



Konzeption und prototypische Umsetzung eines modularen, regelwerkbasierten Systems für nutzerzentrierte Datenerhebung

Bachelorarbeit

vorgelegt von: Mona Kögel

Studiengang: Media Engineering

Fakultät: Elektrotechnik Feinwerktechnik Informationstechnik efi

Matrikelnummer: 3038524

Erstgutachter: Prof. Dr. Hans-Georg Hopf

Zweitgutachter: Prof. Dr. Heinz Brünig

Betreuer: Philipp Wolf

Abgabedatum: 03.05.2021

Abstract

Durch immer mehr technologische Innovationen wird der Handelsplatz Internet und auch der damit verbundene Datenverkehr zunehmend größer. Um an die verschiedenen, immer spezifischer werdenden Daten der Nutzer zu gelangen, werden meist Formulare verwendet. Damit sie dieser Entwicklung standhalten können, müssen auch die Formulare immer komplexer werden, was jedoch ihre Erstellung und Benutzbarkeit erheblich erschwert. Die vorliegende Bachelorarbeit untersucht ein Konzept für ein modulares Baukasten-System für Formulare mit eingebundenem Regelwerk. Dieses soll zum einen dafür sorgen, dass Nutzer nur die Fragen beantworten müssen, welche für sie eine hohe Relevanz haben. Zum anderen soll die Erstellung durch das modulare System so einfach gestaltet sein, dass auch Content-Creator und Designer Formulare erstellen können und somit keine Entwickler mehr benötigt werden. Dieses Konzept wird anhand des Themenkomplexes Steuern, insbesondere Werbungskosten, prototypisch umgesetzt. Dabei soll es möglich sein, ein dynamisches Formular aus einzelnen Bausteinen zusammenzusetzen. Der entstandene Prototyp soll dann mit Hilfe von Usability-Tests evaluiert werden, um herauszufinden, ob es für Nicht-Entwickler möglich ist, damit ein Formular zu erstellen.

Hinweis: Diese Erklärung ist in alle Exemplare der Abschlussarbeit fest einzubinden. (Keine Spiralbindung)

Prüfungsrechtliche Erklärung der/des Studierenden

Angaben des bzw. der Studierenden:

Name: Kögel Vorname: Mona Matrikel-Nr.: 3038524

Fakultät: Elektro-, Feinwerk-, Informationstechnik Studiengang: Media Engineering

Semester: Sommersemester 2021

Titel der Abschlussarbeit:

Konzeption und prototypische Umsetzung eines modularen, regelwerkbasierten Systems für nutzerzentrierte Datenerhebung

Ich versichere, dass ich die Arbeit selbständig verfasst, nicht anderweitig für Prüfungszwecke vorgelegt, alle benutzten Quellen und Hilfsmittel angegeben sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Lauf, den 03.05.2021 

Ort, Datum, Unterschrift Studierende/Studierender

Erklärung zur Veröffentlichung der vorstehend bezeichneten Abschlussarbeit

Die Entscheidung über die vollständige oder auszugsweise Veröffentlichung der Abschlussarbeit liegt grundsätzlich erst einmal allein in der Zuständigkeit der/des studentischen Verfasserin/Verfassers. Nach dem Urheberrechtsgesetz (UrhG) erwirbt die Verfasserin/der Verfasser einer Abschlussarbeit mit Anfertigung ihrer/seiner Arbeit das alleinige Urheberrecht und grundsätzlich auch die hieraus resultierenden Nutzungsrechte wie z.B. Erstveröffentlichung (§ 12 UrhG), Verbreitung (§ 17 UrhG), Vervielfältigung (§ 16 UrhG), Online-Nutzung usw., also alle Rechte, die die nicht-kommerzielle oder kommerzielle Verwertung betreffen.

Die Hochschule und deren Beschäftigte werden Abschlussarbeiten oder Teile davon nicht ohne Zustimmung der/des studentischen Verfasserin/Verfassers veröffentlichen, insbesondere nicht öffentlich zugänglich in die Bibliothek der Hochschule einstellen.

Hiermit genehmige ich, wenn und soweit keine entgegenstehenden Vereinbarungen mit Dritten getroffen worden sind,
 genehmige ich nicht,

dass die oben genannte Abschlussarbeit durch die Technische Hochschule Nürnberg Georg Simon Ohm, ggf. nach Ablauf einer mittels eines auf der Abschlussarbeit aufgebrachten Sperrvermerks kenntlich gemachten Sperrfrist

von 0 Jahren (0 - 5 Jahren ab Datum der Abgabe der Arbeit),

der Öffentlichkeit zugänglich gemacht wird. Im Falle der Genehmigung erfolgt diese unwiderruflich; hierzu wird der Abschlussarbeit ein Exemplar im digitalisierten PDF-Format auf einem Datenträger beigefügt. Bestimmungen der jeweils geltenden Studien- und Prüfungsordnung über Art und Umfang der im Rahmen der Arbeit abzugebenden Exemplare und Materialien werden hierdurch nicht berührt.

Lauf, den 03.05.2021 

Ort, Datum, Unterschrift Studierende/Studierender

[Formular drucken](#)

Datenschutz: Die Antragstellung ist regelmäßig mit der Speicherung und Verarbeitung der von Ihnen mitgeteilten Daten durch die Technische Hochschule Nürnberg Georg Simon Ohm verbunden. Weitere Informationen zum Umgang der Technischen Hochschule Nürnberg mit Ihren personenbezogenen Daten sind unter nachfolgendem Link abrufbar: <https://www.th-nuernberg.de/datenschutz/>

Inhaltsverzeichnis

1 - Einleitung	1
1.1 Motivation	1
1.2 Ziel der Arbeit	2
1.3 Projektträger	2
1.4 Aufbau der Arbeit	3
2 - Grundlagen, Methoden und Tools	4
2.1 Usability	4
2.2 Human Centered Design Prozess	5
2.3 Best Practices für benutzerfreundliches und effizientes UX- und UI-Design von Formularen	5
2.4 Atomic Design	8
2.5 Modulare Systeme	10
2.5.1 Angular	10
2.5.2 Apple Research Kit	11
2.5.3 Storybook	12
2.6 Design Systems	13
2.6.1 Material Design von Google	14
2.6.2 Angular Material	15
2.6.3 Sketch & Illustrator	16
2.7 Methoden zur Implementierung der Barrierefreiheit	16
2.7.1 Persona-Spektrum von Microsoft	17
2.7.2 Entwurf eines inklusiven Farbschemas mit Hilfe des Barrierefreiheits-Tool von Adobe Color	18
3 - Nutzungskontext und -anforderungen	21
3.1 Personas	21
3.2 Value Proposition Canvas	23
3.3 Themenkomplex Steuern und Werbungskosten	28
4 - Gestaltungslösung	29
4.1 Konzeption und Design der Prototypen	29
4.1.1 Konzeption und Design der UI-Elemente	30
4.1.2 Konzeption des Formulars	31
4.1.2 Konzeption des Baukasten-Systems	31
4.2 Prototypische Umsetzung	32
4.2.1 Entwicklung der einzelnen UI-Elemente	32
4.2.1 Entwicklung des Formular-Prototypen	36
4.2.1 Entwicklung des Baukasten-System-Prototypen	37

5 - Evaluation mit Hilfe von Usability-Tests	39
5.1 Entwicklung eines Online-Testkonzepts	39
5.2 Testaufbau	40
5.3 System Usability Scale	42
5.4 Evaluation	44
6 - Schlussbetrachtung	46
6.1 Zusammenfassung	47
6.2 Fazit	47
6.3 Ausblick	48
Literaturverzeichnis	50
A - Anhang	1
A.1 Personas	1
A.2 UI-Elemente & Icons	6
A.3 Formular-Prototyp	8
A.4 Baukasten-System-Prototyp	16
A.5 Test-Skript	18
A.6 Auswertung Pre-Test Fragebogen	21
A.7 Auswertung SUS-Fragebögen	23
A.8 Auswertung des zusätzlichen Feedbacks: Pilot-Test	24
A.9 Auswertung des zusätzlichen Feedbacks: Usability-Test	26
A.10 Weiterführende Links	28

Abkürzungsverzeichnis

VPC - Value Proposition Canvas

UX - User Experience

UI - User Interface

SUS - System Usability Scale

CTA - Concurrent Think Aloud

Abbildungsverzeichnis

- 1.1** Logo mit Slogan der adorsys GmbH & Co.KG (Quelle: [3])
- 2.1** Gruppieren von Formularfeldern beim Entwerfen von Formularen (Quelle: [7])
- 2.2** Standard-zweispaltiges Layout im Vergleich zu einem einspaltigen Layout (Quelle: [7])
- 2.3** Vergleich von links ausgerichteten, rechts ausgerichteten und oben ausgerichteten Labels (Quelle: [7],[8])
- 2.4** Aufbau des Atomic Design Prozesses (Quelle: [10])
- 2.5** Beispiele für anpassbare Screens, die in Umfragen verwendet werden können (Quelle: [14])
- 2.6** Ein Button, erstellt in der Storybook-Anwendung (Quelle: [16])
- 2.7** Beispiele für Komponenten, die Material Design zum Entwerfen und Entwickeln von Interfaces bietet (Quelle: [18])
- 2.8** Microsoft Persona-Spektrum (Quelle: [22])
- 2.9** Die verschiedenen Formen von Farbschwäche (Quelle: [23])
- 2.10** Entworfenes Farbschema (Ansicht im Adobe Color Tool (Quelle: [24]))
- 3.1** Die Persona von “Anton Müller” und “Linda König”
- 3.2** Value Proposition Canvas (eigene Darstellung in Anlehnung an ([27], Kap. 1.3: Fit))
- 3.3** Screens aus der Taxfix-App (Quelle: [29])
- 4.1** Die einzelnen Formular-Elemente dargestellt in Sketch
- 4.2** Der Formular-Prototyp beim Schritt “Fahrtkosten”, dargestellt im Browser
- 4.3** Das Baukasten-Systems, dargestellt im Browser
- 5.1** Erste Testaufgabe für das Building-Kit in Miro für das Online-Testkonzept
- 5.2** System Usability Scale Acceptability Score (Quelle: [33])
- 5.3** Kreisdiagramm zu den Ergebnissen der Frage: “Machst du deine Steuererklärung selbst?”

1 - Einleitung

In der folgenden Arbeit werden die Begriffe Benutzer, Nutzer, Tester und das englische Pendant User gleichgestellt benutzt. Aus Gründen der besseren Lesbarkeit wird bei Personenbezeichnungen und personenbezogenen Hauptwörtern die männliche Form verwendet. Entsprechende Begriffe gelten gleichermaßen für alle Geschlechter.

Die Zitate in dieser Arbeit sind eigene Übersetzungen aus dem Englischen ins Deutsche. Die Quelle bezieht sich dabei auf den jeweiligen Absatz, an dessen Ende sie steht.

1.1 Motivation

Die meisten Interaktionen, die im Web stattfinden, werden durch irgendeine Art von Formular erreicht. Ohne Formulare wird das Web zu einer passiven Erfahrung - lediglich einer Möglichkeit, Inhalte zu konsumieren. Mit Formularen können Benutzer Dinge erstellen, aktualisieren und löschen. Ob es um die Kommunikation per E-Mail, den Kauf eines Produkts, Online-Banking oder die Arbeit an einem digitalen Verwaltungsdienst geht: Formulare stehen immer im Vordergrund. ([1], Introduction S.8)

Die bisherige Handhabung in Projekten der Software Entwicklung besteht meist daraus, dass Entwickler, auf Basis eines Prototypen, fertige Software entwickeln. Dabei werden auch Formulare, welche ähnlich schon in anderen Projekten genutzt werden, oftmals neu programmiert. Dieses Vorgehen erzeugt einen erhöhten Mehraufwand und es wird an Implementierungen gearbeitet, die ähnlich schon existieren. Ein effizienteres und innovationsorientiertes Entwicklungsvorgehen könnte erreicht werden, wenn bereits existierende Software nicht unnötig neu entwickelt wird, sondern bereits entwickelte Lösungen verbessert werden. Dadurch könnte auch die Erfahrung der Benutzer verbessert werden. Durch die Lösung des gleichen Problems auf die gleiche Weise haben Nutzer eine konsistente und kohärente Erfahrung. Der Dienst, die App oder Ähnliches wirkt vertraut. Bekannte Interfaces erfordern weniger Aufwand für ihre Bedienung. ([1], Introduction S.9)

Durch die immer wieder neue Implementierung bereits ähnlich existierender Lösungen geraten andere Probleme außer Sichtweite. Unter anderem werden die meist sehr hohen Usability- und Accessibility-Standards dadurch vernachlässigt.

Durch die Knappheit der Ressourcen in vielen Software-Projekten gerät besonders die Accessibility der Software oft in den Hintergrund. Es gibt selten genug ausgebildete Entwickler, Zeit und Geld, um die Inklusivität der Software auszubauen. Die Dringlichkeit, insbesondere in wachstumsorientierten Unternehmen, ist ein ständiger Druck und Kompromisse zu finden meist nicht einfach. ([2], S.11)

So stellen meist simple Formulare sowohl für den alltäglichen Nutzer und für Menschen mit körperlichen Einschränkungen, als auch für den Ersteller eine Herausforderung dar. Diese Arbeit widmet sich diesem Problem.

1.2 Ziel der Arbeit

Das Ziel der Arbeit besteht darin, ein Design System zu konzipieren, wodurch benutzerfreundlich gestaltete Formular-Elemente einfach wiederverwendet werden können. Dies kann geschehen, ohne dass Entwickler dafür ein neues Formular entwickeln müssen. Dieses Design System soll so gestaltet sein, dass selbst Menschen ohne tiefergehende, technische Kenntnisse ein solches Formular erstellen können. Das Design System soll eine Art Regelwerk beinhalten, womit die Ersteller die Details der einzelnen Bausteine bearbeiten und auch deren Styling anpassen können. Durch ein anpassbares Styling könnte beispielsweise auch das Branding für Firmen vereinfacht werden, da Farben und weitere Styles ganz einfach ausgetauscht werden können. Mit Hilfe eines Prototypen werden jedoch zunächst die Grundlagen für ein solches modulares System betrachtet.

1.3 Projektträger

Die Arbeit wurde von der adorsys GmbH & Co. KG unterstützt und von Philipp Wolf, welcher als UX Consultant in der adorsys tätig ist, betreut. Die adorsys wurde 2006 von Francis Pouatcha gegründet und entwickelt seitdem individuelle Software für Banken und Versicherungen. Durch diese neue und innovative Software verschafft die adorsys ihren Kunden einen großen Wettbewerbsvorteil. Die Teams begleiten die Klienten dabei von der ersten Idee bis zum fertigen

Produkt. Es werden sowohl bestehende Services modernisiert, als auch Lösungen für ein neues Geschäftsmodell geschaffen.



Abbildung 1.1: Logo mit Slogan der adorsys GmbH & Co.KG (Quelle: [3])

Die adorsys hat ihren Hauptstandort in Nürnberg und seit 2017 noch einen weiteren Standort in Frankfurt am Main. Seit der Gründung ist die adorsys stark gewachsen und beschäftigt derzeit über 130 Mitarbeiter, wovon sich der Großteil in Nürnberg befindet.

Zu den Kunden der adorsys zählen hauptsächlich Banken wie beispielsweise die Teambank AG, Bausparkasse Schwäbisch Hall AG, Consorsbank, Wüstenrot Bausparkasse AG sowie Versicherungen wie ERGO Direkt AG und die Nürnberger Versicherung.

Die adorsys hat in allen Bereichen der Software-Entwicklung große Kompetenzen entwickelt, welche auch in die Zusammenarbeit mit den Studierenden bei Abschlussarbeiten mit einfließen.

1.4 Aufbau der Arbeit

Zunächst soll anhand der vorangegangenen Recherche in den Bereichen Atomic Design, Design Systems, Usability und Accessibility ein Konzept für ein Design System erstellt werden. Mit Hilfe dieses Konzepts wird dann ein Prototyp entwickelt, mit welchem man ein Formular für die Erfassung von Werbungskosten erstellen kann. Dieser Formular-Baukasten wird dann in Usability-Tests evaluiert. Dabei wird festgestellt, ob Personen ohne ein breites, technisches Vorwissen mit Hilfe dieses Prototyps ein simples Formular erstellen und die einzelnen Bestandteile ohne große Schwierigkeiten bearbeiten können.

2 - Grundlagen, Methoden und Tools

Im folgenden Kapitel sind alle Grundlagen, Methoden und Tools aufgelistet und näher beschrieben, die zum Konzipieren und Ausarbeiten der in dieser Arbeit entstandenen Gestaltungslösung beigetragen oder diese inspiriert haben. Des Weiteren sei erwähnt, dass diese Arbeit auf dem Grundprinzip von Usability und dessen Kriterien aufbaut.

2.1 Usability

Hier eine kurze Einleitung und Definition zum Thema Usability.

Benutzerfreundlichkeit, oder auch Usability genannt, bezeichnet, inwieweit ein System oder ein Produkt von bestimmten Benutzern verwendet werden kann, um bestimmte Ziele mit Effektivität, Effizienz und Zufriedenheit in einem bestimmten Nutzungskontext zu erreichen. [4]

Nun zur genaueren Erläuterung der Merkmale:

- **Effektivität:** Die Genauigkeit und Vollständigkeit, mit der Benutzer bestimmte Ziele erreichen.
- **Effizienz:** Der Aufwand und die Ressourcen, welche zur Erreichung dieser Ziele aufgewendet werden.
- **Zufriedenheit:** Inwieweit die physischen, kognitiven und emotionalen Reaktionen des Benutzers, die sich aus der Verwendung eines Systems oder Produkts ergeben, den Bedürfnissen und Erwartungen des Nutzers entsprechen.

In dieser Arbeit wurde außerdem ein weiteres Merkmal betrachtet, das der

- **Barrierefreiheit:** Inwieweit Produkte und Systeme von Personen aus einer Bevölkerung mit unterschiedlichen Benutzeranforderungen, -merkmalen und -fähigkeiten genutzt werden können, um identifizierte Ziele in identifizierten Nutzungskontexten zu erreichen.

Die Definition der Usability und der einzelnen Merkmale stammen aus der ISO 9241-11 [4] und wurden für die Erschaffung einer Basis für diese Arbeit genutzt.

2.2 Human Centered Design Prozess

Der Aufbau und die Struktur dieser Arbeit bauen auf dem Prozess des Human Centered Design auf. Dies ist ein Ansatz zur Entwicklung interaktiver Systeme, der darauf abzielt, Systeme nutzbar und nützlich zu machen. Erreicht werden kann dies, indem sich mit diesem Ansatz auf die Benutzer, ihre Bedürfnisse und Anforderungen konzentriert sowie menschliche Faktoren, Ergonomie, Kenntnisse und Techniken zur Benutzerfreundlichkeit anwendet werden. ([5], Introduction)

Den Prozess des Human Centered Design kann man in vier Hauptschritte unterteilen:

1. Das Verstehen und Spezifizieren des Nutzungskontextes
2. Angabe der Nutzeranforderungen
3. Die Erstellung einer Lösung
4. Die Evaluierung der Lösung

Diese Human Centered Design Activities entsprechen den allgemeinen Phasen des Designs und der Entwicklung, von den Anforderungen über das Design hin, bis zur Verifizierung und Validierung. Dieser Prozess kann dazu beitragen, ein tieferes Verständnis für die Anforderungen der Benutzer zu erhalten und erste Rückmeldungen zu den Designkonzepten zu geben. ([5], Kap. 6 Human-centered design activities, S.10)

Dieses Konzept wurde nun auf den Aufbau dieser Arbeit übertragen, das heißt es wurde zunächst recherchiert, welche Methoden und Tools für dieses Projekt wertvoll sind. Diese wurden in den folgenden Kapiteln näher beschrieben. Danach folgte die Betrachtung der Nutzer und deren Bedürfnissen. Mit diesen Grundlagen wurde dann eine Lösung entworfen und prototypisch umgesetzt. Im letzten Schritt wurde diese Lösung mit Hilfe von Usability-Tests evaluiert.

2.3 Best Practices für benutzerfreundliches und effizientes UX- und UI-Design von Formularen

Um das Konzept und die daraus folgenden Prototypen so benutzerfreundlich wie möglich zu gestalten, wurde auch zu den aktuellen Standards und Best Practices im UX- und UI-Design für Formulare recherchiert.

In ihrem Buch “Designing UX: Forms” erwähnt Jessica Enders, dass es bei dem Design von Formularen besonders wichtig ist, minimalistisch zu arbeiten. Am besten ist es, wenn man mit nichts beginnt und dann nur das hinzufügt, was für die Kommunikation mit dem Nutzer erforderlich ist. Jeder einzelne Pixel, der hinzugefügt wird, sollte erforderlich sein, um die Benutzererfahrung zu steuern. ([6], Form Projects)

Wie ist jedoch ein benutzerfreundliches Formular aufgebaut? Dazu folgen nun ein paar Beispiele.

Zum einen ist es von Vorteil, zusammengehörende Informationen in logischen Blöcken zu gruppieren. Das Beantworten von einer Reihe von Fragen, die zum gleichen Thema gehören, ähnelt eher dem Redefluss eines Gesprächs. Die Benutzer können dadurch auch die benötigten Informationen besser und vollständig nachvollziehen, welche sie eingeben müssen. Dass diese Art des Aufbaus auch einen besseren Überblick verschafft, kann man gut am folgenden Kontaktformular (Abb. 2.1) erkennen. [7]

The image displays two side-by-side screenshots of user registration forms, demonstrating different approaches to field grouping.

Left Form (Single Column):

- Fields: First Name, Last Name, Email (with note: "Your email address will be your username"), Re-type Email, Password (with note: "Min. 8 characters, 1 number, case-sensitive"), Re-type Password, Address, City, State (dropdown), Zip Code (with note: "Optional"), Phone (with note: "No spaces or dashes"), Date of Birth (dropdown), Gender (dropdown), Security Question (dropdown), and Security Answer (with note: "Not case-sensitive").

Right Form (Grouped Sections):

- Personal Information:** First Name, Last Name, Date of Birth (dropdown), and Gender (dropdown).
- Account Information:** Email (with note: "Your email address will be your username"), Re-type Email, Password (with note: "Min. 8 characters, 1 number, case-sensitive"), Re-type Password, Security Question (dropdown), and Security Answer (with note: "Not case-sensitive").
- Contact Information:** Address, City, State (dropdown), Zip Code (with note: "Optional"), and Phone (with note: "No spaces or dashes").

Abbildung 2.1: Gruppieren von Formularfeldern beim Entwerfen von Formularen (Quelle: [7])

Auch das Layout spielt bei Formularen eine große Rolle und sollte aufgrund des besseren Verständnisses immer so simpel wie möglich gestaltet sein. Somit sollten mehrspaltige Layouts vermieden und es sollte möglichst immer nur eine Spalte verwendet werden. Wie das den Lesefluss des Users beeinflusst erkennt man gut an folgender Illustration (Abb. 2.2). [7]

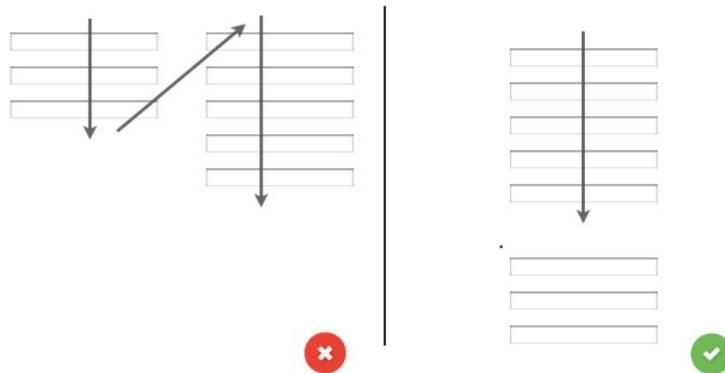


Abbildung 2.2: Standard-zweispaltiges Layout im Vergleich zu einem einspaltigen Layout
(Quelle: [7])

Auch die Platzierung des Labels kann eine große Rolle dabei spielen, wie schnell ein Formular ausgefüllt wird. Befunde aus dem Artikel "Label Placement in Forms", aus dem Jahr 2006 von Matteo Penzo, veranschaulichen dies signifikant. Penzo hat mit Hilfe verschiedener Tests untersucht wie sich die verschiedenen Positionen von Labels auf die User auswirken. Dabei kam heraus, dass das Platzieren eines Labels über dem Eingabefeld dem Nutzer helfen kann, diese beiden als eine Einheit zu erkennen, was die Geschwindigkeit beim Scannen des Formulars deutlich verschnellert. Es sollte jedoch darauf geachtet werden, die Beschriftung für das nächste Eingabefeld visuell vom vorherigen zu trennen. Wie sich die verschiedenen Platzierungen von Labels auf die Augenbewegungen auswirken kann man auch gut auf den folgenden Bildern (Abb. 2.3), welche aus den Tests von Matteo Penzo stammen, erkennen. Die darauf abgebildeten blauen Kreise zeigen, je nach Größe, wo die Augen der Tester hängen blieben bzw. dort länger verweilt sind. [8]

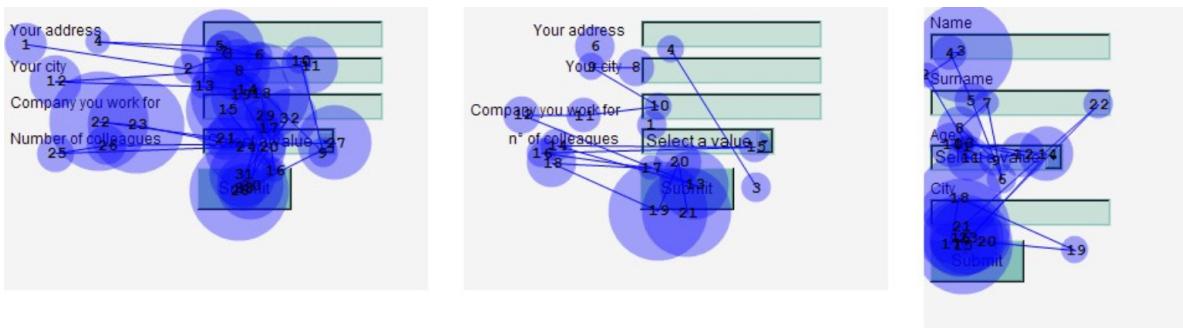


Abbildung 2.3: Vergleich von links ausgerichteten, rechts ausgerichteten und oben ausgerichteten Labels (Quelle: [7],[8])

Ein möglichst simples und unkompliziertes Vorgehen wird auch von vielen weiteren UX/UI-Designern geraten, weil Benutzer mit dem Ausfüllen eines Formulars zögern können. Daher sollte dieser Vorgang so einfach wie möglich gestaltet werden. Kleinere Änderungen, wie zum Beispiel gruppenbezogene Felder und die Angabe, welche Informationen in welche Felder gehören, können die Benutzerfreundlichkeit der Formulare erheblich verbessern. Ein verbessertes Formular führt dann wiederum zu kürzeren Ausfüllzeiten, weniger Formularübermittlungsversuchen und weniger Augenbewegungen. [7],[9]

2.4 Atomic Design

Um ein Design System zu entwickeln ist es von Vorteil, die verschiedenen Vorgehensweisen im Bereich UX- und UI-Design näher zu betrachten. Eine weit verbreitete Vorgehensweise ist die des Atomic Design. Dabei steht besonders die Modularität der einzelnen UI-Elemente im Vordergrund.

Atomic Design ist eine Methodik, die aus fünf verschiedenen, zusammenarbeitenden Phasen besteht. Diese Methodik wird angewendet, um Interface-Design-Systeme bewusster und hierarchischer zu erstellen. Die fünf Stufen des Atomic Design werden bezeichnet als: Atome, Moleküle, Organismen, Templates und Seiten. [10]

Atome sind in diesem Fall grundlegende HTML-Tags wie beispielsweise Labels, Inputs, Buttons und Weitere, die nicht weiter zerlegt werden können, ohne dass sie dadurch funktionsunfähig werden. [10]

Moleküle sind relativ simple Gruppen von Atomen bzw. UI-Elementen, die als Einheit zusammenarbeiten. Beispielsweise können ein Label, ein Search-Input

und ein Button zusammengefügt werden, um ein Suchformular-Molekül zu bilden. [10]

Organismen sind relativ komplexe UI-Komponenten, die aus Gruppen von Molekülen und / oder Atomen und / oder anderen Organismen bestehen. Diese Organismen bilden unterschiedliche Abschnitte eines Interfaces. So kann zum Beispiel aus dem Suchformular-Molekül und weiteren Elementen ein Header-Organismus entstehen. [10]

Templates sind Objekte auf Seitenebene, die Komponenten in ein Layout einfügen und die zugrunde liegende Inhaltsstruktur des Designs artikulieren. Um auf unserem vorherigen Beispiel aufzubauen, können wir den Header-Organismus auf eine Homepage-Template anwenden. [10]

Seiten sind bestimmte Instanzen von Templates, welche zeigen, wie eine Benutzeroberfläche mit repräsentativem Inhalt aussieht. Wenn man also das Beispiel des Homepage-Template verwendet und repräsentativen Text, Bilder und Medien in die Vorlage einfügt, so erhält man eine Seite und damit ein fertiges Gerüst eines Interface Designs. [10]

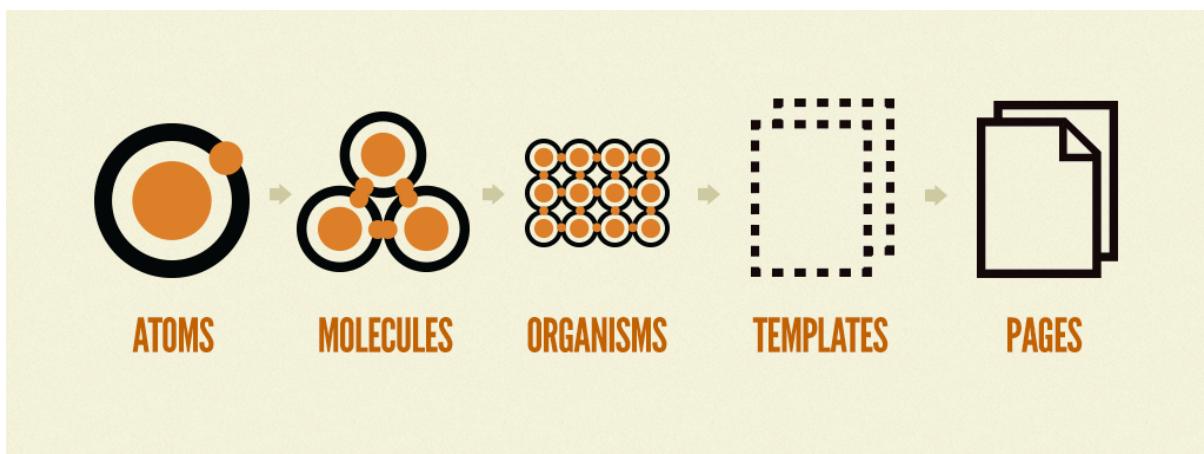


Abbildung 2.4: Aufbau des Atomic Design Prozesses (Quelle: [10])

Die Methodik des Atomic Design hat besonders für den Aufbau eines modularen Baukasten-Systems viele Vorteile und vereinfacht den Prozess der Erstellung. Zum einen wird das Interface in seine kleinsten Bestandteile zerlegt, welche somit als Bausteine für das System verwendet werden können.

Des Weiteren bietet die Vorgehensweise des Atomic Design eine hohe Konsistenz und Skalierbarkeit. Die einzelnen Elemente können sehr einfach wieder verwendet oder angepasst werden.

2.5 Modulare Systeme

Um ein Konzept für ein modulares Design System zu entwickeln, wurde natürlich auch über bereits existierende modulare Systeme recherchiert. Es gibt viele Online-Baukasten-Systeme mit deren Hilfe man Formulare oder Ähnliches erstellen kann. Jedoch sind diese meist inhaltlich sehr simpel und es fehlt das in dieser Arbeit angestrebte Regelwerk, welches einem ermöglichen soll, die modularen Elemente selbst anpassen und auch eine Intelligenz in das Formular mit einbinden zu können, welche dem User irrelevante Fragen erspart.

Jedoch haben einige große Software-Projekte und -Firmen das Potential eines modularen Systems bereits erkannt und nutzen dieses schon sehr erfolgreich. Zum einen wäre da das ResearchKit von Apple zu nennen und zum anderen das UI-Tool von Storybook. Diese beiden modularen Systeme dienten als große Inspiration für diese Arbeit und werden deshalb im Folgenden noch näher beschrieben. Für die Entwicklung des Prototypen wurde Angular verwendet. Dieses Frontend-Framework wurde unter anderem wegen seiner modularen, komponentenbasierten Bauweise ausgewählt.

2.5.1 Angular

Angular ist ein Framework zum Erstellen von Frontend-Anwendungen. Es ist ein Open Source-Projekt, welches von Google betrieben wird. Mitwirkende am Framework kommen aus der ganzen Welt. ([11], Kap. 1: Introduction)

Die Entwicklungsplattform von Angular basiert auf TypeScript. Als Plattform umfasst Angular:

- Ein komponentenbasiertes Framework zum Erstellen skalierbarer Webanwendungen
- Eine Sammlung gut integrierter Bibliotheken, die eine Vielzahl von Funktionen abdecken; darunter Routing, Formularverwaltung, Client-Server-Kommunikation und mehr
- Eine Reihe von Entwicklertools, mit denen man Code entwickeln, erstellen, testen und aktualisieren kann

Angular kann sowohl für kleine Einzelentwicklerprojekte als auch für Anwendungen auf Unternehmensebene genutzt werden. [12]

Die durch Angular gebotene, komponentenbasierte Struktur ist ideal geeignet für die Modularität des Projekts, sowie für das Entwickeln und das Wieder-verwenden der einzelnen UI-Elemente. Auch die Funktionen zum Entwickeln von Formularen, die Angular bietet, bilden einen ausgezeichneten Grundstein für die Implementierung des Prototypen.

2.5.2 Apple Research Kit

ResearchKit ist ein von Apple eingeführtes Open-Source-Framework, mit dem Forscher und Entwickler leistungsstarke Apps für die medizinische Forschung erstellen können. Man kann damit auf einfache Weise visuelle Flows für die Zustimmung der Patienten, dynamische Active Tasks in Echtzeit und Umfragen mithilfe einer Vielzahl anpassbarer Module, auf denen man aufbauen und diese mit der Community teilen kann, erstellen. [13]

Besonders das Aufbauen von Umfragen mit Hilfe von anpassbaren Modulen ist im Zusammenhang mit dieser Arbeit interessant.

Entwickler können eine Vielzahl verschiedener Antwortformate in ihre Umfragen einbauen. Von normalen Ja/Nein-Fragen bis hin zu Fragen, bei denen man Bilder als Antwort auswählen kann. Man kann beispielsweise auch einen Standort abfragen, um zu erfahren, wo der Benutzer sich gerade befindet. Auch der Ablauf der Umfrage kann individuell gestaltet werden, je nachdem was der Benutzer antwortet. So kann zum Beispiel bei einer Ja-Antwort noch eine weitere Frage folgen, anders als dies bei einer Nein-Antwort der Fall wäre. Dieser non-lineare Workflow spielt besonders bei komplexeren Formularen eine immer größere Rolle und war deshalb für die Konzeption des Systems ein wichtiger Punkt.

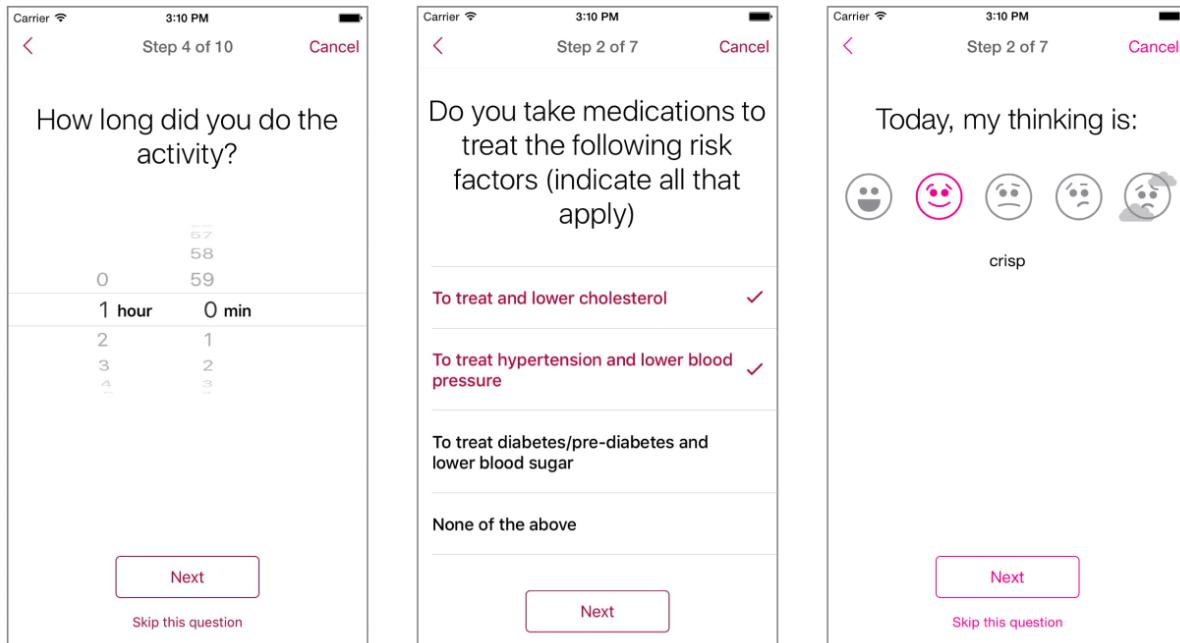


Abbildung 2.5: Beispiele für anpassbare Screens, die in Umfragen verwendet werden können (Quelle: [14])

2.5.3 Storybook

Storybook ist ein Open-Source-Tool zum isolierten Entwickeln von UI-Komponenten für React, Vue, Angular und mehr. Dadurch wird das Erstellen von Benutzeroberflächen organisierter und effizienter. [15]

Das Erschaffen von voneinander unabhängigen Komponenten, deren Eigenschaften durch ein simples Interface bearbeitet werden können, war für den Prototypen ein wichtiger Grundstein. Diese Eigenschaften sind beispielsweise Aussehen, Design und Ähnliches, welche mit ein paar Klicks ganz problemlos individuell angepasst werden können.

Wie man an Abbildung 2.6 erkennt ist Storybook aufgebaut wie ein Design-Tool. Man kann dem Element verschiedene Zustände zuweisen. Beispielhaft weist das Element in der Abbildung eine primäre Form und einen sekundären Form des Buttons auf. Außerdem gibt es auch zwei verschiedene Varianten von Größen: "large" und "small".

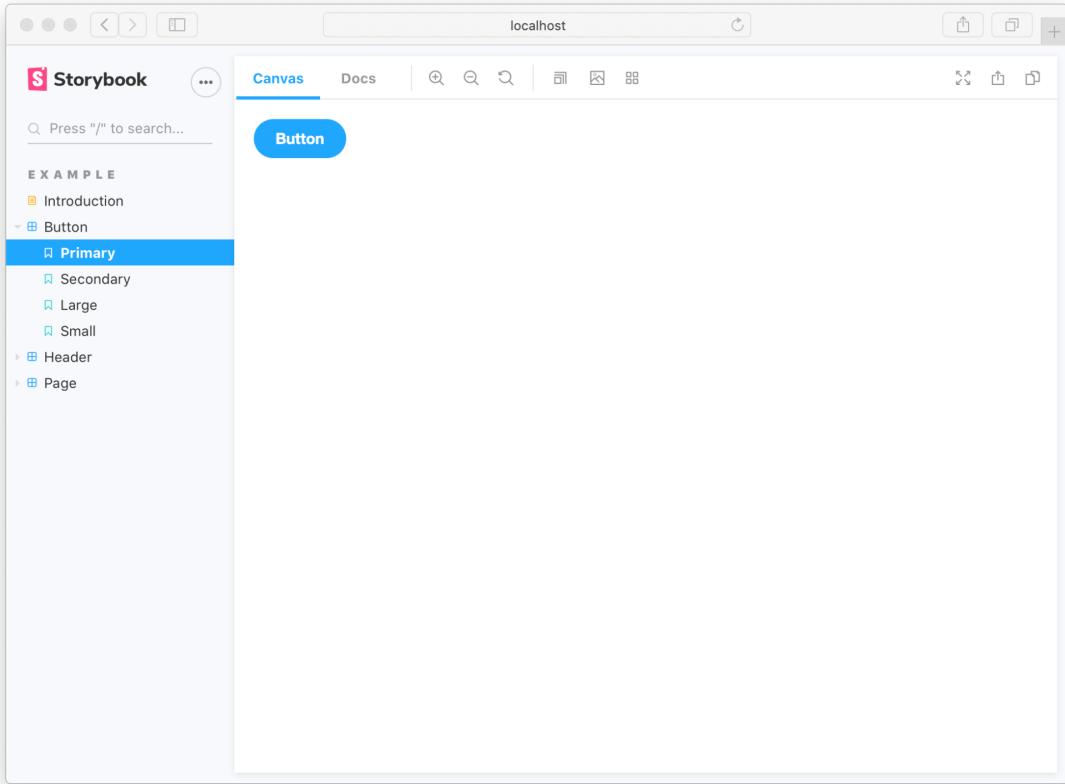


Abbildung 2.6: Ein Button, erstellt in der Storybook-Anwendung (Quelle: [16])

2.6 Design Systems

Um ein Design System zu entwickeln ist es von Vorteil, bereits existierende Design Systems und deren Funktionsweise näher zu betrachten. Ebenso ist es wichtig zu erfassen, welche Vorteile das Nutzen eines solchen Systems mit sich bringt.

Das Kernelement eines Design Systems ist eine UI-Komponentenbibliothek. Diese ist wichtig für die Designer. Des Weiteren enthält ein Design System den gebrauchsfertigen Code der verschiedenen UI-Elemente, welcher für die Entwickler von hoher Wichtigkeit ist. Außerdem ist in einem Design System noch eine Dokumentation der Prinzipien, sozusagen ein Regelwerk inbegriffen. [17]

Bezüglich den Vorteilen eines Design Systems lässt sich festhalten, dass sich die Konsistenz des User Interfaces und auch der einzelnen Elemente verbessert. Da die verschiedenen Komponenten einfach wiederverwendet werden können, ist das fertige Design einheitlich und ordentlich. Dies führt zum nächsten Vorteil, denn ein konsistentes Design erzeugt eine höhere Qualität. Mit Hilfe der

Dokumentation des Design Systems kann das implementierte Interface problemlos überwacht werden und Unregelmäßigkeiten treten seltener auf. Auch die Schnelligkeit des Entwurfsprozesses wird durch ein solches System verbessert. Neue Elemente können ganz einfach in das System eingepflegt und implementiert werden, oder auch von bereits existierenden Elementen abgeleitet werden. Daraus lässt sich ein weiterer Vorteil erschließen. Mit der Verwendung eines Design Systems ist das Aussehen der einzelnen UI-Elemente klar definiert, da diese visuellen Elemente zuvor im System beschrieben wurden. Die Zeit, welche normal für das Gestalten der Elemente verwendet wird, kann nun genutzt werden, um sich mehr auf eine höhere Benutzerfreundlichkeit und eine angenehmere Erfahrung für den User zu konzentrieren. [17]

Im Folgenden wird ein bestimmtes Design System näher betrachtet, da dieses für die weitere Arbeit einen wichtigen Grundstein bildet. Im Anschluss wird noch auf weitere Tools eingegangen, welche dabei geholfen haben, ein eigenes Design System aufzusetzen.

2.6.1 Material Design von Google

Um das Wissen in diesem Bereich zu vertiefen, wurde über verschiedene bereits erfolgreiche Design Systems recherchiert. Eines der wohl bekanntesten Design Systems, welches seit 2015 von Google verwendet wird, wird im Folgenden noch näher erläutert.

Material ist ein von Google entwickeltes Design-System, mit dem man hochwertige digitale Erlebnisse für Android, iOS, Flutter und das Web erstellen kann. Material Design ist inspiriert von der physischen Welt und ihren Texturen, einschließlich der Art und Weise, wie diese Licht reflektieren und Schatten werfen. Material-Komponenten sind interaktive Bausteine zum Erstellen einer Benutzeroberfläche. Mit Material-Theming kann man Material Design mühelos an das gewünschte Erscheinungsbild für Brand- und Corporate-Design anpassen. Dies ist durch die integrierte Unterstützung und Anleitung zum Anpassen von Farben, Typografiestilen und Eckformen unkompliziert machbar. [18]

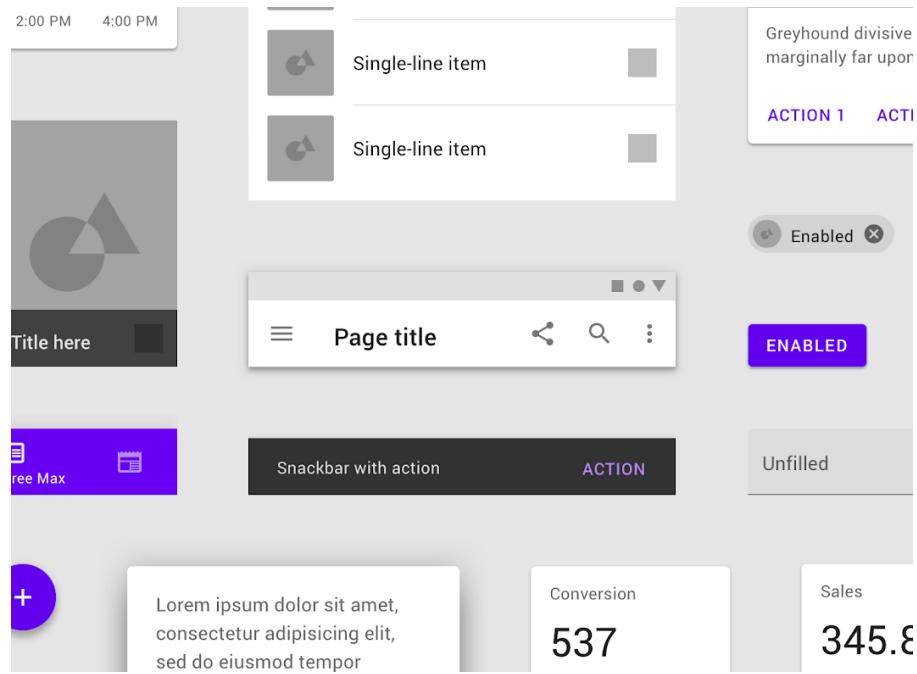


Abbildung 2.7: Beispiele für Komponenten, die Material Design zum Entwerfen und Entwickeln von Interfaces bietet (Quelle: [18])

Dieses Design System wurde jedoch nicht nur aufgrund seiner Bekanntheit durch die Marke Google näher betrachtet. Im anschließenden Teil wird die Bedeutung von Material Design für die Entwicklung konkretisiert.

Für die Implementierung einiger Features des Prototypen wurde die auf Material Design basierende UI-Komponentenbibliothek Angular Material verwendet.

2.6.2 Angular Material

Angular Material basiert auf dem Material Design System von Google. Die Angular Material-Bibliothek bietet eine Liste von Komponenten, welche vorgefertigte Funktionen in Datenformularen, Navigation, Layouts und anderen Aspekten beinhaltet. ([1] Kap. 1: Introduction)

Angular Materials oder UI-Komponenten helfen einem dabei, Anwendungen strukturiert zu gestalten. Sie ziehen Nutzer an und erleichtern den Zugriff auf die in der Anwendung vorhandenen Elemente oder Komponenten. Sie helfen auch dabei, die Anwendung auf attraktive Weise mit einzigartigen Stilen und Formen zu gestalten. Diese Komponenten tragen dazu bei, Anwendungen konsistenter, schneller, vielseitiger und sogar reaktionsschneller zu gestalten. [19]

Besonders zwei Features, welche ohne Angular Material recht kompliziert und zeitintensiv in der Umsetzung wären, wurden für die beiden Prototypen verwendet. Für den Formular-Prototypen wurde die Stepper-Komponente verwendet, um ein Formular mit mehreren Schritten gestalten zu können. Für den Baukasten-Prototypen wurde das Drag-and-Drop-Modul verwendet. Damit wurde sichergestellt, dass die einzelnen Elemente unkompliziert frei verschiebbar sind.

2.6.3 Sketch & Illustrator

Für das Design des Prototypen wurde unter anderem Sketch verwendet.

Sketch ist eine Software für macOS. Es ist als digitales Design Tool bekannt und wird hauptsächlich für das UX- und UI-Design von Mobil-, Web- und Desktop-Applikationen verwendet. Es ist ein vektorbasiertes Tool, wodurch alle Grafiken leicht skalierbar sind. Sketch ist ein universelles Designwerkzeug für alle Arten von digitaler Designarbeit. Seine Hauptaufgabe besteht darin, beim Entwerfen von Benutzeroberflächen zu helfen. Zu den Verwendungszwecken von Sketch zählt jedoch nicht nur das UX- und UI-Design, sondern auch das Erstellen von UI-Bibliotheken für Design-Systeme. [20]

Besonders dieses Feature sprach für die Verwendung von Sketch, da damit das Erstellen des Design Systems, mit Hilfe der Funktion eine Symbol-, Farb- und Schrift-Bibliothek anlegen zu können, um ein vielfaches vereinfacht wurde.

Des Weiteren wurde für die Gestaltung der Icons für den Prototypen Adobe Illustrator verwendet.

Illustrator ist die branchenführende Vektorgrafiksoftware. Mit ihr kann man alles von Web- und Mobilegrafiken bis hin zu Logos, Symbolen und mehr erstellen. [21]

Um nun also simple Icons zu erstellen, welche später für Buttons und Ähnliches verwendet wurden, eignete sich dieses vektorbasierte Tool am besten. Auch die Einbindung der Icons in Sketch funktionierte reibungslos.

2.7 Methoden zur Implementierung der Barrierefreiheit

Die Barrierefreiheit von Software ist ein großer Teil der Software-Entwicklung, jedoch hat sie im großen Zeitdruck der meisten Projekte oft eine niedrige Priorität und wird daher häufig nur wenig oder gar nicht betrachtet. ([2], S.11)

Jedoch ist die Barrierefreiheit ein sehr wichtiger Punkt, denn fast jeder Mensch wird in seinem Leben mit der ein oder anderen Einschränkung konfrontiert, welche ihn oftmals auch beim Verwenden von verschiedener Software einschränkt.

In ihrem Buch "Mismatches" beschreibt Kat Holmes, wie alle Menschen mit zunehmendem Alter Fähigkeiten gewinnen und verlieren. Das sich deren Fähigkeiten verändern durch Krankheit und Verletzung. Letztendlich wird jeder eingeschränkte Nutzer von der Interaktion sämtlicher Software ausgeschlossen, welche sich nicht an die Einschränkungen des Benutzers anpasst. ([2], S. 29)

2.7.1 Persona-Spektrum von Microsoft

Für diese Arten von Einschränkungen hat Microsoft ein Persona-Spektrum entworfen, um dauerhafte, vorübergehende und situative Behinderungen abzubilden. Es ist ein schnelles Tool, welches Empathie fördern und zeigen soll, wie eine Lösung für ein breites Publikum skaliert werden kann. [22]

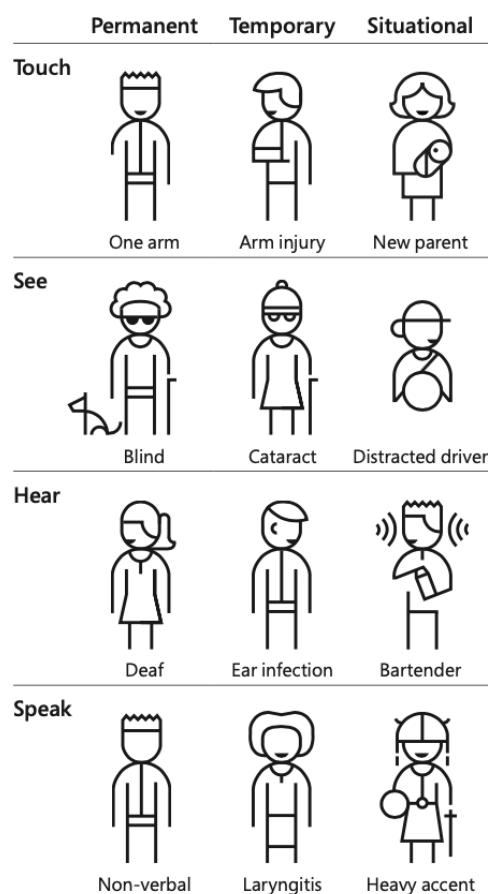


Abbildung 2.8: Microsoft Persona-Spektrum (Quelle: [[22]])

Wie man auf der Abbildung 2.8 erkennen kann sind nicht immer alle Einschränkungen permanent.

Manchmal ist es nur eine vorübergehende Behinderung von Körper oder Sinnen. Selbst eine kurzfristige Verletzung beeinflusst die Art und Weise, wie Menschen mit der Welt um sie herum interagieren, wenn auch nur für kurze Zeit. Zu solchen temporären Einschränkungen zählen beispielsweise auch grelles, blendendes Licht, ein Gips oder eine Kehlkopfentzündung, welche das Sprechen erschwert. Manchmal sind die Einschränkungen auch von der Situation abhängig. Wenn sich Menschen durch verschiedene Umgebungen bewegen, können sich auch ihre Fähigkeiten dramatisch ändern. In einer lauten Menge können sie nicht gut hören. In einem Auto sind sie visuell eingeschränkt. Neue Eltern verbringen einen Großteil ihres Tages damit, Aufgaben mit einer Hand zu erledigen. Ein erschöpfender Tag kann zu sensorischer Überlastung führen. Was möglich, sicher und angemessen für den Nutzer ist, ändert sich ständig. [22]

Diese Sicht auf Einschränkungen mit denen Menschen tagtäglich zu kämpfen haben, egal ob zeitlich begrenzt oder nicht, zeigt noch einmal, wie wichtig das Thema der Barrierefreiheit ist. Es zeigt auch, dass Barrierefreiheit besonders bei der Entwicklung von Software, die für alle Menschen verfügbar und nutzbar sein sollte, nicht zu kurz kommen darf.

2.7.2 Entwurf eines inklusiven Farbschemas mit Hilfe des Barrierefreiheits-Tool von Adobe Color

Wann immer man Design für Menschen entwickelt sollte man darauf achten, es für alle von ihnen, egal ob eingeschränkt oder nicht, zugänglich zu machen. Dies gilt auch für Farben. Farbschwäche, auch allgemein bekannt als Farbenblindheit, betrifft zwischen 3% und 8% der Weltbevölkerung. Diese wird normalerweise durch bestimmte Genetik verursacht, wobei Männern weitaus häufiger von Farbschwäche betroffen sind, als Frauen. In der kauasischen Bevölkerung zum Beispiel ist schätzungsweise einer von 12 Männern betroffen, aber nur eine von 200 Frauen. Dies bedeutet, dass Farben die man für ein Design wählt nicht immer auf die gleiche Weise gesehen werden. [23]

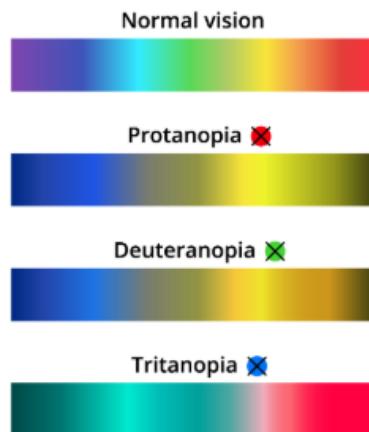


Abbildung 2.9: Die verschiedenen Formen von Farbschwäche (Quelle: [23])

Adobe hat diesbezüglich jedoch ein neues Feature zu ihrer Web-App Adobe Color hinzugefügt. Um nun also ein inklusives Farbthema für das Design des Prototypen zu entwickeln, wurde das Barrierefreiheits-Tool der Web-App genutzt.

Dieses Tool hilft einem dabei, für jeden Menschen zugängliche Farbschemata zu erstellen. Das bedeutet nicht, dass ein Betrachter mit Farbenblindheit die gleichen Farben wie ein normaler Betrachter sieht. Es wird jedoch sichergestellt, wenn jeder Benutzer fünf verschiedene Farben wahrnehmen muss, dass auch jemand mit einer Farbfehlsehigkeit diese fünf Farben als unterschiedlich wahrnimmt. Dies ist es, was man dann unter einem Farbthema, welches für Farbenblinde geeignet ist, versteht. Denn bei Menschen, die von Farbenblindheit betroffen sind, können bestimmte Kombinationen von Farbtönen und Schattierungen zu Verwirrung führen, wodurch manche Farben praktisch nicht mehr zu unterscheiden sind. [23]

Mit Hilfe dieser Barrierefreiheit-Tools kann man überprüfen, ob die gewählten Farben auch für Menschen mit einer eingeschränkten Farbwahrnehmung klar unterscheidbar sind.

Das Farbrad von Adobe Color zeigt Probleme für alle drei Arten von Farbenblindheit (Deutanopie, Protanopie und Tritanopie) an. Unterhalb des Rads wird eine simulierte Ansicht bereitgestellt um zu zeigen, wie das Farbthema für die Betroffenen der einzelnen Arten von Farbfehlsehigkeit angezeigt wird. Mit dem Farbrad werden außerdem problematische Farbkombinationen durch Konfliktlinien gekennzeichnet. Diese Hervorhebungen zeigen, welche Farbfelder

für jemanden mit Farbenblindheit möglicherweise nicht zu unterscheiden sind. Auf diese Weise können Farbpaletten mit fünf Farbfeldern erstellt werden, die sich für alle Menschen, egal ob mit oder ohne Farbfehlensichtigkeit, voneinander unterscheiden. Der Schweregrad des Mangels variiert von Person zu Person, das Farbrad ist jedoch so konzipiert, dass alle gleich angesprochen werden, denn die simulierten Ansichten zeigen die stärkste Schwere für eine Farbfehlensichtigkeit. Infolgedessen werden Konflikte als "potenzielle Konflikte" gekennzeichnet, um darauf hinzuweisen, dass bei Personen mit einer leichteren Form von Farbschwäche möglicherweise nicht dasselbe Problem auftritt. [23]

Dieses Tool ist ein großer Schritt in Richtung inklusives Design. Es ermöglicht UX/UI-Designern, deren Farbschemata auf mögliche Konflikte für Farbenblinde zu testen. Somit kann direkt beim Design-Prozess ausgeschlossen werden, dass Menschen mit derartigen Einschränkungen ausgesperrt werden.

Das für den Prototypen entwickelte Farbschema besteht, wie alle Farbschemata, die mit dem Adobe Color Tool entwickelt wurden, aus fünf Farben: einem hellen Blau als primäre Grundfarbe, einem dunkleren Blau als sekundäre Grundfarbe und drei verschiedenen Blau-Grau Schattierungen. Diese Farben wurden so abgestimmt, dass das Farbthema seitens des Tools als "Für Farbenblinde geeignetes Thema" eingestuft wird.

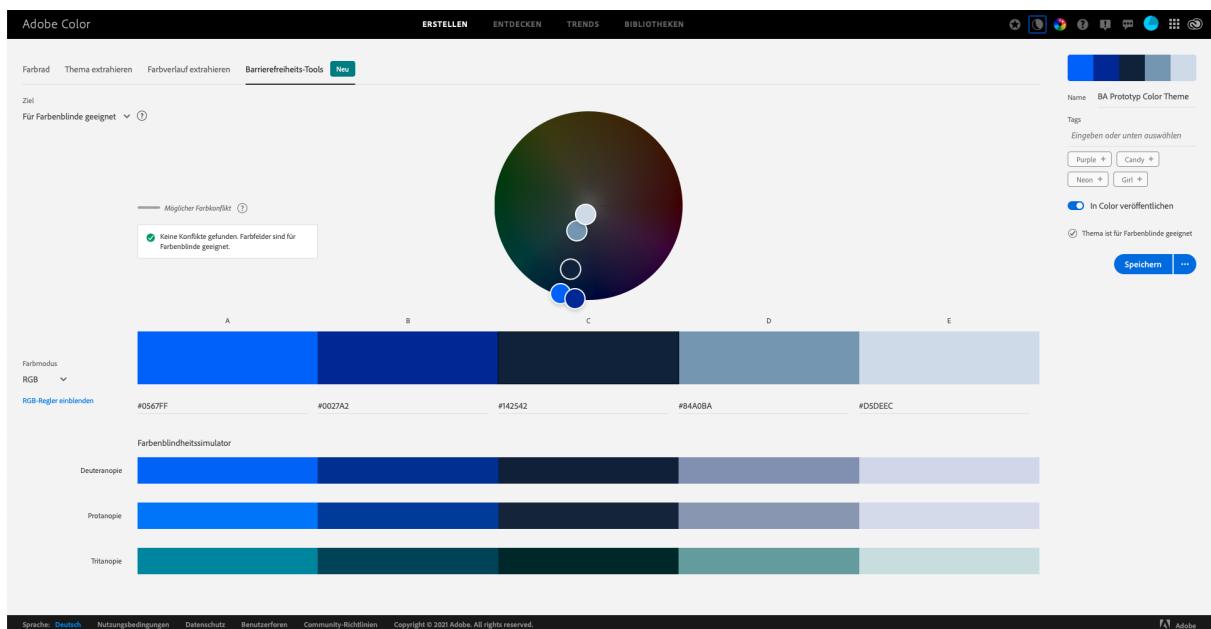


Abbildung 2.10: Entworfenes Farbschema (Ansicht im Adobe Color Tool (Quelle: [24]))

3 - Nutzungskontext und -anforderungen

Der Nutzungskontext und die Nutzungsanforderung sind hauptsächlich implizit in der Arbeit enthalten. Jedoch wurden sie durch die Personas und den Value Proposition Canvas noch konkretisiert und ausgeführt, um daraus eine Gestaltungslösung entwickeln zu können.

3.1 Personas

Der erste Schritt zur Festlegung von Nutzungskontext und -anforderungen, bestand darin, Personas für die Benutzer des Systems zu erstellen.

Personas sind fiktive Charaktere, die basierend auf den Recherchen erstellt werden, um die verschiedenen Benutzertypen darzustellen, welche das System nutzen könnten. Das Erstellen von Personas dient dazu, die Bedürfnisse, Erfahrungen, Verhaltensweisen und Ziele der Benutzer zu verstehen.

Es hilft außerdem dabei zu erkennen, dass unterschiedliche Personen unterschiedliche Bedürfnisse und Erwartungen haben und es kann dabei unterstützen, sich mit dem Benutzer zu identifizieren, für den man das System entwirft. Personas machen die vorliegende Entwurfsaufgabe weniger komplex, leiten Ideenfindungsprozesse und können dabei helfen eine gute Benutzererfahrung für die Zielbenutzergruppe zu schaffen. [25]

In diesem Fall wurden sogenannte Proto-Personas erstellt. Diese werden verwendet, wenn weder Geld noch Zeit vorhanden sind, um echte forschungsbasierte Personas zu erstellen. [26]

Sie basieren auf Annahmen und Vermutungen, welche auf den langjährigen Erfahrungen der adorsys im Bereich der Softwareentwicklung und der Zusammenarbeit mit verschiedensten Kunden aufbauen.

Um den Grundstein für die Personas zu legen wurden zunächst Namen, Jobs, Wohnort, Familienstand und grundlegende Charakterzüge und Eigenschaften entworfen.

Folgende Charakteristika sollten in die Personas mit einfließen, um eine möglichst variable Darstellung von verschiedenen Personen zu haben:

- **Der/Die Unerfahrene:** Macht seine/ihre Steuererklärung selbst, hat jedoch kaum Erfahrung und benötigt viel Zeit dafür.
- **Der/Die Analoge:** Macht seine Steuererklärung selbst und hat viel Erfahrung damit, nutzt jedoch keine digitalen Tools und hat eine Farbfehlensichtigkeit.
- **Der/Die Uninteressierte:** Hat noch nie eine Steuererklärung gemacht, dementsprechend auch keine Erfahrung und wenig Interesse am Thema Steuern.
- **Der/Die Digitale:** Nutzt eine moderne App für die Steuererklärung und hat damit keinerlei Probleme.

Die genauen Details der einzelnen Personas wurden in einem Workshop zusammen mit weiteren Personen entwickelt, um mit den daraus entstandenen Daten die Personas zu finalisieren. Dieser Workshop musste aufgrund der Corona-Pandemie und den damit verbundenen Lockdowns online stattfinden, was jedoch kein großes Problem darstellte. Für die gemeinsame Zusammenarbeit wurde die kollaborative Whiteboard Applikation Miro benutzt. Diese ermöglicht es dem Nutzer unter anderem wie an einem normalen Whiteboard, Post-Its zu platzieren und zu beschriften. Um nun also ein Miro Board für den Persona-Workshop zu gestalten, wurden zunächst die bereits vorher ausgewählten Eckdaten dargestellt. Im Anschluss wurden die Personas in die Punkte "Hobbies und Leidenschaften", "Berufserfahrung", "Emotionen", "Ziele", "Erfahrungen mit Steuererklärungen", "Erwartungen an Informations- austausch", "Soft Skills", "Technologien" und "Zitate" unterteilt. Im Workshop sollten dann die insgesamt fünf Teilnehmer in jeweils kurzen Zeitintervallen Post-Its zu den einzelnen Unterpunkten der verschiedenen Personas verfassen. Im Anschluss wurden dann die einzelnen Ideen bewertet, ob sie zu der jeweiligen Persona passen oder nicht. Diese Herangehensweise ermöglicht es, eine Persona zu entwerfen, welche in sich stimmig ist und keine Inkonsistenzen aufweist, da diese durch die Betrachtung und Ideen von mehreren Personen direkt erkannt und eliminiert werden können.

Die Daten aus dem Workshop wurden dann zusammengefasst und evaluiert, um daraus vier Personas zu erstellen. In Abbildung 3.1 sind zwei dieser vier Personas dargestellt. Alle Personas befinden sich in Anhang A1.



Abbildung 3.1: Persona von Anton Müller und Linda König

Diese Personas halfen dabei die Gestaltung des Prototypen genauer auf die verschiedenen Typen von Benutzern abstimmen zu können. Somit konnte besser entschieden werden, welche Fragen für das Formular relevant sind und welche nicht. Dadurch konnte auch ermittelt werden, welche Fragen für den Nutzer erforderlich bzw. welche optional sind und damit nicht allen Nutzern angezeigt werden müssen.

Es wurden nur Personas zu den Usern und nicht auch zu den Service-Providern erstellt, da diese nicht für den Aufbau des Formulars in Bezug auf das Themengebiet benötigt wurden. Außerdem wurden in den Usability-Tests Personen ausgewählt die zu den im folgenden Kapitel beschriebenen Bedürfnissen der Service-Provider passen. Deswegen waren nur diese vier Personas für die Arbeit wichtig, da diese später auch für eine der Usability-Testaufgaben benötigt wurden.

3.2 Value Proposition Canvas

Um den Nutzungskontext und die Nutzungsanforderungen des Systems weiter auszuarbeiten wurde ein Value Proposition Canvas erstellt.

Dieser beruht auf dem Konzept des "Value Proposition Design", welches im gleichnamigen Buch beschrieben wird. Dieses Konzept dient unter anderem dazu, ein tieferes Verständnis für die Wertschöpfung der Kunden zu entwickeln und zu verstehen, was Kunden, oder in diesem Falle User und Service Provider, wirklich wollen. Man kann dadurch ein genaues Gesamtbild darüber erstellen, worin die Werte des Produkts und die Bedürfnisse des Kunden an das Produkt liegen. Dadurch kann man die gewonnenen Informationen und Daten organisierter nutzen. ([27], Kap. 0: Value Proposition Design)

Der Value Proposition Canvas besteht aus zwei Seiten.

Die **Value (Proposition) Map** beschreibt die Merkmale einer bestimmten Value Proposition des Geschäftsmodells strukturierter und detaillierter. Es unterteilt die Value Proposition in Products & Services, Pain Relievers und Gain Creators.

- **Gain Creators** beschreiben, wie die Products & Services Kundengewinne erzielen.
- **Pain Relievers** beschreiben, wie die Products & Services die Pains der Kunden lindern.
- **Products & Services** ist eine Liste aller Produkte und Dienstleistungen, um die sich eine Value Proposition dreht.

Das **Customer (Segment) Profile** beschreibt ein bestimmtes Kundensegment im Geschäftsmodell strukturierter und detaillierter. Es zerlegt den Kunden in seine Customer Job(s), Pains und Gains.

- **Gains** beschreiben die Ergebnisse, die Kunden erzielen möchten oder die konkreten Vorteile, die sie suchen.
- **Pains** beschreiben schlechte Ergebnisse, Risiken und Hindernisse im Zusammenhang mit Customer Jobs
- **Customer Jobs** beschreiben, was Kunden versuchen, in ihrer Arbeit und in ihrem Leben zu erledigen, ausgedrückt in eigenen Worten.

Der Zustand **Fit** wird erreicht, wenn die Value Map dem Customer Profile entspricht. Mit anderen Worten ist der Zustand erreicht, wenn die Products & Services Pain Reliever und Gain Creators produzieren, die mehreren Customer Jobs, Pains und Gains entsprechen, die für den Kunden wichtig sind. ([27], Kap. 1: Canvas)

Value Proposition Canvas

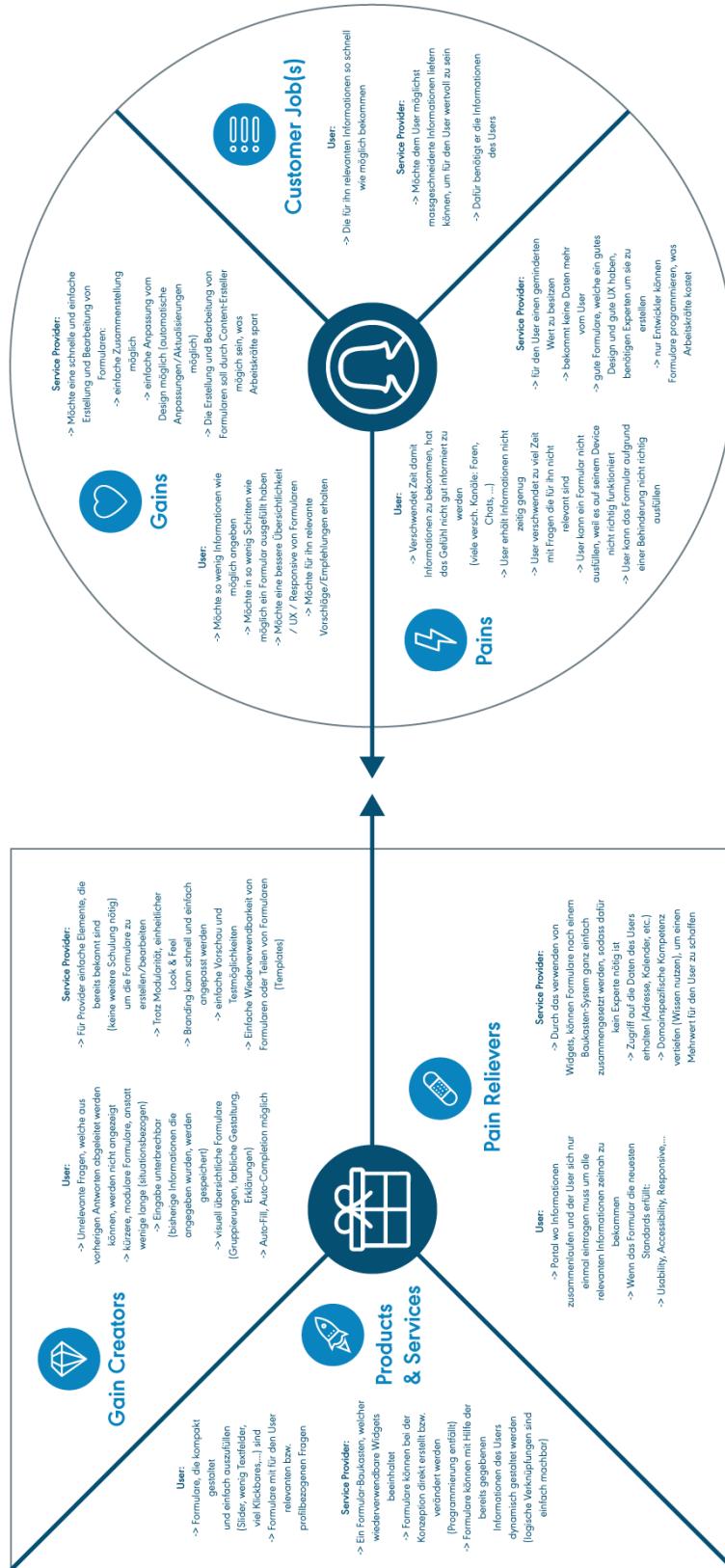


Abbildung 3.2: Value Proposition Canvas (eigene Darstellung in Anlehnung an ([27], Kap. 1.3: Fit))

Um ein Value Proposition Canvas zu erstellen wurde zunächst die Zielgruppe ermittelt. In diesem Fall gab es nicht nur eine Zielgruppe, sondern zwei: Die User und die Service Provider.

Die Erkenntnisse und Daten, welche aus dem Value Proposition Canvas gewonnen wurden basieren auf Annahmen, welche auf der langjährigen Erfahrungen in der Softwareentwicklung und der Arbeit mit den Kunden der adorsys aufbauen.

Der Use-Case, welcher betrachtet wurde bezog sich auf den Informationsaustausch zwischen User und Service Provider. Dieser erfolgt in den meisten Fällen in Form von Formularen.

Diese Art der Kommunikation spielt für beide Zielgruppen eine sehr bedeutende Rolle. Der User möchte für ihn relevante Informationen schnell und einfach erhalten und der Service Provider möchte diese massgeschneiderten Informationen unkompliziert liefern können, um somit an Wert für den User zu gewinnen. Doch dafür benötigt der Service Provider wiederum Informationen über den User, welche er erstmal erfragen muss.

Diese Art der Kommunikation bringt bei immer komplexeren Daten einige Pains für die beiden Zielgruppen mit sich.

Der User verschwendet viel Zeit damit, die Informationen in den verschiedenen Kanälen zu suchen, welche er haben möchte und hat am Ende doch das Gefühl, nicht optimal informiert zu werden. Dabei verwendet er auch viel Zeit damit, Fragen zu beantworten, welche für ihn gar nicht relevant sind. Auch das Endgerät oder eine Behinderung können dazu führen, dass der User ein Formular nicht richtig nutzen kann.

Pain Reliever für den User wären also eine Anlaufstelle, wo er seine Informationen einmal angibt und dann alle relevanten Informationen zeitnah erhält. Das Formular sollte außerdem den teilweise hohen und immer komplexer werdenden Standards in Sachen Usability, Barrierefreiheit und Responsivität entsprechen.

Der Service Provider hingegen hat die Befürchtung durch die Pains, die beim User auftreten, seinen Wert für ihn zu mindern und letztendlich keine Daten mehr vom User zu erhalten. Um jedoch die Probleme des Users zu verringern muss er kostspielige Experten und Designer engagieren, welche ein gutes Design und eine gute UX für die Formulare entwickeln. Da diese jedoch das Formular nur

gestalten und nicht umsetzen können, muss dieses dann noch von Entwicklern implementiert werden, was wiederum Arbeitskräfte und Geld kostet.

Ein Pain Reliever für den Service Provider wäre das Verwenden eines Baukasten-Systems. Dadurch könnten Formulare einfach zusammengesetzt werden, sodass keine Entwickler und Experten für die Erstellung mehr nötig sind. Weitere Pain Reliever wären Zugriff auf weitere Daten des Users, wie beispielsweise Kalender oder Adressdaten. Hierdurch könnten benötigte Informationen einfach berechnet werden und müssten nicht mehr vom User selbst erfragt werden. Das Wissen, welches bereits über den User existiert kann also dazu genutzt werden, um für diesen einen Mehrwert zu schaffen.

Ein Gain für den User ist bei einem Formular, so wenig Informationen in so wenigen Schritten wie möglich anzugeben. Es soll übersichtlich, einfach und kompakt gestaltet sein, um den Prozess des Ausfüllens so einfach wie möglich zu machen. Er möchte außerdem nur für ihn relevante Vorschläge und Empfehlungen erhalten.

Ein Gain Creator für den User ist es also, die Arbeit mit den Formularen für ihn so angenehm wie möglich zu machen. Irrelevante Fragen, welche aus den vorherigen Antworten abgeleitet werden können, sollten gar nicht erst angezeigt werden. Es sollten kürzere, modulare Formulare statt langen, ausführlichen verwendet werden. Die Eingabe sollte unterbrechbar sein und bereits angegebene Informationen gespeichert werden. Auch visuell sollten die Formulare ansprechend gestaltet und Autocompletion möglich sein, um die Benutzung so einfach und unkompliziert wie möglich zu gestalten.

Damit der Service Provider dem User genau so etwas bieten kann, sind seine Gains Formulare schnell und einfach erstellen zu können. Er möchte, dass Anpassungen am Design und Aktualisierungen einfach vorgenommen und eventuell auch automatisiert werden können. Die Erstellung von Formularen sollte so trivial sein, dass diese leicht zusammengestellt werden können und die Erstellung und Bearbeitung direkt durch die Content-Ersteller möglich wird, was wiederum Arbeitskräfte sparen würde.

Ein Gain Creator für den Service Provider ist also ein System, mit dem man schnell und unproblematisch Formulare aufbauen kann. Um ein System für die Erstellung von Formularen einfach und unkompliziert zu gestalten sollten Elemente verwendet werden, welche bereits bekannt sind, damit keine zusätzlichen Schulungen für die Mitarbeiter nötig werden. Trotz der Modularität soll jedoch ein einheitliches Look & Feel entstehen und auch das Branding sollte schnell und

mühelos angepasst werden können. Auch eine simple Vorschau und Testmöglichkeiten sind ein wichtiger Punkt. Zudem spielt die Wiederverwendbarkeit von Formularen oder Teilen davon eine sehr große Rolle, da hierdurch Mehrarbeit deutlich verringert werden kann.

Das Formular, welches den User zufrieden stellen würde ist also eines, das kompakt gestaltet und einfach auszufüllen ist. Das bedeutet, dass vor allem viel Klickbares und wenig Textfelder genutzt werden sollten. Außerdem sollte das Formular nur für den User relevante bzw. profilbezogene Fragen beinhalten.

Für den Service Provider zufriedenstellend ist ein Formular-Baukasten, welcher wiederverwendbare Widgets beinhaltet, mit denen man Formulare direkt bei der Konzeption erstellen bzw. bearbeiten kann. Dadurch entfällt die Programmierung jener Formulare. Des Weiteren sollen die damit gestalteten Formulare mit Hilfe der bereits gegebenen Daten des Users dynamisch generiert werden können. Das bedeutet, dass simple Verknüpfungen einfach umsetzbar sind und keine komplizierte Logik benötigt wird.

Mit Hilfe des Value Proposition Canvas wurden nun also die Probleme, Wünsche und Ansprüche der beiden Zielgruppen genauer erläutert und dokumentiert. Sie bilden damit den Grundstein für die weitere Konzeption.

3.3 Themenkomplex Steuern und Werbungskosten

Wie bei den Personas schon deutlich wurde, ist der Themenkomplex für den Prototypen Steuern. Da dies ein sehr großes Themengebiet ist, liegt der Fokus auf dem Bereich Werbungskosten. Dieses Thema ist besonders im Kontext von Formularen sehr interessant, da vom Benutzer viele Fragen beantwortet werden müssen und dies somit für die Umsetzung viele Möglichkeiten bot.

Um einen Ausgangspunkt für die Entwicklung eines Prototypen zu setzen ist es interessant, sich bereits existierende Steuersoftware anzusehen. Besonders ansprechend war in diesem Fall das 2016 gegründete Start-Up Taxfix und dessen gleichnamige App. Mit Hilfe einfacher Fragen in Form eines Formulars kann man damit ganz simpel und schnell seine Steuererklärung erledigen. [28]

Besonders das UI-Design und der Aufbau des Formulars sind sehr benutzerfreundlich gestaltet. Die Umsetzung der Formularelemente von Taxfix diente somit als ausgezeichnete Inspiration für die Umsetzung der verschiedenen Elemente für den Prototypen.

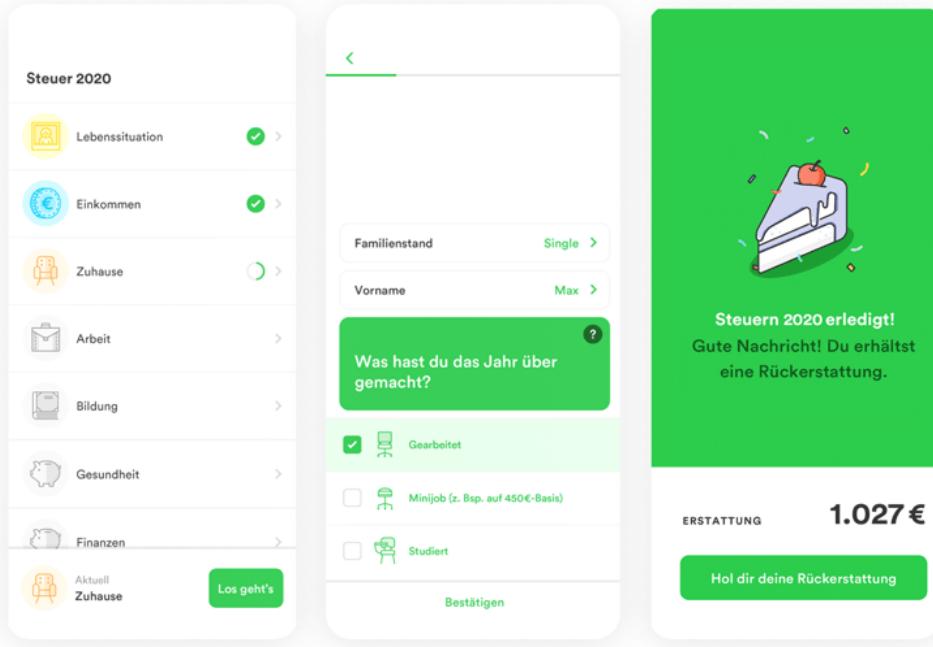


Abbildung 3.3: Screens aus der Taxfix-App (Quelle: [29])

4 - Gestaltungslösung

Die Gestaltungslösung umfasst zum einen die Konzeption der einzelnen UI-Elementen und der Prototypen. Zum anderen wird in diesem Kapitel auch die prototypische Umsetzung und Entwicklung der Prototypen beschrieben.

4.1 Konzeption und Design der Prototypen

Zunächst war nur ein Prototyp des Baukasten-Systems geplant. Da jedoch die Seite der User und der Ersteller in Betracht gezogen wurde, sind im Laufe des Konzeptionsprozesses zwei Prototypen entstanden. Beide nutzen jedoch dieselben UI-Elemente. Der erste Prototyp umfasst das Baukasten-System, während der zweite ein schon fertiges Formular darstellt.

4.1.1 Konzeption und Design der UI-Elemente

Ausgangspunkt des Designs der Prototypen war das Konzept des Atomic Design. Es wurden also zuerst alle Elemente, die später benötigt wurden in ihre kleinstmögliche Form zerlegt und erstellt. Diese Herangehensweise sorgte dafür, dass die Elemente so gestaltet wurden, dass man sie einfach wiederverwenden kann, was für die Modularität von hoher Bedeutung ist.

So entstanden mit Hilfe von Sketch und der in Adobe Color erstellten Farbpalette die verschiedenen Elemente. Als erste Instanz wurden normale Buttons, Textfelder und eine Dropdown-Liste entworfen und auch die verschiedenen Schriftarten und -stile, die später verwendet wurden. Um das Design noch intuitiver zu gestalten wurden die klassischen Radio- und Checkbox-Buttons durch Buttons mit Icons ersetzt. Außerdem wurde ein Toggle-Button für Ja/Nein-Fragen der Sammlung an UI-Elementen hinzugefügt. Als Nächstes folgten komplexere Elemente, wie eine Stepper-Anzeige für das später in Einzelschritte aufgeteilte Formular, sowie ein Logo, ein Header und ein Footer für die Website.

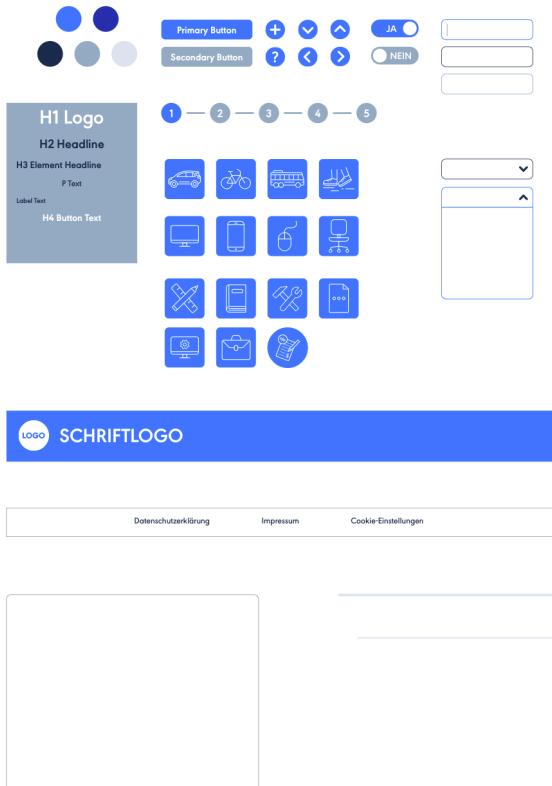


Abbildung 4.1: Die einzelnen Formular-Elemente dargestellt in Sketch

4.1.2 Konzeption des Formulars

Um den ersten Prototyp zu erstellen wurden die verschiedenen Ergebnisse aus den Recherchen zur Zielgruppe, zum Themengebiet und weiteren Recherchebereichen zusammengetragen, um ein simples Formular zu erstellen. Zunächst wurden Fragen ausgewählt, bei denen man möglichst viele verschiedene UI-Elemente zum Einsatz bringen konnte. Es wurden verschiedene Teilbereiche des Themas Werbungskosten ausgewählt, um das Formular so kompakt wie möglich zu gestalten und die spätere prototypische Implementierung zu vereinfachen.

Das Formular wurde in 5 Schritte unterteilt: "Allgemeines", "Fahrtkosten", "Home-Office", "Arbeitsmittel" und "Arbeitskleidung".

Auch in diesen Teilbereichen wurden beispielsweise Fragen, bei denen Geldbeträge oder Ähnliches angegeben werden soll, ausgelassen, da diese für den Prototypen und die späteren Tests irrelevant waren.

Nun wurde aus den zuvor erstellten, verschiedenen UI-Elementen das fertige Formular zusammengesetzt. Durch die Wiederverwendung der Elemente entstand automatisch ein einheitliches, klar strukturiertes und schlichtes Design.

4.1.2 Konzeption des Baukasten-Systems

Um den zweiten Prototypen zu erstellen wurde zunächst überlegt, wie man ein solches Baukasten-System am benutzerfreundlichsten gestalten kann. Hier wird bei bereits vorhandenen Lösungen oft ein Drag-and-Drop-System verwendet. Somit wurde der Prototyp zunächst in zwei Teile aufgeteilt. Eine Spalte für die verschiedenen UI-Elemente und eine andere, in die man die Elemente, welche man verwenden möchte per Drag-and-Drop verschieben kann, um dort das fertige Formular darzustellen. Um den Inhalt und auch das Aussehen der Elemente anpassen zu können, wurde eine weitere Spalte für einen Editor im Design hinzugefügt.

4.2 Prototypische Umsetzung

Um die Konzepte der Prototypen funktional umzusetzen bestand die Möglichkeit, das Prototyping-Tool von Sketch zu nutzen. Da dieses jedoch in seiner Funktionsweise die Umsetzung sehr eingeschränkt hätte, fiel die Entscheidung auf einen HTML-Prototypen, welcher mit Angular umgesetzt wurde.

Der Einsatz eines HTML-Prototypen bringt viele Vorteile mit sich, wovon einige im folgenden noch näher erläutert werden. Die Bausteine des Internets entwickeln sich ständig rasant weiter. Dies kann man jedoch bei einem HTML-Prototypen ausnutzen, um die neuesten Tools und Techniken in den Prototypen zu integrieren. Des Weiteren gibt es viele Ressourcen, die einem bei der Entwicklung helfen können, beispielsweise Tutorials, Tools und Frameworks. Mit einem HTML-Prototypen kann man außerdem komplexe Interaktionen mit Hilfe der Webtechnologien erstellen. Ein Beispiel für eine solche komplexe Interaktion repräsentiert das Speichern von Daten, die ein Benutzer in einen Prototypen eingibt und die später angezeigt werden. Ein weiteres Beispiel stellt, wie im Falle dieser Arbeit, das Ändern des Inhalts dar, den ein Benutzer aufgrund seiner Auswahl oder Eingabe sieht. ([30], Kap. 7: Building HTML-Prototypes, S.147)

4.2.1 Entwicklung der einzelnen UI-Elemente

Um mit der Entwicklung zu beginnen wurde zunächst ein Angular Projekt aufgesetzt. Es wurde hierfür die neueste Version von Angular verwendet, um auch alle Komponenten von Angular Material problemlos verwenden zu können. Des Weiteren wurde das Styling des Projekts mit SCSS anstatt CSS geschrieben.

HTML beispielsweise hat eine klare, verschachtelte und visuelle Hierarchie, CSS hingegen nicht. Mit Sass und SCSS kann man CSS-Selektoren so verschachteln, dass sie der gleichen visuellen Hierarchie ihres HTML-Codes folgen. Des Weiteren kann man mit SCSS Variablen definieren und abrufen. Diese Variablen bieten die Möglichkeit, Informationen zu speichern, welche man im Stylesheet wiederverwenden möchte. Diese sind beispielsweise Farben, Schriften oder andere CSS-Werte. Das ist eine sehr effektive Möglichkeit, um Markenfarben auf der gesamten Website konsistent zu halten und, wenn nötig, auch einfach austauschen zu können. [31]

Folglich wurden zunächst alle Farben, die für die beiden Prototypen verwendet wurden als Variablen angelegt. Im folgenden Code-Block sieht man diese Variablen, welche später in der gesamten Anwendung verwendet wurden.

```
$WHITE: #FFF;  
$BLACK: #000;  
  
$BLUE-PRIMARY: #0567FF;  
$BLUE-SECONDARY: #0027A2;  
  
$RED-ERROR: #DF400B;  
  
$GRAY-DARK: #142542;  
$GRAY-MEDIUM: #84A0BA;  
$GRAY-LIGHT: #D5DEEC;  
  
$BOX-SHADOW: rgba(0,0,0,.2);
```

Code-Block 4.1: Die Variablen, welche für die verschiedenen Farben verwendet wurden

Als Nächstes wurden dann alle UI-Elemente als einzelne Komponenten angelegt. Diese wurden so gestaltet, dass alle wichtigen Informationen und Stylings des Elements veränderbar sind.

Dieser Aufbau der Elemente wird nun am Beispiel eines einfachen Textfeldes noch genauer erläutert. Zunächst kann man den Text des Labels für das Textfeld anpassen. Ebenso kann man die Größe des Textfeldes anpassen. Hierfür wurden verschiedene SCSS-Klassen angelegt, zwischen denen man wählen kann. Hier hätte man auch noch weitere Klassen für beispielsweise verschiedene Farben oder Ähnliches anlegen können, um die Elemente an das Branding einer Firma anzupassen zu können. Dies war jedoch für den Prototyp nicht notwendig. Für die Entwicklung mussten auch der "formControlName" und die dazugehörige "formGroup" anpassbar sein um später eine Validation der einzelnen Element möglich zu machen. Auf diese oder ähnliche Weise wurden alle Elemente aufgebaut, um später in den beiden Prototypen verwendet werden zu können.

```
<div [formGroup]="textFieldParentForm" [ngClass)="textFieldClass">  
  <label  
    class="text-field_label"
```

```

    for="txt-field"
  >
    {{textFieldLabel}}
  </label>
  <input
    class="text-field__input"
    type="text"
    id="txt-field"
    [FormControlName]="textFieldControl"
    required
    [readonly]="textField Readonly"
  >
</div>

```

Code-Block 4.2: HTML-Code der Text-Feld-Komponente

Wie man im Code-Block 4.2 sehen kann wurden die Daten, welche später veränderbar sein sollten mit Hilfe des sogenannten “two-way data binding” mit Variablen verknüpft. Außerdem wurde mit Hilfe von “ngClass” auch die SCSS-Klasse des Elements mit einer Variable verknüpft um auch das Design später anpassen zu können.

```

export class TextFieldComponent implements OnInit {

  @Input() textFieldLabel:string;
  @Input() textFieldClass:string;
  @Input() textFieldControl:string;
  @Input() textFieldReadonly:boolean;
  @Input() textFieldParentForm:FormGroup;

  constructor() { }

  ngOnInit() {
    console.log(this.textFieldControl);
  }

}

```

Code-Block 4.3: Typescript-Code der Text-Feld-Komponente

Hier im Code-Block 4.3 sieht man nun die verschiedenen Variablen für die anpassbaren Parameter der Text-Feld-Komponente. Durch “@Input” wird sichergestellt, dass die Variablen auch in anderen Komponenten verwendet und dort einen Wert zugewiesen bekommen können. Somit konnten später im Formular-Prototypen beispielsweise diese Komponenten eingefügt werden und es mussten ausschließlich den verschiedenen Variablen Werte zugewiesen werden. Somit mussten keine weiteren Änderungen vorgenommen werden, um das Textfeld funktionsfähig fertig zu stellen.

```
.text-field {  
  display: flex;  
  flex-direction: column;  
  
  &--tiny {  
    width: 170px;  
  }  
  
  &--small {  
    width: 270px;  
  }  
  
  &--medium {  
    width: 370px;  
  }  
  
  &--large {  
    width: auto;  
  }  
  
  &__label {  
    margin: 0 0 0.3em 1.5em;  
  }  
  ...  
}
```

Code-Block 4.4: SCSS-Code der Text-Feld-Komponente

Im Code-Block 4.4 sieht man nun die verschiedenen SCSS-Klassen, welche verwendet wurden, um die Größe der Textfelder später anpassen zu können. Dabei wäre auch möglich gewesen, mit noch mehr Klassen, wie bereits oben erwähnt, nicht nur die Größe, sondern auch Farbe, Form und Ähnliches veränderbar machen zu können.

4.2.1 Entwicklung des Formular-Prototypen

Durch die bereits entwickelten Formular-Elemente gestaltete sich die Umsetzung des Formular-Prototypen relativ unproblematisch. Es wurden nach und nach die einzelnen Elemente eingesetzt und deren Werte so angepasst, dass sie zum vorher festgelegten Design passen. Wie im vorherigen Kapitel erklärt, konnte auf diesem Weg die Veränderbarkeit der Komponenten unkompliziert implementiert werden.

Lediglich die Implementierung eines Formulars mit einzelnen Schritten schien anfangs schwer umsetzbar. Diese konnte jedoch durch die Stepper-Komponente von Angular Material schnell gelöst werden. Somit konnte das Formular in mehrere Schritte unterteilt werden, was die Benutzung erleichtern sollte.

Des Weiteren wurde bei optionalen Stufen, wie "Home-Office", "Arbeitsmittel" und "Arbeitskleidung" mit Hilfe eines Toggle-Buttons und einer simplen Ja/Nein-Frage vorher abgeklärt, ob weitere Fragen zu diesem Schritt notwendig sind. Wenn die Antwort "Nein" war, dann wurden diese automatisch ausgeblendet, um irrelevante Fragen für den Nutzer zu vermeiden.

Außerdem musste die Strecke des Arbeitsweges nicht angegeben werden, da diese automatisch aus den Angaben der Adressen von Wohn- und Arbeitsort berechnet wurde. Dazu sollte auch noch ein Feature eingebaut werden, welches die Adressen mit Hilfe von Autocompletion und Google Maps automatisch in das Formular einträgt. Dieses Feature konnte jedoch aufgrund seiner Komplexität nicht fertig gestellt werden.

Einzelne Daten, wie beispielsweise weitere Angaben zu den Kosten von Fortbewegungsmitteln oder den Arbeitsmitteln wurden zudem schon automatisch angeben, da diese für den Prototypen und die Usability-Tests nicht relevant waren.

Das Design des Formulars ist an die verschiedenen Best Practices für ein benutzerfreundliches und effizientes Formular-Design aus Kapitel 2.3 angepasst. Es besteht aus einem einspaltigen Layout und mehreren Schritten, um die verschiedenen Fragen in verschiedene Bereiche gruppieren zu können. Auch innerhalb der verschiedenen Schritte wurden die einzelnen Fragen noch einmal zu verschiedenen Themen gruppiert. Des Weiteren sind die Labels für die verschiedenen Elemente immer über diesen angeordnet, um den Lesefluss noch weiter zu erleichtern. Es wurde versucht, das Design so simpel wie möglich zu halten, um den User nicht abzulenken.

SCHRIFTLOGO

1 — 2 — 3 — 4 — 5

FAHRTKOSTEN

Adresse des Arbeitsplatzes

Straße Hausnummer

PLZ Ort

Adressen weiterer Arbeitsstätten

+ Weitere Adresse hinzufügen

Fortbewegungsmittel

Auto Bus Fahrrad Zu Fuß

< Zurück Weiter >

[Datenschutzerklärung](#) [Impressum](#) [Cookie-Einstellungen](#)

Abbildung 4.2: Der Formular-Prototyp beim Schritt “Fahrtkosten”, dargestellt im Browser

4.2.1 Entwicklung des Baukasten-System-Prototypen

Die Umsetzung des Baukasten-Prototyps schien zunächst etwas anspruchsvoller, da hier viel Funktionalität mit einfließen sollte. Jedoch half ein Modul von Angular Material dabei, einen großen Teil der Funktionalität zu implementieren. Das Drag-and-Drop Modul von Angular Material unterstützt viele Features, um einfache Drag-and-Drop Interfaces zu erstellen.

Der Baukasten-Prototyp war in drei Spalten aufgeteilt. In der ersten Spalte, dem Element Editor, kann man die Daten und das Aussehen der einzelnen Elemente bearbeiten. Jedes Element hat seinen eigenen Bereich im Editor.

In der zweiten Spalte befinden sich die verschiedenen UI-Elemente. Dort wird auch in Echtzeit angezeigt, was man im Editor anpasst. Diese Elemente kann man dann mit Hilfe von Drag-and-Drop in die dritte Spalte verschieben. Die besagte Spalte zeigt einem dann, wie das fertige Formular aussehen wird.

Das Drag-and-Drop Feature ist so implementiert, dass die Elemente immer an eine vorgegebene Stelle springen und somit nicht frei verschiebbar sind. Die beiden Spalten sind sozusagen zwei Listen, zwischen denen die Elemente flexibel verschoben werden können. In der folgenden Abbildung 4.3 sieht man einen Teil des fertigen Baukasten-Systems. Die komplette Ansicht befindet sich in Anhang A.4.

The screenshot displays a user interface for a 'Baukasten-System' (Baukasten-System) in a browser. The interface is divided into three main columns:

- 1. Element Editor:** This column contains various input fields and settings for editing elements. It includes:
 - A section for Headline 1 with a text input field containing "Headline 1".
 - A section for Headline 2 with a text input field containing "Headline 2".
 - A section for Text-Feld 1 with a label "Label Text-Feld 1" and a dropdown menu showing "Klein" (radio button), "Mittel" (radio button, selected), and "Groß".
 - A section for Text-Feld 2 with a label "Label Text-Feld 2" and a dropdown menu showing "Klein" (radio button), "Mittel" (radio button, selected), and "Groß".
- 2. Elemente:** This column displays a list of UI elements that can be dragged and dropped. The elements shown are:
 - Headline 1
 - Headline 2
 - Label Text-Feld 1
 - Label Text-Feld 2
- 3. Formular:** This column shows a preview of the final form. It features a dashed rectangular area with the text "DROP ELEMENTS HERE". Above this area, there is descriptive text: "Hier sehen Sie die Vorschau, wie das fertige Formular aussehen wird. Verschieben Sie die Elemente nach oben oder unten um sie an die gewünschte Position zu setzen."

Abbildung 4.3: Ein Teil des Baukasten-Systems, dargestellt im Browser

5 - Evaluation mit Hilfe von Usability-Tests

Um sicherzustellen, dass das Konzept der beiden Prototypen auch wie geplant funktioniert und um sie auf ihre Benutzerfreundlichkeit zu testen wurden Usability-Tests geplant. Um vor den eigentlichen Tests noch kleinere Ungeheimtheiten auszubessern, fand zunächst ein Pilottest mit einem Teilnehmer statt. Der eigentliche Usability-Test wurde mit vier Testern durchgeführt.

5.1 Entwicklung eines Online-Testkonzepts

Aufgrund der Corona-Pandemie und den damit verbundenen Lockdowns musste ein Testkonzept entworfen werden, welches auch online funktioniert. Ursprünglich war geplant, die User-Tests in den Räumlichkeiten der adorsys durchzuführen. Da dies jedoch zu dem geplanten Zeitpunkt nicht möglich war, weil die Büroräume geschlossen waren und sich alle Mitarbeiter im Home-Office befanden, wurde entschlossen, die Tests online durchzuführen. Für diese Zwecke gibt es bereits einige Software-Lösungen, welche für das Remote-Testing entwickelt wurden. Da diese jedoch recht kostspielig und mit einer gewissen Einarbeitungszeit verbunden sind, wurde ein eigenes Testkonzept aufgesetzt. Hierfür wurde der Testablauf und die Testaufgaben mit Hilfe eines Miro-Boards dargestellt, da die Testaufgaben nicht wie üblich dem Tester auf Papier gegeben werden konnten. Wie dies letztendlich ausgesehen hat sieht man auf der Abbildung 5.1. Dort erkennt man auch, dass auf dem Miro-Board letztendlich auch die verschiedenen Links zu den Fragebögen und Prototypen angegeben waren. Der User wurde noch gebeten seinen Bildschirm zu teilen, damit gesehen werden konnte, was er gerade tut.

Wie bereits erwähnt konnten die beiden Prototypen nicht wie üblich auf dem Entwicklerrechner getestet werden, sondern mussten für die Tester online per Link verfügbar gemacht werden. Dies wurde mit Hilfe des Devops-Teams der adorsys umgesetzt. Die Prototypen wurden auf die Container-Plattform OpenShift geladen und konnten dann mit einem Link aufgerufen werden. Somit konnten die Tester problemlos auf die beiden Prototypen zugreifen und die Usability-Tests konnten trotz anfänglich bestehender Hindernisse stattfinden.

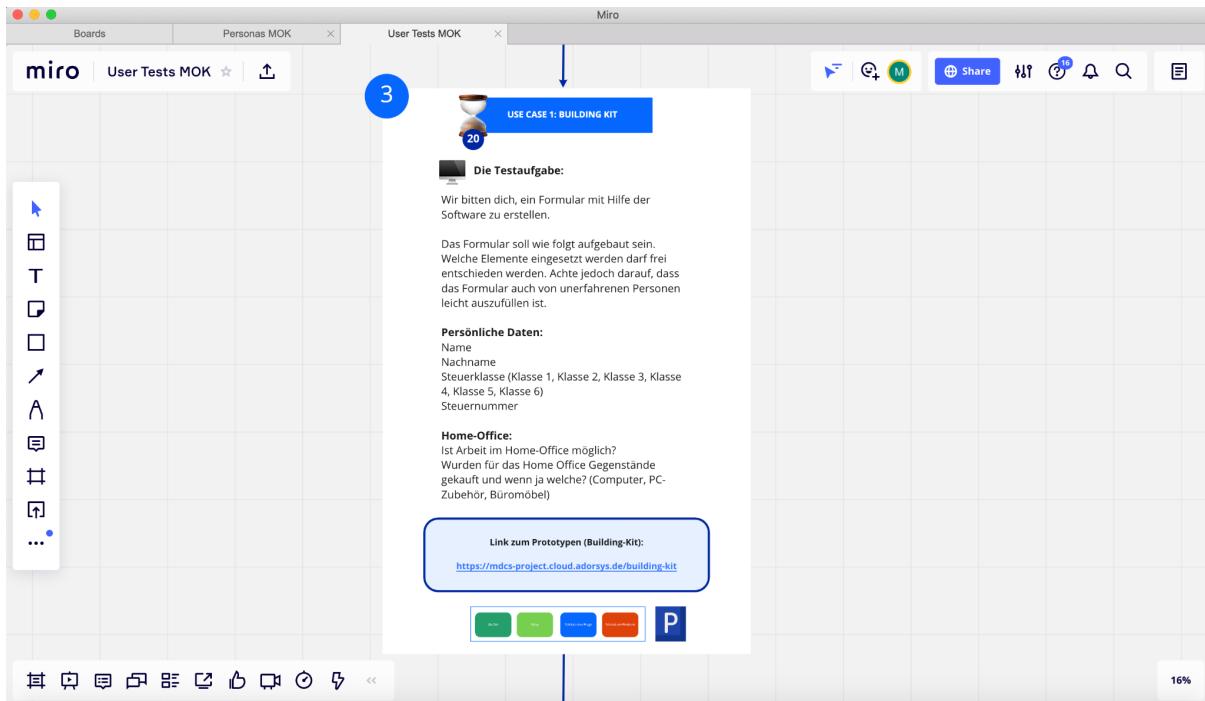


Abbildung 5.1 Erste Testaufgabe für das Building-Kit in Miro für das Online-Testkonzept

5.2 Testaufbau

Die Usability-Tests waren wie folgt aufgebaut: Im Vorfeld zum Test wurde den Teilnehmern ein Fragebogen zugeschickt. Dieser enthielt simple Standardabfragen zu ihrer Person und ihren Kenntnissen bezüglich Steuern und dem Erstellen von Umfragen bzw. Formularen (siehe Anhang A.6).

Der eigentliche Test begann mit einer kurzen Einführung und Einverständniserklärung für die Aufzeichnung. Diese Aufzeichnung war wichtig für die spätere Auswertung des Tests, da die Tester gebeten wurden, während den zwei Aufgaben jeweils ihre Gedanken laut auszusprechen.

Dabei handelt es sich um die Concurrent-Think-Aloud-Methode. Dies ist eine Forschungsmethode, bei welcher der Testteilnehmer eines Usability-Tests darum gebeten wird, während er mit der Testapplikation interagiert, seine Gedanken zu verbalisieren. Auf diese Weise können Moderator und Protokollant besser nachvollziehen, mit was sich der Nutzer gerade beschäftigt und was er fühlt. Damit wird unter anderem erfasst, ob er irritiert, überrascht oder vielleicht sogar frustriert ist. Dadurch können Usability-Probleme schnell identifiziert werden und die Methode ist zudem noch zeitsparend, da sie gleichzeitig mit der Testaufgabe stattfindet. [32]

Nun wurde mit der ersten Testaufgabe begonnen. Diese bestand daraus, ein simples Formular mit Hilfe des Baukasten-Systems zu erstellen. Die genaue Testaufgabe wird im Testskript im Anhang A.5 noch genauer erläutert. Dabei wurde es den Nutzern selbst überlassen, welche Elemente sie verwendeten. Jedoch sollten sie darauf achten, dass das zu erstellende Formular leicht ausfüllbar, also benutzerfreundlich ist. Die Daten für die Aufgaben wurden so angegeben, dass der Bedarf für zwei Überschriften ersichtlich wird. Unter der ersten Überschrift waren drei Textfelder und eine Dropdown-Liste notwendig, um die Aufgabe zu erfüllen. Der zweite Teil war etwas komplexer. Die User sollten darin den Toggle-Button verwenden, um die Ja/Nein-Frage einzubauen und im Anschluss die Icon-Buttons für die zweite Frage verwenden.

Daraufhin sollte ein System-Usability-Scale (SUS) Fragebogen ausgefüllt werden. Auf diesen wird im folgenden Kapiteln noch näher eingegangen.

Als Nächstes folgte die zweite Testaufgabe. Bei dieser sollte ein Formular mit Hilfe von vorgegebenen Daten ausgefüllt werden. Auch diese Testaufgabe ist im Testskript im Anhang A.5 in ausführlicher Form vorhanden. Hierfür wurde die Persona von Linda König als Grundlage für die Aufgabe hinzugezogen. Damit war der Teil "Home-Office" für den Nutzer irrelevant, aber die Teile "Arbeitsmittel" und "Arbeitskleidung" wurden in die Aufgabe mit aufgenommen. Somit konnte evaluiert werden, ob das System mit dem Weglassen von irrelevanten Fragen funktioniert.

Auch nach dieser Aufgabe sollte wieder ein SUS-Fragebogen ausgefüllt werden.

Für die beiden Testaufgaben wurde jeweils eine maximale Bearbeitungszeit festgelegt und es wurde die Zeit gestoppt, um zu beobachten, wie lange die Tester benötigen, um die Aufgaben zu lösen. Außerdem wurde bei den SUS-Fragebögen am Ende noch Platz für weitere Anmerkungen geschaffen, damit die User auch persönliches Feedback und Verbesserungsvorschläge abgeben konnten.

5.3 System Usability Scale

Der 1986 von John Brooke erstellte System Usability Scale bietet eine schnelle und effektive Möglichkeit, die Benutzerfreundlichkeit von Produkten und Designs zu bewerten. SUS ist ein praktisches und zuverlässiges Tool zur Messung der wahrgenommenen Benutzerfreundlichkeit. Im Gegensatz zu einem Usability-Bericht ist SUS keine Diagnose und wird verwendet, um eine allgemeine Messung der Usability-Bewertung gemäß ISO 9241-11 [4] bereitzustellen.

Nach jeder Usability-Testaufgabe wurde dem Benutzer ein SUS-Fragebogen zum Ausfüllen bereitgestellt. Die darin enthaltenen Fragen sollten schnelles und ungefiltertes Feedback vom Benutzer für jede Testaufgabe erfassen und ohne lästige Interaktion beantwortet werden. Einer der Hauptvorteile der Verwendung von SUS besteht darin, dass das Feedback zuverlässig und wiederholbar ist. [33]

Der SUS-Fragebogen ist aus jeweils 10 Fragen aufgebaut, die immer gleich bleiben. Diese werden dann mit Hilfe einer 5-Punkte-Likert-Skala von "Stimme ich überhaupt nicht zu" bis "Stimme ich voll und ganz zu" vom Benutzer beantwortet. Jeder Antwort wird ein Wert für die SUS-Score-Berechnung zugewiesen. Die Punkteverteilung für die Antworten lautet wie folgt:

- **Stimme ich überhaupt nicht zu** = 1 Punkt
- **Stimme ich nicht zu** = 2 Punkte
- **Neutral** = 3 Punkte
- **Stimme ich zu** = 4 Punkte
- **Stimme ich voll und ganz zu** = 5 Punkte

Nun kann man mit Hilfe dieser Punkte und folgenden Formeln den individuellen SUS-Score berechnen. Man addiert die Gesamtpunktzahl für alle ungeradzahligen Fragen und subtrahiert dann 5 von der Gesamtpunktzahl um (X) zu erhalten. Dann addiert man die Gesamtpunktzahl aller geradzahligen Fragen und subtrahiert diese Summe von 25 um (Y) zu erhalten. Am Ende addiert man die Gesamtpunktzahl der neuen Werte (X + Y) und multipliziert sie mit 2,5. Wenn man dies befolgt, so erhält man einen Wert von 0 - 100. [33]

System Usability Score



Abbildung 5.2: System Usability Scale Acceptability Score (Quelle: [33])

Diese Punktzahl gibt nun an, wie benutzerfreundlich und erlernbar die Website oder ein anderes Produkt ist.

Der durchschnittliche SUS-Score beträgt 68. Das bedeutet wenn der SUS-Score über 68 liegt, so ist die Punktzahl überdurchschnittlich. Auf der anderen Seite ist die Punktzahl unterdurchschnittlich, wenn sie unter 68 fällt. Eine Punktzahl von 75 ist besser als etwa 73% der SUS-Scores. Gleichzeitig ist eine Punktzahl von 52 schlechter als rund 85% der Scores. [34]

Daraus kann man nun, wie auf der Abbildung 5.2 erkennbar ist, das SUS-Ergebnis in eine Reihe von Kategorien einteilen. Daran kann wiederum direkt erkannt werden, ob der jeweilige User-Test eher negativ oder eher positiv ausgefallen ist. Durch die Verwendung von Zahlen und die immer gleiche Berechnungsweise erhält man einen genauen Wert, welcher dem Entwickler auch im direkten Vergleich von mehreren Prototypen helfen kann.

Der System Usability Scale ist eine schnelle und zuverlässige Methode zur Bewertung der Benutzerfreundlichkeit von Designlösungen. Das Tool bietet zwar keine konkreten Einblicke in spezifische Probleme, jedoch einfaches Feedback zur allgemeinen Benutzerfreundlichkeit einer Website oder App aus der Sicht eines Benutzers. [33]

5.4 Evaluation

Zunächst wurden die beiden Prototypen in einem Pilottest mit einer Person vorab getestet. In diesem Tests sind kleinere Details im Design aufgefallen, welche vor den eigentlichen Tests noch angepasst wurden. Beispielsweise ein genauerer Hinweistext beim Baukasten-System.

Die Auswertung des Pre-Test Fragebogens ergab, dass die vier Teilnehmer im Durchschnitt zwischen 23 und 35 Jahre alt sind. Des Weiteren waren alle Tester weiblich und hatten wenig bis mittelmäßige Kenntnisse über das Thema Steuern. Jedoch machen 75% von ihnen ihre Steuererklärung selbst. Davon machen 100% Gebrauch von einer Steuersoftware. Keiner der Tester hat bisher Erfahrungen mit mobilen Steuer-Apps, wie beispielsweise Taxfix, gemacht.

Bei dem Erstellen von Umfragen haben 75% angegeben, bereits Erfahrung damit zu haben. Jedoch erstellt niemand der Tester regelmäßig Umfragen, nur gelegentlich oder selten. Die genauen Werte der Umfrage sind in Form von Tortendiagrammen im Anhang A.6 angegeben.

Machst du deine Steuererklärung selbst?

4 Antworten

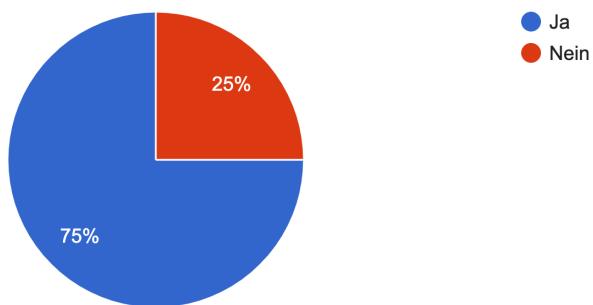


Abbildung 5.3: Kreisdiagramm zu den Ergebnissen der Frage: "Machst du deine Steuererklärung selbst?"

Nun zu den Einzelheiten, die während des Tests aufgrund der CTA-Methode aufgefallen sind. Zum einen mussten sich die Tester beim Baukasten-System zunächst orientieren. Dies war beim Formular nicht der Fall, da diese Art des Interfaces den Testern bereits bekannt ist und regelmäßig genutzt wird.

Doch auch beim Formular sind einige Aspekte aufgefallen. Zum einen wurde nicht direkt erkannt, dass die Strecke des Arbeitsweges automatisch berechnet wurde. Dieses Feld hätte man mit Hilfe von Design-Änderungen stärker von den normalen Textfeldern abgrenzen sollen. Zum Anderen waren die Farben der Icon-Buttons etwas irritierend und man hätte die Farben von aktiv und inaktiv tauschen sollen, um dies klarer darzustellen. Außerdem wurde oft nicht erkannt, dass es für den Teil "Arbeitskleidung" einen extra Schritt gibt, obwohl explizit im vorherigen Schritt "Arbeitsmittel" darauf hingewiesen wird. Dass Hinweistexte oft übersehen werden scheint nicht unbekannt zu sein. Darauf wird später noch näher eingegangen.

Beim Baukasten-System hingegen war meist das anfängliche Verständnis das Problem und die Herangehensweise an die Aufgabe war teils recht unterschiedlich. Einige begonnen zunächst mit dem Editor, andere verschoben direkt die Elemente ins Feld des Formulars. Auch die grundlegenden Lösungsansätze bezüglich der Gestaltung waren unterschiedlich. Einige verwendeten beispielsweise erst die Dropdown-Liste für die Ja/Nein-Frage und anstelle einer Dropdown-Liste im ersten Abschnitt ein Textfeld. Teilweise wurde auch beim Toggle-Button ein anderer Text verwendet, als vorgesehen war. Jedoch waren diese verschiedenen und teilweise unvorhergesehenen Lösungskonzepte nicht zwingend weniger nutzerfreundlich als andere. Die Aufgabe konnte von allen Testern zufriedenstellend gelöst werden, jedoch wurden dafür im Durchschnitt 9,5 Minuten benötigt. Zum Vergleich für den Formular-Prototypen wurden im Durchschnitt nur rund 3,5 Minuten benötigt. Einer der Gründe dafür könnte sein, dass die Tester meist direkt in die Aufgabe eingestiegen sind, ohne sich das Tool zunächst genauer anzusehen. Auch die nach dem Pilottest eingefügten Hinweistexte wurden nicht oder meist erst nach dem Auftreten von Problemen gelesen.

Dies scheint jedoch ein bekanntes Problem zu sein. Nach etwas Recherche wurde klar, dass Menschen im Web nur Wort für Wort lesen, wenn sie wirklich an den Inhalten interessiert sind. Normalerweise überfliegen sie die Seiten und suchen nach hervorgehobenen Schlüsselwörtern, aussagekräftigen Überschriften, kurzen Absätzen und einer durchsuchbaren Liste. Man sollte also nicht erwarten, dass Benutzer Inhalte lesen, die für sie weder leicht zu scannen, noch relevant erscheinen. Lange Textblöcke, unnötige Anweisungen usw. sollten vermieden werden. [35]

Dies wurde auch mit diesem Usability-Test noch einmal bestätigt. Hinweistexte waren sowohl für den Formular-Prototypen als auch den Baukasten-Prototypen keine durchschlagend erfolgreiche Lösung. Kürzere Hinweistexte zu den einzelnen Elementen oder eine Verdeutlichung mit Farben und Icons hätten dieses Problem vermutlich effektiver gelöst.

Kommen wir nun zu den SUS-Fragebögen und dem daraus errechneten System Usability Score. Wie im Anhang A.7 zu sehen ist erreichte der Prototyp des Baukasten-Systems einen Score von 89,376 und der Prototyp des Formulars einen leicht höheren Wert von 91,25. Wie man auf Abbildung 5.2 erkennt sind damit beide Werte im ausgezeichneten Bereich und damit akzeptabel. Die Werte der beide Prototypen liegen auch sehr nah beieinander. Wie kommt es jedoch, dass der Formular-Prototyp einen leicht höheren Wert aufweist? Einerseits war die Aufgabe des Ausfüllens weitaus weniger anspruchsvoll, als das eigenhändige Erstellen eines Formulars. Des Weiteren ist das Verwenden eines Formulars etwas mit dem die meisten Nutzer bereits tiefergehende Erfahrungen haben. Dieser Prototyp ist deswegen für die Tester einfacher zu benutzen als der des Baukasten-Systems.

Was das zusätzliche Feedback der User angeht wurden besonders die Icons und das einheitliche und übersichtliche Design als sehr positiv bewertet. Jedoch gab es besonders beim Baukasten-System noch einiges an Verbesserungsbedarf, wie man an den Erkenntnissen während des Tests und in Anhang A.8 und A.9 erkennt. Die Umsetzung dieser Ideen war jedoch im Zeitrahmen dieser Arbeit nicht möglich. Auch beim Formular sind während des Tests noch Feinheiten im Design aufgefallen. Diese sind in auch Anhang A.8 und A.9 festgehalten und konnten aufgrund der Zeit nicht umgesetzt werden. Dennoch ist das Feedback allgemein sehr positiv ausgefallen.

6 - Schlussbetrachtung

Um diese Arbeit abzuschließen wird sie im Folgenden kurz zusammengefasst, bewertet und es wird ein Ausblick auf die zukünftigen Weiterentwicklungs-möglichkeiten des vorgestellten Konzepts gegeben.

6.1 Zusammenfassung

Die Verwendung von Formularen spielt im World Wide Web eine immer größer werdende Rolle, wodurch auch die Formulare immer komplexere Strukturen aufweisen. Da jedoch in der Software-Entwicklung meist immer neue Lösungen entwickelt werden, anstatt auf bereits existierenden Lösungen aufzubauen und diese wiederzuverwenden, geraten besonders die Benutzerfreundlichkeit und Barrierefreiheit der Formulare in den Hintergrund. Um für dieses Problem eine Lösung zu finden wurde im Laufe dieser Arbeit ein Konzept für ein modulares Baukasten-System für Formulare entwickelt. Dieses Konzept baute auf der Recherche in den verschiedenen Bereichen von Design, Usability und Accessibility auf. Auf Grundlage dieses Konzepts wurden dann zwei Prototypen entwickelt. Beim Themengebiet der beiden Prototypen fiel die Entscheidung auf den Themenkomplex Steuern mit Fokus auf den Bereich Werbungskosten. Dieser Teil bietet bei einem Formular besonders viele Möglichkeiten, da vom Nutzer viele Informationen benötigt werden. Als Erstes wurden einzelne UI-Elemente entwickelt, aus denen dann ein Formular und ein Baukasten-System entstanden ist. Diese beiden Prototypen wurden dann in einem Usability-Test evaluiert. Dieser beinhaltete einen Fragebogen, welcher vorab ausgefüllt wurde und zwei Testaufgaben mit den zwei verschiedenen Prototypen, deren Ergebnisse dann mit Hilfe der Concurrent-Think-Aloud-Methode und SUS-Fragebögen beurteilt wurden.

6.2 Fazit

Da der Bereich von Usability und Accessibility sehr weitreichend ist und es sehr viele Aspekte gibt, die man betrachten kann, konnte nicht jeder Teilaспект beleuchtet werden und auch die entwickelten Prototypen wiesen noch einige Usability-Probleme auf. Jedoch wurde versucht, so viele verschiedene Facetten wie möglich mit in diese Arbeit einzubringen, um ein benutzerfreundliches Interface zu entwickeln. Besonders das Konzept von einem System, welches es auch Nicht-Entwicklern ermöglicht, an der Implementierung von Software mitwirken zu können, ist ein großer Schritt in Richtung eines noch nutzerfreundlicheren Internets. Dies würde auch Entwicklern ermöglichen, sich auf neue, wichtigere Aufgaben zu konzentrieren und nicht bei jedem Software-Projekt vor einer leeren Leinwand zu stehen. Besonders die Usability-Tests zeigten, dass ein Baukasten-System durchaus leicht zu benutzen

und zu erlernen ist. Durch die eingebundene Flexibilität der Elemente, mit deren Hilfe man Aussehen und Werte der Elemente anpassen kann, könnten diese bei Bedarf immer wieder neu verwendet und müssten nicht in anderen Projekten, bei denen ein Formular benötigt wird, neu implementiert werden.

6.3 Ausblick

Um einen Ausblick für diese Arbeit zu geben, sei zuerst gesagt: Es wird wahrscheinlich nie eine universelle Lösung geben, die für jeden in jeder Situation funktioniert. Ein gemeinsames Anliegen von Designern ist es, ein Design mit dem kleinsten gemeinsamen Nenner zu erstellen. Der Versuch, alle zufrieden stellen zu wollen, ist weder für das Produkt noch für die Nutzer erstrebenswert, geschweige denn sinnvoll. Die zugrunde liegende Herausforderung ist die enorme Komplexität der menschlichen Vielfalt. Es gibt endlose Nuancen, Ansichten und Überlegungen beim Entwerfen für Menschen. Es gibt keine Antwort, die jedem passt und alle gleichermaßen zufrieden stellt. Inklusion ist imperfect und erfordert Demut. Es ist eine Gelegenheit, neugierig zu sein und Herausforderungen mit dem Wunsch zu lernen anzugehen. ([2], S.9)

Das bedeutet auch wenn es nie eine perfekte Lösung geben kann, so sollte man sich dennoch der Herausforderung stellen, sowohl für die Ersteller von Software als auch für die Benutzer, immer nach dem bestmöglichen und benutzerfreundlichsten Weg zu suchen, den es gibt.

Diese Arbeit ist nur ein kleiner Teil von dem, was bezüglich UX/UI-Design rund um das Thema Formulare möglich ist, besonders da sich das Internet und dessen Bestandteile stets weiterentwickeln. Um sich jedoch auf diese Themen und die kontinuierliche Veränderung konzentrieren zu können braucht es mehr Zeit. Diese kann durch modulare Systeme geschaffen werden. Jedoch sind auch modulare Systeme in ihrem Design und der Benutzerfreundlichkeit noch lange nicht hundertprozentig ausgereift. In diesen Bereichen sollte die Forschung in den nächsten Jahren noch weiter vorangetrieben werden, um die Entwicklung von Software und besonders das Design für so viele Menschen wie möglich zugänglich zu machen. Auch die Verwendung von Formularen sollte immer intuitiver und so simpel wie möglich gestaltet werden - denn ohne sie wäre das Internet eine sehr einseitige Welt.

Danksagung

Zum Abschluss möchte ich mich für die Zusammenarbeit mit der Firma adorsys GmbH & Co. KG und besonders bei Philipp Wolf für die kompetente Betreuung und Unterstützung während der Arbeit bedanken. Auch bei allen Weiteren, die mich beim Konzipieren, Entwickeln und Testen der Arbeit unterstützt haben, möchte ich mich bedanken.

Literaturverzeichnis

- [1] A. Silver, *Form design patterns: a practical guide to designing and coding simple and exclusive forms for the web*. Freiburg: Smashing Media AG, 2018.
- [2] K. Holmes und J. Maeda, *Mismatch: how inclusion shapes design*. Cambridge, Massachusetts ; London, England: The MIT Press, 2018.
- [3] adorsys GmbH & Co. KG Company 2021
- [4] „ISO 9241-11:2018(en), Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts“. <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en> (zugegriffen Apr. 22, 2021).
- [5] „ISO 9241-210:2010(en), Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems“. <https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en> (zugegriffen Apr. 29, 2021).
- [6] J. Enders, *Designing UX: Forms: Create Forms That Don't Drive Your Users Crazy*. SitePoint, 2016.
- [7] N. Babich, „Form Design Best Practices: Structure, Inputs, & Labels | Adobe XD Ideas“, *Ideas*, März 12, 2020
<https://xd.adobe.com/ideas/principles/web-design/best-practices-form-design/> (zugegriffen Jan. 06, 2021).
- [8] Matteo Penzo, „Label Placement in Forms :: UXmatters“, Juli 12, 2006
<https://www.uxmatters.com/mt/archives/2006/07/label-placement-in-forms.php> (zugegriffen Mai 01, 2021).
- [9] N. Babich, „10 Rules For Efficient Form Design“, Medium, Mai 10, 2020.
<https://uxplanet.org/10-rules-for-efficient-form-design-e13dc1fb0e03> (zugegriffen Jan. 06, 2021).
- [10] „Atomic Design Methodology | Atomic Design by Brad Frost“. <http://atomicdesign.bradfrost.com/chapter-2/> (zugegriffen Jan. 21, 2021).
- [11] V. K. Kotaru, *Angular for Material Design Leverage Angular Material and TypeScript to Build a Rich User Interface for Web Apps*. Berkeley, CA: Apress : Imprint: Apress, 2020.
- [12] „Angular - What is Angular?“ <https://angular.io/guide/what-is-angular> (zugegriffen Apr. 18, 2021).
- [13] „ResearchKit“. <http://researchkit.org/> (zugegriffen Dez. 15, 2020).
- [14] „iOS Human Interface Guidelines: Research Apps“. <http://researchkit.org/hig/index.html> (zugegriffen Mai 01, 2021).
- [15] „Storybook: UI component explorer for frontend developers“. <https://storybook.js.org> (zugegriffen Dez. 15, 2020).
- [16] „What's a Story“. <https://storybook.js.org/docs/react/get-started/whats-a-story/> (zugegriffen Mai 01, 2021).
- [17] Thalion, „What is a Design System? Definition & Examples | UXMISFIT.COM“, März 26, 2019.
<https://uxmisfit.com/2019/03/26/what-is-a-design-system-everything-you-need-to-know/> (zugegriffen Dez. 15, 2020).

- [18] „Introduction“, *Material Design*. <https://material.io/design/introduction> (zugegriffen Apr. 18, 2021).
- [19] „What is Angular Material | UI Component Library in Angular 9“, *Edureka*, Feb. 26, 2020. <https://www.edureka.co/blog/what-is-angular-material/> (zugegriffen Apr. 18, 2021).
- [20] Thalion, „Quick Answer – What do you use Sketch for? | UXMISFIT.COM“, Juli 18, 2019. <https://uxmisfit.com/2019/07/18/quick-answer-what-do-you-use-sketch-for/> (zugegriffen Apr. 08, 2021).
- [21] „Industry-leading vector graphics software | Adobe Illustrator“. <https://www.adobe.com/uk/products/illustrator.html> (zugegriffen Apr. 08, 2021).
- [22] „Microsoft Design“. <https://www.microsoft.com/design/inclusive/> (zugegriffen Jan. 27, 2021), Aus dem PDF: Toolkit Inclusive 101.
- [23] „Adobe Color | Accessibility tools“. <https://helpx.adobe.com/creative-cloud/adobe-color-accessibility-tools.html> (zugegriffen Apr. 01, 2021).
- [24] „Farbrad, ein Farbpaletten-Generator | Adobe Color“. <https://color.adobe.com/de/create/color-wheel> (zugegriffen Mai 01, 2021).
- [25] R. F. Dam und T. Y. Siang, „Personas – A Simple Introduction“, *The Interaction Design Foundation*. <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them> (zugegriffen März 31, 2021).
- [26] Eeva Ilama, „Creating Personas | UX Booth“, Juni 9, 2015 <https://www.uxbooth.com/articles/creating-personas/> (zugegriffen Apr. 22, 2021).
- [27] A. Osterwalder, Y. Pigneur, G. Bernarda, A. Smith, und P. Papadakos, *Value proposition design: how to create products and services customers want. Get started with.* Hoboken, NJ: Wiley, 2014.
- [28] „Das ist Taxfix“, *Taxfix*. <https://taxfix.de/uber-taxfix/> (zugegriffen Apr. 08, 2021).
- [29] „So funktioniert die Taxfix-App“, *Taxfix*. <https://taxfix.de/so-funktioniert-taxfix/> (zugegriffen Mai 01, 2021).
- [30] B. Coleman und D. Goodwin, *Designing UX: prototyping*. 2017.
- [31] „Sass: Sass Basics“. <https://sass-lang.com/guide> (zugegriffen Apr. 22, 2021).
- [32] „Concurrent Think-aloud Method (CTA) in the usability.de glossary“, *usability.de*. <https://www.usability.de/en/usability-user-experience/glossary/concurrent-think-aloud.html> (zugegriffen Apr. 24, 2021).
- [33] Andrew Smyk, „The System Usability Scale & How it's Used in UX | Adobe XD Ideas“, *Ideas*, März 17, 2020 <https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/> (zugegriffen März 20, 2021).
- [34] usabilTEST, „Making Sense of SUS Data“. <https://www.usabilitest.com/blog/making-sense-of-sus-data#usability> (zugegriffen Apr. 22, 2021).
- [35] Z. G. Kollin Zoltan, „Myth #1: People read on the web“, *UX Myths*. <https://uxmyths.com/post/647473628/myth-people-read-on-the-web> (zugegriffen Apr. 28, 2021).

A - Anhang

A.1 Personas



ANTON MÜLLER

„Ich versuche möglichst viel in meinem Leben zu automatisieren, um mich auf die schönen Dinge konzentrieren zu können.“

ZIELE

Anton möchte so viel Zeit wie möglich mit seinen Kindern verbringen und ihnen beim aufwachsen zusehen. Die Zeit mit seiner Familie ist ihm mit das Wichtigste in seinem Leben. Er besitzt zwar grundlegendes Wissen in Sachen Steuern, jedoch muss er viel Aufwand in seine Steuererklärung stecken, was er als sehr störend und nervig empfindet. Er wünscht sich deshalb eine schnelle und unkomplizierte, digitale Lösung, um Zeit zu sparen, um diese wiederum mit seinen Söhnen verbringen zu können.

SOFTSKILLS

Kenntnis Steuererklärung	● ● ● ○ ○
Technik- und Internetaffinität	● ● ● ● ●
Offen für Neues	● ● ● ● ○
Organisation	● ● ● ● ● ○
Visuell	○ ○ ○ ● ○ ○
Textuell	○ ○ ○ ○ ○ ○

PERSONA I



ECKDATEN

34 Jahre alt
Wohnt in Fürth
Ist verheiratet und hat zwei Söhne (2 & 4 Jahre alt)



HOBBIES, LEIDENSCHAFTEN

Mit seinen Söhnen spielen, Radtouren, Playstation spielen mit Freunden



BERUFSERFAHRUNG

Fachabitur im Bereich BWL
Hat erst BWL studiert und ist dann zu Wirtschaftsinformatik gewechselt
Hat 4 Jahre als Business Analyst bei ‚DATEV‘ gearbeitet
Ist aktuell seit 3 Jahren als Frontend-Entwickler bei ‚adorsys GmbH & Co. KG‘



EMOTIONEN

Home Office, Zeit mit seiner Familie
Work-Life-Balance



Wenig Zeit, Überstunden
Lange Meetings, Steuererklärung



ERFAHRUNGEN MIT STEUERERKLÄRUNGEN

- Macht seine Steuererklärung selbst, hat jedoch kaum Erfahrung damit
- Empfindet es als sehr kompliziert und anstrengend
- Es ist viel Aufwand nötig und kostet sehr viel Zeit
- Hat bedenken, dass er noch nicht alles an Geld wieder zurückholt, was möglich wäre



ERWARTUNGEN AN INFORMATIONS AUSTAUSCH

- Kurz, einfach und effizient
- Verständlich und präzise formuliert
- Das meiste sollte automatisch funktionieren
- Nichts überflüssiges oder umständliches





CHRIS ZIMMERMANN

„Meine Steuererklärung mache ich seit 30 Jahren immer auf die selbe Art und Weise.“

ZIELE

Chris will seine Karriere und auch seine körperliche Fitness noch so lange wie möglich erhalten und vielleicht sogar noch weitere Fitnessstudios, eventuell sogar eine eigene Kette eröffnen. Er kommt mit seinen Steuern sehr gut klar, jedoch ist er kein technikaffiner Mensch und am Computer unter anderem eingeschränkt durch seine Farbenblindheit. Er wünscht sich eine Software die auch für Menschen wie ihn verständlich und einfach zu verwenden ist, um seine Steuererklärungen in Zukunft noch effizienter erledigen zu können.

SOFTSKILLS

Kenntnis Steuererklärung	● ● ● ● ●
Technik und Internetaffinität	● ○ ○ ○ ○
Offen für Neues	● ● ○ ○ ○
Organisation	● ● ● ● ○
Visuell	○ ○ ○ ○ ○
	Textuell

PERSONA II



ECKDATEN

53 Jahre alt
Wohnt in Schnaittach
Ist verheiratet und hat eine Tochter (24 Jahre alt)
Hat eine Farbfehlsehigkeit (Deutanopie)



HOBBIES, LEIDENSCHAFTEN

Sport in allen Variationen, Gesunde Ernährung, Wandern gehen mit der Familie



BERUFSFAHRUNG

Fachabitur im Bereich Gesundheit
Hat eine Ausbildung als Fitnesscoach abgeschlossen
Vor 10 Jahren hat er sein eigenes Fitnessstudio eröffnet
Ist aktuell immer noch Trainer und Leiter seines Fitnessstudios „Life Gesundheitszentrum“



EMOTIONEN

Sport, Gesundes und gutes Essen
Zeit mit seiner Tochter, Schönes Wetter



Fast Food, Seine Farbfehlsehigkeit, Computer & Technik, Kaputte Fitnessgeräte



ERFAHRUNGEN MIT STEUERERKLÄRUNGEN

- Macht seine Steuererklärung seit 30 Jahren alleine und hat dementsprechend sehr viel Erfahrung damit
- Kopiert meist seine Eingaben vom Vorjahr um Zeit zu sparen
- Überlegt ab und an, ob er sich einen Steuerberater nehmen sollte, um noch mehr rauszuholen zu können



ERWARTUNGEN AN INFORMATIONSAUSTAUSCH

- Erklärungen für nicht technikaffine Menschen
- Papierformular in digitaler Form
- Daten vom Finanzamt sollten automatisch ausgefüllt werden
- Große Felder mit klaren Farben, sodass auch jemand mit Sehschwäche alles gut erkennen kann



SportScheck

GARMIN **Huel®**



LINDA KÖNIG

„Das Leben ist viel zu schön für Steuererklärungen.“

ZIELE

Linda hat sehr wenig Zeit durch ihren Job und ihr großes Engagement in ehrenamtlichen Projekten. Sie liebt es anderen Menschen helfen zu können und möchte immer ein Vorbild für andere sein. Sie wünscht sich das ihr Job in Zukunft mehr Wertschätzung erhält und Krankenpfleger besser bezahlt werden. Sie könnte jedoch heute schon eine Menge Geld durch ihre Steuererklärung zurück bekommen. Allerdings hat sie sich noch nie wirklich damit beschäftigt und traut sich nicht so richtig an die Sache ran. Sie hofft, dass ihre Steuererklärung vielleicht durch eine Software einfach erledigt werden kann, ohne das sie viel Aufwand und Zeit darin investieren muss.

SOFTSKILLS

Kenntnis Steuererklärung	● ○ ○ ○ ○
Technik- und Internetaffinität	● ● ● ○ ○
Offen für Neues	● ● ● ● ○
Organisation	● ● ● ○ ○
Visuell	○ — ● ○ ○ ○
	Textuell

PERSONA III



ECKDATEN

28 Jahre alt
Wohnt in Nürnberg
Ist ledig und hat keine Kinder



HOBBIES, LEIDENSCHAFTEN

Ehrenamtliche Arbeit, Kochen,
Ihr Pflegepferd, Mädelsabende



BERUFSEXPERIENZ

Mittlere Reife
Hat eine Ausbildung zur Pflegekraft gemacht
Hat viele Fortbildungen abgeschlossen
Arbeitet aktuell seit 4 Jahren als Pflegekraft im „Martha Maria“ in Nürnberg



EMOTIONEN

Anderen helfen, Zeit bei ihrem Pferd
 Ihre Kollegen, Chillige Abende mit Freunden

Gehalt von Pflegekräften, Rücksichtslose Menschen
Corona Leugner, Steuererklärung



ERFAHRUNGEN MIT STEUERERKLÄRUNGEN

- Hat noch nie eine Steuererklärung gemacht und dementsprechend keine Erfahrungen damit
- Wüsste garnicht wo sie am Besten anfangen sollte und benötigt viel Hilfe
- Hat Angst, dass sie etwas falsch machen könnte
- Empfindet es als sehr kompliziert und hat wenig Geduld sich darauf konzentrieren



ERWARTUNGEN AN INFORMATIONSAUSTAUSCH

- Viele Beispiele und Tipps (evtl. auch mit Bildern & Illustrationen)
- Einfach gestaltet, damit auch jemand ohne Erfahrungen alles versteht
- So wenig wie möglich selbst machen müssen, die Software soll das meiste übernehmen
- Möglichkeit für Hilfe sollte gegeben sein





MIA MARIE MEIER

„Steuererklärungen lassen sich ganz easy online erledigen.“

ZIELE

Mia Marie will noch hoch hinaus in ihrem Leben. Sie ist nicht sehr zufrieden mit ihrem aktuellen Stand und möchte bald nach Berlin umziehen. Sie hofft irgendwann vielleicht als Influencer arbeiten zu können oder durch das Fernsehen berühmt und erfolgreich zu werden. Sie hat keine großen Probleme und bis jetzt ging ihr alles immer einfach von der Hand. Auch ihre Steuererklärung erledigt sie mit Leichtigkeit, meist am Smartphone, während sie nebenbei Serien schaut. Jedoch findet sie es immer toll, wenn sie ein neue, noch einfachere und bessere Software oder Apps findet, welche ihr noch mehr Arbeit abnehmen.

SOFTSKILLS

Kenntnis Steuererklärung	● ● ● ● ○
Technik- und Internetaffinität	● ● ● ● ●
Offen für Neues	● ● ● ● ●
Organisation	● ● ● ○ ○
Visuell	● ○ ○ ○ ○
	Textuell

PERSONA IV



ECKDATEN

26 Jahre alt
Wohnt in Nürnberg
Ist ledig und hat keine Kinder



HOBBIES, LEIDENSCHAFTEN

Social Media, Auf Instagram surfen
Mit Freunden essen gehen, Netflix



BERUFSEXPERIENZ

Mittlere Reife
Hat eine Ausbildung zur Speditions-Kauffrau gemacht
Ist nach ihrer Ausbildung auf die BOS und hat Abitur im Bereich BWL & Marketing
Ist aktuell seit 3 Jahren Speditions-Kauffrau bei „Müller - Die lila Logistik AG“
Sie will eigentlich irgendetwas mit Medien machen



EMOTIONEN

Zeit am Smartphone, Neue Trends entdecken
Viele Follower, Neue Serien auf Netflix schauen



Altmodisch Technik und Software, Langeweile
Konkurrenz, Ihren jetzigen Job



ERFAHRUNGEN MIT STEUERERKLÄRUNGEN

- Macht ihre Steuererklärung mit Hilfe von Online-Tools und Apps und besitzt viel Wissen über Steuern
- Ist sehr überzeugt von den Tools, die sie nutzt
- Testet öfter neue Apps und Software, um eventuell eine noch einfachere Lösung zu finden
- Hat keine großen Probleme und findet sich sehr einfach zurecht



ERWARTUNGEN AN INFORMATIONSAUSTAUSCH

- Ist schon relativ überzeugt von den Tools die sie bisher verwendet hat
- Doppelte Eingaben sollten vermieden werden
- Es sollte möglichst viel automatisiert ablaufen
- Die Daten vom Vorjahr sollten gespeichert werden und wieder übernommen werden können



NETFLIX

ABOUT YOU°

Bildlizenzen

ANTON MÜLLER

https://www.freepik.com/free-photo/dad-working-laptop-from-home-during-quarantine-with-child_10858604.htm#page=2&query=dad+kids+laptop&position=9
< a href="https://www.freepik.com/photos/home">Home photo created by freepik - www.freepik.com

<https://icons8.com/icon/71257/angularjs>

<https://icons8.com/icon/108784/javascript>

<https://icons8.com/icon/123603/react-native>

<https://icons8.com/icon/63192/bitcoin>

CHRIS ZIMMERMANN

https://www.freepik.com/free-photo/woman-gym-with-coach_1612387.htm#query=gym+coach&position=14
< a href="https://www.freepik.com/photos/background">Background photo created by senivpetro - www.freepik.com

<https://icons8.com/icon/34552/adidas-trefoil>

<https://www.sportscheck.com/presse/images/logo-1542869>

<https://www.garmin.com/de-DE/newsroom/images/garmin-logo-1570020>

<https://discourse.huel.com/t/huel-vector-logo/7739>

LINDA KÖNIG

https://www.freepik.com/free-photo/smiley-female-nurse-office-with-stethoscope-notepad_10893013.htm#page=1&query=nurse&position=5
< a href="https://www.freepik.com/photos/woman">Woman photo created by freepik - www.freepik.com

<https://icons8.com/icon/17836/android-os>

<https://icons8.com/icon/31079/ebay>

<https://logo-logos.com/kleiderkreisel-logo-83.html>

<https://vereinsplaner.at/a/c/kraemer-pferdesport-asten>

MIA MARIE MEIER

https://www.freepik.com/free-photo/glad-fair-haired-woman-trendy-outfit-doing-her-job-office-holding-notebook-pencil_10785880.htm#page=2&query=laptop+woman&position=25
< a href="https://www.freepik.com/photos/business">Business photo created by lookstudio - www.freepik.com

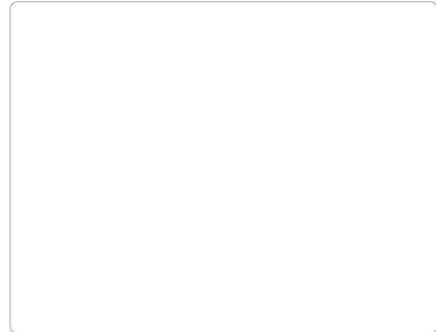
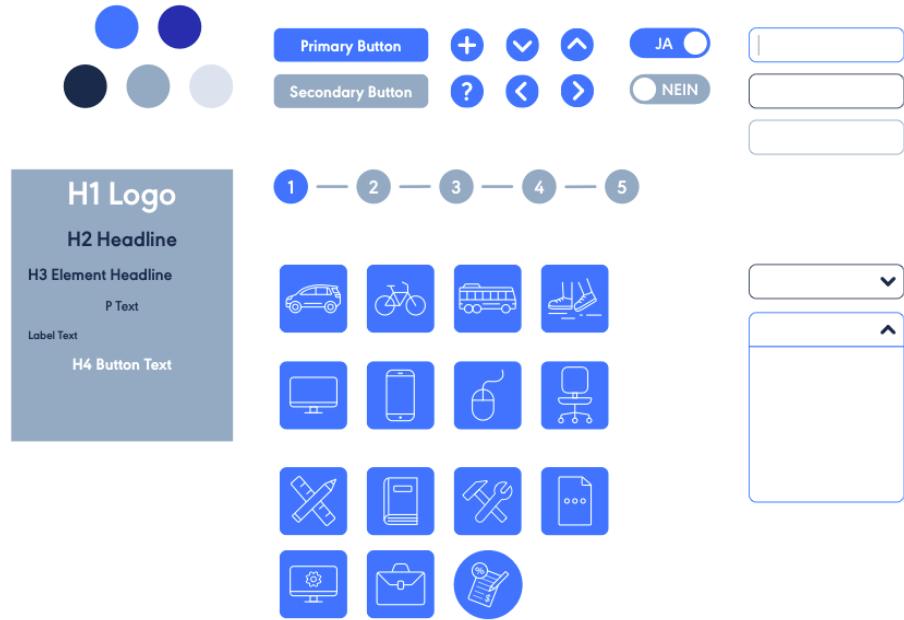
<https://icons8.com/icon/30840/apple-logo>

<https://icons8.com/icon/20519/netflix>

<https://icons8.com/icon/Xy10Jcu1L2Su/instagram>

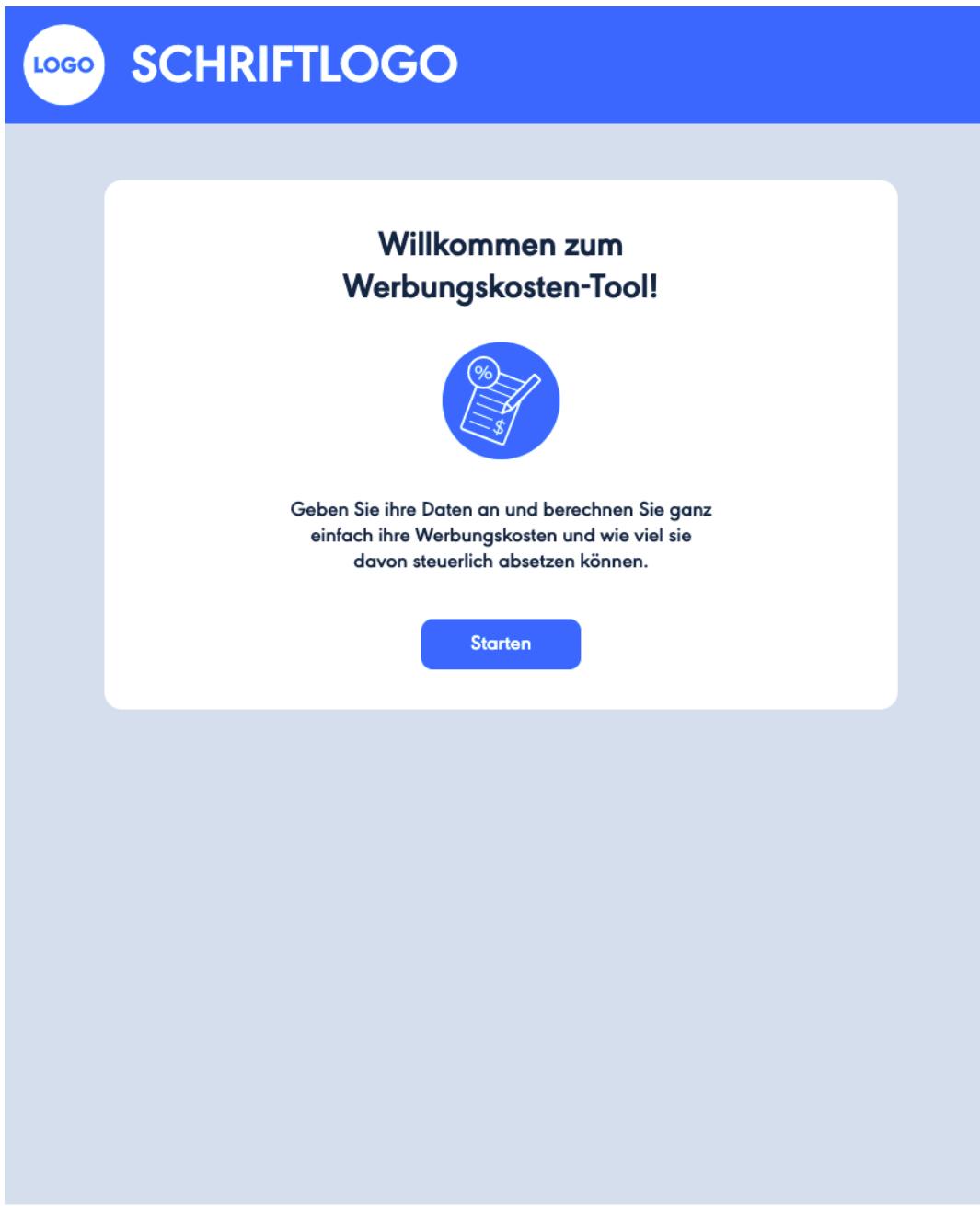
https://de.wikipedia.org/wiki/Datei:ABOUT_YOU_Logo.png

A.2 UI-Elemente & Icons





A.3 Formular-Prototyp



The image shows a wireframe prototype of a web application for calculating advertising costs. At the top, there is a blue header bar with a white circular logo containing the word "LOGO" and the text "SCHRIFTLOGO" in white. Below the header is a light gray main area. In the center of this area is a white rounded rectangle containing the following content:

**Willkommen zum
Werbungskosten-Tool!**

Below this text is a blue circular icon with a white outline, featuring a stylized document with a percentage sign (%) and a dollar sign (\$).

Geben Sie ihre Daten an und berechnen Sie ganz
einfach ihre Werbungskosten und wie viel sie
davon steuerlich absetzen können.

At the bottom of the white box is a blue button with the text "Starten".

[Datenschutzerklärung](#)

[Impressum](#)

[Cookie-Einstellungen](#)



SCHRIFTLOGO

1

2

3

4

5

ALLGEMEINES

Persönliche Informationen

Name

Nachname

Steuerklasse

Steuernummer

Adresse

Straße

Hausnummer

PLZ

Ort

Informationen zum Beruf

Branche

Beschäftigt als

Weiter >



SCHRIFTLOGO

1 — 2 — 3 — 4 — 5

FAHRTKOSTEN

Adresse des Arbeitsplatzes

Straße

Hausnummer

PLZ

Ort

Adressen weiterer Arbeitsstätten



Weitere Adresse hinzufügen

Fortbewegungsmittel

Auto



Bus



Fahrrad



Zu Fuß



< Zurück

Weiter >

[Datenschutzerklärung](#)

[Impressum](#)

[Cookie-Einstellungen](#)



SCHRIFTLOGO

1

2

3

4

5

HOME-OFFICE

Ist Home-Office bei Ihrem Job möglich?

NEIN

< Zurück

Weiter >

[Datenschutzerklärung](#)

[Impressum](#)

[Cookie-Einstellungen](#)



SCHRIFTLOGO

1

2

3

4

5

HOME-OFFICE

Ist Home-Office bei Ihrem Job möglich?

JA

Haben Sie dafür Einrichtung gekauft?

Computer



Smartphone



PC-Zubehör



Büromöbel



< Zurück

Weiter >

[Datenschutzerklärung](#)

[Impressum](#)

[Cookie-Einstellungen](#)



1

2

3

4

5

ARBEITSMITTEL

Benötigen Sie Arbeitsmittel? (Keine Arbeitskleidung)

JA

Haben Sie neue Arbeitsmittel gekauft?

Materialien



Fachliteratur



Werkzeuge



Aktentasche



Software



Sonstiges



< Zurück

Weiter >



SCHRIFTLOGO

1 — 2 — 3 — 4 — 5

ARBEITSKLEIDUNG

Benötigen Sie Arbeitskleidung?

JA

Haben Sie neue Arbeitskleidung gekauft?

NEIN

< Zurück

Zur Überprüfung >

[Datenschutzerklärung](#)

[Impressum](#)

[Cookie-Einstellungen](#)



SCHRIFTLOGO

1

2

3

4

5

ARBEITSKLEIDUNG

Benötigen Sie Arbeitskleidung?

JA

Haben Sie neue Arbeitskleidung gekauft?

JA

3er Pack weiße T-Shirts

50er Pack FFP2 Mund-Nasen-Masken

69,99€

< Zurück

Zur Überprüfung >

[Datenschutzerklärung](#)

[Impressum](#)

[Cookie-Einstellungen](#)

A.4 Baukasten-System-Prototyp

1. Element Editor:

Im Editor können Sie Texte und weitere Merkmale der Formular-Elemente anpassen. Jedes Formular-Element hat seinen eigenen Editor-Bereich.

Text für Headline 1
Headline 1

Text für Headline 2
Headline 2

Label für Text-Feld 1
Label Text-Feld 1
 Klein Mittel Groß

Label für Text-Feld 2
Label Text-Feld 2
 Klein Mittel Groß

Label für Text-Feld 3
Label Text-Feld 3
 Klein Mittel Groß

Label für Text-Feld 4
Label Text-Feld 4
 Klein Mittel Groß

Label für Dropdown-Feld 1
Label Dropdown-Feld 1

Werte für Dropdown-Feld 1
1,2,3,4,5
Dropdown-Werte durch Kommas getrennt eingeben!

Label für Toggle-Button 1
Label Toggle-Button 1

Headline für Icon-Buttons
Headline Icon-Buttons

Label für Icon-Button 1
Label Icon-Button 1

Label für Icon-Button 2
Label Icon-Button 2

Label für Icon-Button 3
Label Icon-Button 3

2. Elemente:

Hier befinden sich die verschiedenen Elemente die Sie verwenden können. Verschieben Sie diese einfach mit Hilfe von Drag and Drop zum weißen Feld bei "3. Formular:" um sie dem Formular hinzuzufügen.

Headline 1

Headline 2

Label Text-Feld 1

Label Text-Feld 2

Label Text-Feld 3

Label Text-Feld 4

Label Dropdown-Feld 1

Label Toggle-Button 1 NEIN

Headline Icon-Buttons

Label Icon-Button 1 Label Icon-Button 2 Label Icon-Button 3

3. Formular:

Hier sehen Sie die Vorschau, wie das fertige Formular aussehen wird. Verschieben Sie die Elemente nach oben oder unten um sie an die gewünschte Position zu setzen.

DROP ELEMENTS HERE

1. Element

Editor:

Im Editor können Sie Texte und weitere Merkmale der Formular-Elemente anpassen. Jedes Formular-Element hat seinen eigenen Editor-Bereich.

Text für Headline 1

Persönliche Daten:

Text für Headline 2

Home-Office:

Label für Text-Feld 1

Name

Klein Mittel Groß

Label für Text-Feld 2

Nachname

Klein Mittel Groß

Label für Text-Feld 3

Steuernummer

Klein Mittel Groß

Label für Text-Feld 4

Label Text-Feld 4

Klein Mittel Groß

Label für Dropdown-Feld 1

Steuerklasse

Werte für Dropdown-Feld 1

Klasse 1, Klasse 2, Klasse

Dropdown-Werte durch Kommas getrennt eingeben!

Label für Toggle-Button 1

Ist Arbeit im Home-Off

Headline für Icon-Buttons

Wurden für das Home-

Label für Icon-Button 1

Computer

Label für Icon-Button 2

PC-Zubehör

Label für Icon-Button 3

Büromöbel

2. Elemente:

Hier befinden sich die verschiedenen Elemente die Sie verwenden können. Verschieben Sie diese einfach mit Hilfe von Drag and Drop zum weißen Feld bei "3. Formular:" um sie dem Formular hinzuzufügen.

Label Text-Feld 4

3. Formular:

Hier sehen Sie die Vorschau, wie das fertige Formular aussehen wird. Verschieben Sie die Elemente nach oben oder unten um sie an die gewünschte Position zu setzen.

Persönliche Daten:

Name

Nachname

Steuerklasse

Klasse 1

Steuernummer

Home-Office:

Ist Arbeit im Home-Office möglich?

NEIN

Wurden für das Home-Office Gegenstände gekauft?

Computer

PC-Zubehör

Büromöbel



A.5 Test-Skript

User Test Skript

1.) Einführung

- > Begrüßung
- > Der Test wird aufgezeichnet (Einverständnis klären)
 - > Aufzeichnung starten und beginnen mit: „Herzlichen Dank, das Sie sich mit der Aufzeichnung einverstanden erklärt haben“
- > Grund für die Tests:

Durch immer mehr technologische Innovationen wird der Handelsplatz Internet und auch der damit verbundene Datenverkehr immer größer. Um an die verschiedenen, immer spezifischer werdenden Daten der Nutzer zu gelangen, werden meist Formulare verwendet. Damit sie dieser Entwicklung standhalten können, müssen auch die Formulare immer komplexer werden, was jedoch ihre Erstellung und Benutzbarkeit erheblich erschwert.

Im Rahmen meiner Bachelorarbeit untersuche ich ein Konzept für ein modulares Baukasten-System für Formulare im Themen Bereich Steuern, genauer Werbungskosten. Durch den Test möchte ich nun herausfinden ob ein solches System die Erstellung und Benutzbarkeit von Formularen, sowohl für die Ersteller als auch für die Nutzer vereinfachen kann.
- > Der Test besteht aus 2 Teilen:
 - > Im ersten Teil soll ein Formular mit Hilfe eines Baukasten-Systems erstellt werden
 - > Im zweiten Teil soll ein Formular mit Hilfe vorgegebener Daten ausgefüllt werden
- > Standardabfragen durchgehen:
 - Testnummer
 - Alter
 - Kenntnisse in Sachen Steuern
 - Wird die Steuererklärung selbst erledigt
 - Wenn Ja, wie wird die Steuererklärung erledigt
 - Erfahrung mit Steuer-Apps (z.B. tatfix, ...)
 - Erfahrung mit dem Erstellen von Umfragen und Ähnlichem (z.B. mit Google Forms, ...)

- > Anweisung vor den Aufgaben:

-> „Ganz wichtig ist, wir evaluieren / testen nicht Sie sondern Sie evaluieren / testen für uns das User Interface der beiden Prototypen. Damit helfen Sie uns die User Experience von Formularen und Formular-Baukästen zu verbessern.

-> Während Sie das System bedienen bitte ich Sie, laut zu denken und Ihren Eindruck vom System zu schildern, dabei ist jede Bemerkung gleich wertvoll, egal ob negativ oder positiv. Wir möchten wissen, was Sie überrascht, was Sie freut, was Sie irritiert oder sogar frustriert. Wenn Sie an einen Punkt kommen, an dem Sie nicht sicher sind was Sie tun sollen oder Sie versuchen herauszufinden was Sie hier tun sollen, dann sagen Sie uns bitte, was Sie denken.

2.) Hauptteil

- > Sie können jederzeit Fragen stellen, haben Sie im Moment Fragen?
- > Stoppuhr starten, wenn der User beginnt

3.) Use Case Building Kit (max. 25 Minuten)

-> Wir bitten Sie, ein Formular mit der Software zum Thema Werbungskosten zu erstellen

-> Das Formular soll wie folgt aufgebaut sein. Welche Elemente eingesetzt werden darf frei entschieden werden. Achten Sie jedoch darauf, dass das Formular auch von unerfahrenen Personen leicht auszufüllen ist.

Persönliche Daten:

Name

Nachname

Steuerklasse (Klasse 1, Klasse 2, Klasse 3, Klasse 4, Klasse 5, Klasse 6)

Steuernummer

Home-Office:

Ist Arbeit im Home-Office möglich?

Wurden für das Home Office Gegenstände gekauft?

4.) Nachbefragung Building Kit

-> Stoppuhr stoppen, wenn der User fertig ist (oder nach 25 Minuten abbrechen, wenn die Aufgabe bis dahin nicht erfüllt wurde)

-> Der User muss nach dem jeweiligen Use Case folgende Fragen beantworten:

SUS-Fragebogen:

- Ich kann mir sehr gut vorstellen, das Baukasten-System regelmäßig zu nutzen.
- Ich empfinde das Baukasten-System als unnötig komplex.
- Ich empfinde das Baukasten-System als einfach zu nutzen.
- Ich denke, dass ich technischen Support brauchen würde, um das Baukasten-System zu nutzen.
- Ich finde, dass die verschiedenen Funktionen des Baukasten-Systems gut integriert sind.
- Ich finde, dass es im Baukasten-System zu viele Inkonsistenzen gibt.
- Ich kann mir vorstellen, dass die meisten Leute das Baukasten-System schnell zu beherrschen lernen.
- Ich empfinde die Bedienung als sehr umständlich.
- Ich habe mich bei der Nutzung des Baukasten-Systems sehr sicher gefühlt.
- Ich musste eine Menge Dinge lernen, bevor ich mit dem Baukasten-System arbeiten konnte.

Sonstige Anmerkungen:

Positiv:

Negativ:

Verbesserungen:

5.) Use Case Formular (max. 15 Minuten)

-> Stellen Sie sich vor sie benutzen die Software um damit ihre Steuererklärung zu erledigen. Sie beginnen mit dem Punkt Werbungskosten und müssen dazu das nun folgende Formular ausfüllen.

-> Hierbei geht es darum die Nutzerfreundlichkeit eines solchen Steuerformulars zu testen

-> Einige Dinge wurde schon vorausgefüllt um den Prototypen zu vereinfachen.

-> Sie sollen dafür die Daten der hier genannten Person benutzen um das Formular auszufüllen:

Name: Linda König

Steuerklasse: Steuerklasse 1

Steuernummer: 99 145 672 874

Adresse:
Hasensprung 15
90411 Nürnberg
Branche: Gesundheit und Sozialwesen
Beschäftigt als: Angestellte(r) im öffentl. Dienst

Adresse des Arbeitsplatzes:
Krankenhaus Martha-Maria Nürnberg
Stadenstraße 58, 90491 Nürnberg

Fortbewegungsmittel: Bus

(Strecke: 3,8 km)

Keine Arbeit im Home-Office

Hat Fachliteratur für ihre Arbeit gekauft

Benötigt Arbeitskleidung und Masken für ihre Arbeit

6.) Nachbefragung Formular

-> Stoppuhr stoppen, wenn der User fertig ist (oder nach 15 Minuten abbrechen, wenn die Aufgabe bis dahin nicht erfüllt wurde)

-> Der User muss nach dem jeweiligen Use Case folgende Fragen beantworten:

SUS-Fragebogen:

- Ich kann mir sehr gut vorstellen, das Formular regelmäßig zu nutzen.
- Ich empfinde das Formular als unnötig komplex.
- Ich empfinde das Formular als einfach zu nutzen.
- Ich denke, dass ich technischen Support brauchen würde, um das Formular zu nutzen.
- Ich finde, dass die verschiedenen Funktionen des Formular gut integriert sind.
- Ich finde, dass es im Formular zu viele Inkonsistenzen gibt.
- Ich kann mir vorstellen, dass die meisten Leute das Formular schnell zu beherrschen lernen.
- Ich empfinde die Bedienung als sehr umständlich.
- Ich habe mich bei der Nutzung des Formular sehr sicher gefühlt.
- Ich musste eine Menge Dinge lernen, bevor ich mit dem Formular arbeiten konnte

Sonstige Anmerkungen:

Positiv:

Negativ:

Verbesserungen:

7.) Abschluss

-> Abschluss des Tests

-> Vielen Dank, wir sind jetzt fertig mit den Aufgaben. Haben Sie abschließend noch Fragen oder weitere Anmerkungen.

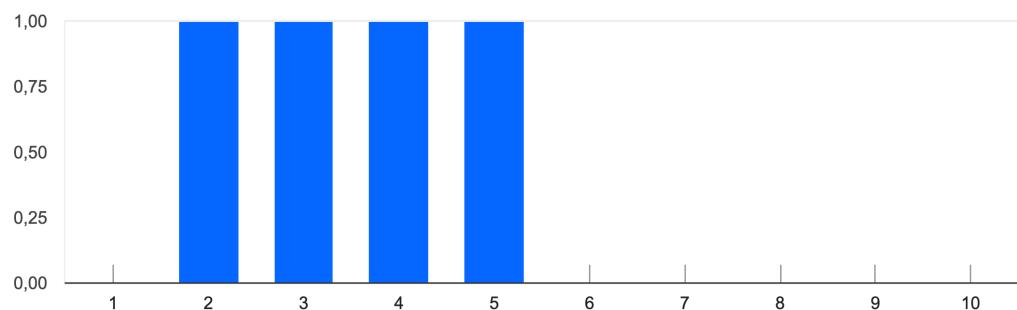
-> Ich bedanke mich sehr für die Teilnahme

-> Aufnahme beenden!

A.6 Auswertung Pre-Test Fragebogen

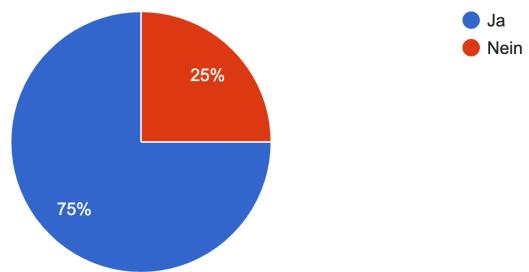
Wie gut sind deine Kenntnisse in Sachen Steuern?

4 Antworten



Machst du deine Steuererklärung selbst?

4 Antworten



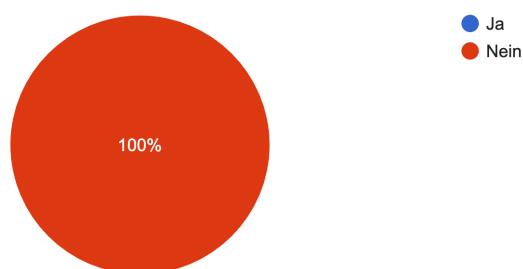
Wenn Ja, wie erledigst du deine Steuererklärung?

3 Antworten



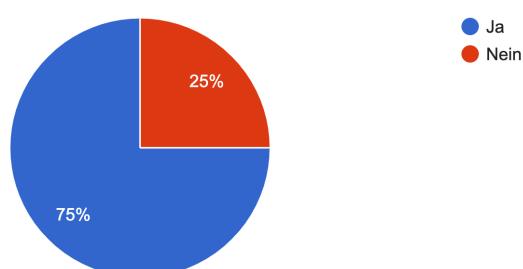
Hast du Erfahrung mit Steuer-Apps? (z.B. taxfix, ...)

4 Antworten



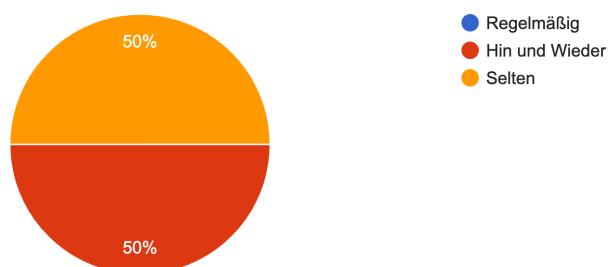
Hast du Erfahrung mit dem Erstellen von Umfragen und Ähnlichem? (z.B. Google Forms, ...)

4 Antworten



Wenn Ja, wie oft erstellst du Umfragen oder Formulare?

4 Antworten



A.7 Auswertung SUS-Fragebögen

Usability-Test

System Usability Scale: Baukasten-System



	Tester 1A	Tester 2B	Tester 3C	Tester 4D	Gesamt
System Usability Scale Fragebogen					
1) Ich kann mir gut vorstellen, das Baukasten-System regelmäßig zu nutzen.	3	4	5	5	
2) Ich empfinde das Baukasten-System als unnötig komplex.	1	1	1	1	
3) Ich empfinde das Baukasten-System als einfach zu nutzen.	4	5	4	5	
4) Ich denke, dass ich technischen Support brauchen würde, um das Baukasten-System zu nutzen.	1	1	2	2	
5) Ich finde, dass die verschiedenen Funktionen des Baukasten-Systems gut integriert sind.	4	5	5	5	
6) Ich finde, dass es im Baukasten-System zu viele Inkonsistenzen gibt.	1	2	2	1	
7) Ich kann mir vorstellen, dass die meisten Leute das Baukasten-System schnell zu beherrschen lernen.	5	5	4	5	
8) Ich empfinde die Bedienung als sehr umständlich.	1	1	2	2	
9) Ich habe mich bei der Nutzung des Baukasten-Systems sehr sicher gefühlt.	3	5	4	5	
10) Ich musste eine Menge Dinge lernen, bevor ich mit dem Baukasten-System arbeiten konnte.	1	1	1	2	
System Usability Score	85	95	85	92,5	89,375

Bachelor-Vortrag von Mona Kögel 28

Usability-Test

System Usability Scale: Formular



	Tester 1A	Tester 2B	Tester 3C	Tester 4D	Gesamt
System Usability Scale Fragebogen					
1) Ich kann mir gut vorstellen, das Formular regelmäßig zu nutzen.	2	5	5	4	
2) Ich empfinde das Formular als unnötig komplex.	1	1	1	1	
3) Ich empfinde das Formular als einfach zu nutzen.	3	5	5	5	
4) Ich denke, dass ich technischen Support brauchen würde, um das Formular zu nutzen.	1	1	1	1	
5) Ich finde, dass die verschiedenen Funktionen des Formulars gut integriert sind.	4	5	4	5	
6) Ich finde, dass es im Formular zu viele Inkonsistenzen gibt.	1	1	1	1	
7) Ich kann mir vorstellen, dass die meisten Leute das Formular schnell zu beherrschen lernen.	5	5	4	5	
8) Ich empfinde die Bedienung als sehr umständlich.	3	1	1	1	
9) Ich habe mich bei der Nutzung des Formulars sehr sicher gefühlt.	4	5	4	4	
10) Ich musste eine Menge Dinge lernen, bevor ich mit dem Formular arbeiten konnte.	1	1	1	1	
System Usability Score	77,5	100	92,5	95	91,25

Bachelor-Vortrag von Mona Kögel 29

A.8 Auswertung des zusätzlichen Feedbacks: Pilot-Test

Sonstige Anmerkungen: Building-Kit

Positives? 1 Antwort

Wenn ich mir das Szenario Sachbearbeiter/Admin im Amt vorstelle, welcher täglich einige Formulare erstellt und/oder bearbeitet/oder aktualisiert, stelle ich mir diese Lösung schon charmant vor einfach einen Link mit dem neuen Formular an den jeweiligen Klienten zu versenden. Spannend wäre es da sicher auch wie die Schnittstellen zu bereits bestehenden System aussehen könnten

Negatives? 1 Antwort

Hab etwas lange gebraucht um den Use Case zu verstehen, bereits vorhandene Elemente wurden nicht gekennzeichnet, USP der Anwendung wurde nicht auf einen Blick klar

Verbesserungsvorschläge? 1 Antwort

Den step-by-step Workflow von links nach Rechts im UI/UX deutlicher herausstellen z.B. durch 1 2 3 oder Pfeile in der Navigation , die Fragezeichen-icons in Infoicons abändern oder direkt als Hinweisfeld anzeigen, Bereits verwende Elemente kennzeichnen, Variation an Elementen deutlicher herausstellen und ohne das man großartig scrollen muss, 1 Satz was der USP der Anwendung ist deutlich lesbar anzeigen

Sonstige Anmerkungen: Formular

Positives? 1 Antwort

Der Wizard hat mir gut gefallen, und die Farbgebung sowie Typo des UIs

Negatives? 0 Antworten

Auf diese Frage liegen noch keine Antworten vor.

Verbesserungsvorschläge? 1 Antwort

Infoicons neben relevanten Fragen anzeigen, Resultscreen am Ende anzeigen

A.9 Auswertung des zusätzlichen Feedbacks: Usability-Test

Sonstige Anmerkungen: Building Kit

Positives? 4 Antworten

schnell zu bedienen

die Icons

- Meine Textänderungen waren direkt bei dem entsprechenden Baukasten sichtbar und ich konnte sehen wie es aussieht.
- Drag&Drop hat sehr einfach und schnell funktioniert
- Ich konnte das Formular gleich für mich testen durch das Eingeben von Daten oder das Klicken der Buttons

Übersichtliche, klare Darstellung

Negatives? 2 Antworten

- Die Spalte 2 war nicht mit Spalte 1 verknüpft

Bei der 1. Anwendung, erweckt es den Eindruck, dass es sehr viele zu befüllende Felder gibt.

Verbesserungsvorschläge? 2 Antworten

Kurze Einleitung / Satz zu den Punkten oder das Wort 'Label' anders nennen / Überschrift

- Spalte 2 und Spalte 1 irgendwie connecten, damit ich die Änderungen direkt neben mir sehe ohne Scrollen, wenn ich bereits Bausteine verschoben habe
- Befehl um Bausteine einfach ans Ende vom Formular schieben zu lassen
- Schieben und Scrolling: Hier kann das Handling für eine finale Version noch abgeändert werden

Sonstige Anmerkungen: Formular

Positives? 3 Antworten

- einheitliches Design, jede Textbox hat den gleichen Rahmen und ähnliches
- Der blaue Rahmen wenn man in einem Feld ist bzw. es noch ausgewählt ist
- Formular an sich war sehr gut aufgebaut

Sehr übersichtlich gestaltet, einfache Handhabung, einfache Führung
einfach zu bedienen

Negatives? 2 Antworten

- Spontan ist mir dabei nichts aufgefallen
- ist nicht ganz ersichtlich, ob beim Auswahl der Icons blau oder grau die Auswahl ist

Verbesserungsvorschläge? 2 Antworten

Eine Übersicht über die bereits ausgefüllten Seiten wäre gut, damit man schnell switchen kann, wenn man zu Beginn zB etwas vergessen hat oder etwas prüfen möchte auf einer vorherigen Seite.

- Felder die nicht editierbar sind, sollten etwas anders aussehen, der Rahmen ist nicht so gut zu erkennen. Ich würde eher eine Einfärbung des Felder bemerken
- Die Farbe von Toggle und Icon Button mit einander abstimmen

A.10 Weiterführende Links

Miro-Board zum Persona-Workshop: https://miro.com/app/board/o9J_IG1kSdo=/
Passwort: ado34567

Miro-Board zu den Usability-Tests: https://miro.com/app/board/o9J_lOOnUHo=/
Passwort: ado34567

Github Repository der Arbeit:

<https://github.com/adorsys-academic/modular-data-collection-system>

Link zum Formular-Prototyp: <https://mdcs-project.cloud.adorsys.de/start>

Link zum Baukasten-Prototyp: <https://mdcs-project.cloud.adorsys.de/building-kit>