schnirring	12:00 - 13:00 Noch frei: 0	
Vergeben an:		
Theodor Test		Kanal: personal
Beratungstermin Bei: Johannes Schnirring	14:00 - 15:00 Noch frei: 1	Vergeben

Abbildung 4.7: Detail-View: Ein Zeitslot wurde nun vergeben und ist für den entsprechenden Kunden reserviert. Hilfskräfte können nur den Namen des Ratsuchenden einsehen.

Ist der Termin nun erfolgreich vergeben, können alle Nutzenden der Software einsehen, an welche Person dieser Termin vergeben wurde. Meldet sich ein Ratsuchender beispielsweise einige Tage später noch einmal bei der Erstinformation und möchte wissen, wann sein Beratungstermin stattfindet, können die Mitarbeitenden der Erstinformation diese Auskunft aus der Software ablesen. Aus Datenschutzgründen können allerdings keine weiteren personenbezogenen Daten des Beratungstermins ausgelesen werden. Lediglich der Studienberatende, bei dem der Termin stattfindet, bekommt beim Aufruf des

Detail-Views weitere Details wie Kontaktdaten und Anliegen der ratsuchenden Person angezeigt.

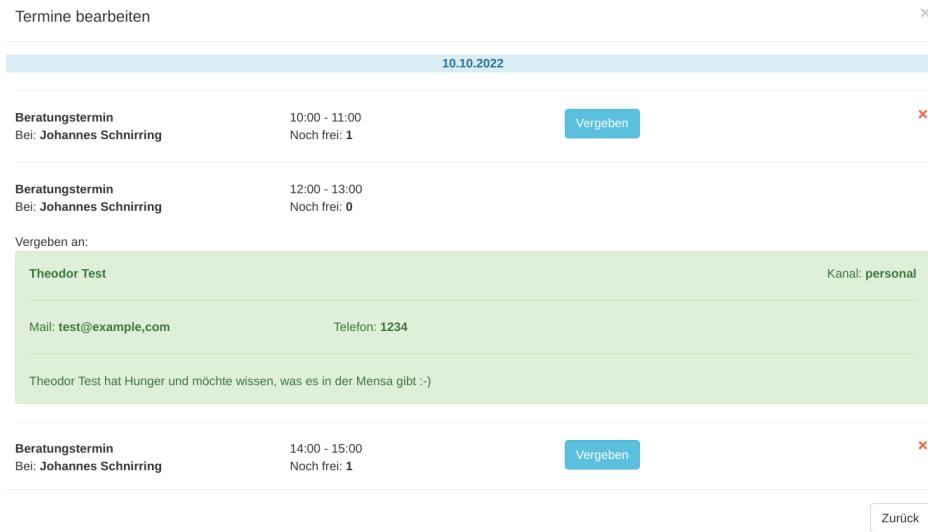


Abbildung 4.8: Detail-View: Der verantwortliche Beratende kann weitere personenbezogene Details einsehen.

Herr Maier hat nun den zweistufigen Workflow zur Terminvergabe einmal komplett durchgespielt und mich auf viele Details hingewiesen. Während Herr Maier mir gezeigt und erzählt hat, wie die Terminvergabe in der aktuellen Softwareversion abläuft, habe ich in Stichworten mitgeschrieben, welche Bemerkungen und Auffälligkeiten er besonders betont hat.

4.2 Auswertung des Interviews

Während bisher der detaillierte Ablauf des Interviews geschildert wurde, sollen im Folgenden die wesentlichen Kernaspekte nochmals zusammengefasst werden, die während des Interviews notiert wurden. Das Augenmerk liegt hierbei auf Beobachtungen, die Konsequenzen für den Designprozess des neuen Kalendermoduls zur Terminvergabe hervorbringen.

Methode der Auswertung

Während dem Interview habe ich mir alle relevant erscheinenden Aussagen von Herr Maier auf einem iPad notiert. Wurden im weiteren Gesprächsverlauf noch ergänzende Informationen zu den einzelnen Punkten deutlich, habe ich diese in den Notizen stichpunktartig an die entsprechenden Themen angefügt. Im Nachgang des Interviews mussten diese Notizen nun sorgfältig analysiert und ausgewertet werden. Hierzu bin ich die einzelnen Themen durchgegangen und habe die entsprechenden Ansichten und Klickpfade in der Software nochmals nachgespielt. In einem neuen Dokument habe ich nun die herausgearbeiteten Problematiken zusammengefasst, um die zu Grunde liegenden Zusammenhänge klarzustellen und zu spezifizieren. Dies entspricht dem zweiten Schritt *Nutzungsanforderungen spezifizieren* im iterativen Design Zyklus des Human Centered Design nach ISO 9241[8].

Spannende Erkenntnisse

Das Interview im Kontext hat viele spannende Beobachtungen geliefert, die hier nicht alle detailliert ausgeführt werden können. Daher sollen im Folgenden nun drei verschiedene Themen exemplarisch vorgestellt werden, die während des Interviews aufgefallen sind. Im weiteren Verlauf dieser Arbeit wird in den unterschiedlichen Phasen des Gestaltungsprozesses immer wieder auf diese Themen Bezug genommen.

Kompakte Monatsübersicht

In der alten Softwareversion, die an der Uni Kassel bisher zum Einsatz kam, werden alle freien und vergebenen Zeitslots der Beratungstermine in einer tabellarischen Monatsansicht dargestellt.



Abbildung 4.9: Tabellarische Ansicht der Zeitslots mit Einfärbungen der einzelnen Tage.

Durch die farblichen Markierungen der einzelnen Tage können die Nutzenden auf einen Blick erfassen, ob an diesem Tag Beratungsslots eingetragen wurden und ob unter den eingetragenen Zeitslots noch freie Termine vorhanden sind. Ein grün markierter Tag bedeutet, dass an diesem Tag noch mindestens ein freier Beratungsslot vorhanden ist. Ein rot markierter Tag bedeutet, dass an diesem Tag Beratungstermine stattfinden, diese allerdings bereits alle an ratsuchende Personen vergeben sind. In der überarbeiteten Version der Stubegru Software, die in Zusammenarbeit mit der Hochschule Bremen entstanden ist, wurde diese kompakte tabellarische Übersicht durch eine größere umfangreiche Ansicht ausgetauscht, die durch die Bibliothek *fullcalendar* bereitgestellt wird.

Mon	Tue	Wed	Thu	Fri
	31	1	2	3 4
	7	8 9 ● 10:00 Beratungste ● 12:00 Beratungste	10 ● 10:00 Beratungste ● 12:00 Beratungste	11
14 ● 10:00 Beratungste ● 12:00 Beratungste	15	16	17 ● 10:00 Beratungste	18
21 ● 12:00 Beratungste	22	23	24	25
	28	29 30	1	2
	5	6	7	8 9

Abbildung 4.10: Monatsübersicht der Beratungstermine in der *Bremer Version*.

Diese neue Ansicht ermöglicht auf den ersten Blick zu sehen, zu welcher Uhrzeit die Termine stattfinden und einzelne Termine aus der Monatsübersicht direkt anzuklicken. Allerdings bietet diese Ansicht keine Möglichkeit, Tage je nach freien Plätzen rot oder grün darzustellen. Dies ist jedoch ein wichtiges Feature für die zweistufige Terminvergabe an der zentralen Studienberatung der Universität Kassel. An dieser Stelle braucht es eine Lösung um den Hilfskräften der Erstinformation auf den ersten Blick anzusehen, ob sie an diesem Tag noch freie Terminslots vergeben können.

Suche nach Teilnehmern

Manchmal kommt es vor, dass Ratsuchende, die bereits einen Beratungstermin vereinbart haben, nochmals in Kontakt mit der Erstinformation treten, um weitere Fragen zum Termin zu stellen. Auch kommt es vor, dass das

genaue Datum oder die Uhrzeit vergessen wurden. In diesem Fall sollen die Hilfskräfte der Erstinformation möglichst schnell Auskunft über die angefragten Details geben können. Hierfür immer alle vergebenen Beratungstermine manuell durchzulesen, ist zeitlich ein großer Aufwand. Es braucht also ein Feature, sodass die Mitarbeitenden der Erstinformationen direkt nach Terminen und weiteren organisatorischen Daten dieser Termine suchen können. Wenn Ratsuchende beispielsweise am Telefon ihren Namen nennen, werden sie manchmal nicht einwandfrei verstanden. Ein Suche nach Teilnehmernamen der Termine sollte also auch funktionieren, wenn der Name nicht exakt in der gleichen Schreibweise eingegeben wird, wie er im Datensatz des Beratungstermins in der Datenbank hinterlegt ist.

Darstellung der Telefonnummern

In der Regel wird bei einer Terminvergabe die Telefonnummer der ratsuchenden Person erfasst. Der zuständige Studienberatende kann den Datensatz bei Bedarf aufrufen und diese Telefonnummer einsehen. Dies passiert in der Regel, wenn der Beratende vor einem Beratungstermin nochmals telefonisch Details mit der ratsuchenden Person abklären möchte. Der Berater wählt also die angezeigte Telefonnummer in seinem Telefon. Während des Interviews im Kontext zeigte sich, dass die Eingabe längerer Telefonnummern manchmal Fehler mit sich bringt, da Ziffern vertauscht oder vergessen werden. Den Beratenden wäre hier eine wertvolle Hilfe an die Hand gegeben, wenn eine Darstellung langer Telefonnummern möglich wäre, die ein direktes und intuitives Eintippen in die Telefontastatur erleichtert.

4.3 Gestaltungslösungen entwickeln

Nachdem nun die Problematiken und Herausforderungen des neuen Softwaremoduls verdeutlicht wurden, sollen im nächsten Schritt konkrete Ideen entwickelt werden, wie die erkannten Problematiken und Anforderungen in der Praxis umgesetzt werden können. Alan Dix betitelt diese Phase in *Human Computer Interaction* als *Requirements specification* und betont, dass der Fokus in diesem Schritt darauf liegt, die notwendigen Funktionalitäten und Features der Software grob zu beschreiben. Von besonderer Bedeutung in diesem Schritt des Designzyklus sind Zusammenhänge und Abhängigkeiten zwischen einzelnen Komponenten. Exakte Implementierungsdetails hingegen

sind in dieser Phase noch nicht von großer Bedeutung und sollten erst im nächste Schritt genauer betrachtet werden[4].

Methode der Erarbeitung

Durch die Auswertung des Interviews im Kontext sind Nutzungsanforderungen an das neue Modul zur Terminvereinbarung entstanden. Um diese lose formulierten Nutzungsanforderungen später implementieren zu können, werden sie in diesem Schritt weiter konkretisiert. Es sollen erste Ideen entstehen, wie die Bedürfnisse der Nutzenden durch einzelne Komponenten der Software umgesetzt werden können. In diesem Fall wird mit Skizzen der einzelnen *Views* und Formulare gearbeitet. Für jedes Szenario, dass Nutzende beim späteren Verwenden der Software durchlaufen, wird eine Skizze erstellt. Durch Markierungen und Notizen an der Skizze werden die Funktionen und Beziehungen der Elemente definiert.

Kompakte Monatsübersicht

Die Übersicht aller Termine eines Monats ist die Ansicht, die Nutzende beim Aufruf der Software als erstes sehen. Dies soll auch in der neuen Softwareversion erhalten bleiben: Den größten Raum nimmt die tabellarische Ansicht der einzelnen Tage des Monats ein. In den einzelnen Feldern werden Terminslots, nach Uhrzeit sortiert, aufgelistet. Neben der Uhrzeit des Termins wird der Titel eines jeden Termins angezeigt. Die einzelnen Termine werden farblich entweder grün oder rot eingefärbt, um auf den ersten Blick zu kennzeichnen, ob es sich um einen freien Terminslot (grün) oder um einen bereits vergebenen Termin (rot) handelt. Wenn an einem Tag viele Zeitslots angelegt werden, wird das Feld für diesen Tag automatisch größer, sodass alle Termine Platz finden. Sollten an jedem Tag sehr viel Termine angelegt werden, könnte die tabellarische Monatsansicht so lang werden, dass sie unter Umständen nicht mehr vollständig auf den Bildschirm passt. Dies wäre unpraktisch, da dann nicht mehr alle Termine eines Monats auf einen Blick erfasst werden könnten. In der Phase der Evaluation sollte diese Problematik berücksichtigt werden und eine Abschätzung getroffen werden, wie viele Termine im praktischen Einsatz tatsächlich pro Tag angelegt werden.

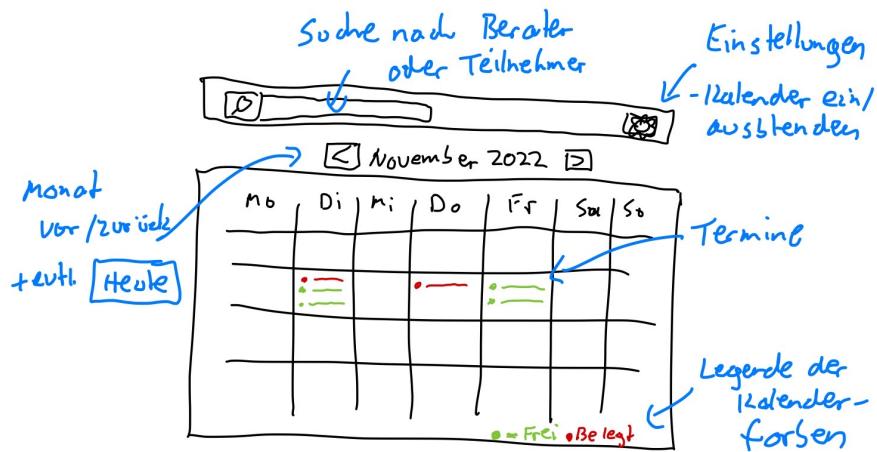


Abbildung 4.11: Monatsübersicht der Beratungstermine mit farblichen Markierungen.

Über der tabellarischen Ansicht der Tage befindet sich eine horizontale Leiste, die den aktuell angezeigten Monat betitelt und Kontrollelemente beinhaltet, um in den vorherigen bzw. nächsten Monat zu wechseln. Über der Leiste mit dem Monat befindet sich eine weitere Kontrollleiste. Diese enthält einen Button, um einen neuen Termin anzulegen. Dieser Button sollte nur für Nutzeraccounts von Beratern sichtbar sein. Hilfskräfte der Erstinformation sollen Zeitslots nur vergeben, aber nicht selbst anlegen können. Daneben befindet sich eine Suchleiste, um schnell nach Namen von ratsuchenden Personen suchen zu können. Ganz rechts gibt es schließlich noch einen Button, um weitere Einstellungen vorzunehmen. Durch einen Klick auf diesen Button mit dem Zahnrad Symbol soll ein Dropdown-Menü aufklappen, in dem Filter für die Ansicht der Termine gesetzt werden können.

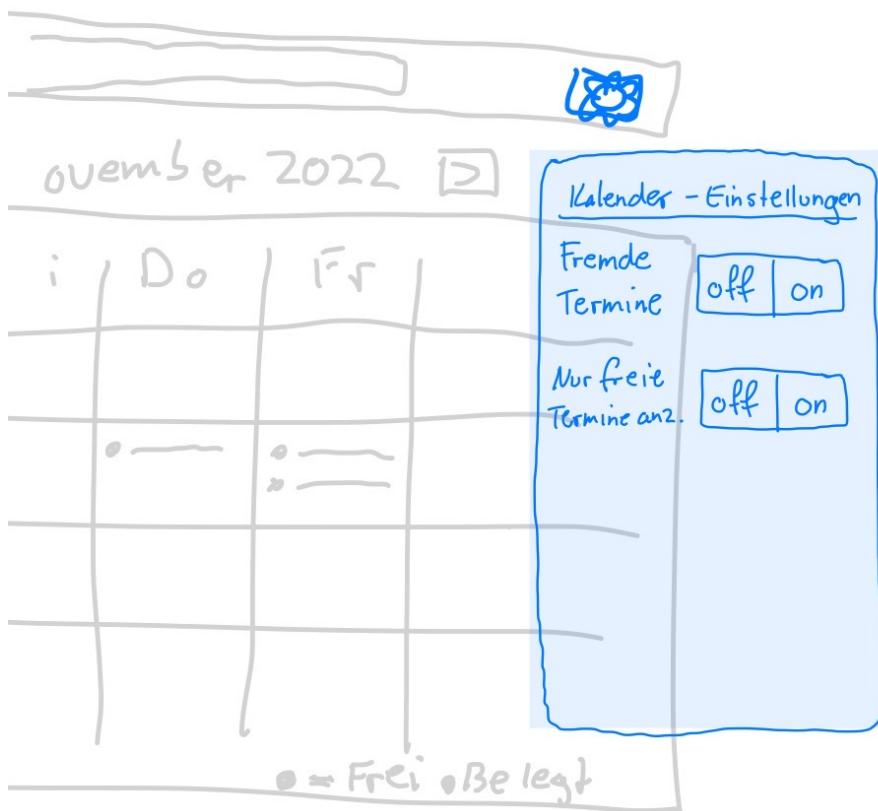


Abbildung 4.12: Filtereinstellungen der Kalenderansicht. Das Dropdown Menü öffnet sich durch Klick auf den Zahnrad Button.

In diesem Menü kann über Toggles eingestellt werden, ob nur eigene Termine oder auch fremde Termine in der Monatsansicht dargestellt werden sollen. Mit *eigenen Terminen* sind Termine gemeint, die den eigenen Benutzeraccount als zuständigen Beratenden hinterlegt haben. Außerdem kann ein Filter gesetzt werden, um ausschließlich freie Termine anzuzeigen. Dies kann besonders für Hilfskräfte bei der Vergabe freier Termine relevant sein, da bereits vergebene Termine in diesem Fall irrelevante Informationen sind, die von freien Zeitslots ablenken.

Suche nach Teilnehmern

Die Suchfunktion ist ein weiterer Aspekt, dem in dieser Ausarbeitung besondere Aufmerksamkeit gewidmet ist. Über das Freitextfeld in der oberen Kontrollleiste können Nutzende nach Namen von Ratsuchenden suchen, an die bereits Termine vergeben wurden. Tippt man einige Buchstaben in das Suchfeld ein, klappt eine Box mit Ergebnisvorschlägen unter der Suchleiste auf und schiebt den restlichen Inhalt (die tabellarische Monatsansicht) nach unten. In dieser Box werden zur Suchanfrage passende Termine dargestellt. Für jeden Termin wird in einer Zeile der Titel, der Name des Ratsuchenden, der Name des Beratenden sowie Datum und Uhrzeit aufgelistet. Neben jedem Datensatz erscheint ein Button mit einem Augensymbol. Durch einen Klick darauf wird der entsprechende Termin in der Detailansicht geöffnet.

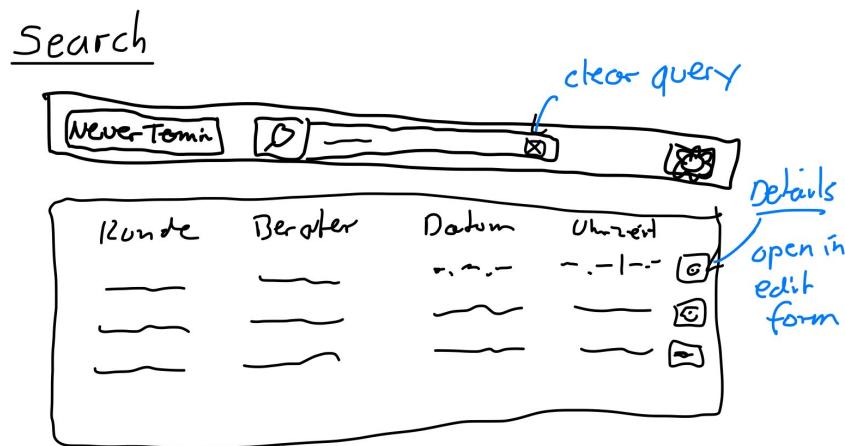


Abbildung 4.13: Suche nach Terminen eines Ratsuchenden mit Ergebnisliste.

Wichtig für die Suchfunktion ist, dass passende Ergebnisse auch angezeigt werden, wenn die Eingabe in der Suchleiste eventuell Fehler enthält oder noch nicht vollständig ist. Durch solche automatischen Ergebnisvorschläge wird das Suchen für die Nutzenden erleichtert und Fehlerquellen minimiert. Dadurch, dass Nutzende schon während dem Tippen der ersten Buchstaben ein aktives und konstruktives Feedback erhalten, fühlt sich die Nutzung der Software dynamischer und flüssiger an[16]. Wenn Mitarbeitende der Erstinformation ihre Kunden am Telefon beispielsweise nicht ganz genau verstehen,

können sie mit diesen automatischen Ergebnisvorschlägen trotzdem den passenden Termin finden. Allerdings muss bei solchen automatisierten Vorschlägen darauf geachtet werden, dass nicht zu viele unnötige oder unpassende Vorschläge angezeigt werden. Dies würde Nutzende von den eigentlich gesuchten Ergebnissen ablenken und sich somit nachteilig auf die User-Experience auswirken[17].

Darstellung der Telefonnummern

Durch einen Klick auf den Termin in der Monatsübersicht öffnet sich die Detailansicht des zugehörigen Termins und weitere Eigenschaften des Datensatzes werden angezeigt. Alternativ kann ein Termin auch über die Suchfunktion gefunden und dann in der Detailansicht aufgerufen werden. In dieser Ansicht können Nutzeraccounts mit der entsprechenden Berechtigung nochmals Details des Termins bearbeiten oder den Termin löschen.

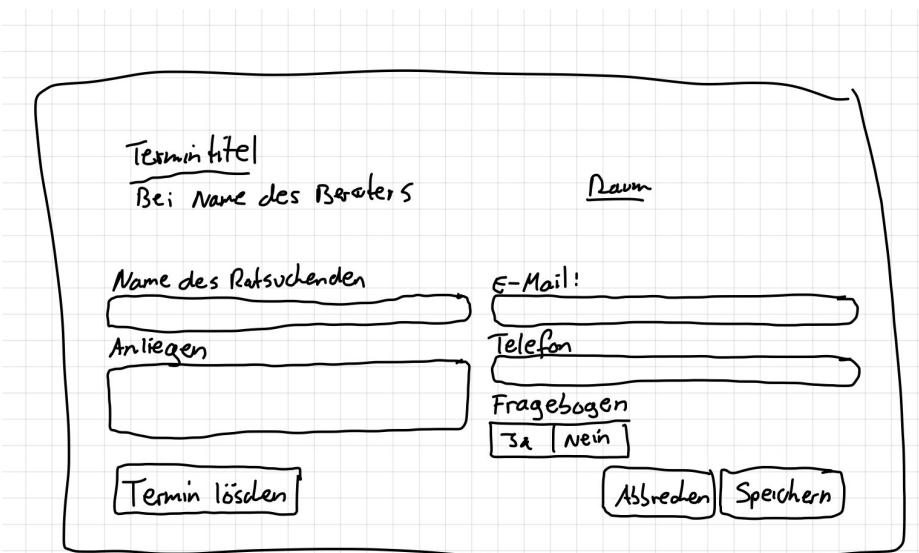


Abbildung 4.14: Detailansicht eines Termins, der bereits an eine ratsuchende Person vergeben wurde.

Wenn Beratende nach der Vereinbarung eines Termins nochmals auf telefonischem Weg Absprachen oder Vorgespräche mit den Ratsuchenden erleben möchten, können sie die Telefonnummer der entsprechenden Person

in der Detailansicht eines Beratungstermins einsehen. Die Beobachtung des Nutzungsverhaltens während des Interviews im Kontext hat gezeigt, dass es umständlich ist, lange Telefonnummern zu erkennen und korrekt in die Tastatur des Telefons einzugeben. Im Gespräch mit Herr Maier kam der Wunsch auf, Telefonnummern an dieser Stelle so zu formatieren, dass sie intuitiver erfasst und abgetippt werden können. Der Standard für das Formatieren von Telefonnummern in Deutschland wird durch DIN 5008 geregelt. Diese Norm beschäftigt sich mit Formatierungsstandards für Briefe und Anschreiben. Hier wird das Trennen der Vorwahl vom Rest der Nummer durch ein Leerzeichen vorgeschrieben. Weitere Formatierung, wie beispielsweise das Aufteilen der Ziffern in kleinere Blöcke wird hier nicht thematisiert[18].

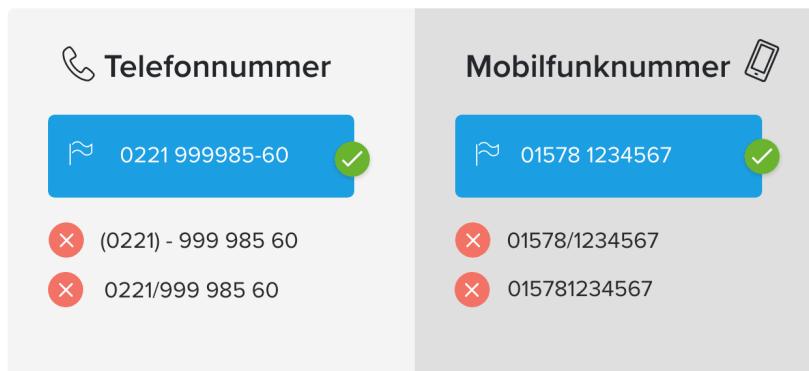


Abbildung 4.15: So werden nationale Festnetz- und Mobilfunknummern nach DIN 5008 richtig geschrieben. Quelle: [19].

Wissenschaftliche Untersuchungen zeigen, dass das Eingeben und Ablesen von Telefonnummern in Interaktion mit den entsprechenden Maschinen ein relevantes Detail ist. Dieser Prozess sollte durch die technischen Systeme möglichst intuitiv und nutzerfreundlich gestaltet werden[20]. Eine Unterteilung der Ziffern in kleinere Blöcke, von beispielsweise vier Ziffern pro Block, erhöht die Lesbarkeit deutlich und ermöglicht es dem menschlichen Gehirn, einen Ziffernblock in einem Blick direkt zu erfassen und auf die Telefontastatur zu übertragen[19][21][22].

Kapitel 5

Gestaltungslösungen implementieren

Im vorherigen Schritt wurden konkrete Nutzungsanforderungen herausgearbeitet und Ideen für passende Gestaltungslösungen formuliert. Das folgende Kapitel soll sich mit der praktischen Umsetzung der Anforderungen beschäftigen. Dieser Prozess entspricht dem dritten Schritt *Gestaltungslösungen entwickeln, die die Nutzungsanforderungen erfüllen* des Designzyklus im Human Centered Design nach ISO 9241[8].

5.1 Kundenwünsche und technische Machbarkeit

Auch wenn im Human Centered Design immer die Nutzenden und nicht das technische System im Fokus stehen sollte, kommt man nicht drumherum, die technischen Details genauer zu betrachten. Dadurch ist es ganz besonders wichtig, in dieser Phase die Bedürfnisse und Erwartungen der Nutzenden nicht aus den Augen zu verlieren. Eine klare und intuitive Schnittstelle für die Nutzenden sollte in diesem Prozess eine höhere Priorität einnehmen als eine Lösung, die technisch am einfachsten umzusetzen ist. K. Holtzblatt verwendet hier den Begriff *Kohärenz* und meint damit, dass sich ein System für den Nutzer so zusammenhängend und natürlich wie möglich anfühlen soll: „The challenge is to keep the system work model coherent, so that it supports the users and fits with their expectations while extending and transforming their work practice as prescribed by the vision.“[9] Gleichzeitig müssen al-

le Lösungsansätze natürlich auch tatsächlich programmiert werden können. Hierbei kann nicht ignoriert werden, dass viele externe Faktoren die Machbarkeit bestimmter Lösungskonzepte in der Praxis einschränken. G.A. Boy erwähnt in *The Handbook of Human-Machine Interaction* einige dieser Faktoren: „Design work is often constrained by various external factors in the development organization and the marketplace (policies, standards, competitive products, past and planned products, schedules, resource budgets).“[3] So ermöglichen verwendete Programmiersprachen, eingesetzte Frameworks und bereits existierende Softwareteile es, bestimmte Konzepte sehr elegant umzusetzen. Andere Gestaltungsideen sind, beschränkt durch äußere Faktoren, unter Umständen gar nicht realisierbar[3].

5.2 Technische Grundlagen Stubegru

In dem konkreten Fall des Moduls zur Terminvergabe für die Studienberatung ist hier ein wichtiger Gesichtspunkt, dass es sich um ein Modul innerhalb eines bereits bestehenden Softwarepakets handelt. **Tech Stack**, Programmiersprache und Schnittstellen zu anderen Modulen sind hier bereits vorgeben und können nicht allein durch die Wünsche der Nutzenden geformt werden.

Tech Stack

Im Folgenden sollen nun die technischen Gegebenheiten und Grundlagen der Software Stubegru vorgestellt werden. Das neue System zur Terminvereinbarung muss sich als Modul in das Softwarepaket Stubegru einbinden lassen und somit einige Standards und Schnittstellen der Software zur Verfügung stellen.

Die Software Stubegru ist webbasiert und wird über einen Browser aufgerufen. Dementsprechend sind die verwendeten Technologien im Frontend **Html**, **Css** und **Javascript**. Die Kommunikation mit dem Backend wird durch asynchrone Requests umgesetzt, die durch Javascript Methoden initiiert werden. Das Backend besteht aus vielen einzelnen **Php** Dateien, welche die per **Http** übermittelten Daten auslesen und aufarbeiten. Als Datenbank kommt eine relationale **MySQL** Datenbank zum Einsatz, die von den Php Skripten über **Php Data Objects** angesprochen wird. Die abgerufenen Daten der Php Skripte werden, als **JSON** codiert zurück an das Frontend geschickt und dort von Javascript Methoden weiterverarbeitet. Mithilfe der Javascript

Library **jQuery** werden die entsprechenden Elemente im **Document Object Model** angepasst und somit für den Nutzenden grafisch dargestellt. Um den Nutzenden eine einheitliche und vertraute Nutzungsoberfläche zu bieten, kommt das Framework **Bootstrap** zum Einsatz.

Das Softwarepaket Stubegru ist grundlegend modular aufgebaut, sodass für jede Funktionalität oder jeden Prozess ein eigenes Modul programmiert wird, das genau diese Aufgabe übernimmt. Jedes Modul besteht aus den entsprechenden Anzeigeelementen, die in einer Html Datei formuliert werden. Des weiteren aus grafischen Designregeln, die als CSS Datei angelegt werden und der eigentlichen Logik, die in Javascript implementiert werden muss. Im Modul zur Terminvereinbarung kommt zusätzlich die Javascript Library **fullcalendar** zum Einsatz. Diese bietet eine einfache Schnittstelle, um Termindaten in einer monatlichen Übersicht darzustellen und bietet den Nutzenden standardisierte Kontrollelemente zum Interagieren mit der Kalenderansicht. Funktionalitäten wie das Rendern einer Monatsübersicht oder das Wechseln zwischen den einzelnen Monaten sind über diese Bibliothek bereits abgedeckt und müssen nicht eigenständig implementiert werden. Gleichzeitig schränkt die Verwendung dieser Bibliothek die Möglichkeiten in der Darstellung der Termine ein, sodass lediglich die angebotenen Ansichten (Monatsansicht, Wochenansicht, Tagesansicht, Terminliste) eingesetzt werden können.

5.3 Nutzungsanforderungen formalisieren

Einführung Sequenzdiagramme

Um die erarbeiteten Nutzungsanforderungen in praktische Programmierung umzusetzen, wurden zunächst Sequenzdiagramme erstellt. Diese Diagramme werden jeweils für einen zusammenhängenden Workflow gezeichnet und skizzieren alle kleineren Teilschritte, die in diesem Workflow nacheinander abgearbeitet werden sollen. Ziel eines Sequenzdiagramms ist es, die einzelnen Aktivitäten darzustellen, die ein Nutzender beim Verwenden der Software durchläuft[23]. An solchen Diagrammen können auch die Zusammenhänge dieser Aktivitäten untereinander abgelesen werden. Karen McGraw verweist in *User-centered Requirements* darauf, dass diese Art von Diagrammen sehr hilfreich sein kann, um primäre Workflows und zusammenhängende Prozesse zu identifizieren[5]. Ian Alexander führt in *Scenarios, Stories, Use Cases* ganz ähnliche Diagramme ein, die er *Scenario Process Models* nennt. Auch

hier geht es darum, ein Abbild der Teilprozesse zu schaffen, die ein Nutzender beim Verwenden einer Software durchläuft. I. Alexander nennt als weiteren Vorteil dieser Diagramme, das noch unklare oder fehlende Daten innerhalb eines Workflows schnell zu erkennen sind. Über eingehende und ausgehende Pfeile kann an den Sequenzdiagrammen abgelesen werden, durch welche Aktionen der jeweilige Workflow angestoßen wird, beziehungsweise welche anderen Workflows durch Nutzerinteraktionen gestartet werden können. Teile einzelner Elemente der grafischen Oberfläche werden skizzenhaft dargestellt, um einen intuitiven Eindruck festzuhalten, wie die Nutzenden durch diesen Workflow navigieren und welche Steuerelemente sie auf dem Bildschirm verwenden können. Durch farbige Anmerkungen an einzelnen Elementen oder Arbeitsschritten wird auf bestimmte Details hingewiesen, die bei der Implementierung zu beachten sind. Die Sequenzdiagramme sind bewusst formlos gehalten und nur grob skizziert. Dies spiegelt die schnelle und intuitive Erstellung dieser Diagramme wieder. Es geht noch nicht darum, den Prozess in einer sehr strukturierten Darstellungsform aufzuzeichnen, die man direkt in Code übersetzen könnte. Vielmehr soll intuitiv und spontan festgehalten werden, welche Schritte ein Nutzender durchlaufen könnte, wenn er jenen Workflow anstößt.

Sequenzdiagramm *Vergebenen Termin aufrufen*

Beispielhaft für den Entstehungsprozess dieser Sequenzdiagramme wird im Folgenden ein Diagramm gezeigt und näher erläutert. Dieses Diagramm beschreibt den Workflow, wenn ein Nutzender die Detailansicht eines bereits vergebenen Termins aufruft.

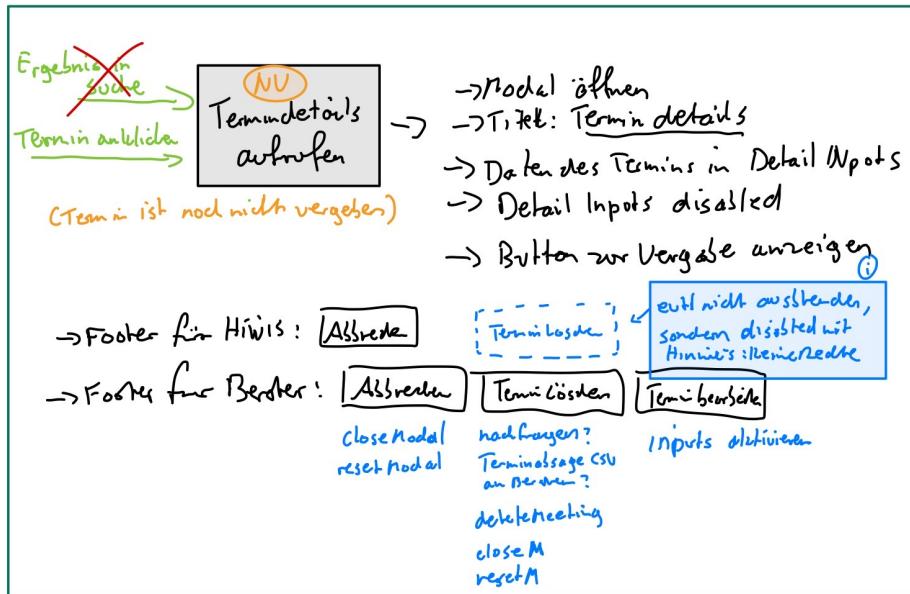


Abbildung 5.1: Sequenzdiagramm: Laden der Detailansicht für einen vergebenen Termin.

Die grünen Pfeile links oben beschreiben, auf welchem Weg die Nutzenden diesen Workflow anstoßen können. Die Detailansicht eines vergebenen Termins kann in diesem Fall auf zwei verschiedenen Wegen geöffnet werden. Entweder klickt ein Nutzender auf einen Termin in der Monatsübersicht oder er klickt auf ein Suchergebnis in der Ergebnisliste der Suche nach Teilnehmenden. Im rechten oberen Teil des Diagramms wird stichpunktartig festgehalten, welche Teilschritte notwendig sind, um die Termindetails sinnvoll darstellen zu können: Zunächst muss das **Modal** über dem bestehenden Bildschirminhalt eingeblendet werden. Im nächsten Schritt muss der Titel des Modals angepasst werden. Der Standardtitel *Termin erstellen* macht in diesem Kontext keinen Sinn, da der Termin bereits existiert. Daher wird der Titel des Modals in diesem Fall auf *Termindetails* geändert. Im weiteren Verlauf müssen die Daten des Termsins (Datum, Uhrzeit, Titel) in die entsprechenden Input Felder geladen werden. Diese Input Felder sollen als *disabled* dargestellt werden. Das bedeutet, dass man den eingetragenen Inhalt lesen, ihn jedoch nicht bearbeiten kann. Ein bereits angelegter Kundenvertrag kann nicht bearbeitet werden, solange der Datensatz des Kunden nicht entfernt wurde. So soll vermieden werden, dass beispielsweise das Da-

tum des Beratungstermins nachträglich geändert wird, ohne dass der Kunde darüber informiert wird. Hier wird den Nutzenden also bewusst die Funktionalität des nachträglichen Änderns verboten, um keine Missverständnisse in der Kundenkommunikation aufkommen zu lassen. Sollte tatsächlich einmal das Datum eines Beratungstermins verändert werden, muss der Kundendatensatz zunächst entfernt und dann, nach dem Anpassen des Datums neu hinzugefügt werden. Dies hat den Vorteil, dass die entsprechenden Informationsmails (Terminabsage, Neue Terminbestätigung) korrekt an den Kunden und den Beratenden gesendet werden und somit für beide Parteien nachvollziehbar ist, an welchem Datum der Termin nun tatsächlich stattfinden soll.



Abbildung 5.2: Verschiedene Bereiche des Modals mit Kontrollelementen.

Im unteren Teil des Diagramms wird durch kleine Skizzen festgehalten, welche Kontrollelemente den Nutzenden im Workflow zur Verfügung stehen. Die *Vergabeleiste* bezeichnet den Bereich, in dem Buttons zum Interagieren mit den Kundendaten dargestellt werden. In diesem Fall ist lediglich der Button *Kundendaten löschen* sichtbar. Ein nachträgliches Bearbeiten der Kunden-

daten soll nicht möglich sein, da auch hier inkonsistente Status beim Versand der Mails an den Kunden entstehen könnten. Wird beispielsweise die Mailadresse des Kunden geändert, nachdem er für den Versand eines Feedback-Fragebogens eingetragen wurde, kann der Link für den Fragebogen nicht mehr an die korrekte Adresse verschickt werden. Diese Problematik wird im Diagramm grafisch durch die orangene Infobox am rechten Rand verdeutlicht. Schlussendlich wird im Diagramm dargestellt, welche Buttons in der Fußleiste des Modals angezeigt werden sollen. Hier soll lediglich der Button *Abbrechen* aktiv sein. Über diesen Button wird das Modal ohne Übernahme von Änderungen geschlossen und ein anderer Termin kann aufgerufen werden. Die Buttons zum Speichern und Löschen des Termins sollten nicht angeklickt werden können. Wie bereits erwähnt, soll ein Ändern oder Löschen der Termindaten nur möglich sein, wenn die verknüpften Kundendaten entfernt wurden. Die Buttons sollen jedoch nicht vollkommen unsichtbar sein, sondern lediglich ausgegraut dargestellt werden. Die blaue Box im Diagramm ergänzt, dass in diesem Fall ein Hinweis angezeigt werden soll, in dem den Nutzenden erklärt wird, dass zunächst die Kundendaten entfernt werden müssen, bevor diese Buttons wieder genutzt werden können. Dass die Buttons ausgegraut und mit einem Hinweis dargestellt werden, soll dazu führen, dass Nutzende eine einheitliche Benutzeroberfläche vorfinden. Bei dem Versuch, mit den deaktivierten Buttons zu interagieren, sollen sie automatisch darauf hingewiesen werden, welche Schritte notwendig sind, um die gewünschte Funktion nutzen zu können. Dies soll dazu beitragen, dass auch unerfahrene Nutzende schnell und intuitiv verstehen, wie eine Interaktion mit dem System gedacht ist. Im Anhang (Kapitel 11.2) finden sich alle angefertigten Sequenzdiagramme, die auch weitere Prozesse beschreiben: *Vergebenen Termin aufrufen*, *Neuen Termin erstellen* und *Termin an Kunden vergeben*.

5.4 Details der Implementierung

Im folgenden Abschnitt soll der Fokus nun auf die technischen Details der Implementierung gelegt werden. Wesentliche Fragestellungen sind hier: Wie werden die Nutzungsanforderung in der Praxis umgesetzt? Welche Designpatterns und Softwarekonzepte werden verwendet? Wie werden Schnittstellen zwischen Frontend und Backend gestaltet? Und über welche Protokolle werden Daten ausgetauscht? Wichtig ist es, an dieser Stelle wieder im Blick zu behalten, dass es sich um ein Modul in einer bereits bestehenden Software

handelt. Somit sollten bereits existierende Konzepte und Schnittstellen nach Möglichkeit wieder verwendet werden. Durch das Aufgreifen bestehender Lösungsstrategien und Entwurfsmuster wird das gesamte Softwarepaket einheitlich strukturiert und ist somit leichter zu warten. Gleichzeitig müssen einige Funktionen nicht von Grund auf neu implementiert werden, somit kann Zeit und Arbeit in diesem Prozess erspart werden[14]. Im Folgenden werden einige Designentscheidungen, Grafiken und Codeschnipsel präsentiert, um einen groben Einblick in die technische Umsetzung zu erlauben. Der vollständige Sourcecode, sowie alle Commits zur Programmierung dieses Moduls sind im entsprechenden GitHub Repository einzusehen. Weitere Details dazu im Anhang unter Abschnitt 11.1.

Die Klasse Meeting

Die eigentlichen Datenmodelle, mit denen in dem Modul zur Terminvereinbarung gearbeitet wird, sind relativ simpel gehalten. Es gibt drei relevante Typen von Datensätzen: Termine, Mailtemplates und Beratungsräume. Die Datensätze, die einen Termin beschreiben, enthalten außerdem auch Informationen über den Ratsuchenden, der diesen Termin gebucht hat. Diese Datensätze werden im Folgenden als **Meeting** bezeichnet und sollen anhand eines **UML** Diagramms detailliert veranschaulicht werden.



Abbildung 5.3: UML Diagramm der Klasse Meeting. Stellt alle Methoden und Eigenschaften eines Objektes dar, das einen Termin repräsentiert.

Die Klasse `Meeting` stellt zunächst einige statische Eigenschaften und Methoden zur Verfügung. Das Array `meetingList` enthält zur Laufzeit alle aktuell bekannten Termine in einer unsortierten Liste. Durch die statische Funktion `fetchMeetings()` werden alle Datensätze aus der Datenbank geladen und in der `meetingList` gespeichert. Über die Funktion `getById()` kann ein bestimmtes Meeting aus der `meetingList` angesprochen werden. Die weiteren Funktionen dienen dazu, die Eigenschaften eines Meetings an das Backend zu senden, um die neuen Werte in der Datenbank zu speichern oder zu löschen. Die statische Methode `createOnServer(properties)` sendet die Daten eines neuen Meetings an den Server. Diese Methode erstellt bewusst keine neue Instanz vom Typ `Meeting`, da es keinen Sinn macht, ein Meeting Objekt zu verwenden, solange der Server dieses Meeting noch gar nicht kennt. Im Anschluss an die `createOnServer(properties)` Methode sollte stets die Methode `fetchMeetings()` aufgerufen werden, um auf das neu erstellte Meeting zugreifen zu können. Die Funktion `updateServer()` wird auf einem bestehenden Objekt vom Typ `Meeting` aufgerufen und sendet die aktuellen lokalen Eigenschaften an den Server. Hierbei wird die Funktion `toFormData()` verwendet, um die Eigenschaften des Objektes in das passende Format zu bringen. Die so aufbereiteten Daten werden über den asynchronen Aufruf der **fetch API** mit einer Http Request an das entsprechende Php Skript auf dem Server gesendet.

Listing 5.1: Senden von lokalen Änderungen eines Meetings an den Server

```
1 /**
2  * Updates an existing meeting on the server for storage in
3  * database
4 */
5 async function updateOnServer() {
6   let formData = this.toFormData();
7
8   const url = `${stubegru.constants.BASE_URL}/modules/
9   calendar/dates/update_calendar_date.php`;
10  let meetingResp = await fetch(url, {
11    method: 'POST',
12    body: formData
13  });
14  meetingResp = await meetingResp.json();
15  return meetingResp;
```

14}

Über die Methode `deleteOnServer()` kann ein bestehendes Meeting auf dem Server gelöscht werden. Durch einen anschließenden Aufruf von `fetchMeetings()` wird das Meeting dann auch in der lokalen `meetingList` gelöscht. Außerdem stellt ein Objekt vom Typ `Meeting` noch zwei Methoden zur Verfügung, um Kundendaten zu verknüpfen, beziehungsweise zu löschen. Diese Funktionen stoßen in den jeweiligen Php Skripten auf dem Server weitere Workflows an, wie beispielsweise das Versenden einer Bestätigungsmaile an die ratsuchende Person.

Listing 5.2: Php Skript zum Erstellen eines neuen Meetings in der Datenbank

```

1 <?php
2
3 // Dieses Script speichert einen neuen Kalendereintrag
4 $BASE_PATH = getenv("BASE_PATH");
5 require_once "$BASE_PATH/utils/auth_and_database.php";
6 require_once "$BASE_PATH/modules/user_utils/user_utils.
    php";
7 permissionRequest("MEETINGS_WRITE");
8 $creatorId = $_SESSION["id"];
9
10 $date = $_POST["date"];
11 $ownerId = $_POST["ownerId"];
12 $room = $_POST["roomId"];
13 $start = $_POST["start"];
14 $end = $_POST["end"];
15 $title = $_POST["title"];
16 $template = $_POST["templateId"];
17
18 //Channel attribute will only be set by calendar2
    frontend
19 $channel = isset($_POST["channel"]) ? $_POST["channel"] :
    "unknown";
20
21 $ownerName = getUserName($ownerId);
22
23 try {
    $insertStatement = $dbPdo->prepare("INSERT INTO `Termine` ('date','owner','ownerId','room','start
        
```

```

        ', 'end', 'title', 'template', 'channel') VALUES (:date, :ownerName, :ownerId, :room, :start, :end, :title, :template, :channel);";
25 $insertStatement ->bindValue(':date', $date);
26 $insertStatement ->bindValue(':ownerName', $ownerName);
27 ;
28 $insertStatement ->bindValue(':ownerId', $ownerId);
29 $insertStatement ->bindValue(':room', $room);
30 $insertStatement ->bindValue(':start', $start);
31 $insertStatement ->bindValue(':end', $end);
32 $insertStatement ->bindValue(':title', $title);
33 $insertStatement ->bindValue(':template', $template);
34 $insertStatement ->bindValue(':channel', $channel);
35 $insertStatement ->execute();
36 //Id des neu hinzugefügten Termins abrufen
37 $dateId = $dbPdo->lastInsertId();
38 } catch (Exception $e) {
39     echo json_encode(array("status" => "error", "message" => $e->getMessage()));
40     exit;
41 }
42 echo json_encode(array("status" => "success", "message" => "Der Termin wurde erstellt", "dateId" => $dateId));

```

Vergebenen Beratungstermin aufrufen

In Kapitel 5.3 wurden exemplarisch alle Teilschritte erklärt, die nötig sind, um einen bereits vergebenen Beratungstermin aufzurufen und in der Detailansicht darzustellen. Für jeden dieser einzelnen Schritte wurde eine entsprechende Funktion in der Klasse `CalendarModal` implementiert. Die Funktion `setModalVisible(isVisible)` kann beispielsweise aufgerufen werden, um das Modal ein- oder auszublenden. Die Klasse `CalendarController` verwendet nun all diese Funktionen und bildet so den kompletten beschriebenen Workflow Schritt für Schritt ab. Das folgende Listing beschreibt den Ablauf aller einzelnen Schritte durch den sequenziellen Aufruf der einzelnen Funktionen:

Listing 5.3: Öffnen der Detailansicht eines vergebenen Beratungstermins

```
1 static function openAssignedMeeting(meetingId) {
```

```

2   let C = CalendarController;
3   let m = C.modal;
4
5   m.setModalVisible(true);
6   m.setModalTitle("Termindetails (Termin vergeben)");
7   m.resetAllForms();
8
9   const meeting = Meeting.getById(meetingId);
10  m.setMeetingDetailData(meeting);
11  m.enableDetailMeetingForm(false);
12
13  m.showAssignButtons(false, false, true, false);
14
15  m.setClientVisible(true);
16  m.enableClientForm(false);
17  m.setClientData(meeting.teilnehmer);
18
19  m.enableFooterButtons(false, false, false, true);
20
21  m.setInfoAlert('<i class="fas fa-info-circle"></i> Dieser
Termin ist bereits an einen Kunden vergeben. Bearbeiten des
Termins ist nur möglich, nachdem die Kundendaten
gelöscht wurden.');
22}

```

Monatsansicht mit fullcalendar darstellen

Als abschließendes Beispiel der konkreten Implementierung wird die Klasse `CalendarView` vorgestellt. Diese ist für das Darstellen der Terminübersicht zuständig. An dieser Stelle wird die Javascript Bibliothek `fullcalendar` verwendet, um den Aufbau der Monatsansicht deutlich zu vereinfachen. Die Klasse `CalendarView` referenziert auf eine Instanz vom Typ `FullCalendar` und kann darüber die wichtigsten Funktionen zum Löschen und Hinzufügen weiterer Termine aufrufen. Im folgenden Listing wird die Funktion `addMeetings(meetingList)` gezeigt. Diese Funktion wird vom `CalendarController` aufgerufen und bekommt als Parameter eine Liste von `Meetings` übergeben. Aus den übergebenen Objekten vom Typ `Meeting` werden zunächst Objekte generiert, die alle

Eigenschaften erhalten, welche die fullcalendar Bibliothek benötigt, um die Termine in der Übersicht korrekt anzeigen zu können. Die so erstellten Datensätze werden anschließend in *freie Termine* und *vergebene Termine* sortiert. Über die Funktion `fullCalendar.addEventSource(eventSource)` werden die Listen mit den freien und den vergebenen Terminen hinzugefügt. Über den weiteren Parameter `color` wird in diesem Fall die farbliche Darstellung der Termine (grün = frei, rot = vergeben) in der Monatsübersicht gesteuert.

Listing 5.4: Hinzufügen von Terminen in die Monatsübersicht von fullcalendar

```

1 function addMeetings(meetingList) {
2     //Generate events for fullcalendar
3     let ownUserId = stubegru.currentUser.id;
4     let ownEvents = { free: [], assigned: [] };
5     let othersEvents = { free: [], assigned: [] };
6
7     for (let inMeeting of meetingList) {
8         let outMeeting = {
9             title: inMeeting.owner,
10            start: `${inMeeting.date}T${inMeeting.start}`,
11            end: `${inMeeting.date}T${inMeeting.end}`,
12            extendedProps: inMeeting
13        };
14
15        //Sort meetings by free/assigned and own/others
16        if (inMeeting.ownerId == ownUserId) {
17            (inMeeting.teilnehmer && inMeeting.teilnehmer != "") ?
18                ownEvents.assigned.push(outMeeting) :
19                ownEvents.free.push(outMeeting);
20        } else {
21            (inMeeting.teilnehmer && inMeeting.teilnehmer != "") ?
22                othersEvents.assigned.push(outMeeting) :
23                othersEvents.free.push(outMeeting);
24        }
25    }
26
27 //Generate and add Eventsource

```

```
28     this.fullCalendar.addEventSource({
29         id: "stubegru-own-free-events",
30         events: ownEvents.free,
31         color: "#5cb85c",
32         classNames: ["pointer"]
33     });
34     this.fullCalendar.addEventSource({
35         id: "stubegru-own-assigned-events",
36         events: ownEvents.assigned,
37         color: "#d9534f",
38         classNames: ["pointer"]
39     });
40     this.fullCalendar.addEventSource({
41         id: "stubegru-others-free-events",
42         events: othersEvents.free,
43         color: "#5cb85c",
44         classNames: ["pointer"]
45     });
46     this.fullCalendar.addEventSource({
47         id: "stubegru-others-assigned-events",
48         events: othersEvents.assigned,
49         color: "#d9534f",
50         classNames: ["pointer"]
51     });
52 }
```

Kapitel 6

Gestaltungslösungen evaluieren

Im vorherigen Kapitel wurden einzelne Aspekte der konkreten Implementierung der Gestaltungslösungen vorgestellt. Im nächsten Schritt sollen die so entwickelten Softwareteile mit einem Nutzer getestet werden. Dies entspricht dem vierten Schritt *Gestaltungslösungen aus der Benutzerperspektive evaluieren* im menschzentrierten Gestaltungsprozess der ISO Norm[8]. Christian Moser erklärt in *User Experience Design* den möglichen Ablauf eines solchen Usertests wie folgt: „Bei Tests mit Benutzern wird den Testteilnehmern der Kontext anhand des Szenarios erklärt und dann werden die Aufgaben gestellt. Dabei wird beobachtet, wie gut sie diese mit Hilfe des Prototyps lösen können.“[1]. In diesem Fall wird der Test gemeinsam mit Herr Maier von der allgemeinen Studienberatung der Universität Kassel durchgeführt. Gemeinsam haben wir uns eine Stunde Zeit genommen, um das neu entwickelte Modul zur Terminvereinbarung zu testen. Wir sitzen gemeinsam im Büro von Herr Maier und können die Software so direkt an seinem Dienstreichner testen, um den Nutzungskontext möglichst authentisch zu gestalten.

6.1 Durchführung des Usertests

Der Usertest beginnt damit, dass Herr Maier die Software Stubegru im Browser aufruft und den View mit dem neuen Kalendermodul zur Terminvergabe öffnet. Sein erster Blick fällt auf die Monatsübersicht der Termine im November.

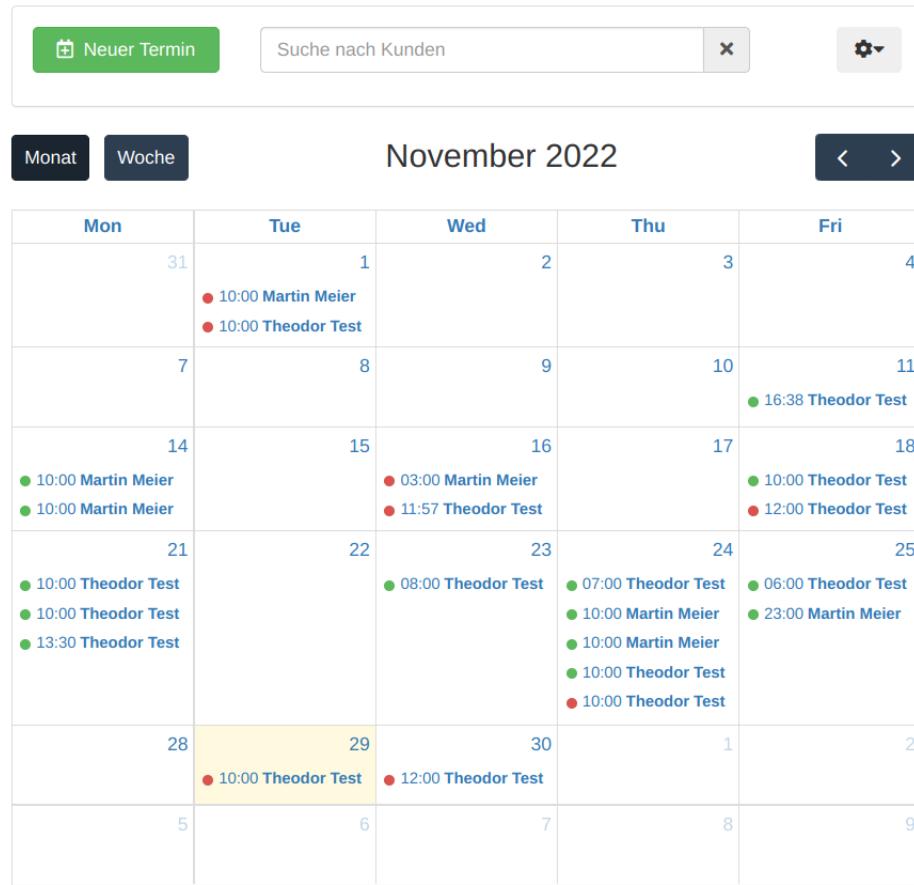


Abbildung 6.1: Monatsübersicht der Termine im November.

In dieser Übersicht kann er einsehen, welche Termine für den aktuellen Monat eingetragen sind. Anhand der farbigen Punkte kann Herr Maier auf einen Blick erkennen, welche Termine noch frei sind und welche bereits an rätsuchende Kunden vergeben wurden. Neben dem farbigen Punkt kann für jeden Termin eingesehen werden, bei welchem Studienberatenden dieser Termin stattfindet. Herr Maier merkt an, dass es im Vergleich zur alten Version möglich ist, direkt in der Übersicht zu sehen, zu welchem Zeitpunkt die Termine stattfinden und wie viele Termine an einem Tag angeboten werden. Er testet, was passiert, wenn viele Termine am gleichen Tag angelegt werden. Zufrieden stellt er fest, dass die Zelle für den entsprechenden Tag automatisch größer wird, wenn viele Termine eingetragen werden. Herr Maier überlegt weiterhin, dass die Monatsübersicht dadurch eventuell so lang werden könnte, dass

nicht mehr alle Zeilen gleichzeitig auf den Bildschirm passen könnten. Wir diskutieren, wie viele Termine üblicherweise pro Tag angeboten werden und einigen uns darauf, die aktuelle Ansicht zunächst beizubehalten und in der Praxis zu testen.

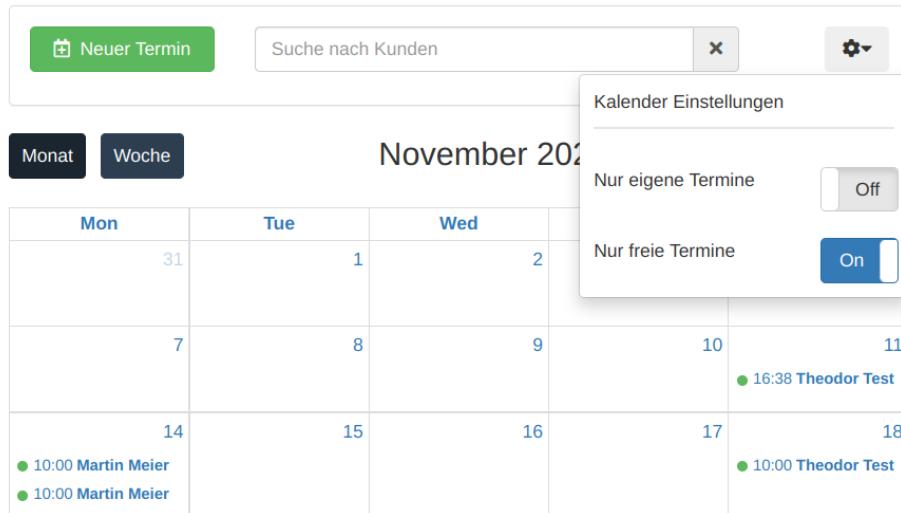


Abbildung 6.2: Toggles im Dropdown Menü um die angezeigten Termine zu filtern. Hier werden nur freie Termine dargestellt.

Ich weise Herr Maier darauf in, dass er über den Button mit dem Zahnrad weitere Filtermöglichkeiten hat, um eine überfüllte Monatsübersicht auf die wesentlichen Termine zu reduzieren. Den Button hat er zunächst nicht bemerkt. Nachdem er ihn einmal gefunden hat, versteht er die Bedienung über die beiden Toggles aber schnell. Er merkt positiv an, dass Änderungen an den Filtereinstellungen direkt übernommen werden und die Auswirkung der Filter somit leicht zu begreifen ist. Ihm fällt auf, dass der Filter *nur eigene Termine anzeigen* für die Hilfskräfte wenig Sinn ergibt, da diese keine Berechtigung haben, um selber Termine zu erstellen. Sie können lediglich eingestellte Zeitslots an Ratsuchende vergeben. Wir diskutieren über die Möglichkeit, diesen Filter für Accounts von Hilfskräften auszublenden, entscheiden uns aber dafür, dass dies nicht notwendig ist. Der Filter ist für diese Nutzergruppe zwar überflüssig, aber auch nicht störend.

Als nächstes gebe ich Herr Maier die Aufgabe, einen neuen Zeitslot zu erstellen. Ich verrate nicht weiter, wo er diese Funktion finden kann und lasse ihn ausprobieren. Nach kurzer Zeit entdeckt Herr Maier den Button *Neuer Ter-*

min und es öffnet sich das Modal zum Erstellen eines freien Zeitslots. Herr Maier erklärt seine Gedanken: „Ich kenne es von anderer Software so, dass man auf den entsprechenden Tag in der Monatsübersicht klickt, um einen neuen Termin anzulegen. Aber aus der alten Version von Stubegru erinnere ich mich noch an den grünen Knopf zum Erstellen eines Termins. Früher war der Knopf *unter* der Monatsübersicht, aber so fügt er sich elegant in die Leiste mit der Suche und den neuen Filtern ein.“[24]



Abbildung 6.3: Navigationsleiste über der Monatsansicht. Über den grünen Button kann eine neuer Zeitslot erstellt werden.

In dem leeren Formular füllt Herr Maier Schritt für Schritt die Input Felder aus. Nach einem Klick auf das Feld für die Startzeit ist er enttäuscht. In der alten Version öffnete sich hier ein **Timepicker**, mit dessen Hilfe man die Uhrzeit ohne Tastatur nur mit der Maus eingeben konnte. Auch ich bin überrascht, denn in meinen Tests hatte sich hier auch ein Timepicker geöffnet. Nach kurzer Recherche stellt sich heraus, dass das verwendete Html5 Input Feld für Uhrzeiten in jedem Browser unterschiedlich dargestellt wird[25]. Während der Implementierung habe ich die Ergebnisse immer in *Google Chrome* getestet, dort erscheint nach dem Klick in ein Uhrzeit-Input ein Timepicker. Die Abteilung der Studienberatung verwendet in der Regel Firefox als Browser, da dieser auf den Dienstrechnern standardmäßig vorinstalliert ist. Ich verspreche, mich dem Problem anzunehmen und einen zusätzlichen Timepicker einzubinden, der unabhängig vom verwendeten Browser funktioniert.

The screenshot shows a web-based appointment creation form. At the top right, there is a red circular icon with a white exclamation mark and the text "Ungespeicherte Änderungen" (Unsaved changes). The form fields include:

- Datum:** 08/12/2022
- Titel:** Studienberatung
- Raum:** Bitte wählen...
- Kontaktkanal:** Bitte wählen...

A prominent feature is a time picker modal window titled "Beginn" (Start) and "Ende" (End). It displays a grid of hours from 17:00 to 23:00. The hour 17:00 is highlighted in blue, indicating it is selected. Below the grid are buttons for "+ New" and "Edit". To the right of the grid, there is a "Mailvorlage" (Email template) section with a dropdown menu and a "Bitte wählen..." button.

At the bottom of the form, there are two buttons: "Termin löschen" (Delete appointment) in red, and "Schließen" (Close) and "Speichern" (Save) in green.

Abbildung 6.4: Erstellen eines neuen Termins und Ausfüllen der Eingabefelder. In diesem Fall wird in Google Chrome ein Timepicker angezeigt.

Alle anderen Formularfelder füllt Herr Maier zügig aus. Da die abgefragten Daten und die Anordnung der Eingabefelder genauso wie in der alten Version gestaltet sind, merkt man hier, dass Herr Maier bereits viel Erfahrung mit der alten Softwareversion von Stubegru gesammelt hat. Den unten rechts platzierten grünen Button zum Speichern der Daten findet er sofort. Der Datensatz wird in der Datenbank gespeichert und eine kleine grüne Meldung am linken unteren Rand verkündet, dass der Termin erfolgreich gespeichert wurde. Herr Maier ist trotzdem verunsichert, ob der Vorgang erfolgreich war. In der alten Version, hat sich das Modal nach dem Speichern automatisch geschlossen. Herr Maier merkt an, dass in der neuen Version nun jedes Mal ein zusätzlicher Klick auf den Button *Schließen* notwendig ist. Wenn das Modal sich nach dem Speichern direkt schließen würde, wäre es für ihn einfacher. Ich erkläre, dass das Modal geöffnet bleibt, damit ein neu erstellter Termin direkt an einen Kunden vergeben werden kann. Nach dem Speichern erscheint hierfür der große blaue Button mit der Aufschrift *Termin vergeben*. Herr Maier erklärt, dass die Termine eigentlich immer von den Hilfskräften der Erstinformation vergeben werden und es für die Studienberatenden nicht nötig sei, den erstellten Termin direkt vergeben zu können. Wir einigen uns darauf, dass ein Klick auf den *Speichern* Button das Modal, nach erfolgreichem Sichern der Daten, schließt.

Und noch eine weitere Sache fällt Herr Maier auf: Wenn er bereits einen Termin eingetragen hat und dann einen zweiten Termin erstellen möchte, werden alle Eingabefelder zurückgesetzt und sind wieder leer. „Das war in der alten Version praktischer“, sagt Herr Maier „Oftmals trage ich viele Termine für den gleichen Tag, mit den gleichen Attributen ein. Das einzige, was ich bisher anpassen musste, war dann die Uhrzeit. Nun brauche ich viel

mehr Klicks, weil ich jedes Mal wieder den Raum und das Mailtemplate auswählen muss.“[24] Es stellt sich heraus, dass es in der alten Softwareversion einen Bug gab, wodurch in einigen Fällen das Zurücksetzen des Formulars nicht aufgerufen wurde. Dieser Bug hat sich in der Praxis dann allerdings als sehr praktisch etabliert. Die Formularfelder nie zurückzusetzen kann allerdings inkonsistente Status erzeugen, wenn zuvor beispielsweise ein bereits vergebener Termin im Modal angezeigt wurde. Wir einigen uns auf eine gemeinsame Idee: Wenn das Modal zum Erstellen eines Termins angezeigt wird, erscheint in der Fußleiste ein weiterer Button mit der Aufschrift: *Speichern und nächster*. Dieser Button speichert den aktuellen Termin und lässt das Modal ohne Zurücksetzen der Formularfelder geöffnet. Somit kann direkt ein neuer Termin mit ähnlichen Attributen eingetragen werden.

Abbildung 6.5: Ein freier Termin. Studienberatende können Änderungen am Termin vornehmen oder einen Kunden einbuchen.

Als nächstes soll Herr Maier einen erstellten Zeitslot an einen Kunden vergeben. Intuitiv klickt er hierfür auf den entsprechenden Termin in der Monatsübersicht. Es öffnet sich das Modal mit der Detailansicht der Termindaten. Sehr schnell findet Herr Maier den großen blauen Knopf zur Terminvergabe. Wie er es bereits aus der alten Softwareversion kennt, öffnet sich das Formular, um die weiteren Kundendaten einzutragen. Herr Maier prüft, was passiert, wenn er hier relevante Felder einfach leer lässt. Beruhigt stellt er fest, dass beim Versuch die Kundendaten zu speichern ein Hinweis am entsprechenden leeren Eingabefeld auftaucht und das Speichern unvollständiger Daten verhindert.

Nachdem alle Felder vollständig ausgefüllt sind, speichert Herr Maier die Kundendaten und sieht eine leicht veränderte Detailansicht des Termsins:

The screenshot shows a software interface for managing customer data and appointments. At the top, there are filters for 'Titel' (Beratungstermin), 'Berater*in' (Martin Meier), 'Kontaktkanal' (Alle), 'Raum' (Das absolut geilste Büro), and 'Mailvorlage' (Testvorlage für Mails). Below these are sections for 'Kundendaten' (Customer Data) and 'Anliegen' (Concerns). The 'Kundendaten' section contains fields for 'Vor- und Nachname' (Johannes), 'Fragebogen' (Nein), 'Kanal' (Webmeeting), 'Telefon' (0561 2345 6789), and 'E-Mail' (johannes@johannes.de). A note below the telephone number states: "*** Wird nur dem Beratenden angezeigt ***". A red button labeled 'Kundendaten löschen' (Delete customer data) is present. A blue info box at the bottom left states: 'Dieser Termin ist bereits an einen Kunden vergeben. Bearbeiten des Termins ist nur möglich, nachdem die Kundendaten gelöscht wurden.' (This appointment is already assigned to a customer. Editing the appointment is only possible after deleting the customer data). At the bottom right are buttons for 'Termin löschen' (Delete appointment), 'Schließen' (Close), and 'Speichern' (Save).

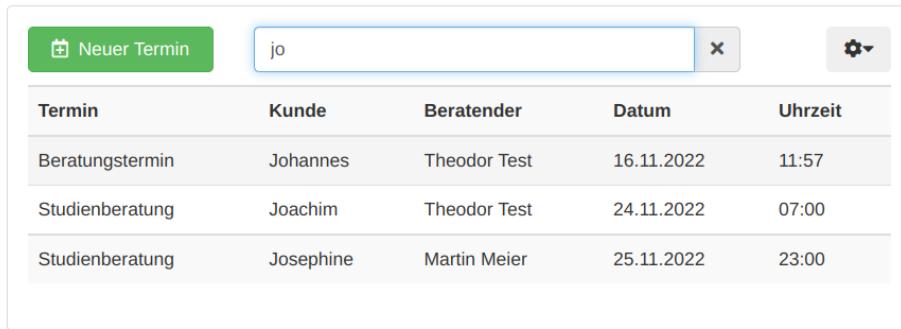
Abbildung 6.6: Ein vergebener Termin. Terminattribute und neue Kundendaten können erst nach Löschen der aktuellen Kundendaten hinterlegt werden.

Die Terminattribute und Kundendaten können nicht weiter bearbeitet werden. Dies erkennt Herr Maier schnell daran, dass die entsprechenden Eingabefelder ausgegraut angezeigt werden. Zusätzlich verrät der Hinweis in der blauen Infobox, dass Änderungen nur vorgenommen werden können, wenn zuvor die Kundendaten gelöscht werden. Herr Maier realisiert, dass diese Einschränkung Sinn macht, da somit keine Änderungen am Termin vorgenommen werden können, ohne dass der Kunde durch ein erneutes Einbuchen darüber informiert wird.

Herr Maier freut sich, dass die Telefonnummer nun in übersichtlichen Blöcken von jeweils vier Ziffern angezeigt wird. Das Eintippen ins Telefon sei somit deutlich einfacher. Allerdings merkt er an, dass die Telefonnummer nur in dem Account des zugeordneten Beratenden angezeigt werden soll. So kann ein höherer Datenschutz gewährleistet werden und Hilfskräfte haben keinen Zugriff auf die Kontaktdaten aller ratsuchenden Kunden. Ich erkläre, dass ich diese Funktion bereits für das Beratungsanliegen umgesetzt habe. Wir loggen uns mit einem anderen Account ein und sehen, dass das Beratungsanliegen zensiert dargestellt wird. Diese Funktionalität soll in Zukunft auch für die Telefonnummer und Mailadresse der Kunden eingesetzt werden.

Abschließend betrachtet Herr Maier nochmals die Monatsübersicht. „Wäre es technisch aufwendig, bei den vergebenen, roten Terminen statt dem Namen

des Beratenden den Namen des Kunden anzuzeigen? So könnten die Hilfskräfte den entsprechenden Termin eines Kunden schneller finden.“, fragt er mich[24]. Ich antworte, dass dies technisch kein großer Aufwand wäre. Allerdings könnte es Verwirrung stiften, wenn bei manchen Terminen der Name des Beratenden und bei anderen der Name des Kunden angezeigt wird. Für den Fall, dass der Termin eines bestimmten Kunden schnell gefunden werden muss, wurde außerdem die Suchfunktion entwickelt. Herr Maier erinnert sich an die Suchfunktion und will diese direkt ausprobieren.



The screenshot shows a user interface for searching appointments. At the top, there is a green button labeled "Neuer Termin" (New Appointment) and a search input field containing the text "jo". To the right of the input field is a clear button (X) and a settings icon. Below the search bar is a table with five columns: "Termin" (Appointment), "Kunde" (Customer), "Beratender" (Consultant), "Datum" (Date), and "Uhrzeit" (Time). The table contains three rows of data:

Termin	Kunde	Beratender	Datum	Uhrzeit
Beratungstermin	Johannes	Theodor Test	16.11.2022	11:57
Studienberatung	Joachim	Theodor Test	24.11.2022	07:00
Studienberatung	Josephine	Martin Meier	25.11.2022	23:00

Abbildung 6.7: Suchfunktion, um bereits vergebene Termine anhand des eingebuchten Kunden zu finden. Die Suchanfrage „jo“ ergibt drei Treffer.

Das Eingabefeld für die Suchanfrage findet er schnell in der oberen Navigationsleiste. Er gibt einige Buchstaben ein und direkt werden passende Suchvorschläge angezeigt. Herr Maier ist überzeugt, dass dies eine gute Lösung ist, um direkt nach Namen von ratsuchenden Kunden zu suchen. Er betont, dass es wichtig ist, dass passenden Suchvorschläge bereits beim Tippen der ersten Buchstaben angezeigt werden und kein Bestätigen der Suche mit Enter notwendig ist.

Der Usertest ist nun beendet und Herr Maier bedankt sich für die entgegengebrachte Aufmerksamkeit. Er ist sichtlich begeistert, dass die Nutzenden des Systems in diesem Designprozess im Fokus stehen und freut sich auf die Einführung der neuen Softwareversion: „Wir müssen dann sicherlich eine Schulung für die Hilfskräfte machen, um sie mit den neuen Features vertraut zu machen. Aber eigentlich ist ja fast alles so, wie sie es schon gewohnt sind.“, beendet Herr Maier unser Gespräch[24]. Ich bedanke mich ebenfalls für die Zeit und die konstruktive Kritik, die eine Diskussion über Verbesserungspotenziale erst möglich gemacht hat.

Kapitel 7

Reflexion und Fazit

Nachdem in den vorherigen Kapiteln die Durchführung des praktischen Designprozesses ausführlich beschrieben wurde, soll dieses Kapitel die Ausarbeitung abschließen, indem es die Ergebnisse zusammenfasst und hinterfragt. Zunächst wird im Abschnitt 7.1 das Ergebnis des Gestaltungs- und Entwicklungsprozesses resümiert. Hier wird nochmals darauf eingegangen, wie der Nutzungskontext erfasst wurde, die Nutzungsanforderungen spezifiziert, die entsprechenden Gestaltungslösungen umgesetzt und evaluiert wurden. Somit wird der in den Kapiteln 3 bis 6 ausführlich dokumentierte Entwicklungsprozess nochmals anhand der vier Phasen des Human Centered Design aufgegriffen und eingeordnet[8]. Zusätzlich wird kurz präsentiert, welche Änderungen an der Software nach der Auswertung des Usertests noch vorgenommen wurden. Als nächstes werden die eingesetzten Methoden im Abschnitt 7.2 aufgegriffen und kritisch hinterfragt. Der Fokus liegt dabei auf der Methode des Interviews im Kontext und auf dem Designzyklus des Human Centred Design nach ISO9241. Hierbei soll an die Fragestellung der Einleitung angeknüpft werden und eine Verbindung zu den gewonnenen Resultaten hergestellt werden. Als Letztes wird in Abschnitt 7.3 ein Ausblick gegeben, wie die entwickelte Software nun in der Praxis eingesetzt werden kann und welche Schritte dazu noch notwendig sind. Außerdem wird auch skizziert, welche weiteren wissenschaftlichen Methoden und Prozesse man noch anwenden könnte, um die hier gewonnenen Ergebnisse zu vertiefen und weiter zu verwenden.

7.1 Zusammenfassung der Ergebnisse

Was war geplant, was wurde umgesetzt?

Ziel dieses Abschnittes soll es sein, den Bogen zu spannen zwischen den anfänglich erarbeiteten Nutzungsanforderungen und den am Ende tatsächlich umgesetzten Ergebnissen.

Wenn man sich die Auswertung des Interviews im Kontext in Abschnitt 4.1 ansieht, wird klar, dass eines der wichtigsten Bedürfnisse der Nutzenden stets das schnelle und intuitive Nutzen der Software ist. Es sollten möglichst viele Informationen auf einen Blick sichtbar sein, ohne die Nutzenden von den wesentlichen Dingen abzulenken. Wenn Workflows durch die Nutzenden abzuarbeiten sind, sollten diese möglichst schnell und mit möglichst wenigen Klicks zu erledigen sein. Ein großer Detailgrad der Auswahl- und Konfigurationsmöglichkeiten sind dagegen in diesem Kontext nicht relevant. Wo die Software eine halbwegs sinnvolle Vorauswahl treffen kann, sollte dies automatisch passieren und den Nutzenden somit unnötige Interaktion ersparen. Um hierfür konkrete Anhaltspunkte zu geben, werden im Folgenden die drei exemplarisch vorgestellten Nutzungsanforderungen aus Abschnitt 4.2 nochmals aufgegriffen.

Gefordert wurde da eine kompakte Kalenderansicht, die durch farbliche Markierungen auf den ersten Blick darstellt, an welchen Tagen noch freie Termine verfügbar sind. Diese Anforderung konnte durch die tabellarische Monatsansicht mithilfe der Javascript Bibliothek *fullcalendar* umgesetzt werden. Durch die roten und grünen Punkte neben jedem angezeigten Termin kann direkt auf den ersten Blick erkannt werden, ob es sich um einen bereits vergebenen oder freien Termin handelt. Durch die verwendete Bibliothek zum Darstellen der Monatsübersicht ist dieser tabellarische Überblick nicht mehr ganz so kompakt wie in der alten Softwareversion. Dies ist auch Herr Maier beim Usertest negativ aufgefallen. Ein positiver Ausgleich hierfür ist allerdings, das in der neuen Übersicht direkt die einzelnen Termine eines Tages dargestellt werden. In der alten Version war es nötig, zunächst den Tag anzuklicken und erst in dem danach erscheinenden Modal konnten die einzelnen Termine und die entsprechenden Uhrzeiten eigensehen werden. Der etwas ausführlichere neue View nimmt also mehr Platz auf dem Bildschirm ein und kann bei sehr vielen Terminen an einzelnen Tagen sehr lang werden, bietet dafür aber eine umfassende Übersicht über freie und vergebene Termine und stellt weiterführende Informationen mit nur einem Klick in der Detailansicht zur

Verfügung.

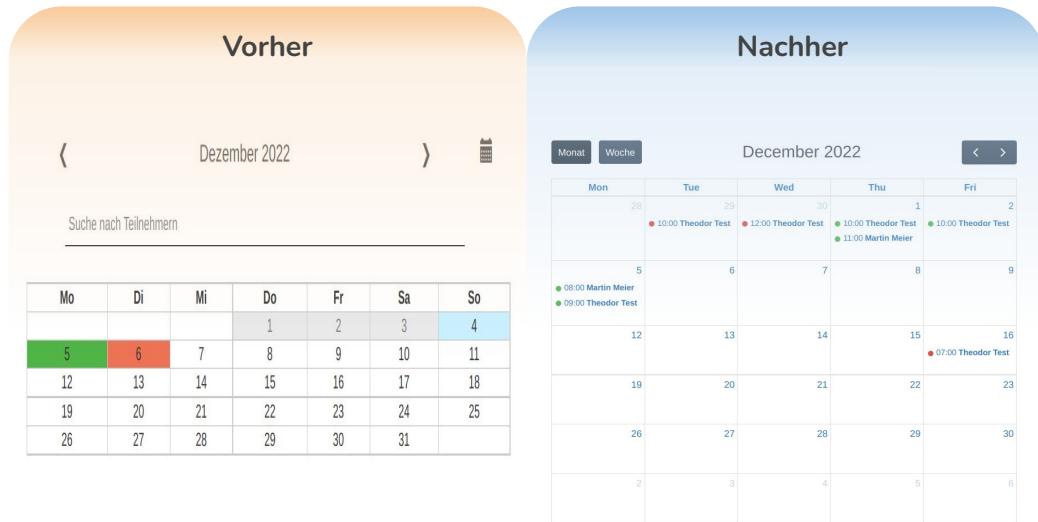


Abbildung 7.1: Vergleich der Monatsübersicht in der alten und neuen Softwareversion.

Als nächstes wurde beim Interview im Kontext deutlich, dass eine Funktion notwendig ist, um im Nachhinein den Termin eines Ratsuchenden schnell und unkompliziert finden zu können. Hierfür wurde die Suchfunktion neu implementiert. Bereits nach dem Eingeben einiger Buchstaben aus dem Namen der ratsuchenden Person schlägt diese neue Suchfunktion passende Ergebnisse vor und bietet in einer tabellarischen Übersicht direkt die wichtigsten Daten zum zugehörigen Beratungstermin. Durch einen Klick auf ein Suchergebnis kann der Termin schnell und einfach in der Detailansicht geöffnet werden. In der Planungsphase war hierfür ein extra Button mit einem Augensymbol vorgesehen. Ein einfacher Klick auf die entsprechende Zeile in der Tabelle hat sich aber als praktischer erwiesen. Im Vergleich zur alten Softwareversion sind die Suchergebnisse sehr viel ausführlicher und strukturierter dargestellt. Dies wurde auch im Usertest positiv wahrgenommen. In weiteren Tests entsteht der Eindruck, dass die alte Suchfunktion etwas schneller gearbeitet hat. Dies kann aber erst objektiv bewertet werden, wenn die neue Implementierung auch auf den gleichen Produktiv-Servern installiert und im Einsatz ist.

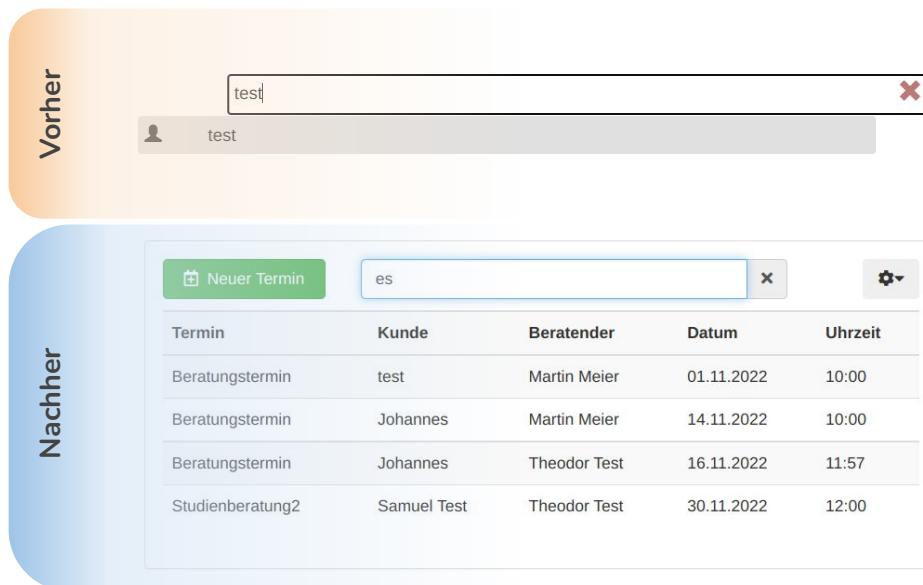


Abbildung 7.2: Vergleich der Suchfunktion in der alten und neuen Softwareversion.

Als letztes wurde im Abschnitt 4.2 das elegante Darstellen der Telefonnummern gefordert. Dieses Feature ließ sich in der Praxis leicht implementieren. In die aus der Datenbank abgerufene Telefonnummer wird an jeder vierten Stelle ein Leerzeichen eingefügt. Somit kann die Telefonnummer von den Studienberatenden bei Bedarf in leicht zu merkenden Vierer-Blöcken in das Telefon eingeben werden. Für andere Nutzeraccounts als den zuständigen Beratenden werden persönliche Details der Ratsuchenden nicht angezeigt. Diese Funktion gab es bereits in der alten Version, allerdings wird in der neuen Version statt den eigentlichen Daten ein Hinweis angezeigt. In der alten Version wurde an dieser Stelle gar nichts angezeigt, was manchmal zu Unsicherheiten geführt hat, ob hier überhaupt Daten eingetragen und korrekt gespeichert wurden. Das Feedback aus dem Usertest zeigt: Die neue Variante braucht etwas mehr Platz auf dem Bildschirm, ist dafür aber klarer zu verstehen und intuitiver zu erfassen.

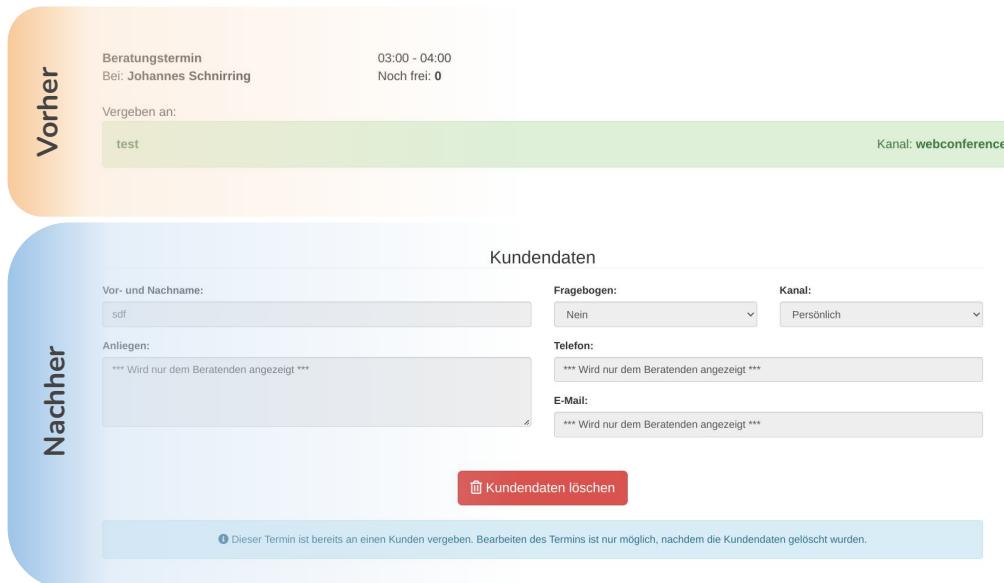


Abbildung 7.3: Vergleich der Detailansicht eines vergebenen Termins in der alten und neuen Softwareversion.

Umsetzung von Feedback des Usertests

In den Kapiteln 3 bis 6 wurde das Durchlaufen des Designzyklus anhand einer Iteration ausführlich beschrieben. Ein essentielles Grundkonzept des Human Centered Design beinhaltet allerdings auch das Durchlaufen weiterer Iterationen[2]. K. Holtzblatt betont in *Contextual Design: Evolved*, dass der Entwicklungsprozess mit einem durchgeführten Usertest noch nicht als abgeschlossen gelten sollte. Die herausgearbeiteten Kritikpunkte und Verbesserungsvorschläge, die sich aus dem Usertest ergeben haben, sollten auch in den Gestaltungsprozess der Software mit einfließen[23]. Dies ist im Schaubild nach ISO9241 durch die gestrichelten Pfeile erkennbar. Nach der Evaluierung sollte der Prozess erneut durchlaufen werden. Der Einstieg erfolgt dabei je nach Feedback der Nutzenden in Phase eins bis drei[8]. Diese weiteren Iterationen ausführlich zu beschreiben würde den Rahmen dieser Arbeit überstrapazieren. Daher werden im Folgenden lediglich die Ergebnisse eines zweiten Durchlaufes nach dem Usertest exemplarisch präsentiert:

Ein neuer Button *Speichern und Nächster* wird beim Erstellen eines neuen Termins angezeigt. Somit kann mit einem Klick der aktuelle Termin gespeichert werden und direkt ein neuer Datensatz eingetragen werden. Die Beson-

derheit ist, dass die Formularfelder in diesem Fall nicht auf Standardwerte zurückgesetzt werden. Somit können viele Termine mit ähnlichen Attributen besonders schnell hintereinander eingetragen werden.



Abbildung 7.4: Neuer Button: Mit *Speichern und Nächster* kann direkt der nächste Termin eingetragen werden.

Herr Maier hat während des Usertests angemerkt, dass nicht nur das Beratungsanliegen der ratsuchenden Personen besonderem Datenschutz unterliegt. Auch persönliche Kontaktdaten wie Mailadresse und Telefonnummer sollten nur dem zuständigen Beratenden angezeigt werden. In der überarbeiteten Version werden somit nun auch die Telefonnummer und die Mailadresse des Kunden zensiert.

Vor- und Nachname:	Fragebogen:	Kanal:
Josephine	Nein	Telefon
Anliegen:	Telefon:	E-Mail:
*** Wird nur dem Beratenden angezeigt ***	*** Wird nur dem Beratenden angezeigt ***	*** Wird nur dem Beratenden angezeigt ***

Dieser Termin ist bereits an einen Kunden vergeben. Bearbeiten des Termins ist nur möglich, nachdem die Kundendaten gelöscht wurden.

Abbildung 7.5: Details eines vergebenen Termins aus Ansicht einer Hilfskraft.

7.2 Reflexion der eingesetzten Methoden

Um dem wissenschaftlichen Anspruch dieser Arbeit Genüge zu tragen, sollen die vorgestellten Methoden und Theorien nicht nur in der Praxis erprobt, sondern auch kritisch hinterfragt und eingeordnet werden. Im folgenden Abschnitt werden die Methoden des Human Centered Design und des Interviews im Kontext nochmals aufgegriffen und deren Praxistauglichkeit reflektiert. Des Weiteren wird auch die Umsetzung und Implementierung der Gestaltungslösungen in Frage gestellt und unter Einbeziehung des Feedbacks aus dem Usertest analysiert.

Human Centered Design und Interview im Kontext

Die Theorie des Human Centered Design wurde zu Beginn dieser Arbeit motiviert mit der technischen Entwicklung von interaktiven Systemen. Software ist im Laufe der letzten Jahrzehnte immer interaktiver geworden. Der Schnittstelle zwischen Mensch und Maschine kommt dabei eine immer größer werdenden Bedeutung zu[4]. Der Ansatz des Human Centered Design versucht an dieser Stelle alle Beteiligten einzubeziehen und eine einfache, intuitive Bedienung der Systeme zu ermöglichen, die ganz bewusst aus Perspektive der Nutzenden gestaltet wurde[5]. Diese Theorie wurde in diesem Fall anhand des Designzyklus nach ISO9241 in die Praxis umgesetzt. Für die Durchführung der ersten Phase *Den Nutzungskontext verstehen und Beschreiben* wurde hier die Methode des Interviews im Kontext verwendet. Hierbei ging es darum, alle Menschen, die mit dem System zu tun haben einzubeziehen und ihre Arbeitsabläufe zu verstehen. Wie der Name der Methode bereits impliziert, liegt der Fokus hierbei auch auf dem Kontext, in dem diese Arbeitsabläufe stattfinden[10]. Das Interview im Kontext wurde deshalb mit Herr Maier in seinem Büro an seinem Dienstrechner durchgeführt, um eine möglichst realistische Alltagssituation abzubilden.

Eine zentrale Fragestellung dieser Arbeit war folgendermaßen beschrieben worden: *An welchen Stellen können die theoretischen Grundlagen des Human Centered Design den Entwicklungsprozess in der Praxis tatsächlich sinnvoll unterstützen?* Der Einsatz dieser Methoden lässt sich rückblickend als sehr ertragreich beschreiben. Durch den engen Austausch mit Herr Maier konnte der Nutzungskontext sehr gut verstanden und aufgearbeitet werden. Das Interview im Kontext hat überraschend viele neue Ideen hervorgebracht, die wohl keiner der Beteiligten im Vorhinein hätte formulieren können. Durch

das gemeinsame Benutzen der alten Softwareversion vor Ort wurde ganz klar deutlich, welche Features wirklich wichtig sind für den täglichen Einsatz, an welchen Stellen es schnell gehen muss und wo detaillierte Einstellungsmöglichkeiten unbedingt notwendig sind. Die Erkenntnisse und Beobachtungen aus dem Interview im Kontext wurden unstrukturiert schriftlich festgehalten. Beim Auswerten dieser Notizen ist aufgefallen, dass viele Details aus dem Gespräch nicht so schnell aufgeschrieben werden konnten. Die Notizen waren somit nicht ganz vollständig. Daher war es sehr wichtig, die Auswertung des Interviews im Kontext zeitlich sehr nah an der Durchführung zu planen. Somit konnten viele Details aus der Erinnerung an das Interview selbst bei der Auswertung ergänzt werden. Der Austausch mit Menschen aus der allgemeinen Studienberatung hat sich auf den Kontakt mit Herr Maier beschränkt. Dieser Austausch war sehr ertragreich, dennoch wäre es interessant gewesen, auch noch Nutzungsanforderungen anderer Nutzergruppen wie beispielsweise der Hilfskräfte der Erstinformation mit einzubeziehen. Die Erfahrungen und Ideen anderer Nutzenden hätten noch weitere hilfreiche Aspekte in den Gestaltungsprozess einbringen können. Somit hätte ein System entstehen können, das noch inklusiver und reibungsloser alle beteiligten Personen mit einbezieht. In der Praxis war es allerdings schon sehr aufwendig, Termine mit Herr Maier zu organisieren. Wären noch weitere Nutzungsgruppen mit einbezogen worden, wäre der organisatorische und zeitliche Aufwand der Software deutlich höher gewesen. Dies kann gerade in gewinnorientierten Softwarehäusern einen hemmenden Faktor für den Einsatz von Human Centered Design mit sich bringen.

Um noch einmal Bezug auf die vier Phasen des Gestaltungsprozesses nach ISO9241 zu nehmen, kann man festhalten, dass die klare Zuordnung der Arbeitsschritte in eine dieser Phasen sehr gut funktioniert hat. Somit wurde jede Arbeitsphase klar strukturiert und war einem definierten Ergebnis zugeordnet. Ein zielführendes und effizientes Arbeiten wurde somit erleichtert.

Implementierung und Usertests

Im Folgenden soll nun auch die Implementierung der Software kritisch hinterfragt werden. Die Ergebnisse der Usertests sollen hierbei mit einbezogen werden und das Meinungsbild mit ersten Testerfahrungen bestärken. Zunächst fällt allgemein auf, dass besonders nutzungsfreundliche Lösungen oftmals technisch verhältnismäßig aufwendig umzusetzen sind. Wird Software von Seite der technischen Möglichkeiten und Datenstrukturen her gestaltet,

drängen sich bestimmte Herangehensweisen oftmals fast unumgänglich auf. Diese Herangehensweisen sind dann meist sehr technisch geprägt und für unerfahrene Nutzende manchmal schwierig zu verstehen. Wenn man diesen Prozess im Sinne des Human Centered Design bewusst umdreht und die Gestaltung bei den Nutzenden und ihrem Umfeld beginnt, können oftmals deutlich passendere und einfachere Workflows und Views geschaffen werden. Diese sind in der Implementierung allerdings oftmals aufwendig zu realisieren. Als nächstes soll nochmal in Erinnerung gerufen werden, dass es sich bei dieser Umsetzung um ein Modul in einer bereits bestehenden Software handelt. Es mussten also alle bestehenden Schnittstellen und Programmierkonzepte der bereits existierenden Module eingehalten werden. Beispielsweise wurde für die Termindatensätze, sowie für Beratungsräume und Mailtemplates, ein objektorientierter Ansatz gewählt. Ziel war es, damit die Beziehungen dieser Datentypen untereinander klarer modellieren und elegant in Code umsetzen zu können. Im Rahmen der gesamten Software Stubegru hat sich allerdings gezeigt, dass es eher wenig Sinn macht, lokale Instanzen von Objekten zu erstellen und zu referenzieren. Die restlichen Module funktionieren so, dass Daten meist nur vom Server abgerufen und angezeigt werden. Weitere Referenzen oder Methodenaufrufe auf diesen Daten sind nicht vorgesehen und machen in diesem Anwendungsfall auch wenig Sinn. Die erhofften Vorteile der objektorientierten Umsetzung in diesem Modul konnten somit nicht vollständig erreicht werden. Viele Methoden wurden letztendlich als *static* implementiert und werden somit unabhängig von der Instanz einer Klasse aufgerufen.

Im Usertest hat sich herausgestellt, dass es extrem hilfreich ist, wenn man eine Gestaltungslösung vorzeigen und gemeinsam durchspielen kann. Beim gemeinsamen Nutzen der implementierten Funktionen fallen oft wichtige Details auf, die im Vorhinein bei der Planung noch nicht berücksichtigt wurden. Somit war es also extrem hilfreich, für den Usertest bereits das fast vollständig fertig implementierte Modul zur Terminvereinbarung präsentieren zu können und somit eine greifbare Gesprächsgrundlage zu bieten. Vielleicht wäre es hilfreich gewesen, schon früher im Gestaltungsprozess erste Prototypen zu entwickeln, die noch nicht voll funktionsstüchtig sind, allerdings schon die wichtigsten Nutzungsinteraktionen wiedergeben können. Dann hätte man den Nutzenden schon früher im Designprozess ein anschauliches Beispiel für konkrete Gestaltungslösungen präsentieren können. Somit hätten etwaige Mängel schon früher erkannt und direkt korrigiert werden können. Allerdings lässt sich auch festhalten, dass durch das ausführliche Interview im Kontext

bereits ein sehr guter Überblick entstanden ist, wie die Software aussehen sollte, um die Nutzenden möglichst gut zu unterstützen. Somit waren nach dem Usertest gar keine großen, grundlegenden Änderungen mehr notwendig. Dies zeigt auf, dass eine ausführliche und gewissenhafte Durchführung der ersten Phasen im Gestaltungsprozess die Wahrscheinlichkeit für Missverständnisse oder größere Änderungswünsche in späteren Usertests senkt.

7.3 Ausblick

Als abschließender Abschnitt dieser Ausarbeitung soll die weitere Perspektive des Forschungsthemas betrachtet werden. Hierbei ist es interessant, den weiteren Einsatz des praktisch entwickelten Moduls zur Terminvereinbarung zu skizzieren. Wie wird sich das Modul in das gesamte Softwarepaket Stubegru eingliedern? In welchem Kontext kann es tatsächlich produktiven Einsatz finden und welche Schritte sind dafür noch zu bedenken? Außerdem sollen die verwendeten Methoden und Theorien nochmals aufgegriffen werden. An welchen Stellen lassen sich die verwendeten Konzepte eventuell noch verbessern? In welcher Richtung verspricht eine weitere wissenschaftliche Forschung spannende neue Erkenntnisse?

Werdegang der Software Stubegru

Wie bereits in Abschnitt 7.1 erwähnt, sind noch weitere Iterationen und Feedbackrunden mit den Nutzenden nötig, um das neu entwickelte Modul zur Terminvereinbarung tatsächlich sinnvoll im Arbeitsalltag zu verwenden. Es ist konkret geplant, diese weiteren Entwicklungszyklen zu durchlaufen und eine neue Version der Software Stubegru in der Abteilung Studium und Lehre produktiv einzusetzen. Ist das Modul zur Terminvereinbarung fertig, werden noch andere Module und Workflows der Software überarbeitet werden müssen. Wenn all diese Arbeiten abgeschlossen sind, wird es eine intensive Einführungsphase geben. In den ersten Wochen im produktiven Betrieb ist zu erwarten, dass noch einige Bugs entdeckt und behoben werden müssen. Außerdem müssen die Mitarbeitenden der Abteilung auf die neuen Abläufe in der überarbeiteten Softwareversion geschult werden. All diese Schritte sollen im Rahmen eines Dienstleistungsvertrages mit der Universität Kassel weiterhin betreut werden.

Forschungsausblick

Die Durchführung und Auswertung dieser Arbeit hat gezeigt, dass die verwendeten Theorien des Human Centered Design schon sehr ausgereift und vor allem sehr praxisnah formuliert sind. Die Umsetzung der methodischen Grundkonzepte ließ sich somit sehr gut in die Praxis übertragen. Spannend wäre es allerdings, sich noch tiefgreifender mit den verschiedenen Nutzungsgruppen einer Software auseinander zu setzen. In dieser Arbeit bedeutete der Austausch mit den Nutzenden größtenteils der Kontakt mit einem Studienberater der allgemeinen Studienberatung. Dieser repräsentierte eine Nutzungsgruppe, die bereits jahrelange Erfahrung mit den Prozessen und Abläufen in der Abteilung vorweisen kann. Spannend wäre es, noch weitere Nutzungsgruppen in den Fokus zu rücken. Hilfskräfte beispielsweise sind oftmals nur für einen kurzen Zeitraum mit geringen Wochenstunden angestellt. Eine jahrelange Einarbeitung in die Prozesse des Arbeitsalltags kann hier nicht vorausgesetzt werden. Wie könnte man also diese Nutzungsgruppe besser in der Bedienung der verwendeten Software unterstützen? Könnten zum Beispiel kleine interaktive Hilfetexte helfen, die Arbeitsprozesse spielerisch bei der Verwendung der Software zu erlernen? Hierfür wäre sicherlich ein intensiverer und direkter Austausch mit Hilfskräften der Abteilung notwendig gewesen.

Des Weiteren wäre es relevant, die Unterschiede in der direkten Interaktion mit dem Computer der einzelnen Nutzenden genauer zu untersuchen. In den hier verwendeten Grundlagen zum Human Centered Design lag der Fokus immer darauf, eine Gestaltungslösung zu finden, die möglichst nutzungsfreundliche Schnittstellen bietet[2]. Oftmals bevorzugen verschiedene Nutzende allerdings verschiedene Schnittstellen. Ich persönlich, als technisch sehr erfahrener Nutzer, bevorzuge beispielsweise Eingaben über die Tastatur und arbeite gerne mit Shortcuts. In den Usertests hat sich schnell gezeigt, dass der Testkandidat am liebsten alle Formularfelder mit der Maus ausfüllt. Somit ist für ihn beispielsweise ein grafischer Timepicker sehr wichtig für eine gute User Experience. Für mich wäre es allerdings viel wichtiger, die Möglichkeit zu haben, jedes Eingabefeld auch über Tastaturbefehle zu erreichen. Dieses kleine Beispiel soll illustrieren, dass selbst einzelne Menschen einer Nutzungsgruppe oftmals ganz verschiedene Vorstellungen von einer *angenehmen* und *schönen* Nutzungsoberfläche haben. Diese Differenzen weiter zu untersuchen und allen Nutzenden eine adäquate Schnittstelle zu bieten, wäre also eine weitere spannende Richtung, um die Ergebnisse dieser Arbeit

noch weiter zu vertiefen.

Kapitel 8

Glossar

Bootstrap

Bootstrap ist ein ursprünglich von Twitter entwickeltes *Css* Framework, das standardisierte *Html* Elemente definiert. Somit können Softwaredesignende häufig benötigte Gestaltungselemente direkt verwenden und können gleichzeitig eine einheitliche und vertraute User-Experience bieten.

In dieser Arbeit verwendete Version: Bootstrap v3.3.4 [26]
–Begriff kommt vor auf Seite: 38, 74

Css

Abkürzung für Cascading Style Sheets. Css ist eine textbasierte Stylesheet-Sprache und wird in Verbindung mit *Html* genutzt um die Position, Farbe und Größe der einzelnen Html Elemente zu definieren.

In dieser Arbeit verwendete Version: CSS 3 [27]
–Begriff kommt vor auf Seite: 37, 72, 74

Document Object Model

Das Document Object Model (kurz DOM) definiert eine Programmierschnittstelle um *Html* Dokumente als Baumstruktur darzustellen. Über das DOM kann zum Beispiel mit *Javascript* auf einzelne Html Elemente zugegriffen werden, um diese zu verändern oder auszulesen,
–Begriff kommt vor auf Seite: 38

fetch API

Die fetch API ist eine Schnittstelle zum dynamischen Abrufen und Versenden von Daten über **Http** Requests und kann in **Javascript** standardmäßig verwendet werden. Im Vergleich zu den früher gebräuchlichen XMLHttpRequests bietet die fetch API eine deutliche einfachere Syntax sowie mächtige Funktionalitäten um dem asynchronen Charakter von Anfragen an einen entfernten Server gerecht zu werden [28].

–Begriff kommt vor auf Seite: 46

fullcalendar

Fullcalendar ist eine **Javascript** Bibliothek und stellt fertige Funktionen zur Verfügung um Kalenderansichten und Termindaten einfach und übersichtlich in einem **Html** Dokument anzuzeigen.

In dieser Arbeit verwendete Version: FullCalendar v5.8.0 [29]

–Begriff kommt vor auf Seite: 27, 38, 49

Html

Abkürzung für Hypertext Markup Language. Html ist eine textbasierte Sprache um Dokumente, insbesondere Internetseiten zu Strukturieren. Über Html kann der Text auf einer Internetseite dargestellt, formatiert und positioniert werden. Außerdem können grafische Elemente wie Hyperlinks, Bilder oder Videos eingebunden werden.

In dieser Arbeit verwendete Version: HTML 5 [30]

–Begriff kommt vor auf Seite: 37, 72–74

Http

Die Abkürzung Http steht für Hypertext Transfer Protocol. Dieses Protokoll wurde von Tim Berners-Lee entwickelt um **Html** Dokumente über das World Wide Web verfügbar zu machen. Das Hypertext Transfer Protocol definiert Standards, wie eine Website vom Server zum Client übertragen wird. [31]

–Begriff kommt vor auf Seite: 37, 73

Javascript

JavaScript (Abk. JS) ist eine Skriptsprache, welche für interaktive Inhalte in Webbrowsersn entwickelt wurde. Mit Javascript können Inhalte

von **Html** Seiten angepasst oder dynamisch nachgeladen werden. Außerdem können Interaktionen vom Nutzenden (z.B. Klicken, Scrollen, Tastatureingaben) verarbeitet und ausgewertet werden.

In dieser Arbeit verwendete Version: ECMA Script 2022 (ES13) [32]

–Begriff kommt vor auf Seite: 37, 72–74

jQuery

jQuery ist eine **Javascript** Bibliothek und stellt fertige Funktionen zur Verfügung um **Html** Elemente dynamisch und effizient auszulesen, zu verändern oder einzublenden.

In dieser Arbeit verwendete Version: jQuery v3.6.0 [33]

–Begriff kommt vor auf Seite: 38

JSON

JSON steht für JavaScript Object Notation und ist eine BeschreibungsSprache. Mithilfe von JSON können Datensätze in textueller Form dargestellt werden. Diese Darstellung kann sowohl von Menschen, als auch von Computern einfach gelesen werden. Daher wird JSON inzwischen nicht nur im Zusammenhang mit **Javascript** verwendet, sondern hat sich als Standard für den Datenaustausch im Internet durchgesetzt. [34]

–Begriff kommt vor auf Seite: 37

Modal

Im **Css** Framework **Bootstrap** bezeichnet ein Modal eine Art Popup Fenster. Dieses Fenster legt sich über den bisherigen Bildschirminhalt und zieht somit die Aufmerksamkeit der Nutzenden auf sich. Aufforderungen für Nutzereingaben, oder Detailansichten eines Datensatzes werden oftmals mit Modals umgesetzt.

–Begriff kommt vor auf Seite: 22, 40

MySQL

MySQL ist ein Datenbankverwaltungssystem, das von Oracle entwickelt und maintained wird. MySQL ist OpenSource und kann somit kostenfrei von Softwareentwickelnden genutzt werden um relationale Datenbanken aufzusetzen.

*In dieser Arbeit verwendete Version: **10.9.4-MariaDB** [35]*
–Begriff kommt vor auf Seite: 37, 75

Php

Php ist eine Programmiersprache um serverseitige Befehle auszuführen. Php Skripte werden häufig verwendet um Datenbankoperationen auf einem Webserver auszuführen und die Ergebnisse an einen Webbrowser zu schicken.

*In dieser Arbeit verwendete Version: **PHP 8** [36]*
–Begriff kommt vor auf Seite: 37, 75

Php Data Objects

Die Php Data Objects Erweiterung (PDO) definiert eine Schnittstelle, um in **Php** Skripten auf Datenbanken zuzugreifen. **MySQL** definiert beispielsweise eine PDO-Schnittstelle, die es ermöglicht aus Php Skripten heraus auf Datensätze in der Datenbank zuzugreifen und diese zu bearbeiten.

–Begriff kommt vor auf Seite: 37

Stubegru

Stubegru ist ein Softwarepaket für akademische Beratungsstellen. Als webbasierte Groupware kann Stubegru typische Workflows in einer Studienberatung begleiten: Beratungstermine, Wissensdatenbank, Abwesenheitsmanagement und vieles mehr [13]. Die Software Stubegru wurde gemeinsam von Johannes Schnirring und der Universität Kassel entwickelt. In Abschnitt 11.1 finden sich weitere Links und Referenzen zur Software.

–Begriff kommt vor auf Seite: 14

Tech Stack

Ein Tech Stack beschreibt die Technologien, die zur Entwicklung einer spezifischen Software zum Einsatz kommen. Dazu zählen beispielsweise die verwendeten Programmiersprachen, Bibliotheken, Frameworks oder auch Entwicklungswerkzeuge.

–Begriff kommt vor auf Seite: 37

Timepicker

Kleines grafisches Popup, um Uhrzeiten elegant mit der Maus eingeben zu können. Ähnlich wie ein Datepicker, der allerdings zur grafischen Eingabe eines Datum genutzt werden kann. [37]

–Begriff kommt vor auf Seite: 19, 55

UML

Abkürzung für Unified Modeling Language (vereinheitlichte Modellierungssprache). UML beschreibt Konventionen um Datenmodelle und Workflows von Software und technischen Systemen grafische zu modellieren. [38]

–Begriff kommt vor auf Seite: 44

Usertest

Fertige Software oder Prototypen werden den Nutzenden vorgestellt. Diese sollen vorgegebene Prozesse in der Software durchlaufen. Softwareentwickelnde und Gestaltende beobachten wie die Nutzenden mit dem System interagieren. Anschließend kann das Feedback der Nutzenden eingeholt werden.

–Begriff kommt vor auf Seite: 7

View

Ein View beschreibt eine Ansicht, die Nutzende zu einem Zeitpunkt beim Verwenden einer Software auf dem Bildschirm grafisch angezeigt bekommen. Eine tabellarische Übersicht von Datensätzen oder eine Eingabemaske für einen neuen Datensatz sind Beispiele für einzelne Views.

–Begriff kommt vor auf Seite: 22, 30

Kapitel 9

Literaturverzeichnis

- [1] C. Moser, *User Experience Design: Mit erlebniszentrierter Softwareentwicklung zu Produkten, die begeistern* (X.media.press). Springer Berlin Heidelberg, 2012, ISBN: 9783642133633. Adresse: <https://books.google.de/books?id=F5AhBAAQBAJ>.
- [2] M. L. Hofmann, *Human Centered Design*, 2017. Aufl. Paderborn, Deutschland: Brill | Fink, 2017, S. 187. eprint: <https://elibrary.utb.de/doi/pdf/10.5555/9783846762868>. Adresse: <https://elibrary.utb.de/doi/abs/10.5555/9783846762868>.
- [3] G. Boy, *The Handbook of Human-Machine Interaction: A Human-Centered Design Approach*. CRC Press, 2017, ISBN: 9781317029472. Adresse: <https://books.google.de/books?id=t2kQEAAAQBAJ>.
- [4] A. Dix, J. Finlay, G. D. Abowd und R. Beale, *Human Computer Interaction*, 3. Aufl. Harlow, England: Pearson Prentice Hall, 2003, ISBN: 978-0-13-046109-4. Adresse: <https://hds.hebis.de/ubks/Record/HEB216164192/Holdings#tabnav>.
- [5] K. McGraw und K. Harbison, *User-centered Requirements: The Scenario-based Engineering Process*. CRC Press, 2020, ISBN: 9781000106022. Adresse: <https://books.google.de/books?id=i8EIEAAAQBAJ>.
- [6] M. Kurosu, *Human-Computer Interaction: Human-Centred Design Approaches, Methods, Tools and Environments: 15th International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I* (Lecture Notes in Computer Science). Sprin-

- ger Berlin Heidelberg, 2013, ISBN: 9783642392320. Adresse: <https://books.google.de/books?id=6Ei6BQAAQBAJ>.
- [7] International Organisation for Standardisation (IOS), *Schaubild - Iterativer Designprozess nach ISO 9241*, 1214 Vernier, Geneva Switzerland, 2011. Adresse: <https://www.iso.org/standard/77520.html>.
 - [8] International Organisation for Standardisation (IOS). “DIN EN ISO 9241-210.” (2011), Adresse: <https://www.iso.org/standard/77520.html>.
 - [9] K. Holtzblatt und H. Beyer, *Contextual Design: Defining Customer-Centered Systems* (Interactive Technologies). Elsevier Science, 1997, ISBN: 9780080503042. Adresse: <https://books.google.de/books?id=JxQaQg0ONGIC>.
 - [10] J. A. Jacko, *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Third Edition*, 3rd. USA: CRC Press, Inc., 2012, ISBN: 1439829438.
 - [11] Johannes Schnirring (für diese Arbeit selbst erstellt), *Schaubild Interview im Kontext*, 2022.
 - [12] O. Claves, T. Haubrich und A. Braun. “Allgemeine Studienberatung Universität Kassel.” (2022), Adresse: <https://www.uni-kassel.de/uni/studium/kontakt-und-beratung/servicestellen/allgemeine-studienberatung> (besucht am 05.10.2022).
 - [13] J. Schnirring. “Software Stubegru - Offizielle Website.” (2021), Adresse: <https://stubegru.org/> (besucht am 05.10.2022).
 - [14] E. Gamma, *Entwurfsmuster : Elemente wiederverwendbarer objektorientierter Software* (Programmer’s choice), [6. Aufl.] 2011, ISBN: 9783827330437. Adresse: http://deposit.d-nb.de/cgi-bin/dokserv?id=3541416&prov=M&dok_var=1&dok_ext=htm.
 - [15] Studienberater der Universität Kassel, “Mündliches Zitat aus Interview im Kontext,” Kassel, 2022.
 - [16] F. Cai und M. de Rijke, “A Survey of Query Auto Completion in Information Retrieval,” *Foundations and Trends® in Information Retrieval*, Jg. 10, Nr. 4, S. 273–363, 2016, ISSN: 1554-0669. DOI: 10.1561/1500000055. Adresse: <http://dx.doi.org/10.1561/1500000055>.

- [17] T. J. Hazen, A. Olteanu, G. Kazai, F. Diaz und M. Golebiewski, “On the Social and Technical Challenges of Web Search Autosuggestion Moderation,” *CoRR*, Jg. abs/2007.05039, 2020. arXiv: 2007 . 05039. Adresse: <https://arxiv.org/abs/2007.05039>.
- [18] DIN Deutsches Institut für Normung e. V. “DIN 5008 Schreib- und Gestaltungsregeln für die Text- und Informationsverarbeitung.” (2020), Adresse: <https://www.din.de/de/mitwirken/normenausschuesse/nia/veroeffentlichungen/wdc-beuth:din21:318422674> (besucht am 01.11.2022).
- [19] placetel.de/BroadSoft Germany GmbH. “Telefonnummern richtig schreiben.” (2022), Adresse: <https://www.placetel.de/blog/telefonnummer-schreibweise-richtig-schreiben> (besucht am 01.11.2022).
- [20] R. Molich und J. Nielsen, “Improving a Human-Computer Dialogue,” *Commun. ACM*, Jg. 33, Nr. 3, 338–348, 1990, ISSN: 0001-0782. DOI: 10.1145/77481.77486. Adresse: <https://doi.org/10.1145/77481.77486>.
- [21] S. Dehaene, *The Number Sense: How the Mind Creates Mathematics, Revised and Updated Edition*. Oxford University Press, 2011, ISBN: 9780199910397. Adresse: <https://books.google.de/books?id=1p6XYuwpjUC>.
- [22] S. Dehaene, G. Dehaene-Lambertz und L. Cohen, “Abstract representations of numbers in the animal and human brain,” *Trends in Neurosciences*, Jg. 21, Nr. 8, S. 355–361, 1998, ISSN: 0166-2236. DOI: [https://doi.org/10.1016/S0166-2236\(98\)01263-6](https://doi.org/10.1016/S0166-2236(98)01263-6). Adresse: <https://www.sciencedirect.com/science/article/pii/S0166223698012636>.
- [23] K. Holtzblatt und H. Beyer, *Contextual Design: Evolved* (Synthesis Lectures on Human-Centered Informatics). Springer International Publishing, 2014, ISBN: 9783031010798. Adresse: <https://books.google.de/books?id=yCg4zwEACAAJ>.
- [24] Studienberater der Universität Kassel, “Mündliches Zitat aus Nutzer-test,” Kassel, 2022.
- [25] Mozilla Developer Network MDN. “Html Time Input.” (), Adresse: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/time#handling_browser_support (besucht am 12.12.2022).

- [26] Bootstrap Team. “Bootstrap - Offizielle Website.” (), Adresse: <https://getbootstrap.com/> (besucht am 08.12.2022).
- [27] W3C - Bert Bos. “CSS Specifications.” (), Adresse: <https://www.w3.org/Style/CSS/specs.en.html> (besucht am 08.12.2022).
- [28] Mozilla Developer Network MDN. “Javascript Fetch API.” (), Adresse: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API (besucht am 20.11.2022).
- [29] Fullcalendar Team. “Fullcalendar - Offizielle Website.” (2022), Adresse: <https://fullcalendar.io/> (besucht am 28.10.2022).
- [30] WHATWG Community. “HTML Living Standard.” (), Adresse: <https://html.spec.whatwg.org/multipage/> (besucht am 18.12.2022).
- [31] IETF. “RFC 9113 - HTTP/2.” (), Adresse: <https://datatracker.ietf.org/doc/html/rfc9113> (besucht am 18.12.2022).
- [32] ECMA International. “ECMAScript 2022.” (), Adresse: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/> (besucht am 18.12.2022).
- [33] The jQuery Team. “jQuery - Offizielle Website.” (), Adresse: <https://jquery.com/> (besucht am 08.12.2022).
- [34] JSON.org authors. “ECMA-404 The JSON Data Interchange Standard.” (), Adresse: <https://www.json.org/json-en.html> (besucht am 18.12.2022).
- [35] MariaDB Foundation. “MariaDB - Offizielle Website.” (), Adresse: <https://mariadb.org/> (besucht am 18.12.2022).
- [36] PHP Developement Team. “PHP - Offizielle Website.” (), Adresse: <https://www.php.net/> (besucht am 08.12.2022).
- [37] Wikipedia. “Kalendersteuerelement.” (), Adresse: <https://de.wikipedia.org/wiki/Kalendersteuerelement> (besucht am 09.12.2022).
- [38] Wikipedia. “Unified Modeling Language.” (), Adresse: https://de.wikipedia.org/wiki/Unified_Modeling_Language (besucht am 09.12.2022).

Kapitel 10

Abbildungsverzeichnis

2.1	Iteratives Vorgehen im Human Centered Design nach ISO 9241 [7].	10
2.2	Während eines Interviews im Kontext können viele Aspekte beobachtet werden, die über das benutzte System selbst hinausgehen[11].	12
4.1	Datepicker im Formular zur Erstellung eines Zeitslots.	20
4.2	Dropdown zur Auswahl des Beratungsraums. Der eigene Raum wird immer als oberstes angezeigt.	20
4.3	Kalenderübersicht. Grüne gefärbte Tage zeigen noch freie Zeitslots an. Rot gefärbte Tage weisen auf vergebenen Zeitslots hin. .	21
4.4	Bewegt man den Mauszeiger über einen Tag, erscheinen weiteren Informationen zu den Zeitslots an diesem Tag.	22
4.5	Der Detail-View: Eine Liste mit drei freien Zeitslots.	23
4.6	Formular zum Vergeben eines Zeitslots an eine ratsuchende Person.	23

4.7	Detail-View: Ein Zeitslot wurde nun vergeben und ist für den entsprechenden Kunden reserviert. Hilfskräfte können nur den Namen des Ratsuchenden einsehen.	24
4.8	Detail-View: Der verantwortliche Beratende kann weitere personenbezogene Details einsehen.	25
4.9	Tabellarische Ansicht der Zeitslots mit Einfärbungen der einzelnen Tage.	27
4.10	Monatsübersicht der Beratungstermine in der <i>Bremer Version</i>	28
4.11	Monatsübersicht der Beratungstermine mit farblichen Markierungen.	31
4.12	Filttereinstellungen der Kalenderansicht. Das Dropdown Menü öffnet sich durch Klick auf den Zahnrad Button.	32
4.13	Suche nach Terminen eines Ratsuchenden mit Ergebnisliste.	33
4.14	Detailansicht eines Termins, der bereits an eine ratsuchende Person vergeben wurde.	34
4.15	So werden nationale Festnetz- und Mobilfunknummern nach DIN 5008 richtig geschrieben. Quelle: [19].	35
5.1	Sequenzdiagramm: Laden der Detailansicht für einen vergebenen Termin.	40
5.2	Verschiedene Bereiche des Modals mit Kontrollelementen.	42
5.3	UML Diagramm der Klasse Meeting. Stellt alle Methoden und Eigenschaften eines Objektes dar, das einen Termin repräsentiert.	45
6.1	Monatsübersicht der Termine im November.	53
6.2	Toggles im Dropdown Menü um die angezeigten Termine zu filtern. Hier werden nur freie Termine dargestellt.	54
6.3	Navigationsleiste über der Monatsansicht. Über den grünen Button kann eine neuer Zeitslot erstellt werden.	55
6.4	Erstellen eines neuen Termins und Ausfüllen der Eingabefelder. In diesem Fall wird in Google Chrome ein Timepicker angezeigt.	56
6.5	Ein freier Termin. Studienberatende können Änderungen am Termin vornehmen oder einen Kunden einbuchen.	57
6.6	Ein vergebener Termin. Terminattribute und neue Kundendaten können erst nach Löschen der aktuellen Kundendaten hinterlegt werden.	58

6.7 Suchfunktion, um bereits vergebene Termine anhand des eingebuchten Kunden zu finden. Die Suchanfrage „jo“ ergibt drei Treffer.	59
7.1 Vergleich der Monatsübersicht in der alten und neuen Softwareversion.	62
7.2 Vergleich der Suchfunktion in der alten und neuen Softwareversion.	63
7.3 Vergleich der Detailansicht eines vergebenen Termins in der alten und neuen Softwareversion.	64
7.4 Neuer Button: Mit <i>Speichern und Nächster</i> kann direkt der nächste Termin eingetragen werden.	65
7.5 Details eines vergebenen Termins aus Ansicht einer Hilfskraft.	65
11.1 Sequenzdiagramm, Erstellen eines neuen Termins.	85
11.2 Sequenzdiagramm, Laden der Detailansicht für einen freien Termin.	86
11.3 Sequenzdiagramm, Laden der Detailansicht für einen vergebenen Termin.	87
11.4 Sequenzdiagramm, Vergeben eines Termins und Eintragen der Kundendaten.	88

Kapitel 11

Anhang

11.1 Software Stubegru

GitHub Repository

Der komplette Quellcode der Software Stubegru ist als öffentliches Open-Source Repository auf GitHub verfügbar:

<https://github.com/stubegru/stubegru>

Dieses Repository umfasst den Coder des gesamten Softwarepaket. Alle Skripte, die im Rahmen dieser Bachelorarbeit entstanden sind, finden sich unter:

<https://github.com/stubegru/stubegru/tree/main/modules/calendar2>

Alle Issues für das neue Modul zur Terminvereinbarung sind unter folgendem Milestone zusammengefasst:

<https://github.com/stubegru/stubegru/milestone/1?closed=1>

Website und Live-Demo

Zusätzlich zum Repository mit allen technischen Informationen, finden sich auf der offiziellen Website weitere Infos zum Softwarepaket Stubegru:

<https://stubegru.org>

Über die Website lässt sich ebenfalls eine Live-Demo der Software Stubegru aufrufen:

<https://stubegru.org/demo>.

Um das neu entwickelte Modul zur zweistufigen Terminvereinbarung zu testen kann direkt der entsprechende View aufgerufen werden:

<https://stubegru.org/demo?view=caltest>

11.2 Sequenzdiagramme

Hier finden sich die in Kapitel 5.3 erwähnten Sequenzdiagramme für alle vier betrachteten Workflows.

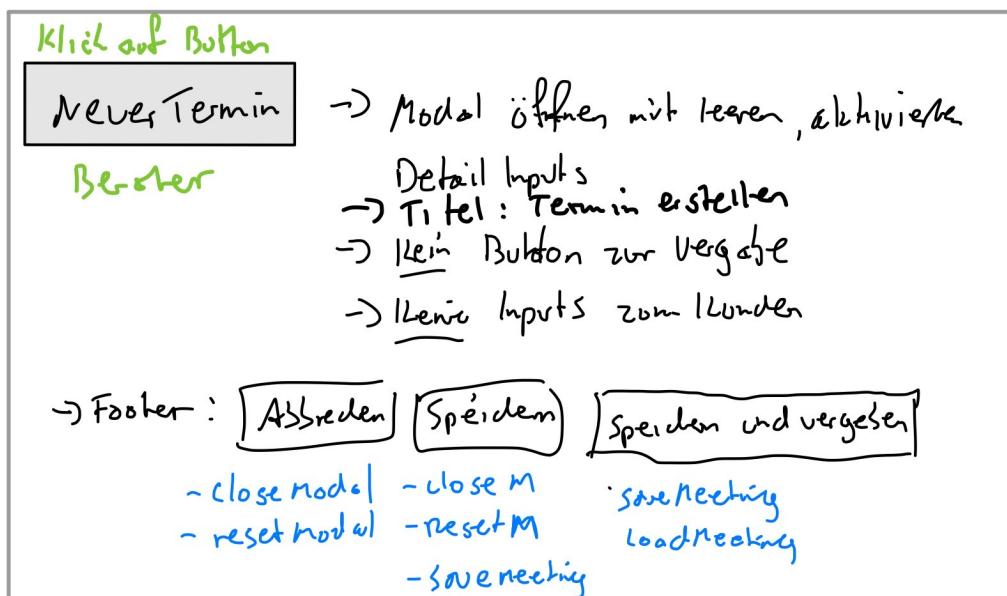


Abbildung 11.1: Sequenzdiagramm, Erstellen eines neuen Termins.

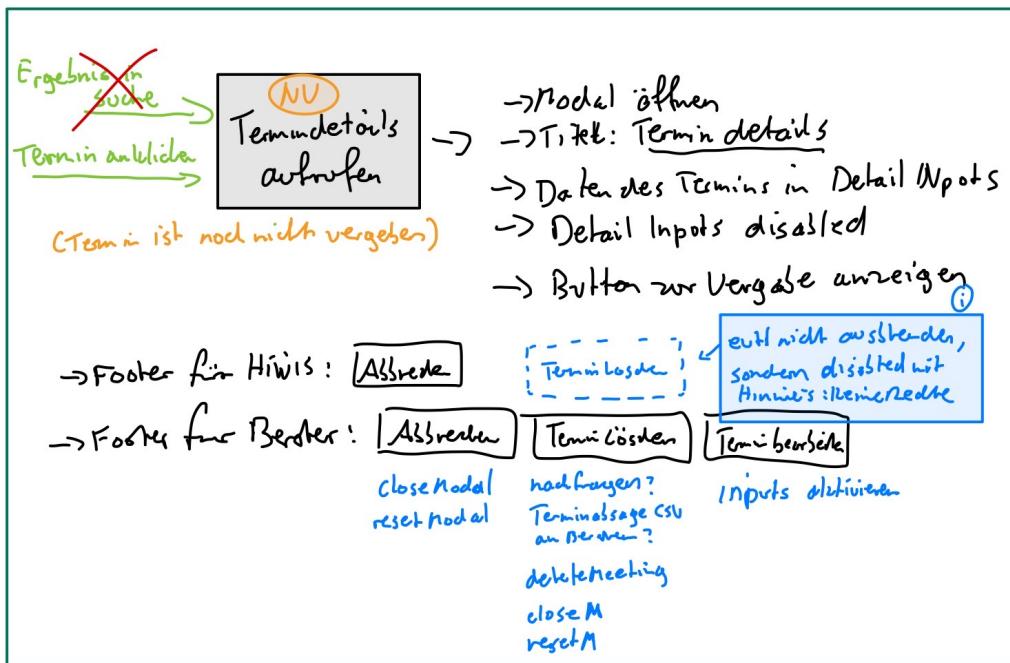


Abbildung 11.2: Sequenzdiagramm, Laden der Detailansicht für einen freien Termin.

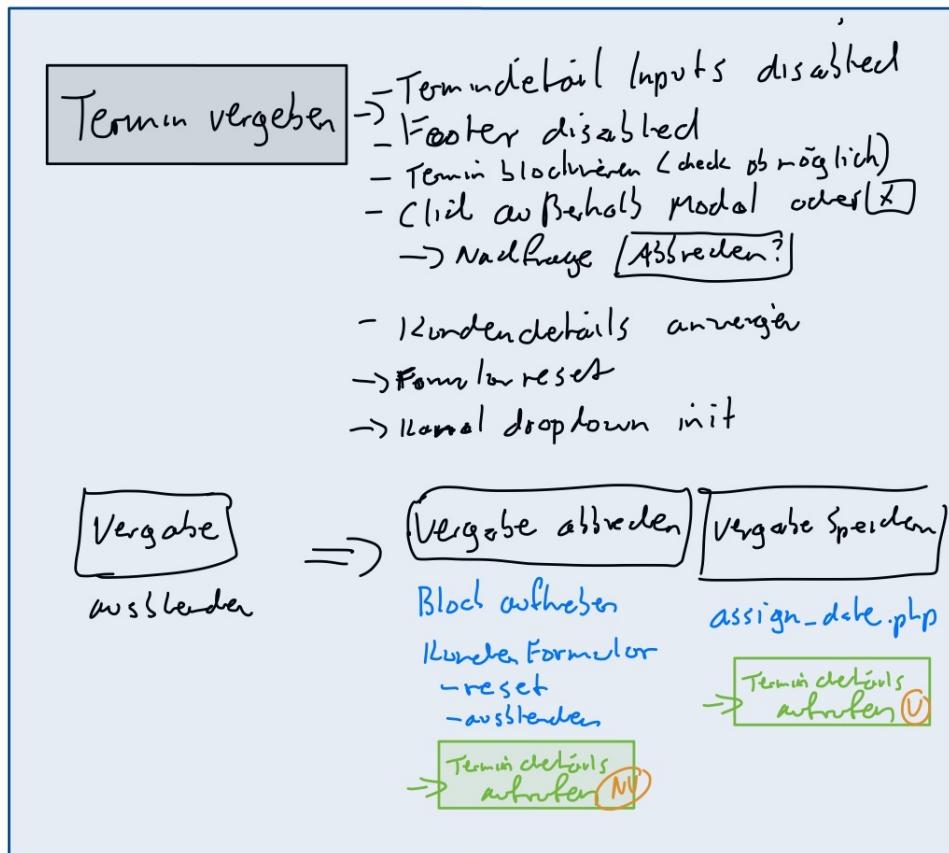


Abbildung 11.3: Sequenzdiagramm, Laden der Detailansicht für einen vergebenen Termin.

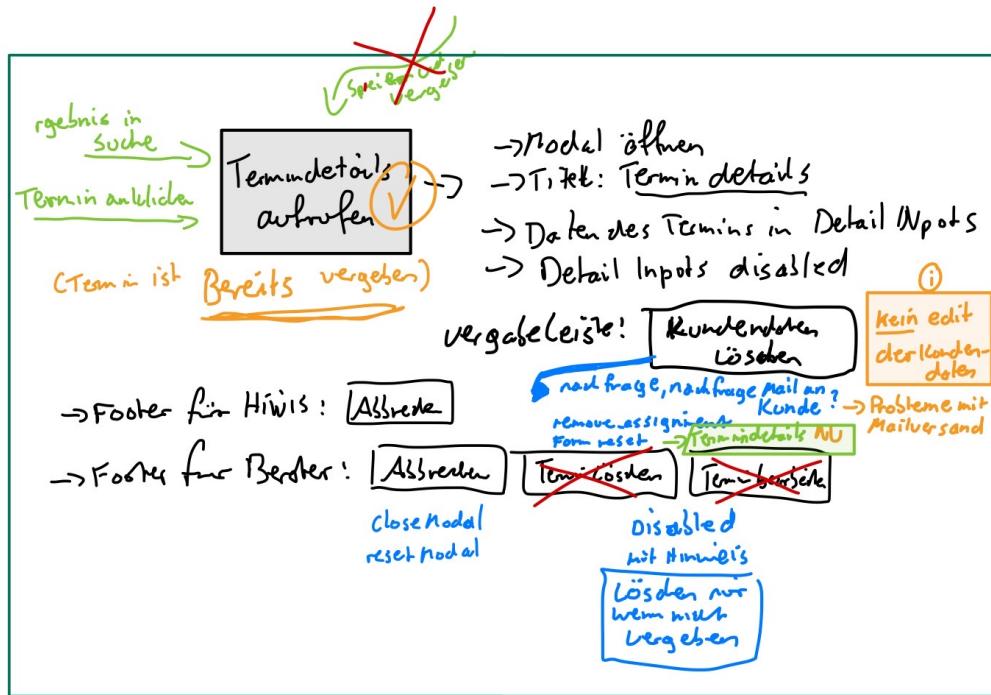


Abbildung 11.4: Sequenzdiagramm, Vergeben eines Termins und Eintragen der Kundendaten.