

# MENSCHZENTRIERTE GESTALTUNG EINER TERMINVEREINBARUNGSSOFTWARE

BACHELORARBEIT

Studierender: Johannes Schnirring

Betreuung: Prof. Dr. Claude Draude

Semester: Wintersemester 2022

Datum: 22. November 2022

Universität Kassel

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Zusammenfassung . . . . .	3
1.2	Motivation . . . . .	3
1.3	Aufbau . . . . .	5
<b>2</b>	<b>Methoden</b>	<b>6</b>
2.1	Human Centered Design . . . . .	6
2.2	Interview im Kontext . . . . .	9
<b>3</b>	<b>Durchführung</b>	<b>10</b>
3.1	Situation in der Abteilung . . . . .	10
3.2	Aktuelle Softwarelösung . . . . .	12
3.3	Grund für Veränderung . . . . .	14
3.4	Nutzungskontext verstehen - Interview im Kontext . . . . .	15
3.4.1	Kontext des Interviews . . . . .	16
3.4.2	Auswertung des Interviews . . . . .	21
3.4.3	Gestaltungslösungen entwickeln . . . . .	25
3.5	Implementierung . . . . .	31
3.5.1	Beschreibung des Prozesses . . . . .	31
3.5.2	Nutzungsanforderungen formalisieren   Sequenzdiagramme . . . . .	34

3.5.3	Technische Umsetzung . . . . .	37
3.6	Testen / User-feedback . . . . .	43
3.6.1	Methode ??? . . . . .	43
3.6.2	Präsentation erster Ergebnisse . . . . .	43
3.6.3	Feedback der Nutzenden . . . . .	43
3.6.4	Ausblick auf weitere Iterationen . . . . .	43
<b>4</b>	<b>Reflektion und Fazit</b>	<b>43</b>
4.1	Beschreibung des Ergebnis . . . . .	43
4.2	Beurteilung der Umsetzungsphase . . . . .	43
4.3	Beurteilung der eingesetzten Methoden . . . . .	44
4.4	Ausblick . . . . .	44

# 1 Einleitung

## 1.1 Zusammenfassung

Diese Arbeit beschäftigt sich mit dem Entwicklungsprozess einer Software zur Terminvergabe von Beratungsterminen der Studienberatung der Universität Kassel. Der Entwicklungsprozess wird mit Methoden des **Human Centred Design** begleitet und umgesetzt. Diese Arbeit erklärt die verwendeten Methoden, sowie die dahinterliegende Theorie. Die Software zur Terminvereinbarung wird implementiert, wobei alle Schritte des Entstehungsprozesses beschrieben, dokumentiert und ausgewertet werden.

Die allgemeine Studienberatung der Universität Kassel setzt die Software Stubegru bereits seit 6 Jahren zum Management ihrer Beratungstermine, Abwesenheiten 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 Herr Maier, als Ansprechpartner aus der Abteilung Studium und Lehre, wird dieses neue Modul entwickelt, designet und implementiert. Die Bachelorarbeit begleitet diesen praktischen Prozess, indem theoretische Grundlagen erläutert werden 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 Entwicklungsprozesses angewandt werden. Hierzu gehört beispielsweise das **Interview im Kontext**, das in dieser Arbeit erklärt wird und dessen praktische Umsetzung dokumentiert, ausgewertet und kritisch hinterfragt wird.

## 1.2 Motivation

### **Historische Entwicklung von Interaktion mit technischen Systemen**

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 haben, diese Systeme gleichermaßen zu nutzen und zu verste-

hen. 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 war meist ein eingeschränkter, speziell dafür ausgebildeter Personenkreis. Diese Situation hat sich gänzlich verändert. Fast jeder Mensch agiert heute viele Male am Tag mit technischen, softwaregetriebenen Systemen, wie zum Beispiel Bankautomaten, Smartphones, Supermarktkassen oder Computern. 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 bedienenden Personen bieten.

**Rolle des Human Centered Design** Der Ansatz des **Human Centered Design** setzt an genau dieser Stelle an, indem er Grundkonzepte und Methoden vermittelt, die das Design von Benutzerschnittstellen thematisieren. Das Kernkonzept des Human Centered Design ist es den Blickwinkel aus der Perspektive der Nutzenden zu wählen. 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.

**Praktische Anwendung Stubegru** Dieser Designansatz soll in dieser Bachelorarbeit genauer untersucht werden. 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 all dieser Methoden und Konzepte dar. Dies soll am Beispiel einer Software zur Terminvergabe in der Studienberatung an der Universität Kassel durchgeführt werden. Das entsprechende Modul für diese Software soll im Rahmen dieser Arbeit geplant, gestaltet, implementiert und getestet werden. All diese Schritte werden schriftlich aufgearbeitet, festgehalten und hinterfragt. Das Ziel dieser Arbeit soll es also sein, den Designzyklus des Human Centered Design am praktischen Beispiel einer Terminvergabe-Software anzuwenden. Die bestehenden wissenschaftlichen Erkenntnisse im Bereich der menschenzentrierten Gestaltung sollen somit erarbeitet, vertieft und auf Praxistauglichkeit geprüft werden. Ein weiterer wichtiger Bestandteil ist schließlich

die theoretischen und methodischen Grundlagen im Hinblick auf die gewonnenen praktischen Erfahrungen zu hinterfragen und deren Praxistauglichkeit zu diskutieren.

### 1.3 Aufbau

Zu Beginn der Arbeit werden zunächst die verwendeten Grundlagen und Methoden des Human Centered Design genauer erläutert und klar definiert. Exemplarisch für eine Methode wird das Konzept des Interviews im Kontext genauer betrachtet und eingeführt. Das nächste Kapitel beschäftigt sich mit der praktischen Durchführung der Entwicklung des Softwaremoduls zur Terminvereinbarung anhand der vorher vorgestellten Methoden. Um dem Lesenden einen Eindruck über den aktuellen Einsatz der betrachteten Software an der Universität zu geben, wird zunächst der Kontext im Team der entsprechenden Abteilung erläutert. Anschließend wird die eingesetzte Software genauer vorgestellt und deren Entwicklungsgeschichte aufgegriffen. Im weiteren Verlauf soll thematisiert werden, aus welchen Gründen eine Überarbeitung dieser Software notwendig ist und welches Ziel die allgemeine Studienberatung damit beabsichtigt. Im nächsten Schritt wird ein Interview im Kontext mit einem Mitarbeitenden der Studienberatung durchgeführt, dokumentiert und ausgewertet. Die aus den Wünschen und Bedürfnissen der Nutzenden entstehenden Anforderungen sollen ebenfalls zusammengefasst und dargestellt werden. Als nächster Schritt soll die praktische Umsetzung der technischen Anforderungen an die Software skizziert werden. Implementierungsdetails, Screenshots und Codeschnipsel sollen in diesem Teil die Dokumentation der Umsetzung veranschaulichen. Als letzter Aspekt dieses Kapitels soll das Feedback der Nutzenden bei Vorstellungen von Prototypen und ersten Ergebnissen festgehalten und aufbereitet werden. Im letzten Kapitel wird schließlich das Ergebnis des durchgeführten 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 eventuell auch Methoden, die in der praktischen Umsetzung problematisch waren oder noch optimiert werden könnten. Der letzte Teil dieser Arbeit gewährt schlussendlich einen Ausblick auf weitere Einsatzmöglichkeiten des Human Centered Design und die weitere Entwicklung der

## 2 Methoden

### 2.1 Human Centered Design

Human Centered Design ist eine Methode die den Entwicklungsprozess von interaktiven Systemen und Software beschreibt. Die Kernthese des Human Centered Design ist, dass die Nutzenden der Systeme im Mittelpunkt aller Entwicklungs- und Designprozesse stehen.

Die Entwicklung von Software war in den Anfängen geprägt von den technisch sehr eingeschränkte Möglichkeiten des Computers. Software wurde also größtenteils so entwickelt, dass sie möglichst problemlos und effizient auf dem entsprechenden Computer ausgeführt werden kann. Alle Entwicklungsprozesse wurden also 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“. [12]

Wie im Kapitel Motivation ausführlich erläutert gewinnt dieser Ansatz in der heutigen Zeit immer größere Bedeutung. Wie Alan Dix in „Human Computer Interaction“ erklärt, hat sich die Art und Weise, wie Menschen und Computer interagieren, in den letzten Jahrzehnten stark verändert. Während Computer früher meist nur mathematische Berechnungen durchführten, sind Softwaresysteme heute hochgradig interaktiv und sollen von allen Teilen der Gesellschaft ungehindert genutzt werden können. [7]

Um die Methodik des Human Centered Design weiter zu spezifizieren ist zunächst festzuhalten, dass der Designprozess von Software als solches eine wichtige Stellung einnimmt. 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 zwi-

schen dem Nutzenden und der Maschine. Man betrachtet, wie die Nutzenden mit dem Computer interagieren. Wie stoßen sie Prozesse an? Welche Reaktionen erwarten sie von dem System? Die Nutzenden sollen mit den Systemen möglichst intuitiv interagieren können. Die Benutzeroberfläche der Software soll selbst erklären, welche Funktionen auf welchem Weg erreicht werden können.

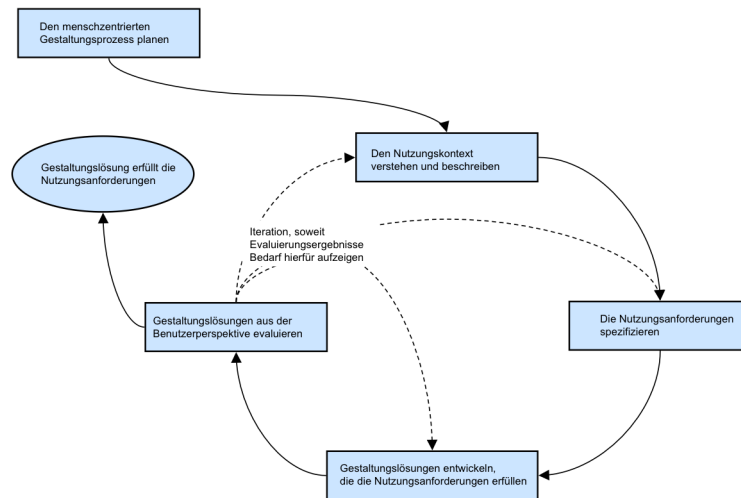
Ein essentieller Bestandteil um die Interaktion zwischen den Nutzenden und der Systeme optimieren zu können, ist zu verstehen, wie die Systeme genutzt werden. Relevante Fragestellung sind hier: In welchem Kontext werden die Systeme eingesetzt? Welche Menschen arbeiten mit den Systemen? Welche Informationen müssen die Nutzenden schnell erfassen können? Um diese Fragen sinnvoll beantworten zu können müssen die Nutzenden 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. [7]

Wie M. Kurosu erwähnt, ist ein weiterer wichtiger Aspekt des Human Centered Design das Durchlaufen mehrere Iterationen. Ein Produkt ist nach einem ersten Entwicklungszyklus selten schon perfekt. Oftmals ist es wichtig den Nutzenden 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 erwähnt, steht der zyklische Ablauf des Human Centered Design auch in der entsprechende ISO Norm im Fokus. [16]

Das Schaubild der Iso Norm verdeutlicht den Ablauf eines Designprozesses in mehreren Phasen. Nachdem geplant ist, was für ein Projekt umgesetzt werden soll beginnt der zyklische Kreislauf des Entwicklungsprozesses. Im ersten Schritt liegt der Fokus auf dem Analysieren 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. Im zweiten Schritt geht es darum aus den geplanten Erwartungen an das System und der Analyse des Nutzungskontextes konkrete Anforderungen zu formulieren. Hierbei geht es noch nicht darum, wie diese Anforderungen technisch umgesetzt werden könnten, sondern darum, welche Anforderungen das fertige System überhaupt erfüllen soll. Im dritten Schritt geht es nun darum diese Anforderungen tatsächlich



Abbildung 1: Iteratives Vorgehen im Human Centered Design nach ISO 9241 [8]



umzusetzen und technische Lösungen zu implementieren. Gibt es erste Ergebnisse wird der vierte Schritt relevant: Die Umgesetzte Lösungen müssen im tatsächlichen Nutzungskontext evaluiert werden. Das setzt wieder eine intensiven Austausch mit den Nutzenden 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 System 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. Neben dem Zuhören und Eingehen auf die Nutzenden und ihr Umfeld ist also auch das Einbringen und zur Diskussion stellen neuer, für die Nutzenden unbekannter Lösungsansätze von hoher Bedeutung. [1]

## 2.2 Interview im Kontext

Ein wesentlicher Bestandteil im Entwicklungsprozess nach den Methoden des Human Centered Design ist der intensive Austausch mit den Nutzenden der Systeme. Hierfür 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 hierfür 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 herkömmliche Interviews begegnen sich Interviewer 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 geht es nicht darum **über** einen Prozess zu sprechen, sondern den Prozess als Interviewender **direkt mitzuerleben**. [13] 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 könne man noch viel mehr praxisnahe Details erfassen, als wenn man mit den Nutzenden nur über das System spricht. 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. [15]

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 beobachtet wie die Personen in ihrer Umgebung, ihrem Kontext agieren. [15] 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: Während eines Interviews im Kontext können viele Aspekte beobachtet werden, die über das benutzte System selbst hinausgehen



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 dabei aus den Beobachtungen des Interviews konkrete Nutzungsanforderung an das zu entwickelnde System zu formulieren. Hier können beispielsweise Featurelisten, Diagramme oder User-Scenarios zum Einsatz kommen. [1]

## 3 Durchführung

### 3.1 Situation in der Abteilung

**Überleitung Situation Abteilung** Um den Bedarf und Entstehungsprozess der Software besser einordnen zu können, werden nun die Workflows und Prozessabläufe im Büroteam 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.

**Grundlegende Beschreibung der Abteilung** Als Nutzergruppe aller Studien dieser Bachelorarbeit werden die Mitarbeitenden einer Abteilung der Hochschulverwaltung an der Universität Kassel dienen. Es handelt sich um die Abteilung "Studium und Lehre", zu deren alltäglichen Aufgaben es gehört, 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 in der Abteilung Studium und Lehre gesetzt.

**Studienberatung** 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 in Gruppen mit Fokus auf bestimmten Fragestellungen austauschen und Qualifikationen im Umgang mit herausfordernden Studiensituationen erlangen. Auch "SSchnupperkurse" für Studieninteressierte und Schüler werden von der allgemeinen Studienberatung angeboten um jungen Menschen mit Interesse an einem Studium einen möglichst unmittelbaren Einblick in den Studienalltag zu gewähren. [4]

**Terminvereinbarung in der ZSB** Eines der zentralen Themen im Alltag der Studienberatung sind Beratungstermine. Studienberatende müssen Ter- gendern?  
mine 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. Selbstverständlich ist auch möglich einen persönlichen Beratungskontakt vor Ort zu vereinbaren. Über all diese Termine muss jeder Studienberatende den Über-

blick 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 SStubegru**" 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. [22]

**Terminvereinbarung in Stubegru** Das wichtigste Modul der eingesetzten Software für die Studienberatung ist der Kalender zur Terminvereinbarung von Beratungsterminen. Über diese 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 Zeitslot 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 eine 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 den Kunden vereinbaren. Wenn ein freier Zeitslot vergeben wird und fest mit einem Ratsuchenden verknüpft ist, wird eine Email an den Beratenden versendet, der über alle Details wie Adresse, Kontaktinformationen und Anliegen der Ratsuchenden informiert wird. Des Weiteren wird eine Mail an die Ratsuchende Person versendet, die auf dem zuvor verknüpften Mailtemplate aufbaut und dynamisch terminrelevante Informationen einsetzt, wie beispielsweise Datum und Zeit 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 Auskunftszwecken 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 eingesetzten Software Stubegru, kann genau gesteuert werden, welche Nutzergruppen Beratungstermine anlegen und vergeben dürfen.

fen.

### 3.3 Grund für Veränderung

**Neue Softwareversion** Die Software Stubegru wurde ursprünglich von einer 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 wiederverwendbarer objektorientierter Software* betont, bieten wiederverwendbar 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. [10] Somit ist der Einsatz der Software Stubegru 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 allgemeinen Studienberatung notwendig.

**Fehlende Features** In der aktuellen 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.

**Vorschau auf Hauptteil der Arbeit** 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 praktische durchgeführten 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.

### 3.4 Nutzungskontext verstehen - Interview im Kontext

**Warum Interview im Kontext?** Das Modul zur Terminvereinbarung der Software Stubegru soll auf den Arbeitsalltag der allgemeinen Studienberatung der Universität Kassel angepasst 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 Nutzenden des Softwaresystems [7] Um den Änderungsbedarf eines bestehenden Softwaresystems einschätzen zu können wird im Human Centered Design häufig die Methode des *Interviews im Kontext* gewählt. [13] Diese Methode eignet sich besonders zu Beginn des Entwicklungsprozesses, da wenig Vorkenntnisse über die eingesetzte Software und das Umfeld, in dem Software 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 Nutzenden der Systeme beeinflussen.

Quelle  
che-  
cken



### 3.4.1 Kontext des Interviews

**Rahmenbedingungen IiK** Als erster Schritt wurde ein Termin für ein Interview im Kontext mit Herr Maier vereinbart. Herr Maier ist einer von drei Mitarbeitenden der allgemeinen 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 erstellt 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 aktuelle nutzt, welche Features für ihn sehr wichtig sind und an welchen Stellen noch Verbesserungspotenzial besteht.

Anonymisierung  
ange-  
ben?

**Detaillierter Ablauf IiK** 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 ein Date-/Timepicker mit der Maus eingegeben werden können. Eine Datumseingabe über die Tastatur würde er nicht bevorzugen.

Beim Eintragen mehrere 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 der 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


Abbildung 3: Datepicker im Formular zur Erstellung eines Zeitslots

Neuer Termin ×


**Titel der Veranstaltung**

Beratungstermin


**Datum der Veranstaltung**

10.10.2022 


**Beginn der Veranstaltung**

11:00 


**Ende der Veranstaltung**

12:00 


**Kanal**

Alle 


**Eigentümer**

Johannes Schnirring 

**Raum**



**Vorlage**



Raum bearbeiten Neuen Raum anlegen


Vorlage bearbeiten Neue Vorlage anlegen

Abbrechen Termin Speichern

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 stattfinden, ist aber auch solch eine Auswahl möglich.

Abbildung 4: Dropdown zur Auswahl des Beratungsraums. Der eigene Raum wird immer als oberstes angezeigt

**Raum**

1603 - Johannes Raum 

1603 - Johannes Raum

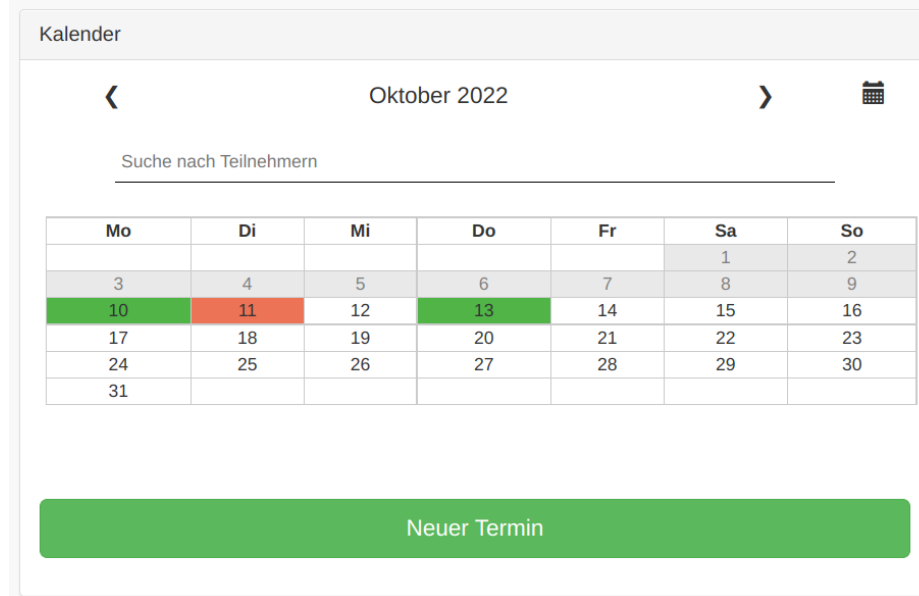
123 - Marias Raum

1428 - Peters Raum

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.

Durch ein Mouseover über den entsprechenden Tag in der Monatsübersicht kann man die genauen Termine mit Informationen über die Uhrzeit, den

Abbildung 5: Kalenderübersicht. Grüne gefärbte Tage zeigen noch freie Zeitslots an. Rot gefärbte Tage weisen auf vergeben Zeitslots hin



zuständigen Beratenden und die Anzahl der freien Plätze sehen. Herr Maier erklärt 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 welche 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 [3].

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 geöffnet 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 Vergabe dieses Zeitslots zur Verfügung.

Abbildung 6: Bewegt man den Mauszeiger über einen Tag, erscheinen weiteren Informationen zu den Zeitslots an diesem Tag

The screenshot displays a booking interface for October 2022. A tooltip is shown over the 10th, providing details for consultation slots. The calendar grid below shows the days of the month, with the 10th, 11th, and 13th highlighted in green. A green button labeled 'Neuer Termin' is at the bottom.

**Tooltip Information:**

- Beratungstermin
- Bei: Johannes Schnirring
- 10:00 - 11:00
- Freie Plätze: 1
- Personliche Notizen

**Calendar Grid:**

Di	Mi	Do	Fr	Sa	So
				1	2
4	5	6	7	8	9
10	11	12	13	14	15
17	18	19	20	21	22
24	25	26	27	28	29
31					

**Buttons:**

- Neuer Termin

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, indem Name, Kontaktdaten und Anliegen der Ratsuchenden erfasst werden können.

Nachdem alle personenbezogenen Daten korrekt erfasst wurden kann der Termin nun endgültig gebucht werden. Hierzu klicken die Hilfskräfte auf den Button "Bestätigen". Herr Maier erklärt mir, dass dies ein sehr wichtiger Schritt ist: Solange eine 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 beispielsweise 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,

Abbildung 7: Der Detail-View: Eine Liste mit drei freien Zeitslots am entsprechenden Datum



dass die terminvergebende Person auf "Abbrechen" klickt, um die Sperre dieses Zeitslots aufzuheben und ihn somit 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 einer ratsuchenden Person wieder absagen zu müssen, als einen Zeitslot versehentlich zu sperren.

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.

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.

Abbildung 8: Formular zum vergeben eines Zeitslots an eine ratsuchende Person

10.10.2022

<b>Beratungstermin</b> Bei: Johannes Schnirring	10:00 - 11:00 Noch frei: 1	Vergeben	✖
<b>Beratungstermin</b> Bei: Johannes Schnirring	12:00 - 13:00 Noch frei: 1	Vergeben	✖

**Vor- und Nachname:**

**E-Mail:**

**Telefon:**

**Fragebogen:** Nein ▾

**Kanal:** Persönlich ▾ Die Auswahlmöglichkeiten bestimmt der Beratende

**Anliegen:**

Abbrechen Bestätigen

Unbedingt Bestätigen oder Abbrechen

### 3.4.2 Auswertung des Interviews

**Einleitung Auswertung** Während bisher der detaillierte Ablauf des Interviews im Kontext 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 überarbeiteten 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 ergänzenden 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

Abbildung 9: Detail-View: Ein Zeitslot wurde nun vergeben und ist für den entsprechenden Kunden reserviert. Hilfskräfte können nur den Namen des Ratsuchenden einsehen

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

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 des iterativen Design Zyklus des Human Centered Design nach ISO 9241 [8]

**Spannende Erkenntnisse** Im Folgenden werden nun drei Punkte exemplarisch vorgestellt, die während des Interviews aufgefallen sind. Anhand dieser drei verschiedenen bestehenden Probleme wird der Designprozess des Human centered Design beispielhaft durchlaufen.

**Kompakte Ansicht Kalender (mit Farben)** In der alten Softwareversion, die an der Uni Kassel bisher zum Einsatz kam, werden alle freien und vergeben Zeitslots der Beratungstermine in einer tabellarischen Monatsansicht dargestellt.

Durch die farblichen Markierungen der einzelnen Tage können 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 al-

Abbildung 10: Detail-View: Der verantwortliche Beratende kann weitere personenbezogene Details einsehen

Termine bearbeiten ✕

10.10.2022

<b>Beratungstermin</b> Bei: Johannes Schnirring	10:00 - 11:00 Noch frei: 1	Vergeben	✕
<b>Beratungstermin</b> Bei: Johannes Schnirring	12:00 - 13:00 Noch frei: 0	Vergeben an: <div> <div>Theodor Test Kanal: personal</div> <div>           Mail: test@example.com           Telefon: 1234         </div> <div>Theodor Test hat Hunger und möchte wissen, was es in der Mensa gibt :-)</div> </div>	
<b>Beratungstermin</b> Bei: Johannes Schnirring	14:00 - 15:00 Noch frei: 1	Vergeben	✕

Zurück

le 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 Javascript Bibliothek *full calendar* [23] gerendert wird.

Dieses 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 Idee um den Hilfskräften der Erstinformation auf den ersten Blick anzuzeigen, 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



Abbildung 11: Tabellarische Ansicht der Zeitslots mit Einfärbungen der einzelnen Tage

Kalender

<

Oktober 2022

>

Suche nach Teilnehmern

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Neuer Termin

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.

**Telefonnummer Anzeige (SSilbentrennung")** 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 Berater vor einem Beratungstermin nochmals telefonisch Details mit der ratsuchenden Person abklären möchte. Der Berater wählt also die angezeigt

Abbildung 12: Monatsübersicht der Beratungstermine in der Bremer Version

Monat

Woche

November 2022

Team

Ich

<

>

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

Telefonnummer in seinem Telefon. Während des Interviews im Kontext zeigt 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 erleichtern.

### 3.4.3 Gestaltungslösungen entwickeln

**Einleitung Gestaltungslösungen** 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

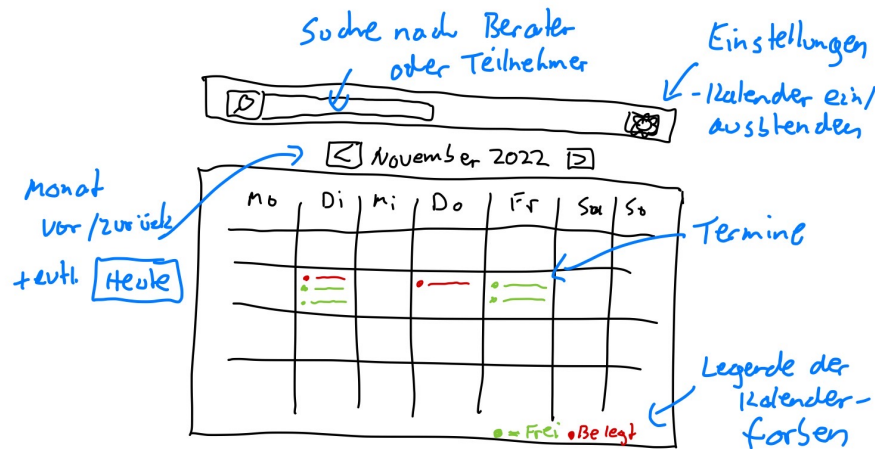
Dix betitelt diese Phase in "Human Computer Interaction als Requirements specification" und betont, dass der Fokus in diesem Schritt darauf liegt, die notwendigen Funktionalitäten und Features der Software grob zu beschreiben. Von besonderer Bedeutung in diesem Schritt des Designzyklus sind Zusammenhänge und Abhängigkeiten zwischen einzelnen Komponenten. Exakte Implementierungsdetails hingegen sind in dieser Phase noch nicht von großer Bedeutung und sollten erst im nächsten Schritt genauer betrachtet werden.

**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, das Nutzende beim späteren Verwenden der Software durchlaufen, wird eine digital gezeichnete Skizze erstellt. Hierbei werden bereits wichtige Elemente wie Buttons, Formularfelder und Hinweisboxen skizziert. Durch Markierungen und Notizen an der Skizze werden die Funktionen dieser Elemente definiert.

**Kompakte Ansicht Kalender (mit Farben)** Die Übersicht aller Termine eines Monats ist die Ansicht, die Nutzende beim Aufruf der Software als erstes sehen. 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 praktische Einsatz tatsächlich pro Tag angelegt werden.

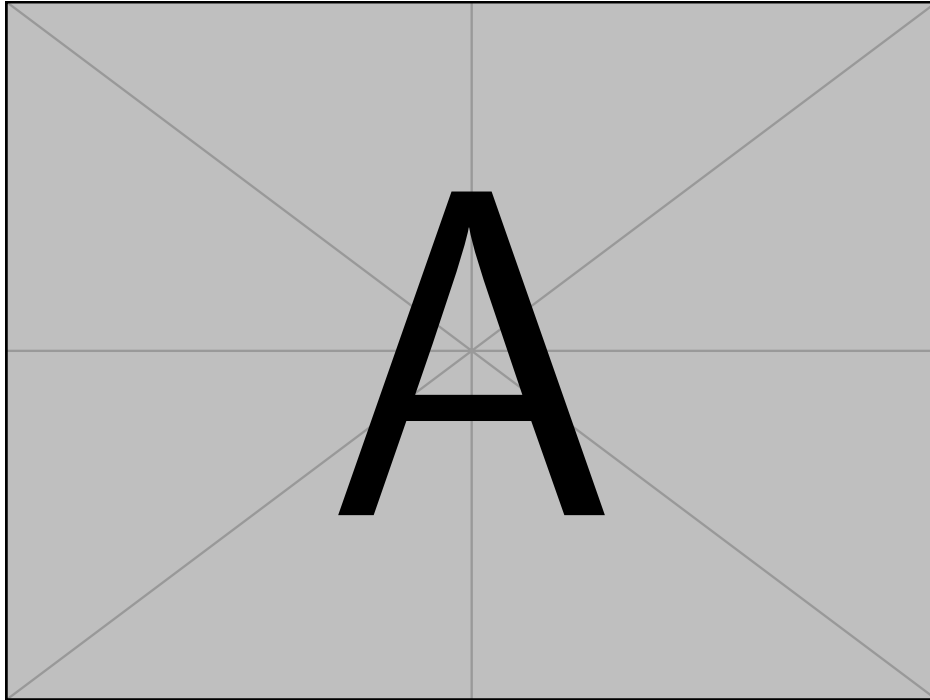
Abbildung 13: Monatsübersicht der Beratungstermine mit farblichen Markierungen



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. Ein Button, um nach einigem hin- und herblättern wieder den aktuelle Monat anzuzeigen, könnte in einigen Anwendungsszenarien viele Klicks ersparen. Über der Leiste mit dem Monat befindet sich eine weitere Kontrollleiste. Diese enthält einen Button um einen neuen Zeitslot anzulegen. Dieser Button sollte nur für Nutzeraccounts von Beratenden 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 einem Zahnrad Symbol soll ein Dropdown-Menü aufklappen, in dem Filter für die Ansicht der Termine gesetzt werden können.

In diesem Menü kann über Toggles eingestellt werden, ob nur eigene Ter-

Abbildung 14: Filtereinstellungen der Kalenderansicht. Das Dropdown Menü öffnet sich durch Klick auf den Zahnrad Button

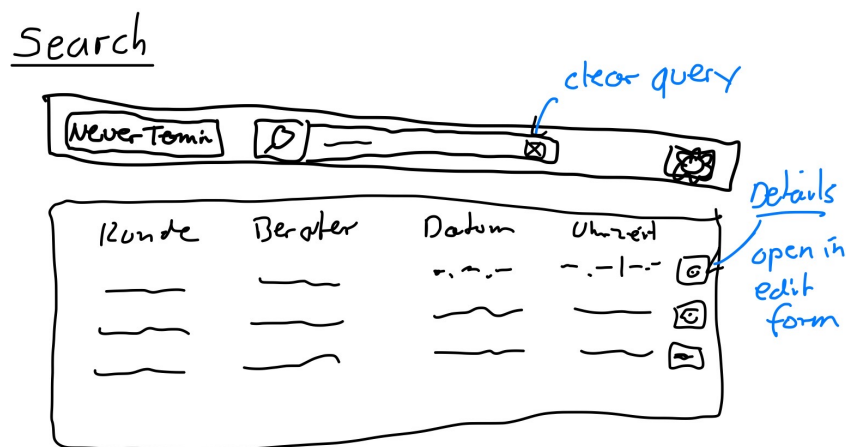


mine oder auch fremde Termine in der Monatsansicht dargestellt werden sollen. Mit *eigenen Terminen* sind Termine gemeint, die den eigenen Benutzeraccount als zuständigen Beratenden hinterlegt haben. Außerdem kann ein Filter gesetzt werden um ausschließlich freie Termine anzuzeigen. Dies kann besonders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

**Suche nach Teilnehmern** Die Suchfunktion ist ein weiterer Aspekt, dem in dieser Ausarbeitung besonderer 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 Suche auf und schiebt den restlichen Inhalt (die tabellarische Monatsansicht) nach unten. In diese Box werden zur Suchanfrage

passenden 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 15: 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 Nutzenden schon während dem Tippen der ersten Buchstaben ein aktives und konstruktives Feedback erhalten, fühlt sich die Nutzung der Software dynamischer und flüssiger an. [2] 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. Diese würden Nutzende von den eigentlich gesuchten Ergebnissen ablenken und sich somit nachteilig auf die User-Experience auswirken. [11]

**Telefonnummeranzeige (SSilbentrennung")** 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 über den Button mit dem Augensymbol in der Detailansicht aufgerufen werden. In dieser Ansicht können Nutzeraccounts mit der entsprechenden Berechtigung nochmals Details des Termins bearbeiten oder den Termin löschen.

Abbildung 16: Detailansicht eines Termins, der bereits an eine ratsuchende Person vergeben wurde

Terminittel  
Bei Name des Beraters

Datum

Name des Ratsuchenden

Anliegen

E-Mail!

Telefon

Fragebogen  
Ja | Nein

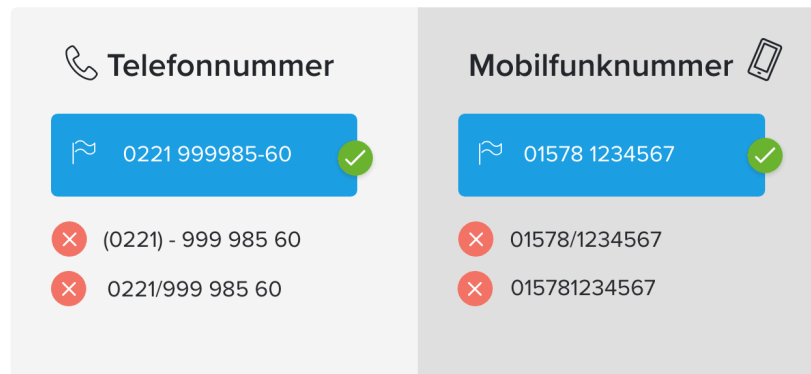
Termin löschen

Absreden | Speichern

Wenn Beratende nach der Vereinbarung eines Termins nochmals auf telefonischem Weg Absprachen oder Vorgespräche mit den Ratsuchenden erledigen 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. [9]

captionSo werden nationale Festnetz- und Mobilfunknummern nach DIN 5008 richtig geschrieben. Quelle: [20]



Wissenschaftliche Untersuchungen zeigen, dass das Eingeben und Ablezen 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. [19]. 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. [20] [5] [6]

## 3.5 Implementierung

### 3.5.1 Beschreibung des Prozesses

**Überleitung** Im vorherigen Schritt wurden konkrete Nutzungsanforderungen herausgearbeitet und Ideen für passende Gestaltungslösungen formuliert. Nun soll das folgende Unterkapitel 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].

**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, das 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.“ [13]. Gleichzeitig müssen alle 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).“ [1] 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. [1]

**Überleitung konkreter Anwendungsfall Stubegru** In dem konkreten Fall des erarbeiteten 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.

**Techstack Stubegru** 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 implementieren.

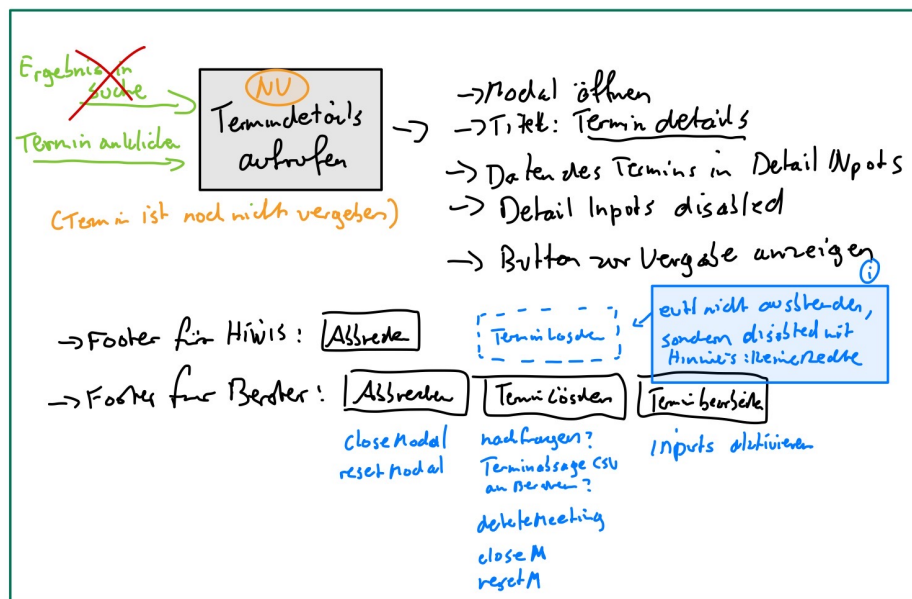
Die Software Stubegru ist webbasiert und wird über einen Browser aufgerufen. Dementsprechend sind die verwendeten Technologien im Frontend Html, Css und Javascript. Die Kommunikation mit dem Backend wird durch asynchrone Http-Requests umgesetzt, die durch Javascript Methoden initiiert werden. Das Backend 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-Database-Objects (PDO) 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 entsprechende Element im Document Object Model (DOM) angepasst und somit für den Nutzenden grafisch dargestellt. 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 grafische 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 dem 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 auch ein, sodass lediglich die angebotenen Ansichten (Monatsansicht, Wochenansicht, Tagesansicht, Terminliste) eingesetzt werden können.

### 3.5.2 Nutzungsanforderungen formalisieren | Sequenzdiagramme

**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 kleinere Teilschritte, die in diesem Workflow nacheinander abgearbeitet werden sollen. Ziel eines Sequenzdiagramms ist es, die einzelnen Aktivitäten, die ein Nutzer beim Verwenden der Software durchläuft, darzustellen. [14] An solchen Diagrammen können auch die Zusammenhänge dieser Aktivitäten untereinander abgelesen werden. Karen L. McGraw verweist in *User-centered Requirements* darauf, dass diese Art von Diagrammen sehr hilfreich sein können, um primäre Workflows und zusammenhängende Prozesse zu identifizieren [17]. Ian Alexander führt in *Scenarios, Stories, Use Cases* ganz ähnliche Diagramm ein, die er Scenario Process Models nennt. Auch hier geht es darum ein Abbild der Teilprozesse zu schaffen, die ein Nutzer beim Verwenden einer Software durchläuft. I. Alexander nennt als weiteren Vorteil dieser Diagramme, dass 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 angestoßen werden können. Teile einzelner Elemente der grafischen Oberfläche werden skizzenhaft dargestellt, um einen intuitiven Eindruck festzuhalten, wie die Nutzer 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. So kann beispielsweise angemerkt werden, dass nach einem Klick auf einen Button, ein Popup mit einer Bestätigungsaufforderung angezeigt werden soll. 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 Nutzer durchlaufen könnte, wenn er jenen Workflow anstößt.

**Beispiel 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 17: Sequenzdiagramm, Laden der Detailansicht für einen vergebenen Termin



Die grünen Pfeile links oben beschrieben, auf welchem Weg 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 *Termin details* geändert. Im weiteren Verlauf müssen die Daten des Termins (Datum, Uhrzeit, Titel) in die entspre-

Modal  
im  
Glossar

chenden 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 an einen Kunden vergebener Termin kann nicht bearbeitet werden, solange der Datensatz des Kunden nicht entfernt wurde. So soll vermieden werden, dass beispielsweise das Datum 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 mit der Kundschaft 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.

Im unteren Teil des Diagramms wird durch kleine Skizzen festgehalten, welche Kontrollelemente den Nutzenden im Workflow zur Verfügung stehen. Die TextitVergabestelle bezeichnet den Bereich, indem Buttons zum Interagieren mit den Kundendaten dargestellt werden. In diesem Fall ist lediglich der Button *Kundendaten löschen* sichtbar. Ein nachträgliches Bearbeiten der Kundendaten 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, das bei einem Mouse-Hover über diese ausgegrauten Buttons ein Hinweis angezeigt werden soll. Dass die Buttons nicht vollständig ausgeblen-

det werden, sondern stattdessen deaktiviert mit einem Hinweis dargestellt werden, soll dazu führen, dass Nutzende eine einheitliche Benutzeroberfläche vorfinden und bei dem Versuch mit den deaktivierten Buttons zu interagieren, automatisch darauf hingewiesen werden, welche Schritte notwendig sind, um die gewünschte Funktion nutzen zu können. Dieser Ansatz soll dazu beitragen, dass auch unerfahrene Nutzende schnell und intuitiv verstehen, wie eine Interaktion mit dem System gedacht ist.

Im Anhang finden sich weitere drei Sequenzdiagramme die folgende Prozesse beschreiben: *Vergebenen Termin aufrufen*, *Neuen Termin erstellen* und *Termin an Kunden vergeben*.

Referenz  
einfü-  
gen

Diagramme  
einfü-  
gen

### 3.5.3 Technische Umsetzung

**Überleitung Technische Details** 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 Designpattern 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 eine 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. [10] 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 des Moduls sind im entsprechenden GitHub Repository einzusehen. [21]

**Datenmodell** 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 Beratungs-

räume. 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.

UML  
erklä-  
ren?

Die Klasse Meeting stellt zunächst einige statische Eigenschaften und Methoden zur Verfügung. Das Array *meetingList* enthält zur Laufzeit alle aktuell bekannten Meetings 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 *fetchApi* [18] mit einer HTTP Request an das entsprechenden PHP Skript auf dem Server gesendet.

Listing 1: Senden von lokalen Änderungen eines Meetings an den Server

```
/**
 * Updates an existing meeting on the server for storage in
 * → database
 */
async function updateOnServer() {
  let formData = this.toFormData();

  const url = `${stubegru.constants.BASE_URL}/modules/
  → calendar/dates/update_calendar_date.php`;
  let meetingResp = await fetch(url, {
```

```

        method: 'POST',
        body: formData
    });
    meetingResp = await meetingResp.json();
    return meetingResp;
}

```

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 Bestätigungsmail an die ratsuchende Person.

Listing 2: PHP Skript zum Erstellen eines neuen Meetings in der Datenbank

```

<?php

// Dieses Script speichert einen neuen Kalendereintrag
$BASE_PATH = getenv("BASE_PATH");
require_once "$BASE_PATH/utils/auth_and_database.php";
require_once "$BASE_PATH/modules/user_utils/user_utils.php";
permissionRequest("MEETINGS_WRITE");
$creatorId = $_SESSION["id"];

$date = $_POST["date"];
$ownerId = $_POST["ownerId"];
$room = $_POST["roomId"];
$start = $_POST["start"];
$end = $_POST["end"];
$title = $_POST["title"];
$template = $_POST["templateId"];

//Channel attribute will only be set by calendar2 frontend
$channel = isset($_POST["channel"]) ? $_POST["channel"] : "
    ↪ unknown";

```



```

$ownerName = getUsername($ownerId);

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);
    ↳ ");
    $insertStatement->bindValue(':date', $date);
    $insertStatement->bindValue(':ownerName', $ownerName);
    $insertStatement->bindValue(':ownerId', $ownerId);
    $insertStatement->bindValue(':room', $room);
    $insertStatement->bindValue(':start', $start);
    $insertStatement->bindValue(':end', $end);
    $insertStatement->bindValue(':title', $title);
    $insertStatement->bindValue(':template', $template);
    $insertStatement->bindValue(':channel', $channel);
    $insertStatement->execute();
    //Id des neu hinzugefuegten Termins abrufen
    $dateId = $dbPdo->lastInsertId();
} catch (Exception $e) {
    echo json_encode(array("status" => "error", "message" => $e
    ↳ ->getMessage()));
    exit;
}
echo json_encode(array("status" => "success", "message" => "Der
↳ Termin wurde erstellt", "dateId" => $dateId));

```

**Workflow: Vergebenen Beratungstermin aufrufen** In Kapitel 3.5.2 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 3: Öffnen der Detailansicht eines vergebenen Beratungstermins

```
static function openAssignedMeeting(meetingId) {
    let C = CalendarController;
    let m = C.modal;

    m.setModalVisible(true);
    m.setModalTitle("Termindetails_(Termin_vergeben)");
    m.resetAllForms();

    const meeting = Meeting.getById(meetingId);
    m.setMeetingDetailData(meeting);
    m.enableDetailMeetingForm(false);

    m.showAssignButtons(false, false, true, false);

    m.setClientVisible(true);
    m.enableClientForm(false);
    m.setClientData(meeting.teilnehmer);

    m.enableFooterButtons(false, false, false, true);

    m.setInfoAlert('<i_class="fas_fa-info-circle"></i>Dieser
        ↳ Termin_ist_bereits_an_einen_Kunden_vergeben.
        ↳ Bearbeiten_des_Termins_ist_nur_moeglich,_nachdem_die_
        ↳ Kundendaten_geloescht_wurden.');
```

**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* [23] verwendet um den Aufbau der Monatsansicht deutlich zu vereinfachen. Die Klasse *CalendarView* referenziert auf eine Instanz vom Typ *FullCalendar* und kann darüber die wichtigs-

ten 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 **eigene Termine** und **fremde Termine** sortiert. Über die Funktion *fullCalendar.addEventSource(eventSource)* werden die Listen mit den eigenen und den fremden Terminen hinzugefügt. Über den weiteren Parameter *color* wird in diesem Fall die farbliche Darstellung der Termine in der Monatsübersicht gesteuert.

Listing 4: Hinzufügen von Terminen in die Monatsübersicht von fullcalendar

```
function addMeetings(meetingList) {
  //Generate events for fullcalendar
  let ownUserId = stubegru.currentUser.id;
  let ownEvents = [];
  let othersEvents = [];

  for (let inMeeting of meetingList) {
    let outMeeting = {
      title: inMeeting.title,
      start: `${inMeeting.date}T${inMeeting.start}`,
      end: `${inMeeting.date}T${inMeeting.end}`,
      extendedProps: inMeeting
    };

    if (inMeeting.ownerId == ownUserId) { ownEvents.push(
      ↪ outMeeting); } else { othersEvents.push(
      ↪ outMeeting); }
  }

  //Generate and add Eventsource
  this.fullCalendar.addEventSource({
    id: "stubegru-own-events",
    events: ownEvents,
    color: "#2196f3",
```

```
        classNames: ["pointer"]
    });

    this.fullCalendar.addEventSource({
        id: "stubegru-others-events",
        events: othersEvents,
        color: "#999ca1",
        classNames: ["pointer"]
    });
}
```

## 3.6 Testen / User-feedback

### 3.6.1 Methode ???

### 3.6.2 Präsentation erster Ergebnisse

### 3.6.3 Feedback der Nutzenden

### 3.6.4 Ausblick auf weitere Iterationen

## 4 Reflektion und Fazit

### 4.1 Beschreibung des Ergebnis

### 4.2 Beurteilung der Umsetzungsphase

Problem: Designpattern bei fertigem Softwareökosystem sehr eingeschränkt  
Fullcalendar Lib: Weniger Arbeit (billiger, robuster) dafür weniger Flexibilität

### 4.3 Beurteilung der eingesetzten Methoden

### 4.4 Ausblick

## Literatur

- [1] G.A. Boy. *The Handbook of Human-Machine Interaction: A Human-Centered Design Approach*. CRC Press, 2017.
- [2] Fei Cai and Maarten de Rijke. A survey of query auto completion in information retrieval. *Foundations and Trends® in Information Retrieval*, 10(4):273–363, 2016.
- [3] Oliver Claves. Interview im kontext. 04.10.2022.
- [4] Oliver Claves, Thomas Haubrich, and Andrea Braun. Allgemeine studienberatung universität kassel, 2022.
- [5] S. Dehaene. *The Number Sense: How the Mind Creates Mathematics, Revised and Updated Edition*. Oxford University Press, 2011.
- [6] Stanislas Dehaene, Ghislaine Dehaene-Lambertz, and Laurent Cohen. Abstract representations of numbers in the animal and human brain. *Trends in Neurosciences*, 21(8):355–361, 1998.
- [7] Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale. *Human Computer Interaction*. Pearson Prentice Hall, Harlow, England, 3 edition, 2003.
- [8] International Organisation for Standardisation (IOS). *DIN EN ISO 9241-210*. International Organisation for Standardisation (IOS), 1214 Vernier, Geneva Switzerland, 2011.
- [9] DIN Deutsches Institut für Normung e. V. Din 5008 schreib- und gestaltungsregeln für die text- und informationsverarbeitung, 2020.
- [10] Erich Gamma. *Entwurfsmuster : Elemente wiederverwendbarer objekt-orientierter Software*. Programmer’s choice. [6. aufl.] edition, 2011.
- [11] Timothy J. Hazen, Alexandra Olteanu, Gabriella Kazai, Fernando Diaz, and Michael Golebiewski. On the social and technical challenges of web search autosuggestion moderation. *CoRR*, abs/2007.05039, 2020.
- [12] Martin Ludwig Hofmann. *Human Centered Design*. Brill | Fink, Paderborn, Deutschland, 2017 edition, 2017.

- [13] K. Holtzblatt and H. Beyer. *Contextual Design: Defining Customer-Centered Systems*. Interactive Technologies. Elsevier Science, 1997.
- [14] K. Holtzblatt and H. Beyer. *Contextual Design: Evolved*. Synthesis Lectures on Human-Centered Informatics. Springer International Publishing, 2014.
- [15] Julie A. Jacko. *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Third Edition*. CRC Press, Inc., USA, 3rd edition, 2012.
- [16] 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. Springer Berlin Heidelberg, 2013.
- [17] K.L. McGraw and K. Harbison. *User-centered Requirements: The Scenario-based Engineering Process*. CRC Press, 2020.
- [18] Mozilla Developer Network MDN. Javascript fetch api.
- [19] Rolf Molich and Jakob Nielsen. Improving a human-computer dialogue. *Commun. ACM*, 33(3):338–348, mar 1990.
- [20] placetel.de/BroadSoft Germany GmbH. Telefonnummern richtig schreiben, 2022.
- [21] Johannes Schnirring. Stubegru.
- [22] Johannes Schnirring. Software stubegru, offizielle website, 2021.
- [23] Fullcalendar Team. Fullcalendar - offizielle website, 2022.

Abbildung 18: Verschiedene Bereiche des Modals mit Kontrollelementen

The diagram illustrates the layout of a modal window, divided into several functional areas:

- Modal Title:** Labeled "Termin Details" in an orange box at the top left.
- Termin details:** A blue-bordered section containing:
  - Input fields for "Datum" (01.01.2022), "Start" (10:00), and "Ende" (12:00).
  - Input fields for "Termin Titel" (Studienberatung), "Berater\*in" (Peter Test), and "Kanal" (Vor Digital).
  - Input fields for "Raum" (Zoom Raum) and "Mailvorlage" (Digitales Meeting), each with "Edit" and "Delete" buttons.
- Teilnehmer\*in:** A green-bordered section for participant information:
  - Fields for "Name des Ratsuchenden" (Marta Müller) and "E-Mail:" (marta.mueller.de).
  - Fields for "Anliegen" (Hat Durst) and "Telefon" (0561-1234 5678).
  - A "Fragebogen" section with "Ja" and "Nein" radio buttons.
- Kundendaten:** Labeled in green, pointing to the participant information section.
- Vergabeleistung:** Labeled in red, pointing to a red-bordered box containing "Abbrechen" and "Termin vergeben" buttons.
- Footer:** A purple-bordered section at the bottom containing:
  - A "Termin löschen" button.
  - "Abbrechen" and "Speichern" buttons.



Abbildung 19: UML Diagramm der Klasse Meeting. Stellt alle Methoden und Eigenschaften eines Objektes dar, das einen Termin repräsentiert

