

Fakultät für Gestaltung

Studienrichtung Interaktive Medien (B.Sc.)

Eine Methodentoolbox zur Unterstützung von
Human-Centered Designprozessen

Bachelorthesis

von

Simon Hofmeister

Datum der Abgabe: 14.09.2021

Erstprüfer: Prof. Dr. Wolfgang Kowarschick

Zweitprüfer: Prof. Daniel Rothaug

In Zusammenarbeit mit: Iconstorm – Designing the Next

Abstract

In strategischen Designprozessen setzen sich die Grundsätze des Human-Centered Design (HCD) zunehmend durch, was sich in der steigenden Verwendung sogenannter HCD-Methoden deutlich macht. Derartige Methoden müssen bisher analog oder verteilt auf diversen Plattformen vorbereitet, durchgeführt und analysiert werden. In meiner Arbeit konzipiere ich die Grundlage für eine digitale und zentralisierte Anwendung, welche asynchrones Arbeiten ermöglicht und dem Nutzer progressive Erkenntnisse bietet. In einer prototypischen Umsetzung werden die Ergebnisse der Thesis gestützt und getestet.

Danksagungen

AN DIESER STELLE MÖCHTE ICH MICH BEI ALL DENJENIGEN BEDANKEN, DIE MICH WÄHREND DER ANFERTIGUNG DIESER BACHELORARBEIT UNTERSTÜTZT UND MOTIVIERT HABEN.

Zuerst gebührt mein Dank der Agentur iconstorm, allem Vorweg Tim Zeidler und Jochen Denzinger, für die technische sowie inhaltliche Unterstützung.

Ich bedanke mich bei meinen Professoren Dr. Wolfgang Kowarschick und Daniel Rothaug bei der Betreuung des Projekts und für die hilfreiche und konstruktive Kritik.

Ebenfalls möchte ich mich bei Alexandra Manta und Florian Kapaun für die Korrekturlesung und mentale Unterstützung bedanken.

Abschließend danke ich meinen Eltern, die mir mein Studium durch ihre Unterstützung ermöglichten.

Inhaltsverzeichnis

1 Einleitung	1
1.1 Themenfeld	1
1.1.1 Grundsätze des Human-Centered Design	1
1.1.2 Gestaltungsaktivitäten im Human-Centered Design	4
1.1.3 Methoden und Werkzeuge zum Human-Centered Design	8
1.1.4 Human-Centered Design bei iconstorm	10
1.2 Maßnahmen	15
1.2.1 Problembeseitigung	15
1.2.2 Mehrwert	17
1.2.3 Umsetzung	18
2 Recherche	19
2.1 Marktanalyse	19
2.1.1 Methodensammlungen	19
2.1.2 Personagenerator	23
2.1.3 Kollaborationsplattformen	25
2.2 Potenziale	28
3 Konzeption	29
3.1 Personae	29
3.2 Szenarios	32
3.3 System	35
3.3.1 Anforderungsanalyse	35
3.3.2 Use-Case-Diagramm	38
3.3.3 Datenmodell	42
3.3.4 Designsystem	46
3.3.5 Screen-Flow	49

Inhaltsverzeichnis

4 Technische Umsetzung	68
4.1 User Interface	68
4.1.1 Vue 	68
4.1.2 Vuetify 	69
4.2 Datenverwaltung und Kommunikation	70
4.2.1 Strapi 	70
4.2.2 GraphQL 	71
4.2.3 Apollo Client 	72
4.2.4 Node 	73
5 Resümee und Ausblick	74
5.0.1 Zusammenfassung	74
5.0.2 Ausblick	74
5.0.3 Iterativer Prozess	75
6 Literatur	77
7 Erstellungserklärung	80
A Szenarios	81
B Anforderungsanalyse und MoSCoW-Priorisierung	97

Abkürzungsverzeichnis

API Application Programming Interface

CLI Command Line Interface

CMS Content Management System

DIN Deutsches Institut für Normung

HCD Human-Centered Design

UCD User-Centered Design

UI User Interface

Abbildungsverzeichnis

1.1	Wechselseitige Abhangigkeit menschzentrierter Gestaltungsaktivitaten [DIN20] (Zur erleichterten Verstandlichkeit, wurden Nummerierungen aus der Grafik entfernt)	5
1.2	Template der Agentur iconstorm zur Bearbeitung von RAW-Personae	12
1.3	Template der Agentur iconstorm zur Bearbeitung von RAW-Szenarios	13
2.1	Screenshot der Methodensammlung von UCDToolbox	21
2.2	Screenshot der Methodensammlung von Design Kit	22
2.3	Screenshot des Personagenerators von UXPressia	24
2.4	Screenshot eines Persona-Templates in Miro	26
2.5	Screenshot der Szenarios in Notion	27
3.1	Persona eines Konzepters	30
3.2	Persona einer Visual Designerin	30
3.3	Persona eines Auszubildenden im Design	31
3.4	Persona eines IT-Leiters	31
3.5	Auszug aus den Szenarios	34
3.6	Anforderungsanalyse des Methodenpanels	38
3.7	Use-Case-Diagramm der Anwendung	39
3.8	ER-Diagramm des Datenmodells	43
3.9	Grafische Darstellung des Beispiels	45
3.10	Integrierte Farbpalette	48
3.11	Wahl der Typografie	49
3.12	Dashboard	50
3.13	Kundenpanel	52

Abbildungsverzeichnis

3.14 Projektpanel I	53
3.15 Projektpanel II	55
3.16 Methodenpanel Szenarios	56
3.17 Ausführungspanel Szenario	58
3.18 Methodenpanel Personae	61
3.19 Ausführungspanel Personae	63
3.20 Features zur Kollaboration	65
4.1 Integrierter Tech-Stack	68
4.2 Screenshot der Dokumentation von Vuetify	69
4.3 Screenshot einer Collectionverwaltung in Strapi	70

1 Einleitung

In Kooperation mit dem Unternehmen iconstorm entstand die Idee einer Plattform, auf welcher diverse Usability- und Gestaltungsmethoden genutzt und verwaltet werden können.

Der Betrieb hat sich auf die Arbeit mit HCD spezialisiert und betreut verschiedenste Projekte mit diesem Ansatz. Um dafür eine optimale Entwicklung zu garantieren und eine Grundlage für Designprozesse zu schaffen, wird eine Vielzahl von Methoden genutzt und weiterentwickelt.

1.1 Themenfeld

Um zu verstehen, welchen Zweck diese Arbeit erfüllt, muss erst einmal geklärt werden, worum es sich bei HCD eigentlich handelt. Im Folgenden gebe ich daher einen Überblick über die wichtigsten Auszüge der Deutsches Institut für Normung (DIN) EN ISO 9241-210: „Ergonomie der Mensch-System-Interaktion – Menschzentrierte Gestaltung interaktiver Systeme“, welche genau diese Thematik behandelt.

1.1.1 Grundsätze des Human-Centered Design

HCD, oder auch „Menschzentrierte Gestaltung“ ist ein weitverbreiteter Gestaltungsansatz und Teil der Mensch-System-Interaktion. Er folgt laut der DIN sechs klar vorgeschriebenen Grundsätzen:

1. Die Gestaltung beruht auf einem umfassenden Verständnis der Benutzer, Aufgaben und Arbeitsumgebungen (siehe 1.1.1.1)

2. Die Benutzer sind während der Gestaltung und Entwicklung einbezogen (siehe 1.1.1.2)
3. Das Verfeinern und Anpassen von Gestaltungslösungen wird fortlaufend auf der Basis benutzerzentrierter Evaluierung vorangetrieben (siehe 1.1.1.3)
4. Der Prozess sieht Iterationen vor (siehe 1.1.1.4)
5. Bei der Gestaltung wird die gesamte User Experience berücksichtigt (siehe 1.1.1.5)
6. Das Gestaltungsteam vereint fachübergreifende Kompetenzen und Gesichtspunkte (siehe 1.1.1.6)[DIN20]

Wenn diese Grundsätze eingehalten werden, können beispielsweise Kosten für Betreuung und Beratung von Kunden verringert werden, da diese das Produkt einfacher verstehen. Die User Experience wird verbessert und die Qualität steigt.

1.1.1.1 Die Gestaltung beruht auf einem umfassenden Verständnis der Benutzer, Aufgaben und Arbeitsumgebungen

Wie der Name schon sagt, steht der Mensch im Mittelpunkt des HCD-Prozesses. Deshalb ist es wichtig, dass alle Nutzer und Stakeholder¹ ermittelt werden. Da jeder Benutzer ein bestimmtes Ziel oder Aufgabe bei der Interaktion mit einem Produkt hat, ist es elementar, dessen Kontext zu verstehen, um eine schlechte User Experience zu vermeiden. So kann beispielsweise eine Anwendung ein optisch herausragendes Interface besitzen, allerdings unbrauchbar werden, wenn sie von Personen mit einer Sehbehinderung bedient werden muss.

1.1.1.2 Die Benutzer sind während der Gestaltung und Entwicklung einbezogen

Die im vorherigen Grundsatz definierten Nutzer sollen aktiv in den Gestaltungsprozess involviert werden. Das kann durch „Mitwirkung an der Gestaltung, als Quelle für relevante Daten oder bei der Evaluierung von Lösungen“[DIN20] geschehen. Die Benutzer, welche voraussichtlich mit dem künftigen System arbeiten

¹ Stakeholder: „Person, für die es aufgrund ihrer Interessenlage von Belang ist, wie ein bestimmtes Unternehmen sich verhält (z. B. Aktionär, Mitarbeiter, Kunde, Lieferant)“[Dudc]

und somit die relevante Zielgruppe bilden, nehmen also direkten Einfluss auf die Gestaltung und Entwicklung des Produkts.

1.1.1.3 Das Verfeinern und Anpassen von Gestaltungslösungen wird fortlaufend auf der Basis benutzerzentrierter Evaluierung vorangetrieben

Das Einbeziehen der Benutzer hat erst dann einen Mehrwert, wenn auch deren Evaluierung der Arbeit im Prozess berücksichtigt wird. Nur ein Nutzer, der bei einer schlechten Erfahrung Rückmeldung gibt, kann dafür sorgen, dass die Gestaltung angepasst und optimiert werden kann.

1.1.1.4 Der Prozess sieht Iterationen vor

Fehler in der Gestaltung können nicht auf Anhieb beseitigt werden. Deshalb ist eine fortwährende Iteration² von enormer Signifikanz. Dies bezieht auch die bisher genannten Grundsätze mit ein, da Benutzer auch in einem späteren Stadium eines Projekts neue Erkenntnisse erhalten können.

1.1.1.5 Bei der Gestaltung wird die gesamte User Experience berücksichtigt

Es ist darauf zu achten, dass das Produkt nicht nur den Anspruch hat, lediglich die gewünschte Aufgabe ausführen zu können. Zur Gebrauchstauglichkeit gehören deutlich mehr Aspekte wie die Instandhaltung oder die unterstützende Betreuung (z. B. das Anfertigen einer Benutzerdokumentation oder das Bereitstellen einer Online-Hilfe). Beginnen kann die User Experience bereits mit der Produktverpackung oder der Installation, welche den Ersteindruck bildet.

Auch der Grad der Automatisierung soll aufgrund der „Stärken, Einschränkungen, Vorlieben und Erwartungen der Benutzer“[DIN20] definiert werden. Es wird also vom Nutzer abhängig gemacht, welche Aufgaben er selbst übernehmen soll und welche ihm das System abnimmt.

Die Sicherheit und Zufriedenstellung der Benutzer sollte zu jeder Zeit im Mittelpunkt stehen („sowohl kurzfristig, z. B. Behaglichkeit und Vergnügen, als auch

² „Iteration bezeichnet in diesem Kontext das Wiederholen einer Folge von Schritten, solange bis das gewünschte Ergebnis erzielt wurde“[DIN20]

langfristig, z. B. Gesundheit, Wohlbefinden und Arbeitszufriedenheit“[DIN20]).

1.1.1.6 Das Gestaltungsteam vereint fachübergreifende Kompetenzen und Gesichtspunkte

Es ist nicht von zentraler Bedeutung, dass ein Team im Gestaltungsprozess von großem Ausmaß ist, allerdings sollte es den nötigen Kompetenzbereichen entsprechend vielfältig besetzt sein. So sind nicht nur Benutzer und Stakeholder wichtig für die Zusammensetzung der Arbeitsgemeinschaft. Beispielsweise können auch Personen aus den Bereichen Arbeitswissenschaft, Verkauf, Wartung, Gesundheit oder Benutzerbetreuung einen Mehrwert bieten.

Durch die Erweiterung der Fachkompetenzen im Team wird die Kreativität gefördert und eine umfassende Kompetenzbasis gebildet.

1.1.2 Gestaltungsaktivitäten im Human-Centered Design

Wurde entschieden, dass die menschzentrierte Entwicklung angewandt wird, müssen vier miteinander verbundene Gestaltungsaktivitäten unter Zuhilfenahme verschiedenster Methoden durchgeführt werden.

1. Verstehen und Beschreiben des Nutzungskontexts (siehe 1.1.2.1)
2. Spezifizieren der Nutzungsanforderungen (siehe 1.1.2.2)
3. Erarbeiten von Gestaltungslösungen (siehe 1.1.2.3)
4. Evaluieren der Gestaltung (siehe 1.1.2.4)[DIN20]

Die vier Aktivitäten werden iterativ durchgeführt und mit jeder Wiederholung werden tiefere Einsichten gewonnen. Im Prinzip entsprechen sie den allgemeinen Phasen der Gestaltung und Entwicklung.

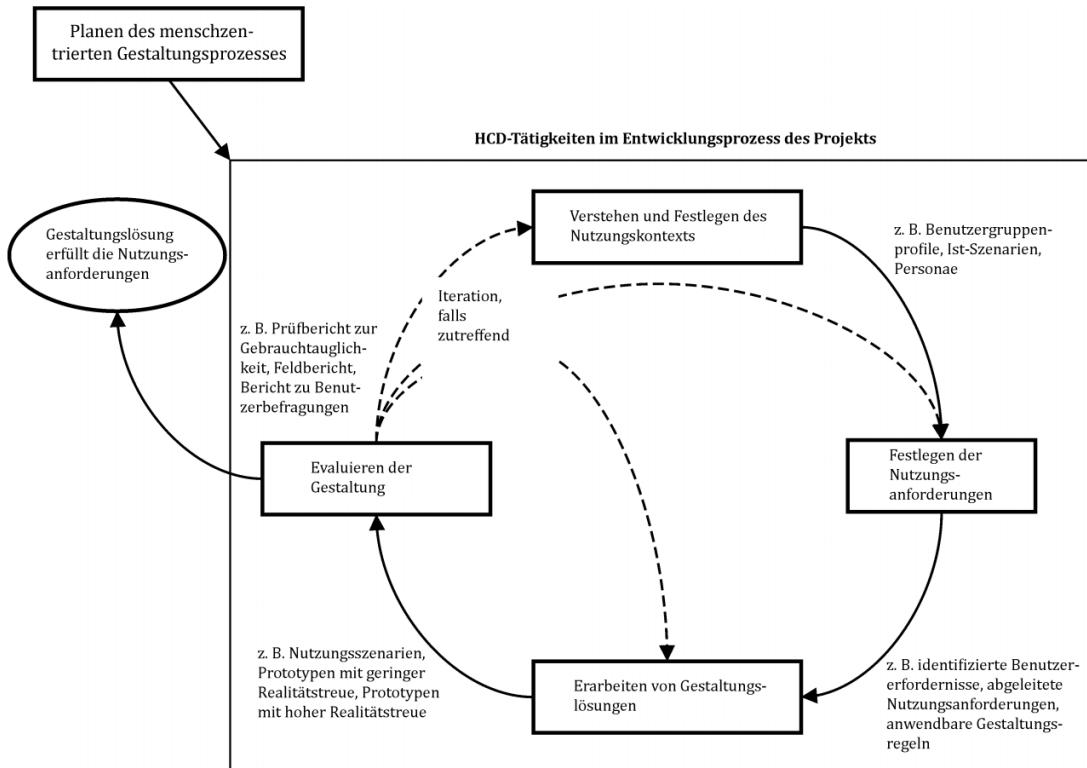


Abbildung 1.1 Wechselseitige Abhängigkeit menschzentrierter Gestaltungsaktivitäten [DIN20]
(Zur erleichterten Verständlichkeit, wurden Nummerierungen aus der Grafik entfernt)

Der Startpunkt des Prozesses ist dabei nicht vorgeschrieben. Wird eine Agentur beispielsweise beauftragt, eine vorhandene Anwendung zu überarbeiten, welche Mängel in der User Experience aufweist, kann durchaus mit der vierten Phase („systematischen Evaluierung des Systems“, siehe 1.1.2.4) begonnen werden. Der Ablauf ist dennoch iterativ und kann meist nicht mit Abschluss der vierten Phase als „fertig“ angesehen werden. Auch den Startpunkt auf die zweite („Festlegen der Nutzungsanforderungen“, siehe 1.1.2.2) oder dritte Phase („Erarbeiten von Gestaltungslösungen“, siehe 1.1.2.3) zu versetzen kann unter den Gesichtspunkten sinnvoll sein, wenn bereits Informationen über die vorherigen Phasen vorhanden und dokumentiert sind.

1.1.2.1 Verstehen und Beschreiben des Nutzungskontexts

Hier wird versucht im Sinne des ersten Grundsatzes (siehe 1.1.1.1) Informationen über relevante Benutzer und sonstige Stakeholder, sowie deren Ziele und Aufgaben zu sammeln.

Für einen brauchbaren Fortschritt ist es wichtig, dass die relevanten Merkmale der Benutzer bzw. Benutzergruppen identifiziert werden. Diese können beispielsweise Alter, Kenntnisse, Kompetenzen, Erfahrung oder Art der Ausbildung sein. Eine klassische Methode, welche diesen Vorgang unterstützt, sind Personae. Aber auch kontextuelle Interviews oder Feldbeobachtungen sind ein valides Mittel zum Erfolg.

Um die Anforderungen im folgenden Schritt definieren zu können, müssen auch die Ziele und Aufgaben aller Nutzer festgelegt werden. Hierzu werden oft Szenarios erstellt, um die Denkweise der unterschiedlichen Nutzer zu verstehen und die Erkenntnisse zu dokumentieren.

Beispielmethoden: Personae (siehe 1.1.3.1), Szenarios (siehe 1.1.3.2)

1.1.2.2 Spezifizieren der Nutzungsanforderungen

Die Nutzungsanforderungen werden von der ersten Phase abgeleitet und dürfen in keinem Widerspruch zu dieser stehen. Wichtig beim Spezifizieren der Nutzungsanforderungen ist, dass klargestellt wird *was* die Benutzer erreichen wollen und nicht *wie* dies ermöglicht wird.

Damit der iterative Ablauf nicht gestört wird, müssen sich die Nutzungsanforderungen über das Projekt hinweg abändern lassen können. Zudem muss die Spezifikation „so abgefasst sein, dass sie eine nachfolgende Prüfung ermöglicht[...][,] durch die relevanten Stakeholder verifiziert [...] [und] in sich widerspruchsfrei sein“[DIN20] kann.

Beispielmethoden: Critical Incident Technique, Illustrative/ informative Modelle

1.1.2.3 Erarbeiten von Gestaltungslösungen

Auf Grundlage der bisherigen Ergebnisse können nun Gestaltungslösungen erarbeitet werden. Hier wird deutlich, wie gewissenhaft und (im Falle einer Übergabe) verständlich die vorherigen Phasen dokumentiert und aufbereitet wurden.

Die Gestaltung beschäftigt sich dabei mit zwei Aspekten:

Gestalten der Aufgaben und Interaktion zwischen Benutzer und System

Hier wird konkretisiert, wie der Nutzer Aufgaben mit dem System ausführt. Beispielsweise wird die Art des Mediums definiert (z. B. grafische Darstellung oder Text). Die detaillierte Beschreibung der optischen Erscheinung findet noch keine Beachtung.

Gestalten der Benutzungsschnittstelle

Hierfür gibt es eine umfangreiche Liste an Methoden. „Doch egal, welche Methode angewandt wird, zwei hauptsächlichen Regeln wird immer Folge geleistet:

- Mehrere Ideen sollten generiert werden. [...]
- Man sollte kreativ sein, ohne irgendwelche Einschränkungen zu berücksichtigen.“[Nor13]

Die Art der Ausarbeitung kann, je nach Fortschritt des Projekts, von Skizzen bis hin zur Anfertigung eines Prototypen inklusive ausformulierter Dokumentation reichen.

Beispielmethoden: Card Sorting, Prototyping

1.1.2.4 Evaluieren der Gestaltung

Hier werden die Stärken und Schwächen des Konzepts herausgearbeitet. Denn auch wenn in der Theorie viele Aspekte durchdacht werden, finden sich in der realen Nutzung eines Produkts, Systems oder einer Dienstleistung Schwachstellen. In der benutzerdefinierten Evaluierung können beispielsweise „neue Informationen über Benutzererfordernisse [...] [gesammelt oder] Vergleiche zwischen Gestaltungsalternativen durchgeführt werden“[DIN20].

Zudem bietet eine gute Evaluierung Lösungsvorschläge, welche für das Gestaltungsteam nützlich werden können.

Eine Evaluierung kann in jedem Stadium des Projekts durchgeführt werden.

So kann schon in einem frühen Zeitraum in diese Phase übergegangen und Szenarios oder Skizzen durch Benutzer bewertet werden. Da der Verlauf iterativ stattfindet, muss auch keine Befürchtung bestehen, bereits ein fertiges Produkt nach der Evaluierung abliefern zu müssen. Im späteren Stadium des Projekts können die Tests dann detaillierter werden und bis hin zu sogenannten Beta³-Prüfungen führen.

Doch auch ohne reale Benutzer ist eine Überprüfung der Anwendung möglich: in einer inspektionsbasierten Evaluierung wird auf die Erfahrung von Fachleuten des Gebiets gebaut, welche sich in die Rolle eines Nutzers begeben. Diese Methode ist zwar kostengünstiger, erzielt allerdings i. d. R. schwächere Ergebnisse.

Bei einem optimalen Evaluierungsprozess werden zudem Langzeitbeobachtungen angefertigt, welche wertvolle Resultate über das aktuelle aber auch künftige Projekte liefern können.

Beispielmethoden: Prüfung mit Benutzern, Inspektionsbasierte Evaluierung

1.1.3 Methoden und Werkzeuge zum Human-Centered Design

Im letzten Kapitel konnten wir schon einen Einblick erhalten, welche Methoden für die einzelnen Phasen nützlich sein können. Die DIN SPEC 91328 („Ressourcen-schonende Anwendung von Methoden und Werkzeugen zur menschenzentrierten Gestaltung gebrauchstauglicher interaktiver IT-Systeme“) gibt einen detaillierten Überblick über viele dieser Methoden.

Im Folgenden möchte ich auf die beiden Methoden eingehen, welche in der Konzeption meines Projekts eine erhebliche Rolle spielen und auch im weiteren Verlauf des Dokuments vermehrt wiederkehren werden. Beide Methoden gehören hauptsächlich der ersten Phase (siehe 1.1.2.1) an.

³ Betaversion: „neues bzw. überarbeitetes Software- oder Hardwareprodukt im zweiten Entwicklungsstadium“[Duda]

1.1.3.1 Personae

Personae „sind Steckbriefe von Benutzern und repräsentieren diese mit ihren Wünschen, Zielen und weiteren Eigenschaften im Entwicklungsprozess“[DIN16]. Im Grunde stellt eine Persona eine Gruppe von Benutzern mit ähnlichen Verhaltensmustern in abstrakter Form dar. Dies können beispielsweise Motivationen, Präferenzen bei der Produktnutzung oder auch Fähigkeiten sein.

Da Personae oft genutzt werden, sind Ergebnisse und Vorlagen in jeglicher Form und Variation vorzufinden. Weil kaum ein Projekt dem anderen gleicht, sind dementsprechend unterschiedliche Eigenschaften einer Persona vom Nutzen. So ist es beispielsweise bei der hier präsentierten Arbeit nicht vom Wert, welches monatliche Einkommen ein Nutzer bezieht. Diese Information würde allerdings eine erhebliche Rolle für eine Firma spielen, welche ein Produkt mit einem Aboservice plant.

Die breite Masse an Variationen finden auch in der DIN SPEC Rücksicht. So wird nur festgesetzt, dass eine Persona „mindestens das Bild der repräsentierten Person sowie den Vor- und Nachnamen“[DIN16] beinhalten muss.

Es ist nicht vorgeschrieben, in welcher Phase die Methode verwendet werden muss, da sie zu jeder Zeit den Prozess unterstützen kann. Den größten Merkwert bietet sie allerdings „insbesondere in den ersten beiden anfänglichen Phasen“[DIN16](siehe 1.1.2.1 und 1.1.2.2).

1.1.3.2 Szenarios

„Ein Nutzungsszenario (oder Use Case) ist die narrative Beschreibung der (gedachten) Interaktion eines (potenziellen) Benutzers mit einem interaktiven IT-System“[DIN16].

In Szenarios werden die Handlungsziele- und aufgaben der Benutzer definiert. Wie aus der Definition bereits hervorgeht, kann diese Methode oft mit den zuvor beschriebenen Personae verknüpft werden. Genau diese Verknüpfung wird im Laufe des Projekts detaillierter untersucht. Dabei spielt keine Rolle, welche der

beiden Methoden zuerst ausgeführt wird - sie können parallel entstehen und auch weiterentwickelt werden. Dies ist am einfachsten mit einem Beispiel zu erklären:

Angenommen *Persona A* wurde zu Beginn des Prozesses definiert. In einem fiktiven Szenario wird nun beschrieben, wie *Persona A* beim Ausführen einer Aktion vorgehen möchte. *Persona A* muss in diesem Szenario in Kontakt mit einer neuen Persona treten, welche bisher nicht definiert wurde: *Persona B*. Da *Persona B* für den Ablauf eine erhebliche Rolle spielt, muss diese nun ausgearbeitet werden.

Die Darstellungsform der Szenarios ist nicht festgeschrieben. „Neben der textlichen Form sind (gezeichnete) Storyboards gebräuchlich, aber auch Flussdiagramme können genutzt werden“[DIN16].

Ein typisches Szenario in textlicher Form bildet beispielsweise „Anmelden beim Online-Dienst“[DIN16], wobei hier noch wenig Informationen für die weitere Bearbeitung bereit gehalten werden.

Oftmals ist es von Vorteil eine Ausformulierung des Templates „Als [Rolle] möchte ich [Wunsch], um [Nutzen]“[Bau14] auszuarbeiten. Auch detailliertere Formulierungen, verschiedene Lösungsansätze oder eventuelle technische Ansätze können an die Szenarios angehängt werden.

Szenarios sind ein äußerst variables Mittel, um schnell kreative Ideen im Gestaltungsprozess zu erhalten und den Nutzungskontext zu verstehen (siehe 1.1.2.1).

1.1.4 Human-Centered Design bei iconstorm

Als ich vor knapp einem halben Jahr als Werkstudent zu iconstorm gestoßen bin, bekam ich meinen ersten Einblick in die Prozesse von HCD. Auch wurde mir erklärt, wie sich Vorgehen der Agentur in Zusammenarbeit mit Projektpartnern definiert. I. d. R. hält die Agentur dazu einen Workshop ab, um die Zielgruppe und die Interessen der Stakeholder besser kennenzulernen. Dafür werden verschiedenste Methoden verwendet, welche über Jahre hinweg genutzt und auf die

eigenen Ansprüche angepasst und erweitert wurden.

1.1.4.1 Vorbereitung

Für die Vorbereitung auf den Workshop ist ein Konzepter oder Strategic-Designer zuständig. Er wählt Methoden aus, welche er für das Projekt als passend empfindet. Ist seine Vorauswahl getroffen, bereitet er diese projektspezifisch vor. Je nach Methode muss er dafür die entsprechenden Templates anpassen oder neue erstellen, um den Anforderungen gerecht zu werden.

Die Workshops finden unter normalen Umständen analog bei einer Zusammenkunft statt, weshalb die ausgearbeiteten Vorlagen mehrfach ausgedruckt und für eine Bearbeitung vorbereitet werden. Da die detaillierte und iterative Ausführung der Methoden die zeitliche Begrenzung eines Workshops überschreiten würde, werden hier sogenannte „RAW-Versionen“⁴ gefertigt:

⁴ RAW (deutsch: roh, unbearbeitet, grob): Die RAW-Versionen sind nicht detailliert, sondern dienen als Grundlage für einen Detaillierungsprozess.

1 Einleitung

The template consists of a grid of fields for defining a persona. At the top, there's a section for a 'My picture' sketch ('Just draw.') and a 'Personal statement' ('Just draw.'), followed by demographic information: Name, Age, Status, and Job Title. Below this, the grid is organized into three columns: 'Mindset' (with a note to put keywords here), 'Social behaviour' (with a note to put keywords here), and 'Skills' (with a note to put keywords here). Further down, there are two more columns: 'My jobs to be done' (with a note about symptoms and causes) and 'I follow' (with examples like people, influencers, media). To the right of these is 'I like' (with examples like hobbies, events, brands, products). At the bottom, there's a column for 'Computer / internet / technology' (with a note about devices and experience) and a final row for 'What's in my bag/car/flat...' (with a note to list typical objects).

Abbildung 1.2 Template der Agentur iconstorm zur Bearbeitung von RAW-Personae

Eine Vorlage für eine RAW-Persona enthält Platz für eine Skizze („Bild der Persona“) und den textlichen Inhalt der vordefinierten Felder, die für das Projekt von Bedeutung sind (z. B. „Name“, „Alter“ oder „Berufsbezeichnung“).

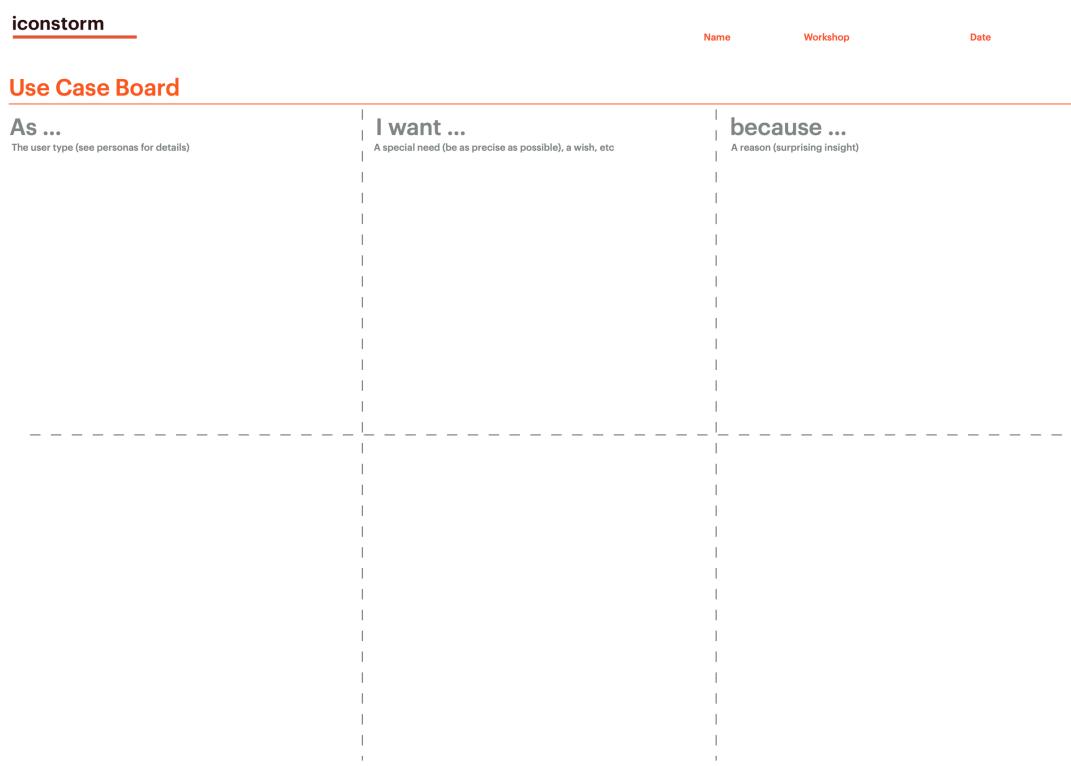


Abbildung 1.3 Template der Agentur iconstorm zur Bearbeitung von RAW-Szenarios

Äquivalent dazu umfasst eine Vorlage für ein RAW-Szenario Felder, in denen die Rolle (z. B. eine Persona), dessen Wunsch und der Nutzen davon eingetragen werden können. Dies entspricht im Prinzip dem Template, welches im Zusammenhang mit der Methode bereits thematisiert wurde (siehe 1.1.3.2).

Im Vorlauf eines Workshops wird meist ein Briefing⁵ mit allen Stakeholdern einberufen, um die Rahmendaten des Projekts zu klären und die geplanten Methoden grob zu erläutern.

1.1.4.2 Workshop

Unter der Leitung des Konzepters werden in Zusammenarbeit mit den Stakeholdern die Vorlagen ausgefüllt. Dabei werden die Methoden bestmöglich iterativ bearbeitet, um optimale Ergebnisse zu erreichen.

Bei Problemen oder Fragen stehen hierbei Mitglieder der Agentur zur Verfügung,

⁵ Briefing: „kurzes Informationsgespräch; Einweisung in etwas“[Dudb]

um Hilfestellungen zu geben. Diese machen sich im Verlauf auch Notizen, um die optimale Weiterverarbeitung der gesammelten Daten zu garantieren.

1.1.4.3 Weiterverarbeitung

Nachdem der Workshop abgehalten wurde, werden die Ergebnisse digitalisiert und aufgewertet. Der Konzepter beschäftigt sich dann damit, den Prozess voranzutreiben und die RAW-Versionen zu detaillieren. Die Personae erhalten also beispielsweise ein echtes Profilbild, gestaltete Reiter und optisch aneinander angepasste Felder. Auch fehlende Informationen werden ergänzt. Die Szenarios werden ausformuliert, geordnet und erste Varianten und technische Ansätze werden beigefügt.

Wie aus der Definition der HCD-Methoden hervorgeht, spielt die anhaltende Kommunikation eine entscheidende Rolle beim Erfolg des Projekts. So bleiben die Projektpartner weiterhin im Arbeitsprozess involviert. Auch sie können fortlaufend Änderungen an den ausgeführten Methoden vorschlagen. Sofern sie die fachliche Kompetenz besitzen, können sie diese sogar selbst vornehmen.

1.1.4.4 Übergabe

Ist das Projekt nun in einem fortgeschrittenen Stadium, werden die Ergebnisse an die Designabteilung übergeben, welche sich mit der Erarbeitung von Gestaltungslösungen befasst (siehe 1.1.2.3). Diese baut darauf, dass die zuvor gesammelten Daten gewissenhaft verwaltet und aufbereitet wurden.

Im Anschluss findet die Evaluierung der Gestaltungsergebnisse (siehe 1.1.2.4) statt und der Kreislauf des Entwicklungsprozesses startet von Neuem (siehe Abbildung 1.1). Im Normalfall wird nicht erneut ein Workshop abgehalten, weil dies dem Ertrag gegenüber einen unverhältnismäßigen Aufwand bereiten würde. Allerdings können die Methodenergebnisse jederzeit überarbeitet und detailliert werden. Dieser Vorgang wird so lange wiederholt, bis ein Ergebnis erzielt wird, welches die Nutzungsanforderungen erfüllt.

1.2 Maßnahmen

Meine Anwendung legt den Grundstein für eine digitale Plattform, welche bestehende Probleme der Agentur im Umgang mit HCD-Methoden behebt und zusätzliche Features integriert. Die Webanwendung bietet die Möglichkeit, mehrere Kunden sowie Projekte und deren Stakeholder zu verwalten. Innerhalb der Projekte können Workshops definiert werden, welche wiederum Methoden (inkl. der Option zur Vorbereitung, Ausführung, iterativen Weiterverarbeitung und Analyse) beinhalten.

1.2.1 Problembeseitigung

Als ich den eben beschriebenen Einblick in die Arbeitsvorgänge der Agentur iconstorm in Zusammenhang mit HCD-Methoden erhielt, wurde mir klar, dass viele der Arbeitsschritte oft unkoordiniert sind. Dies bringt einige Nachteile mit sich, welche ich mithilfe dieses Projekts beheben möchte.

1.2.1.1 Ordnung

Die Unordnung ist ein ständiges Problem, welches die Agentur beim Bearbeiten der Methoden bereits vor Beginn eines Workshops begleitet. Die Aufbereitung der Vorlagen auf verschiedensten Plattformen wie Miro (siehe: 2.1.3.1) oder Notion (siehe: 2.1.3.2) zeugt von Ineffizienz. Der Ursprung liegt darin, dass keine dieser Plattformen eine Ideallösung bildet, welche die Anforderungen aller benötigten Methoden abdecken. Auch einzelne Generatoren für diese Methoden (vgl. 2.1.2) sorgen dafür, dass die Ausführungen an vielen unterschiedlichen Orten gesammelt werden. Dieses Durcheinander resultiert nicht nur in Frustration bei der Anpassung der Vorlagen, sondern birgt das Risiko, dass ganze Methoden im Zuge der Vorbereitung auf den Workshop übersehen oder vergessen werden.

Doch auch während der Durchführung der Workshops, bei der Analyse der gesammelten Daten oder bei der Übergabe an das Designteam entsteht ein wachsendes Sammelsurium an analogen Arbeiten. Spätestens da müssen die Daten in ZIP-Ordnern o. Ä. gesammelt werden, um die Unordnung nicht an die nächste Abteilung weiter zu geben.

Meine zentralisierte Plattform ist in sich schlüssig, sodass kein Stakeholder während der Bearbeitung der Methoden auf eine weitere Anwendung ausweichen muss.

Der Konzepter kann so die komplette Vorbereitung eines Workshops innerhalb des Systems durchführen. Dazu gehört die Auswahl der Methoden und die Aufbereitung der Templates. Da sämtliche Daten digital gesammelt werden, benötigt es keine weiteren Arbeitsschritte, wie beispielsweise das Ausdrucken von Vorlagen.

Jeder Nutzer bekommt lediglich die Teile der Plattform präsentiert, welche für ihn zur Bearbeitung eine Rolle spielen. So kann sich der Workshopteilnehmer weiterhin auf die Ausführung einer Methode konzentrieren, ohne die umfassende Auswahl an weiteren Methoden, Projekten oder Kunden zu sehen, welche der Konzepter benötigt. Auch das Sammeln von Ergebnissen für die Designabteilung ist obsolet, da ein einfacher Zugang für den Designer ausreicht um alle Einträge geordnet aufzurufen und den aktuellen Stand zu begutachten.

1.2.1.2 Iterationen

Das analoge Ausfüllen der Templates bringt den großen Nachteil mit sich, dass das iterative Arbeiten erschwert wird. Einzelne Texte können zwar durchgestrichen und ergänzt werden, allerdings führt dies schnell zu einem unübersichtlichen Ergebnis. Die derzeitige Lösung der Agentur ist es deshalb, nach Durchführung des Workshops die gefüllten Blätter zu sammeln und anschließend zu digitalisieren. So können die Daten immer wieder aufs neue aufbereitet und überarbeitet werden.

Dieser Schritt wird in meiner Anwendung nicht mehr benötigt, da die Arbeiten von Beginn an in digitaler Form vorliegen. Die Methoden können von jedem Nutzer (mit der entsprechenden Berechtigung) bearbeitet und weiterentwickelt werden. Dafür muss der Workshop noch nicht einmal zu Ende sein. Bereits wäh-

rend der Zusammenkunft können die Methoden neu aufbereitet und überarbeitet werden. Der iterative Prozess wird gewährleistet, womit einem der wichtigsten Grundsätze des HCD-Ansatzes Folge geleistet wird.

1.2.1.3 Informationen und Asynchrones Arbeiten

Während des Zeitraums des Workshops müssen dauerhaft Experten der Agentur vorhanden sein, die bei einem möglichen Erinnerungsverlust über den Ablauf einer Methode unterstützend zur Seite stehen können. Diese Methode muss wieder neu erklärt werden und stört eventuell den Arbeitsfluss anderer Stakeholder.

Zudem benötigt man für die physische Präsenz von allen Stakeholdern einen organisatorischen und finanziellen Mehraufwand gegenüber der digitalen Ausführung. Gerade in Zeiten von Corona bemerkte die Agentur, dass Treffen mit einer großen Teilnehmerzahl Probleme hervorrufen können.

Über meine Plattform ist es nicht mehr relevant, dass ein Experte in direktem Kontakt mit einem Nutzer steht, da sämtliche Informationen zur Ausführung einer Methode jederzeit aufgerufen werden können. So werden Kosten für eine Zusammenkunft gespart und jeder Stakeholder kann zeitlich und räumlich unabhängig voneinander arbeiten.

1.2.2 Mehrwert

Die Anwendung beseitigt nicht nur bestehende Probleme, sondern integriert auch neue Features. HCD-Methoden sind, wie in der Einleitung bereits geschildert, häufig miteinander kombinierbar. Diesen Mehrwert der Kombination habe ich nur ansatzweise in vergleichbaren Anwendungen gefunden, weshalb auch hier mit analogen Mitteln nachgeholfen werden muss. Bei meinem zentralisierten System hat der Nutzer neue Möglichkeiten Verknüpfungen verschiedenster Ausführungen zu erzeugen und daraus Informationen zu generieren. So sieht der Benutzer schnell, welche Ausführungen erneut bearbeitet werden müssen und erlangt in kürzester Zeit neue Einblicke um Ideen zu kreieren. Auch kann er mithilfe einer neuen Funktionalität die Relevanz einer Ausführung oder Verknüpfung

verdeutlichen, um die Integrierung von Features zu priorisieren.

1.2.3 Umsetzung

Bei der Erarbeitung des Konzepts habe ich mich auf die beiden Methoden Personae und Szenarios fokussiert, welche bereits in der Einleitung analysiert wurden. Diese Methoden eignen sich gut, um die Vorteile einer geschlossenen Plattform aufzuzeigen. Die Auswahl dieser Methoden hat folgende Beweggründe:

1.2.3.1 Vorkenntnisse

Aufgrund der Tatsache, dass ich die Methoden selbst bereits vor Start des Projekts verwendet hatte, benötigte ich eine kürzere Einarbeitungszeit. Ich konnte meine Recherche detaillierter angehen und mich so tiefer in die Materie einarbeiten.

1.2.3.2 Eigene Verwendung

Dadurch, dass die Methoden beide in einer frühen Phase eines Projekts verwendet werden, konnte ich diese selbst für die Konzeptarbeit meiner Anwendung nutzen. So habe ich andere Plattformen anhand der eigenen Arbeit analysieren und negative Aspekte erkennen können.

1.2.3.3 Präsentation des Mehrwerts

Wie bereits in der Einleitung erklärt, sind Personae und Szenarios miteinander kombinierbar. Um den Mehrwert der Verknüpfungen zu präsentieren eignen sich diese beiden Methoden also besonders.

2 Recherche

2.1 Marktanalyse

Bereits im frühen Stadium meiner Recherche stellte ich fest, dass keine Anwendung aufzufinden ist, welche die vollen Möglichkeiten ausschöpft. Auf eine Kombination aus Methodensammlung, deren kollaborativer Ausführung, sowie die anschließende Analyse konnte ich also nicht antreffen. Lediglich die einzelnen Ansätze sind vorhanden:

2.1.1 Methodensammlungen

Sammlungen, welche sich mit den Usability- und Gestaltungsmethoden beschäftigen, sind sowohl digital als auch analog umfangreich vorhanden. Bei der Umsetzung bauen alle auf diverse Ansätze und Erfahrungswerte, weshalb sich die Sammlung der Methoden immer unterscheidet. Auch die Einteilung dieser in die verschiedenen Phasen ist auf Grundlage unterschiedlicher Ansichten differenziert.

2.1.1.1 UCDtoolbox

Die „UCDtoolbox“ wurde vom niederländischen UX-Designer Tristan Weevers mit seinem eigenen Unmut über aktuelle Methodensammlungen gestartet.“[Weea] Für seine Arbeit suchte er sich Unterstützung dreier Universitäten und diverser internationaler Unternehmen, welche sich im Bereich des User-Centered Design (UCD)¹ spezialisiert haben.

Genauso wie ich war auch Tristan von der Masse an Methoden überschlagen,

¹ Da sich die Ansätze von HCD und UCD (also menschzentrierter und benutzerzentrierter Gestaltung) ähneln, wird in dieser Arbeit auf eine Differenzierung verzichtet.

welche zur Bearbeitung im Gestaltungsprozess genutzt werden können. Denn „wie wählt man bei über 200 verfügbaren Methoden und Techniken aus, welche die Beste ist?“[Weeb]

Um den Anforderungen des UCD gerecht zu werden, stellt seine Plattform eine Auswahl an Methoden zu Verfügung, welche unter Hilfename eines Filtersystems auf die eigenen Ansprüche eingegrenzt wird. Hierfür können beispielsweise die Art des Produkts (z. B. physisches Produkt oder Interface), das Ziel der Anwendung (z. B. Verstehen und Beschreiben des Nutzungskontexts - siehe 1.1.2.1) oder die Verfügbarkeit an Ressourcen (z. B. Zeit oder Personal) Faktoren sein.

Jede Methode kann einzeln aufgerufen werden und enthält eine Schritt-für-Schritt-Anleitung und tiefere Informationen. Darunter befinden sich nützliche Unterpunkte, wie *Beschreibung*, *Vorteile*, *Einschränkungen* und *mögliche Resultate*. Teilweise werden sogar Methoden vorgeschlagen, welche mit der Auswahl kompatibel sind oder als Alternative dienen können. Auch Quellen und weiterführende Links sind meist vorhanden.

2 Recherche

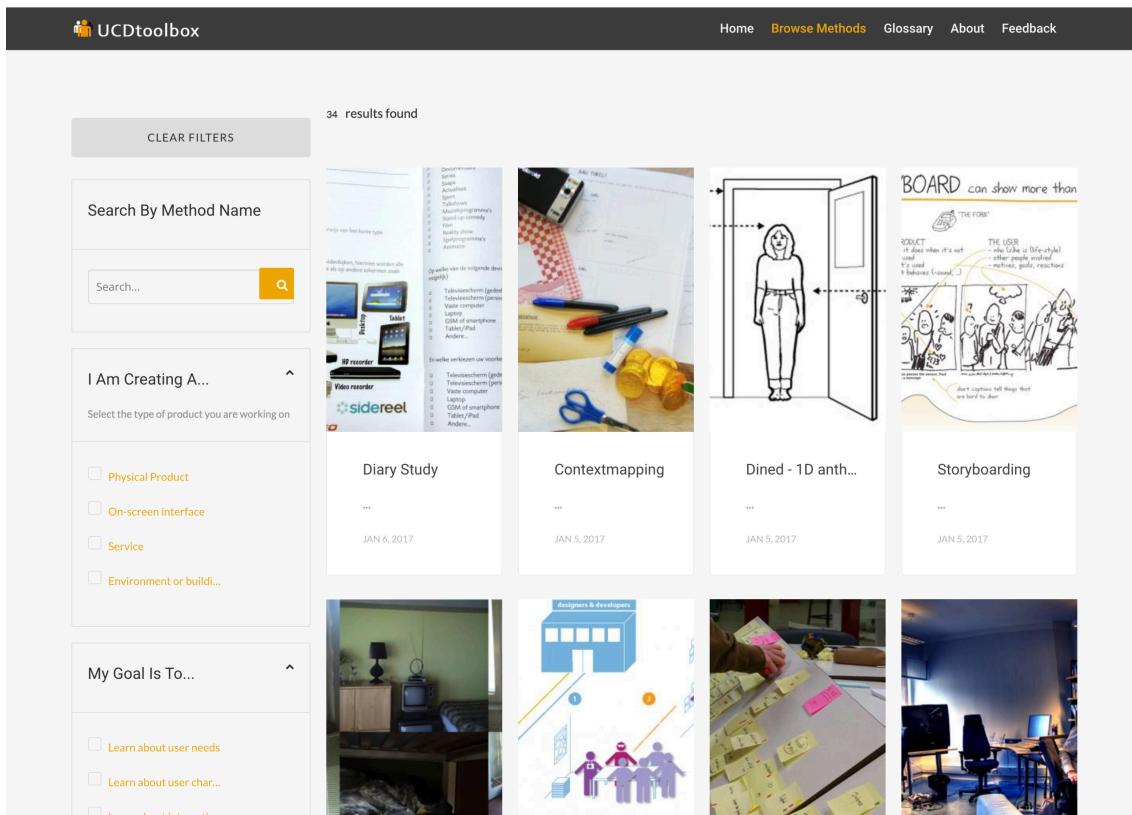


Abbildung 2.1 Screenshot der Methodensammlung von UCDToolbox

Das Projekt befindet sich seit Januar 2017 in einer Beta-Version, weshalb das Design noch nicht optimiert ist. Die UCDtoolbox verschafft den Nutzer aber einen detaillierten Einblick in die Methoden. Auch werden die einzelnen Methoden gut beschrieben, dass sie sogar von unerfahrenen Gestaltern im Designprozess genutzt werden können.

Link: <https://ucdtoolbox.com/>

2.1.1.2 Design Kit

IDEO.org „ist ein gemeinnütziges Designstudio“ und implementiert „Produkte, Dienstleistungen und Erfahrungen, um das Leben und die Lebensgrundlagen von Menschen in armen und gefährdeten Gemeinschaften auf der ganzen Welt zu verbessern.“[IDE]

Die Organisation hat neben einigen Büchern auch eine Plattform geschaffen, welche das Erlernen von HCD-Methoden erleichtern soll. IDEO teilt hierfür die

Methoden in drei Kategorien auf: *Inspiration, Ideation und Implementation.*

Wie auch bei der UCDtoolbox können die einzelnen Methoden im Detail untersucht werden. Auch das Design Kit liefert eine Kurzbeschreibung und eine Schritt-für-Schritt-Anleitung. Die wichtigsten Eckdaten der Methode werden ebenfalls aufgelistet. So findet der Nutzer eine Übersicht darüber, wie lange es dauert die Methode durchzuführen, welchen Schwierigkeitsgrad diese zuordnen ist und welches Material oder welche Art von Teilnehmern für die Ausführung benötigt wird.

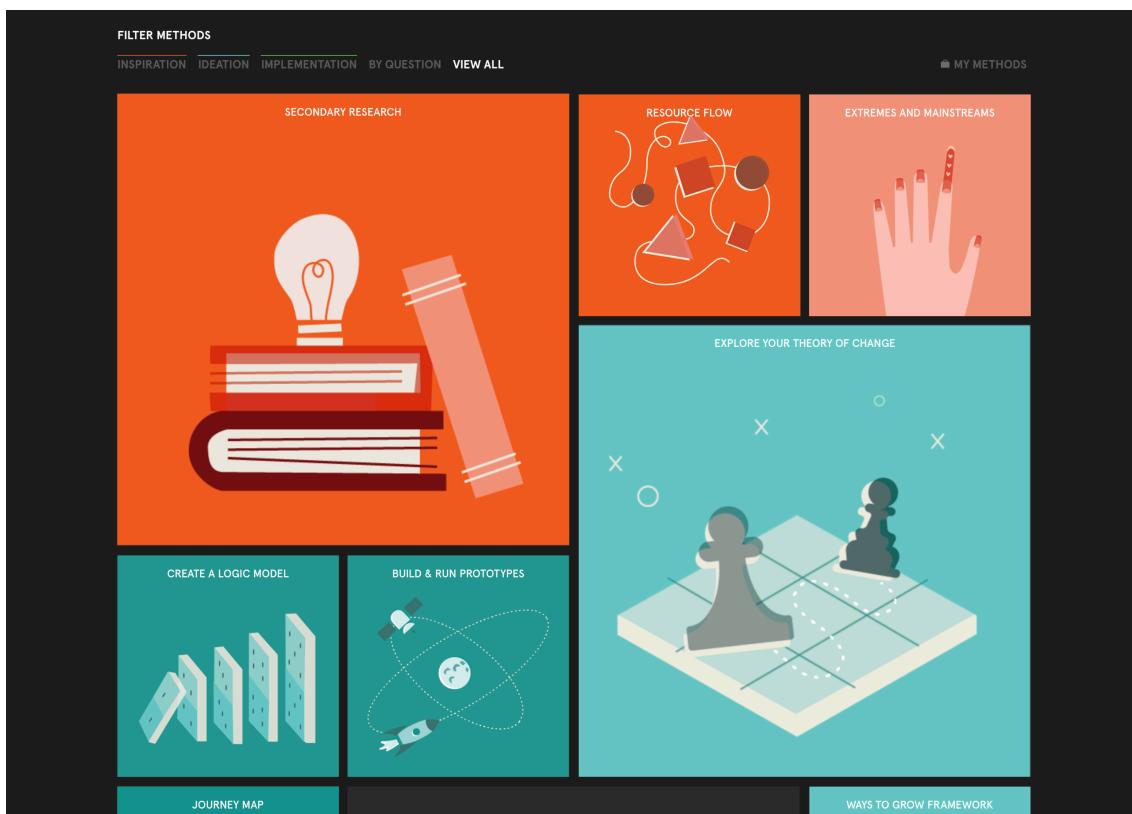


Abbildung 2.2 Screenshot der Methodensammlung von Design Kit

Das Design Kit ist in Sachen Nutzerfreundlichkeit der UCDtoolbox überlegen. Allerdings ist zum jetzigen Stand eine sehr spezielle Auswahl an Methoden getroffen worden, welche vom Hersteller als ideal angesehen wird. So sucht man beispielsweise vergeblich nach Informationen über die Anfertigung von Personae, welche in den meisten Projekten ein Grundbaustein der Arbeit mit HCD sind.

Link: <https://www.designkit.org/>

Ähnliche Projekte

nesta | DIY Toolkit: <https://diytoolkit.org/>

18F | 18F Methods: <https://methods.18f.gov/>

2.1.2 Personagenerator

Für die Anfertigung einzelner Methoden lassen sich im Netz einige Beispiele finden. Die bekanntesten habe ich eingehend analysiert, wobei ein Tool besonders aufgefallen ist:

2.1.2.1 UXPressia

Die Idee von UXPressia ist es, ein Werkzeug zu erzeugen, „das einfach genug für Anfänger ist, um sofort loslegen zu können [...]. Auf der anderen Seite musste es leistungsfähig genug sein, um Profis bei komplexeren Aufgaben zu helfen.“[UXPa]. Der Fokus liegt dabei auf der Erstellung von Personae und Customer Journey Maps².

Im Zuge meiner Recherche habe ich festgestellt, dass UXPressia durchaus gute Ansätze besitzt. Deshalb habe ich die Personae meiner Anwendung auf der Plattform generiert, um einen tieferen Einblick in das System zu erhalten und positive, sowie negative Aspekte herauszuarbeiten.

Bei der Erstellung von Personae bietet UXPressia zunächst verschiedene Templates, welche auf Wunsch schon mit Beispielen befüllt werden. Der Nutzer kann dann den Namen, den Inhalt, die Positionierung und die Größe der einzelnen Felder bearbeiten. Um neue Felder hinzuzufügen, werden dem Nutzer ebenfalls verschiedene Vorlagen angeboten. So sind beispielweise Textfelder bereits mit dem Titel *Goals* oder *Motivations* befüllt. Auch eine Auswahl an Graphen, wie einem klassischen Bereichsschieberegler, oder die Möglichkeit eines Bilduploads, sind vorhanden.

² Da die Plattform als Inspiration für die Methode „Personae“ dient, handelt folgenden Analyse auf dem entsprechenden Feature der Anwendung.

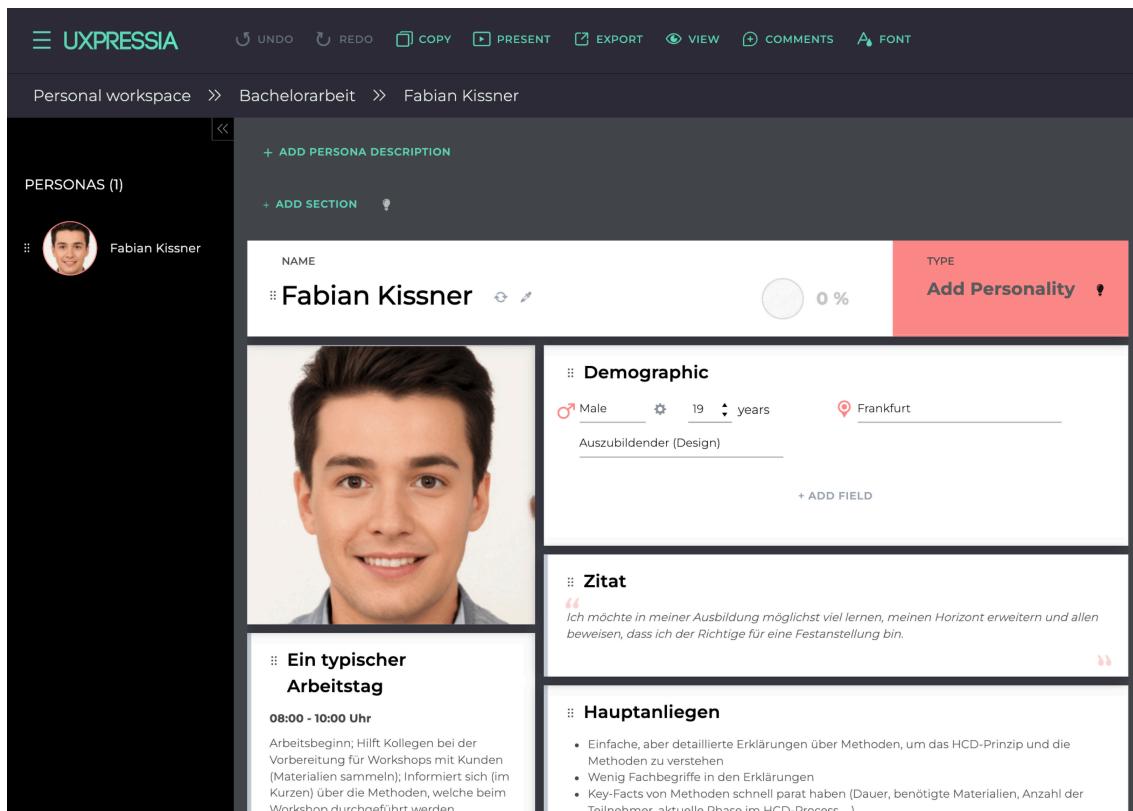


Abbildung 2.3 Screenshot des Personagenerators von UXPressia

Ich war bei der Erstellung meiner ersten Persona überzeugt von der Plattform. Bei der detaillierten Arbeit allerdings fielen mir ein paar negative Kriterien auf. So wurde es beispielsweise etwas unhandlich, als ich mehrere Personae zeitgleich bearbeitete und merkte, dass ein Feld fehlt oder keinen Mehrwert bietet. Möchte man dieses Feld editieren, entfernen oder ein neues hinzufügen, muss dies bei allen Personae manuell erledigt werden.

Auch das Abomodell war im Rahmen der Bachelorthesis ein negativer Aspekt, da in der kostenlosen Version lediglich eine Persona oder eine User Journey Map angelegt werden kann. Dies dient zwar gut als Einblick, was die Plattform zu bieten hat, allerdings kommt man bei einer sinnvollen Integration der Methode nicht daran vorbei, ein Abonnement abzuschließen. Ein Upgrade beginnt hier ab 16\$/Monat (bzw. 160\$/Jahr) für die Erstellung von drei Personae (bzw. Customer Journey Maps)[UXPb]. Da selbst diese Limitierung unvorteilhaft ist, wäre die Pro-Version von 36\$/Monat (bzw. 360\$/Jahr) die vernünftigere Wahl. Mit jeder höheren Stufe des Abomodells sind dann auch weitere Vorteile, wie ein eigenes

Branding oder ein Präsentationsmodus, verfügbar.

Link: <https://uxpressia.com/>

Ähnliche Projekte:

Hubspot | Make my Persona: <https://www.hubspot.de/make-my-persona>

Blueline | Buyer Persona Generator: <https://www.buyer-persona.com/>

2.1.3 Kollaborationsplattformen

Kollaborationsplattformen helfen bei der Zusammenarbeit mehrerer Nutzer. So verwendet auch iconstorm derartige Anwendungen, um gemeinsam mit anderen Stakeholdern an Methoden zu arbeiten.

2.1.3.1 Miro

Miro ist vermutlich das bekannteste Tool dieser Rubrik auf dem Markt und fungiert als eine Art Online-Whiteboard. Das Produkt lässt dem Nutzer freien Spielraum, weshalb es für nahezu jede Tätigkeit in den ersten Phasen eines Projekts mit dem HCD-Prinzip genutzt werden kann. Die User können mit Miro beispielsweise „ein digitales Brainstorming veranstalten, eine Besprechung dokumentieren, einen Kurs leiten oder einen agilen Workflow verwalten - die Möglichkeiten der Zusammenarbeit sind [nahezu] endlos.“[Mir]

Für mein Projekt habe ich vor allem den Fokus darauf gelegt, wie Miro die Zusammenarbeit kennzeichnet. Über das Einblenden der aktiven Nutzer als Pfeil auf dem Board oder das Markieren der bearbeiteten Felder anderer Boardmitglieder während der eigenen Abwesenheit. Die Plattform ist auf diesem Gebiet ein Vorreiter und präsentiert die nützlichsten Informationen ohne den Benutzer der Plattform zu überfordern.

2 Recherche

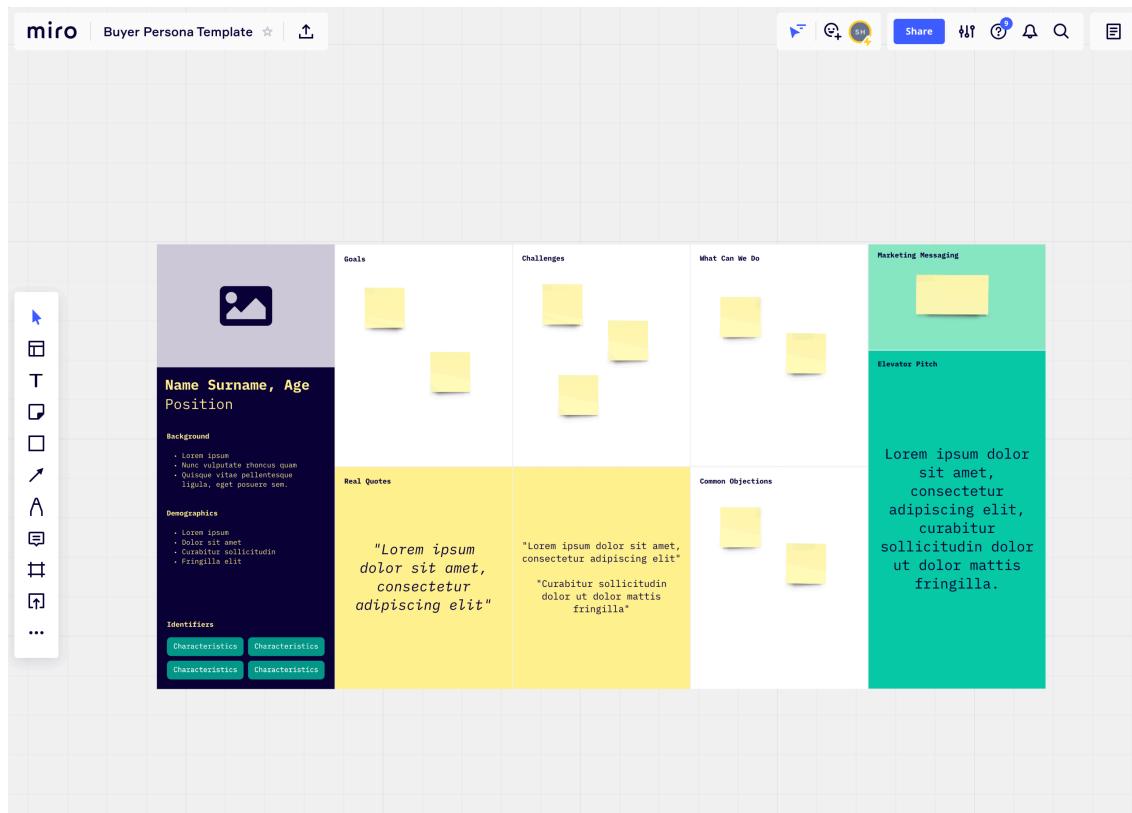


Abbildung 2.4 Screenshot eines Persona-Templates in Miro

Da die Möglichkeiten fast unbegrenzt sind, wird Miro oft als Methodensammlung oder zur Bearbeitung dieser genutzt. Auch iconstorm verwendet die Anwendung in diesem Bereich regelmäßig, vor allem im frühen Stadium einiger Projekte. So werden beispielsweise Ideen gesammelt, um die Templates projektspezifisch aufzubauen. Allerdings birgt die große Freiheit auch Probleme mit sich: Die Vorlagen müssen ständig angepasst werden, befinden sich auf diversen „Boards“ und die Optik lässt zu wünschen übrig. Eine Vereinheitlichung findet nicht statt und meist ist es ein wiederholter Duplizierungsvorgang, welcher erneut Unordnung mit sich bringt.

Link: <https://miro.com/>

2.1.3.2 Notion

Notion bezeichnet sich selbst als „All-in-One-Arbeitsbereich für [...] Notizen, Aufgaben, Wikis und Datenbanken.“ [Not] Ähnlich wie Miro baut die Plattform darauf, dass den Möglichkeiten kaum Grenzen gesetzt werden. Ich selbst habe

für die Anfertigung dieser Arbeit Notion in mehreren Bereichen verwendet. So auch zur Erstellung meiner Szenarios.

Notion fungiert sozusagen als digitales Notizbuch. So ist es einfach Elemente, wie z. B. Texte, Listen, Links, Tabellen oder Bilder hinzuzufügen.



Abbildung 2.5 Screenshot der Szenarios in Notion

Bei all den guten Eigenschaften ist allerdings auch hier die Problematik der Unordnung vorhanden. Für viele Nutzer ist der Einstieg in Notion schwer, da die Boards meist bereits mit Inhalten befüllt sind. Die vielen Unterseiten sorgen dann für Verwirrung beim Anwender. Bei der Anfertigung meiner Szenarios hatte ich große Schwierigkeiten, Ordnung in meine Gruppen und Unterpunkte zu bringen. Mit Fortschreiten des Prozesses fiel es mir zunehmend schwerer, mich in meinem eigenen Dokument zurechtzufinden.

Link: <https://www.notion.so/>

2.2 Potenziale

Wie bereits in der Einleitung des Kapitels angemerkt, gibt es keine Ideallösung auf dem Markt, welche iconstorm weiter helfen kann. Die Sammlungen, Ausführungsmöglichkeiten und Analysen der Methoden müssen auf unterschiedlichsten Plattformen stattfinden. Eine perfekte Lösung kann ohne interne Anfertigung oder spezielle Zusammenarbeit auch nicht existieren, da jede Agentur auf eine andere Sammlung an Methoden baut. Sogar innerhalb der Agenturen bevorzugen die Konzepter und Strategic-Designer unterschiedliche Methoden zur Bearbeitung ihrer zugewiesenen Projekte.

Doch nicht nur die Auswahl an Methoden ist ein Problem, auch deren Bearbeitungsmöglichkeiten. So habe ich in meiner Recherche keine Möglichkeit gefunden, Personae innerhalb eines Projektes einheitlich zu verwalten. I. d. R. sollen diese Personae identische Felder besitzen, da sie sonst nicht miteinander vergleichbar sind. Auch wird der Nutzer mit der Menge an Möglichkeiten oftmals überhäuft, was wiederum Fehler mit sich bringt. So werden Felder genutzt, welche für die Ausführung komplett unnötig sind.

Auch der große Vorteil Verknüpfungen zu schaffen bleibt den bisher genannten Plattformen verwehrt. Zwar sind Anwendungen aufzufinden, welche beispielsweise eine Persona mit einer User Journey verbinden lassen, allerdings gibt es auf dieser keine Möglichkeit Szenarios einzubinden. Auf einer eigenen Plattform können solche fehlenden Möglichkeiten schnell ausfindig gemacht und ergänzt werden. Auch das hinzufügen neuer Methoden ist dann im Laufe der Zeit möglich.

3 Konzeption

Anmerkung: Dieses Kapitel befindet sich auf einer Metaebene. Ich nutze HCD-Methoden, um festzustellen, wie HCD-Methoden am besten auf einer digitalen Plattform auszuführen sind.

3.1 Personae

Wie in der Recherche bereits detailliert Ausgeführt wurde, nutzte ich zur Erstellung meiner Personae den Personagenerator von UXPressia (siehe 2.1.2.1). Ich habe mich auf vier verschiedene Berufsgruppen fokussiert, welche die wichtigsten Zielgruppen abdecken:

Agenturmitglieder

Mitarbeiter der Agentur iconstorm: Konzepter, Visual Designer, Auszubildender

Weitere Workshopteilnehmer

Mitarbeiter des Projektauftraggebers: IT-Leiter

Der Beruf des IT-Leiters ist nur als Beispiel gewählt. Dies soll verdeutlichen, dass die Stakeholder außerhalb der Agentur i. d. R. technikaffiner und deshalb auch über die Dauer eines Workshops hinaus am Projekt beteiligt sind.

3 Konzeption

NAME
Jonathan Dampf



Technologie



Bisherige Erfahrungen (verwandte Arbeiten)

- Gute Tools finden sich nur auf verschiedenen Plattformen verteilt (Personae-Generator, Card-Sorting, etc.)
- Um gewünschte Ergebnisse zu erzielen muss man selbst Templates erstellen
- Unnötiger Zeitaufwand die Ergebnisse zu digitalisieren, um eine Aufbereitung zu ermöglichen

Demografische Informationen

Männlich 50 Jahre Frankfurt
Konzepter

Zitat

„Es gibt dem schönen Satz, dass sich das Design von einer Disziplin der Problemlöser zu einer Problemfindung entwickelt.“

Hauptanliegen (im Bezug auf die Anwendung)

- Verwaltung mehrerer Projekte
- Sammlung aller Methoden, welche ausgeführt werden können
- Unterstützung bei Workshops (Vorbereitung, Ausführung, Auswertung)
- Definition und womöglich Formulare der Methoden digitalisiert auf Abruf (Einsparnis an Arbeitszeit)
- Präferierte kundenbezogenen abändern und vorbereiten
- Methoden digitalisiert ausführen
- Workshops komplett digital abhalten
- Möglichkeit einen (Zusammenhang mit Stakeholdern) auf die Methoden, wenn benötigt Einblick, was die anderen Stakeholder machen
- Übersicht der Ergebnisse an einem Ort
- Außerenstehende die Ergebnisse schnell präsentieren können (optisch ansprechend für Kunden)
- schneller Vergleich von Ergebnissen

Motivationen

- Probleme im Ursprung erforschen
- Zukunftsorientiert arbeiten

Frustrationen

- Unhandliche / unnötig komplexe Tools
- Schlechtes / nicht durchdachtes Design
- Unordnung
- Fehlende Methodenbausteine (z. B. „Whats in my Pocket“ bei Personae)

Designerfahrung

HCD-Methoden: 100
Entwicklungsprozesse: 100
Visual Design: 100

Allgemein

Ordnung: 100
Technisches Verständnis: 100
Organisation: 100
Fachliche Englischkenntnisse: 100
Erfahrung Sportanwendungen: 100

Ein typischer Arbeitstag

0800 - 1000

- Arbeitsbeginn
- Vorbereitung für Workshop mit Kunden
- Aufladen kundenspezifischer Methoden
- Materialien für Methodendurchführung vorbereiten (Post-its, Stifte, Blöcke, ...)

1000 - 1200

- Weg zum Kundentermin mit Firmenauto
- Begrüßung und Vorstellung
- Erklärung des Vorgehens
- Definition von Zielen zusammen mit Kunden
- Ausführung der gewählten Methoden zusammen mit Kunden
- Überprüfung und Unterstützung bei Methodenaufführung
- Dokumentation der Ausführung (Fotos, schriftlich)

1200 - 1300

- Sammeln der Ergebnisse
- Weg zurück in die Agentur
- Organisation der erzielten Ergebnisse (Fotos, ausgefüllte Zettel, etc.)
- Digitalisierung der Ergebnisse (Miro-Board)

Abbildung 3.1 Persona eines Konzepters

NAME
Jana Renk



Technologie



Bisherige Erfahrungen (verwandte Arbeiten)

- Erhält die Ergebnisse ihrer Mitarbeiter auf verschiedenen Plattformen oder Dateipaketen

Demografische Informationen

Weiblich 35 Jahre Frankfurt
Visual Designerin

Zitat

„Ich suche überall nach Potenzialen, die Welt ein bisschen besser zu machen.“

Hauptanliegen (im Bezug auf die Anwendung)

- Informationen, welche Ergebnisse die Methoden überhaupt erzielen sollen
- Schneller und übersichtlicher Einblick in Ergebnisse der Methoden
- Möglichkeit zur Sammlung der Ergebnisse
- Gute Darstellung der miteinander verknüpften Methoden (z. B. Szenarios aufbauend auf Personae)
- Export-Funktion, um den Import in Grafikprogramme zu ermöglichen
- Übersicht der verschiedenen Ergebnisse innerhalb einer Methode (z. B. Vergleich mehrerer Personen)
- Nachfragermöglichkeit bei Ersteller durch z. B. Notizen / Anmerkungen

Motivationen

- Künftige Designarbeiten durch vereinfachte Anwendung von HCD-Methoden verbessern
- Schnelle Informationsbeschaffung

Frustrationen

- Schlechtes Design
- Schlecht präsentierte Ergebnisse
- Unordnung

Designerfahrung

HCD-Methoden: 100
Entwicklungsprozesse: 100
Visual Design: 100

Allgemein

Ordnung: 100
Technisches Verständnis: 100
Organisation: 100
Fachliche Englischkenntnisse: 100
Erfahrung Sportanwendungen: 100

Ein typischer Arbeitstag

0800 - 1000

- Tägliches Meeting am Morgen
- Organisiert die Aufgaben für den Tag
- Verschafft sich Informationen aus der digitalisierten Übersicht der Methodenergebnisse

1000 - 1200

- Sucht nach Zusammenhängen und Erkenntnissen des Workshops
- Leitet daraus ab, welche Ergebnisse das Projekt erzielen muss

1200 - 1300

- Begibt basierend auf den analysierten Ergebnissen den Designprozess

Abbildung 3.2 Persona einer Visual Designerin

3 Konzeption

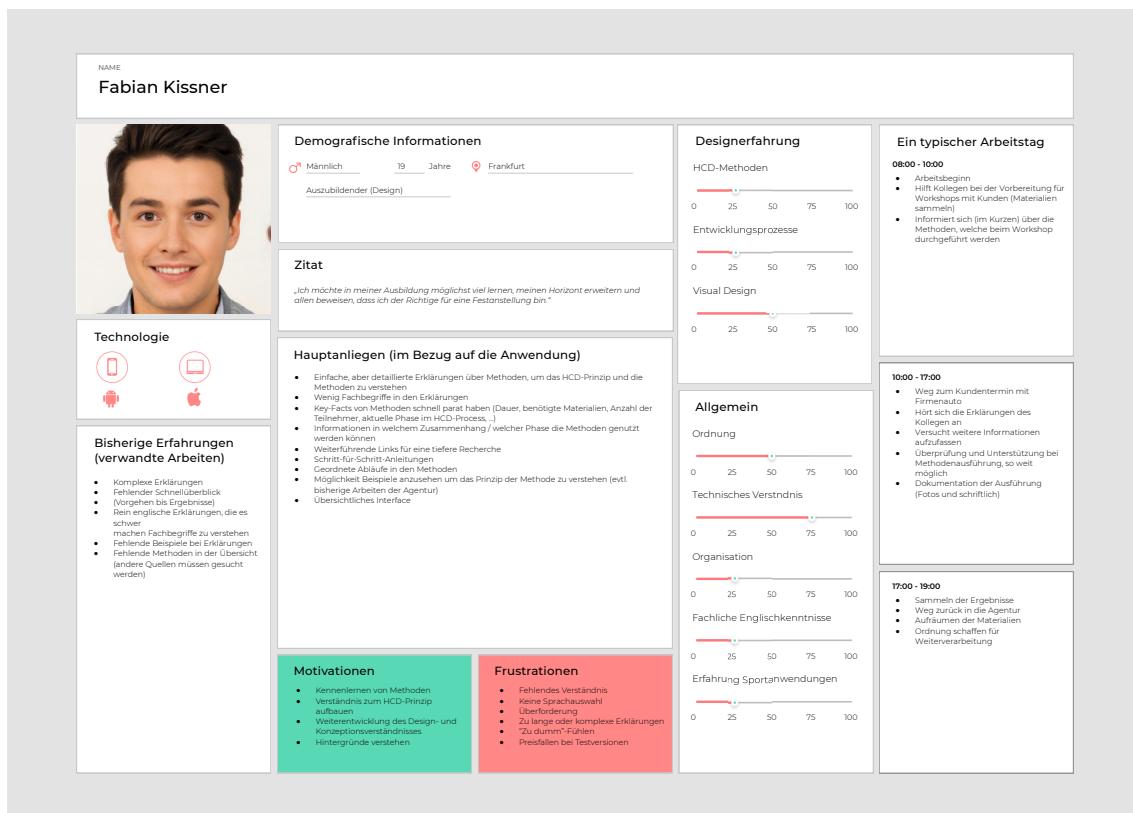


Abbildung 3.3 Persona eines Auszubildenden im Design

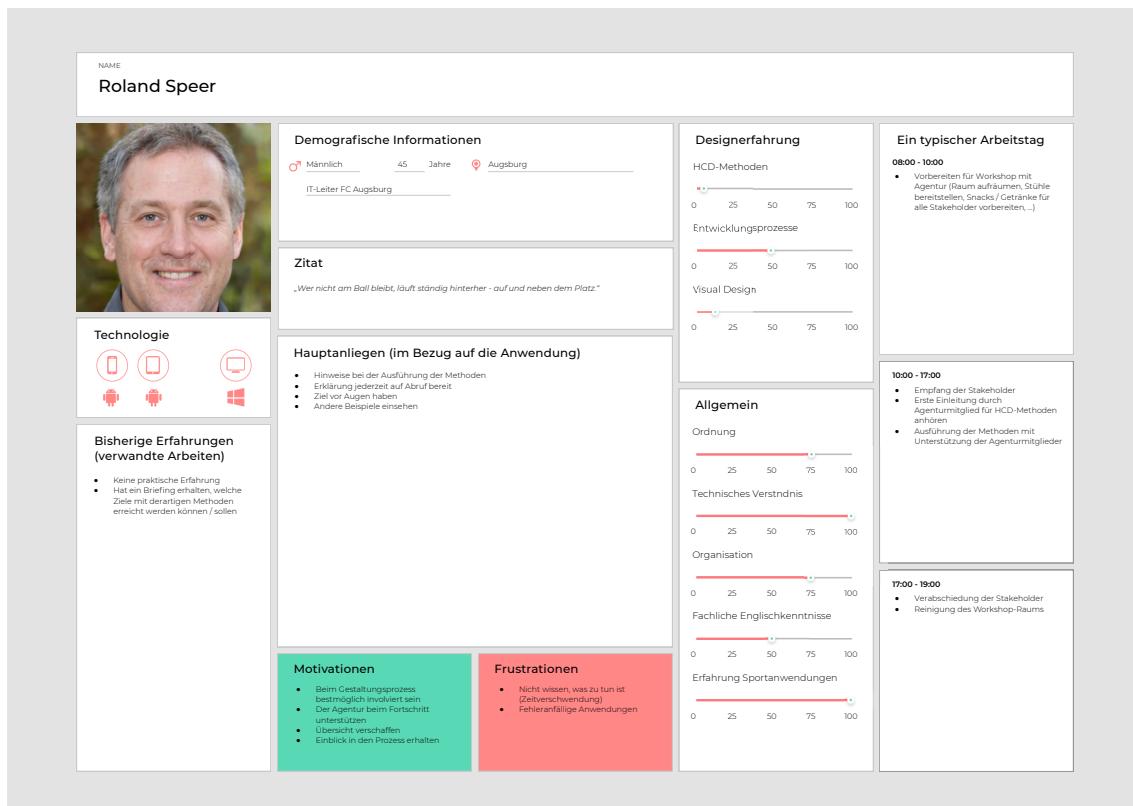


Abbildung 3.4 Persona eines IT-Leiters

3.2 Szenarios

Um meine Szenarios durchzuführen, habe ich ein fiktives Beispiel als Ausgangslage definiert, indem die eben präsentierten Personae eine wichtige Rolle spielen:

Der Fußballverein FC Augsburg möchte ein Projekt entwickeln, um seine Mitglieder auf dem Laufenden zu halten, Präsenz zu zeigen und gegen die fallenden Mitgliederzahlen vorzugehen. Der Verein in Person von Roland (IT-Leiter des Vereins) engagiert die Agentur iconstorm (in Person Jonathan, Jana und Fabian), um eine Strategie zu entwickeln, aus der ein Produkt entstehen soll, welches die genannten Ziele erfüllt.

iconstorm möchte einen Workshop abhalten, um sowohl die Interessen der Stakeholder als auch die Zielgruppe besser kennenzulernen. Dafür nutzen Jonathan und Fabian sogenannte Human-Centered Design Methoden, welche bereits über Jahre hinweg optimiert wurden.

Am Workshop nehmen mehrere Personen aus verschiedenen Positionen des Vereins teil. Diese werden künftig in Person von Roland abgebildet, weil sie im Grunde dieselbe Rolle in dem Szenario einnehmen: Workshopteilnehmer. Die Teilnehmer haben zwar einen unterschiedlichen Erfahrungsschatz, was konzeptuelle Vorgänge betrifft, allerdings ist allen das HCD-Prinzip geläufig, da sie i. d. R. in ähnlichen Abteilungen arbeiten wie Roland. Auch wurde bereits ein Kick-off-Meeting veranstaltet, in welchem geklärt wurde, was die Ziele des Workshops sind.

Der Workshop wird i. d. R. vor Ort abgehalten. Allerdings wird dies in der aktuellen Lage aufgrund der Corona-Pandemie zunehmend erschwert. So ging eine Vision hervor diese Workshops auch remote anbieten zu können, sollte keine andere Möglichkeit bestehen.

Der Ablauf eines Workshops wird definiert vom typischen Vorgehen von Johannes (Konzepter), welcher diese leitet und vorwiegend Daten für Jana sammelt.

Nach Durchführung des Workshops bleibt Roland i. d. R. im engen Kontakt mit Jo-

3 Konzeption

nathan und Jana, um den Prozess gemeinsam voranzutreiben. Da Roland genug Expertise im Umgang mit Methoden besitzt, kann auch er weiterhin auf der Plattform arbeiten.

Die Erkenntnisse des Workshops bilden keine finalen Ergebnisse ab. Es werden lediglich Informationen gesammelt, welche Johannes und seinem Team dabei helfen, ein detailliertes Bild zu schaffen. (Beispiel: Im Workshop werden sogenannte RAW-Personae erstellt. Für eine detaillierte Ausarbeitung fehlt hierbei die Zeit, da viele Bilder gesucht oder Informationen gesammelt werden müssen. Die Detaillierung findet zu einem späteren Zeitpunkt statt).

Anmerkung: Die folgenden Szenarios fungieren lediglich als Ideensammlung. Im späteren Projekt sollen nicht alle angesprochenen Punkte umgesetzt werden, dienen aber als Stoffsammlung für die Bedürfnisse der einzelnen Stakeholder. Eine Priorisierung findet dabei nicht statt.

Auf Grundlage dieser Anforderungen habe ich meine Szenarios aufgebaut und Lösungsanätze verfasst. Da die Szenarios für dieses Projekt durchaus umfangreich ausfielen, findet sich an dieser Stelle nur ein beispielhafter Ausschnitt. Die ausführlichen Szenarios sind gesammelt im Anhang wiederzufinden (siehe A).

a. Workshop und Zusammenarbeit

1. Vorbereitung

1.1 Anmelden

Sowohl Jonathan als auch Fabian, Roland und Jana haben einen eigenen Account von der Agentur bekommen. Sie melden sich an, damit das System weiß, um welchen Nutzer es sich handelt.

1.1.1 Eingabe der Daten

- Benutzername / E-Mail
- Kennwort

1.2 Kunden anlegen

Jonathan möchte einen Kunden im System anlegen, damit Roland nur innerhalb dieses Bereiches Berechtigungen erhält. So werden u. a. Betriebsinterna geschützt und es wird eine bessere Übersicht über die Projekte geboten.

1.2.1 Manuelle Eingabe der Daten

- Definition der wichtigsten Informationen
 - Name der Firma
 - Kurzbeschreibung (Eigenschaften der Firma, Beziehungen, Ansprechpartner, Ziele der Zusammenarbeit, ...)
 - Nutzer hinzufügen
 - Kontaktdaten hinzufügen (evtl. direkt über Nutzer verknüpft)

1.2.2 Rollen verteilen

- Nutzer bekommen Rollen zugewiesen, welche unterschiedliche Rechte besitzen (Guest, Reporter, Editor, Developer, Maintainer, ...)

1.2.3 Rollen verwalten

- Definition der Rollen (welche Rechte besitzen diese)
- z. B. Read-Only, Bearbeitung von Methoden innerhalb eines Projektes eines Kunden, Bearbeitung von Methoden in allen Projekten eines Kunden, Bearbeitung von Methoden von allen Kunden, ...

Abbildung 3.5 Auszug aus den Szenarios

Wie in der Recherche erwähnt, habe ich für die Verwaltung meiner Szenarios die Plattform Notion verwendet (siehe 2.1.3.2). Die Grafik gibt zu erkennen, dass die bereits behandelte Vorlage „Als [Rolle] möchte ich [Wunsch], um [Nutzen]“ mit variiertem Wortlaut als Grundlage für die Definition genutzt wurde.

3.3 System

3.3.1 Anforderungsanalyse

Gemäß dem Kreislauf der Gestaltungsaktivitäten (siehe 1.1.2), wird nun von der ersten in die zweite Phase übergegangen: Dem Festlegen der Nutzungsanforderungen.

Da die Szenarios schon mit ersten Ideen zur Umsetzung befüllt wurden, hatte ich bereits eine grobe Vorstellung meiner Anwendung. Um einen detaillierten Entwurf ableiten zu können, erstellte ich eine Anforderungsanalyse. Dafür sammelte ich alle Anforderungen des Systems in einer großen Tabelle. Aus den Szenarios wurde dabei nicht jede Idee übernommen, da viele bereits bessere Alternativen beinhalteten. Für jede Anforderung notierte ich dabei eine mögliche Umsetzung (z. B. als Dropdown) und wenn benötigt einen „Parent-Component“ (wenn das Element in einem anderen Element positioniert werden kann - z. B. wenn sich der Dropdown im Header befindet). Optional wurden den Einträgen noch Notizen angehängt. Diese Liste an Anforderungen unterzog ich anschließend einer MoSCoW-Priorisierung. Hierbei ordnet man jeder Anforderung mithilfe von Wörtern eine Relevanz für die Umsetzung zu. MoSCoW steht in diesem Fall für:

M: **MUST HAVE** (Muss integriert werden)

S: **SHOULD HAVE** (Sollte integriert werden)

C: **COULD HAVE** (Könnte integriert werden)

W: **WON'T HAVE** (Wird nicht integriert)

„Die 'Muss'-Anforderungen sind nicht verhandelbar, wenn sie nicht geliefert werden, dann ist das Projekt ein Fehlschlag, deshalb ist es im Interesse aller, sich darauf zu einigen, was geliefert werden kann und nützlich sein wird. 'Nice to have'-Funktionen werden in die anderen Kategorien 'Sollte' und 'Könnte' eingeordnet.“ [Col] Die letzte Kategorie spricht für sich.

Diese Analyse habe ich sowohl für die endgültige Anwendung, als auch für die Umsetzung im Bachelorprojekt getätigt. Wurde eine Anforderung mit *Won't*

have gekennzeichnet fügte ich eine weitere Spalte für eine erneute Priorisierung ein, welche angibt, ob das Element angedeutet werden soll. Ist also beispielsweise ein Button in der tatsächlichen Anwendung als *Must have* gekennzeichnet, allerdings in der Analyse für das Bachelorprojekt der Kategorie *Won't have* zugeordnet, so kann der Button in der Umsetzung dennoch angedeutet werden (z. B. im „disabled state“¹ mit einem Hinweis für eine weitere Bearbeitung: „Coming Soon“).

Diese sehr lange Liste an Anforderungen unterteilte ich abschließend in verschiedene Panels, um eine verbesserte Übersicht zu erhalten:

```
Dashboard (Kundenübersicht)
└ Kundenpanel (Projektübersicht)
  └ Projektpanel (Methodenübersicht)
    └ Methodenpanel Szenarios (Ausführungsübersicht)
      └ Ausführungspanel Szenario
    └ Methodenpanel Personae (Ausführungsübersicht)
      └ Ausführungspanel Persona
```

Elemente, die panelübergreifend verfügbar sein müssen, ordnete ich der Gruppierung „Universell“ zu. Welche Inhalte und Eigenschaften die einzelnen Panels im Detail beinhalten, kann im Abschnitt „Screen-Flow“ (siehe: 3.3.5) nachgelesen werden.

Der gesamte Umfang der Anforderungsanalyse ist im Anhang wiederzufinden (siehe B). Im Folgenden wird ein Auszug der Tabelle präsentiert, um einen Einblick zu erhalten:

¹ disabled state (deutsch: deaktivierter Zustand): „Ein deaktiverter Zustand bedeutet, dass eine Komponente oder ein Element nicht interaktiv ist.“ [Goob]

3 Konzeption

Methodenpanel (Ausführungsübersicht)

<u>Aa Anforderung</u>	<u>mögliche Umsetzungen</u>	<u>möglicher Parent Component</u>	<u>MoSCoW-Priorisierung</u>	<u>Umsetzung BP</u>	<u>Andeutung in Anwendung, bei "Won't have"</u>	<u>Kommentar</u>
<u>METHODENPANEL ALLGEMEIN</u>						Hinweis: Diese Tabelle wird unterteilt in allgemeine und methodenbezogene Anforderungen
<u>Methodeninformationen anzeigen</u>	Text		Must Have	Must Have		Kurzinformationen über die Methode
<u>Methodeninformationen (Details) anzeigen</u>	Text, Bilder, Pop-up, Link zu externer Seite		Must Have	Must Have		
<u>Gruppe hinzufügen</u>	Pop-up		Must Have	Should Have		
<u>Gruppe verwalten</u>	Button, Klick auf Textfeld		Must Have	Should Have		Gruppenname und -beschreibung
<u>Gruppe duplizieren</u>	Pop-up, Dropdown		Should Have	Won't Have	Could Have	Duplizierungsmöglichkeiten können mithilfe eines Formulars und einer Abfrage nach gewünschtem Ablageort (Kunde / Projekt) und Inhalt (Auswahl der Ausführungen) dupliziert werden
<u>Gruppe entfernen</u>	Button		Must Have	Must Have		
<u>Ausführung hinzufügen</u>	Button, Pop-up, Link zu externer Seite, Transparente Card	Gruppe	Must Have	Must Have		
<u>Ausführungen auflisten</u>	Cards mit Key-Facts, Liste	Gruppe	Must Have	Must Have		Mögliche Key-Facts: Beschreibung bei Szenarios oder Alter / Beruf bei Personae. Immer: Status
<u>Ausführung anderer Gruppe zuordnen</u>	Drag-and-Drop, Dropdown	Gruppe, Card / Liste (Ausführung auflisten)	Should Have	Should Have		
<u>Ausführung duplizieren</u>	Pop-up, Dropdown	Card / Liste (Ausführung auflisten)	Should Have	Won't Have	Could Have	Duplizierungsmöglichkeiten können mithilfe eines Formulars und einer Abfrage nach gewünschtem Ablageort (Kunde / Projekt / Gruppe) dupliziert werden
<u>Ausführung entfernen</u>	Button	Card / Liste (Ausführung auflisten)	Must Have	Must Have		
<u>Ausführungen filtern</u>	Text-Input	Card / Liste (Ausführung auflisten)	Should Have	Could Have		evtl. nur bei Listenansicht

3 Konzeption

<u>Aa Anforderung</u>	<u>mögliche Umsetzungen</u>	<u>möglicher Parent Component</u>	<u>MoSCoW-Priorisierung</u>	<u>Umsetzung BP</u>	<u>Andeutung in Anwendung, bei "Won't have"</u>	<u>Kommentar</u>
<u>Ausführungen sortieren</u>	Drag-and-Drop, Dropdown	Card / Liste (Ausführung auflisten)	Should Have	Could Have		
<u>Ansicht wechseln</u>	Dropdown	Card / Liste (Ausführung auflisten)	Could Have	Won't Have	Could Have	Ansichten: Galerie, Liste
<u>Warnungen / Key-Facts anzeigen</u>	Text, Icon (Notification), Chip	Card / Liste (Ausführung auflisten)	Should Have	Should Have		z. B. Anzahl ungenutzter Verknüpfungen bei Szenarios oder leerer Felder in Personae
<u>Ausführung aus anderem Projekt / anderen Kunden importieren</u>	Pop-up	Card / Liste (Ausführung auflisten)	Won't Have	Won't Have		Duplizierungsmöglichkeit erfüllt die Anforderung bereits.
<u>METHODENPANEL SZENARIOS</u>						
<u>Verknüpfungen anzeigen</u>	Card, Liste mit Key-Facts	Gruppe	Must Have	Must Have		
<u>Verknüpfungen verwalten</u>	Liste, Checkboxes	Gruppe	Must Have	Must Have		Liste beinhaltet alle Personae (inkl. Name, Gruppe und Status)
<u>METHODENPANEL PERSONAE</u>						
<u>Template wählen / bearbeiten</u>	Pop-up, Link zu externer Seite	Gruppe	Should Have	Won't Have	Could Have	Hinweis: Im BP ist die Ausführung der Personae nur in Teilen geplant und dient lediglich zur Präsentation der Verknüpfungsmöglichkeiten.
<u>Template auf andere Gruppen anwenden</u>	Dropdown, Pop-up	Gruppe	Could Have	Won't Have	Could Have	

Abbildung 3.6 Anforderungsanalyse des Methodenpanels

3.3.2 Use-Case-Diagramm

Aus den Ergebnissen der vorgegangenen Konzeptarbeit konnte ich ein Use-Case-Diagramm erstellen, welches mir bei der Sammlung von Funktionalitäten half. Da sich die Methoden in der Bearbeitung selbst sehr differenzieren, habe ich nur die Cases aufgeführt, welche eine Gemeinsamkeit aufweisen können.

3 Konzeption

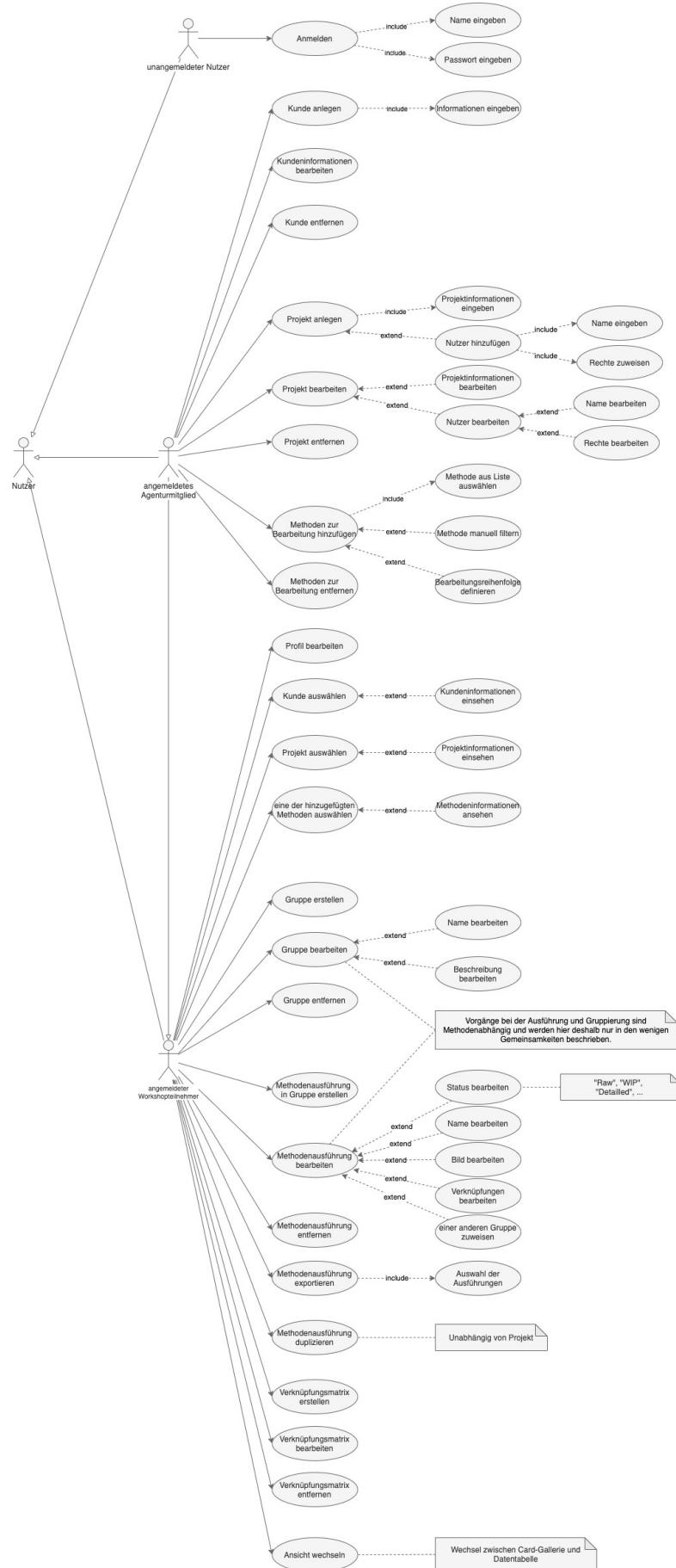


Abbildung 3.7 Use-Case-Diagramm der Anwendung

Im Diagramm ist eine Unterscheidung zwischen Agenturmitgliedern und anderen Workshopteilnehmern vorzufinden. Agenturmitglieder erhalten von Grund auf mehr Berechtigungen als andere Nutzer der Anwendung. Das hat den Ursprung, dass Betriebsinterna und Einstellungen hinterlegt sind, welche externen Anwendern vorenthalten werden müssen. Im Laufe der Entwicklung ist die Implementierung weiterer Arten von Nutzern geplant (z. B. ein Gast, welcher keine Bearbeitungsmöglichkeiten besitzt). Um das Diagramm aber so einfach wie möglich zu halten, werden diese Rollen außenvor gelassen. Nach der Anmeldung weiß das System, welche Art von Nutzer es gerade bedient. Alle Nutzerprofile sowie deren Rollen werden von der Agentur verwaltet.

Ein Agenturmitglied kann folgende Tätigkeiten ausführen:

3.3.2.1 Kunde anlegen, bearbeiten und entfernen

Ein Kunde (nicht zu verwechseln mit Workshopteilnehmer) erhält einen Namen und weitere Informationen wie Kontaktpersonen, Anschrift, eine Beschreibung u. v. m. Diese Angaben können im späteren Verlauf abgeändert werden. Kunden, welche keine Verwendung mehr finden, können aus dem System entfernt werden.

3.3.2.2 Projekt anlegen, bearbeiten und entfernen

Ein Kunde ohne Projekt ist i. d. R. nicht sehr sinnvoll. Deshalb ist es möglich, den Kunden neue Projekte zuzuweisen. Erneut werden dabei variable Informationen hinterlegt. Projekte, welche keine Relevanz mehr besitzen, können entfernt werden. Hinzu kommt die Möglichkeit, jedem Projekt Nutzer mit entsprechenden Rechten zuzuordnen. So erkennt das System, zu welchen Tätigkeiten der Nutzer befugt ist und in welche Informationen er einen Einblick erhält. Die Anzahl der Nutzer kann über die Projektlaufzeit variieren.

3.3.2.3 Methoden zur Bearbeitung hinzufügen und entfernen

Über ein vordefiniertes Sammelsurium können HCD-Methoden zum Projekt hinzugefügt werden. Dies lässt sich von der Grundidee der Planung eines Workshops gleichsetzen. Die Methoden lassen sich über Eigenschaften filtern (vgl.

2.1.1.1). Wurde eine Vielzahl an wichtigen Methoden definiert, so kann auch die Reihenfolge dieser geändert werden. Dies ist beispielsweise hilfreich, um den Ablauf eines Workshops zu definieren. Auch hier wird die übliche Möglichkeit geboten die Methoden zu entfernen, sobald sie keinen Zweck mehr erfüllen.

Darüber hinaus bietet das System weitere Funktionen, bei denen die Art des Nutzers keine Rolle spielt:

3.3.2.4 Profil bearbeiten

Jeder Nutzer kann die üblichen Einstellungen seines Profils ändern: Name, Profilbild, Passwort, u. v. m.

3.3.2.5 Kunden, Projekt und Methode auswählen

Ein angemeldeter Nutzer erhält eine Übersicht über die Kunden, Projekte und Methoden, zu denen er einen Zugriff gewährt bekommt. Diese kann er öffnen und auch die entsprechenden Informationen einsehen.

3.3.2.6 Gruppe oder Ausführung erstellen, bearbeiten, entfernen und exportieren

Jede Methode hat eine andere Möglichkeit der Ausführung impliziert. Grundlegend kann aber jeder Nutzer eine beliebige Anzahl an Ausführungen einer Methode anfertigen. Diese Ausführungen werden in Gruppen unterteilt, welche einen Namen und eine Beschreibung besitzen.

Jede Ausführung besitzt mindestens ein Erkennungsbild, einen Namen und einen Status. Da die Verknüpfungen einen großen Mehrwert dieser Anwendung bilden, soll jede Methode die Möglichkeit haben, diese zu erzeugen. Auch ist es dem Nutzer möglich, die Ausführungen projektunabhängig zu duplizieren. Alle Angaben, die getätigt wurden, sind aufgrund des iterativen Prozesses ersetzbar.

Dem Nutzer werden außerdem verschiedene Möglichkeiten gegeben, die Ausführungen zu exportieren. Dies ist sowohl einzeln als auch in Gruppen möglich. Dabei kann der User frei wählen, welche Ausführungen genau exportiert und in welchem Dateiformat sie gespeichert werden sollen.

3.3.2.7 Verknüpfungsmatrix erstellen, bearbeiten und entfernen

Verknüpfungsmatrizen sind eine Neuerung, welche einen Mehrwert gegenüber bisherigen Anwendungen in diesem Bereich liefern. Da die Erklärung einen erhöhten Grad an Komplexität vorweist, wird diese mit grafischer Unterstützung im Abschnitt des Screen-Flows (genauer: Projektpanel, siehe: 3.3.5.3) vorgenommen. Für das Use-Case-Diagramm ist lediglich wichtig, dass diese Matrizen in beliebiger Anzahl erstellt, bearbeitet und entfernt werden können.

3.3.2.8 Ansicht wechseln

Grundlegend wird dem Nutzer in den meisten Panels eine Gallerieansicht geboten. Dies sorgt für eine schnelle Übersicht. Da die Anzahl der Ausführungen, Gruppen, Methoden, Projekte und Kunden schnell ansteigen kann, wird eine zusätzliche Tabellenansicht implementiert, welche die zusätzliche Möglichkeit für eine Sortierung und Filterung der Einträge bietet.

3.3.3 Datenmodell

Um eine Übersicht über das Datenmodell zu erhalten, habe ich ein ER-Diagramm angelegt. Die Bezeichnungen der Datentypen (z. B. *RICH TEXT*) leiten sich vom Content-Management-System Strapi (siehe 4.2.1) ab.

3 Konzeption

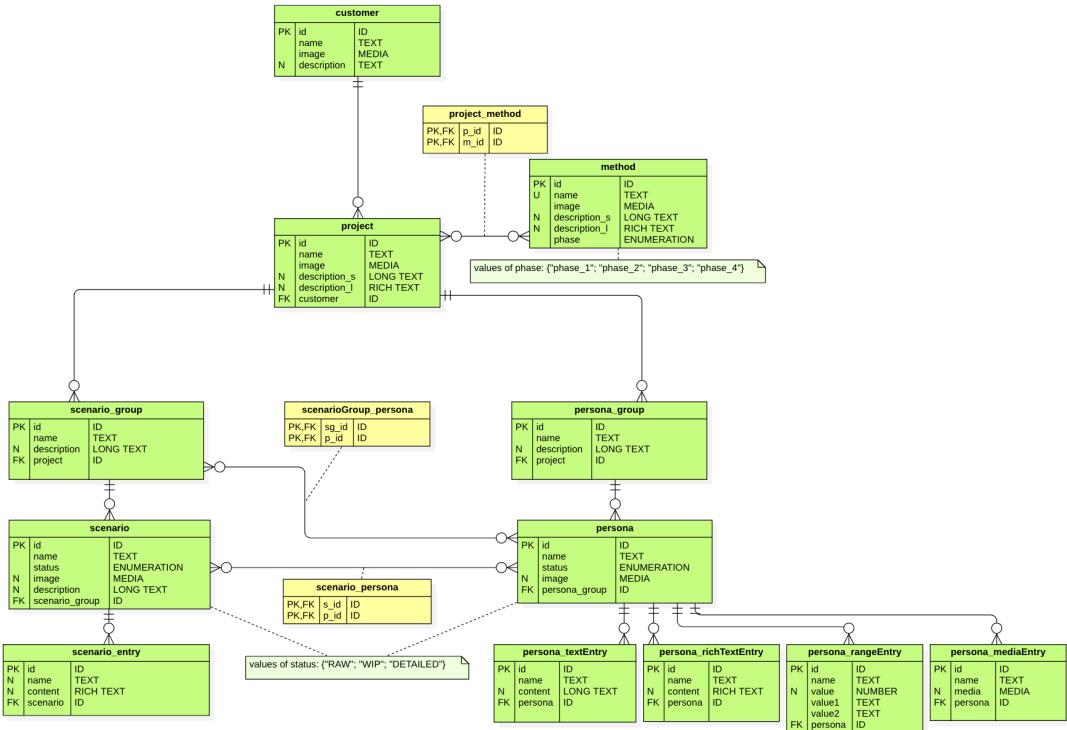


Abbildung 3.8 ER-Diagramm des Datenmodells

Der *Customer* bildet die Grundlage des Modells. Jeder *Customer* hat einen Namen, ein Erkennungsbild und eine Beschreibung, welche der Nutzer der Anwendung frei wählen kann. Diese sind, wie alle vom Nutzer definierten Werte, im Laufe des Prozesses anpassbar. Außerdem besitzt jeder Eintrag eine ID, welche vom System automatisch erzeugt wird. (Die Zuweisung einer ID geschieht unabhängig von der Collection², weshalb sie künftig keine weitere Definition nötig hat.)

Ein *Project* besitzt ähnliche Eigenschaften, welche der Nutzer anpassen kann: Name, Erkennungsbild und eine Beschreibung (in Kurz- und Langform). Jedes *Project* kann einem *Customer* (mithilfe einer Relation³) zugewiesen werden. Ein *Customer* kann allerdings mehrere *Projects* besitzen, weshalb zwischen den beiden Collections eine sogenannte one-to-many Relation (1:n-Beziehung) entsteht.

Eine *Method* (also z. B. Szenarios) erhält einen Namen, ein Bild, eine Beschreibung (ebenfalls in Kurz- und Langform) und eine Phasenbeschreibung (siehe 1.1.2).

2 Eine Collection ist vergleichbar mit einer Tabelle in SQL

3 Eine Relation ist vergleichbar mit einer Beziehung in SQL

In der Datenbank selbst werden die Phasen nur mit „phase_1“ bis „phase_4“ betitelt. Die Bezeichnung wird dann im Frontend in die entsprechende Textform umgewandelt). *Methods* können im Frontend nicht angepasst werden, da diese fest im System verankert werden und keine variablen Werte besitzen.

Ein *Project* kann mehrere *Methods* beinhalten (z. B. Szenarios und Personae). Eine *Method* kann allerdings auch in mehreren *Projects* verwendet werden. Deshalb entsteht zwischen den beiden Collections eine many-to-many Relation (n:m-Beziehung).

Im Laufe der Zeit kann eine beliebige Anzahl an Methodenausführungen entstehen. Da in unserem Fall lediglich Ausführungen von Szenarios und Personae von Relevanz sind, finden sich auch nur Datenmodelleinträge dieser beiden Methoden im ER-Diagramm wieder. Beide können in Gruppen unterteilt werden: *Scenario_groups* bzw. *Persona_groups*. Diese erhalten vom Nutzer jeweils einen Namen und eine Beschreibung.

Zu jeder Gruppe können Ausführungen (*Scenario* bzw. *Persona*) hinzugefügt werden. Diese haben grundsätzlich einen Namen, einen Status (wählbar zwischen „RAW“, „WIP“ (Work in Progress) und „DETAILED“) und ein Bild. Bei *Scenarios* kann zusätzlich eine Beschreibung eingefügt werden. Eine *Persona* bekommt das nicht, weil diese anhand der eingefügten Einträge erzeugt wird (siehe 3.3.5.7). Da jede Ausführung nur einer Gruppe zugehörig ist, sind diese mit einer one-to-many Relation mit der entsprechenden Collection verbunden. Diese Ausführung kann auch auf Wunsch des Nutzers die Zugehörigkeit der Gruppe wechseln.

Die beiden Methodenausführungen unterscheiden sich vor allem bei den möglichen Einträgen. *Scenario_entries* haben immer dieselben Eigenschaften: Name und Inhalt (wobei Letzterer in Markdown⁴ verfasst werden kann).

Komplexer ist die Datenverwaltung bei den Einträgen von Personae. Da unterschiedliche Typen gewählt werden können, gibt es auch verschiedene Col-

⁴ Markdown: „Markdown ist ein einfacher Syntax, um HTML-Code aus Textdokumenten zu generieren.“ [Las16]

lections: *Persona_textEntries* (Name und Inhalt), *Persona_richTextEntries* (Name und Inhalt, vergleichbar mit *Scenario_Entries*), *Persona_rangeEntries* (Name, einen numerischen Wert in Prozent und zwei Definitionswerte - z. B. Introvertiert und Extrovertiert) und *Persona_mediaEntries* (Name und Medium - i. d. R. ein Bild).

Wie bereits thematisiert, sind die Verknüpfungen eine Besonderheit meiner Anwendung. Im Bearbeitungsprozess wurde klar, dass es sinnvoller ist, eine Verknüpfung zwischen *Scenario_groups* und *Personas* zu ermöglichen (many-to-many Relation). Dass die Verknüpfungen nicht gleichermaßen behandelt werden, ging aus der konzeptionellen Vorarbeit und Benutzertests heraus: Bei der Anwendung von Szenarios spielt i. d. R. die Gruppierung und deren Beschreibung eine wichtige Rolle und bekommt daher einen großen Fokus, wohingegen diese bei Personae primär der Organisation dienen.

Die Verknüpfungen sind variabel und können jederzeit vom Nutzer erzeugt oder entfernt werden. Sobald eine Verknüpfung zwischen einer *Scenario_group* und einer *Persona* besteht, gibt es auch die Möglichkeit ein *Scenario*, welches der *Scenario_group* zugewiesen ist, mit selbiger *Persona* zu verknüpfen oder die Verknüpfung wieder zu lösen. Zur Erklärung dieses Zusammenhangs dient folgendes Beispiel:

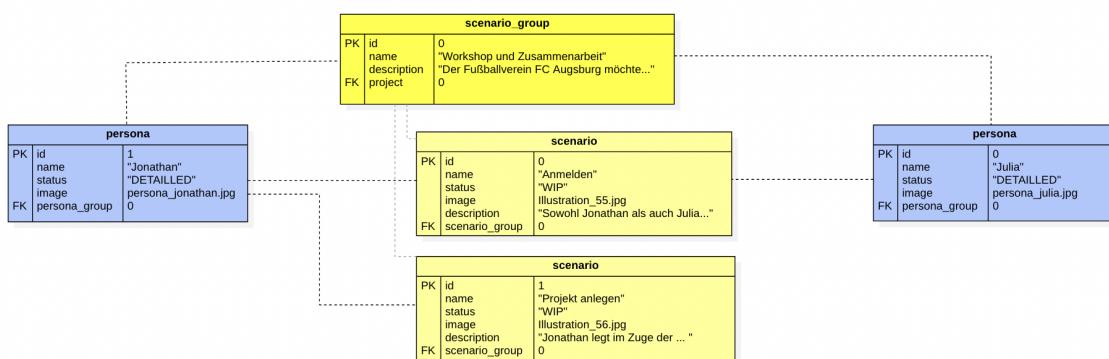


Abbildung 3.9 Grafische Darstellung des Beispiels

Die *Scenario_group* mit dem Namen „Workshop und Zusammenarbeit“ hat Verknüpfungen zu den *Personas* „Jonathan“ und „Julia“, weil diese bei der Bear-

beitung von Relevanz sind. In dieser *Scenario_group* sind die beiden *Scenarios* mit den Namen „Anmelden“ und „Projekt anlegen“ hinzugefügt worden. Standardmäßig besteht für die beiden *Scenarios* ebenfalls eine Verknüpfung zu den eben genannten *Personas*. Nun merkt der Anwender allerdings, dass „Julia“ beim *Scenario* „Projekt anlegen“ keine nennenswerte Rolle spielt. Daher kann er die Verknüpfung kurzerhand innerhalb des *Scenarios* selbst auflösen. Was er nicht kann, ist innerhalb des *Scenarios* eine *Persona* (z. B. eine namens „Fabian“) hinzufügen, sofern diese in der entsprechenden Gruppe nicht als Verknüpfung vorhanden ist.

Sonderfall: Sollte ein *Scenario* die *Scenario_group* wechseln, so wird der Nutzer gefragt, welche Verknüpfungen das *Scenario* erhalten soll. Besteht eine Verknüpfung zwischen dem zu verschiebenden *Scenario* und einer *Persona*, welche nicht in der neuen *Scenario_group* hinterlegt ist, muss der Nutzer auch definieren, ob er diese *Persona* zu den Verknüpfungen der *Scenario_group* hinzufügen möchte.

Andersherum: Beim Bearbeiten einer *Persona* muss erst die entsprechende *Scenario_group* hinzugefügt werden, um ein *Scenario* zu verknüpfen.

Da es in Strapi nicht möglich ist, Regeln für die Relations zu definieren, sind diese auch nicht im ER-Diagramm zu finden. Die Eingrenzung der möglichen Verknüpfungen findet komplett über die Integrierung in das Frontend statt.

3.3.4 Designsystem

Da der Fokus meiner Anwendung auf der Konzeptarbeit liegt, nahm dieser auch viel Zeit in Anspruch. Um einen Prototypen anzufertigen, konnte ich daher kein eigenes Designsystem entwickeln. Ich konzentrierte mich also darauf, ein bereits vorhandenes und dennoch passendes System zu integrieren.

Durch die Wahl meines Frameworks (siehe 4.1) wurde ich nur wenig beschränkt. Jedoch musste ich sichergehen, dass eine nahtlose Integrierung des Systems möglich ist.

Die engere Wahl begrenzte sich nach längerer Recherche auf Spectrum⁵, Eva Design⁶, Carbon Design System⁷ und Material Design. Letztlich entschied ich mich für Material Design, ein von Google entwickeltes Designssystem, das „Teams bei der Entwicklung hochwertiger digitaler Erlebnisse unterstützt“[Gooa].

Das Team rund um Material Design möchte sichergehen, dass ihr System auch richtig verwendet wird, weshalb sie auf der eigenen Homepage die wichtigsten Informationen für den Nutzer bereitstellen:

Design

Im Designabschnitt finden sich die Designrichtlinien des Systems wieder. Also beispielsweise die Definition des Layouts, der Farbpalette, der Typografie oder der Icons.

Components

Da durch die Anforderungsanalyse bereits eine grobe Auflistung an geforderten Komponenten hervorging, wurde auf diese der größte Fokus gelegt. Die integrierten Komponenten werden in der Dokumentation interaktiv präsentiert. So kann der Nutzer bereits gewünschte Anpassungen ansehen, bevor er sie in die Anwendung einbindet. Auch vorgegebene Beispiele werden präsentiert und geben einen detaillierten Eindruck. Des Weiteren sind erneut wichtige Designregeln (z. B. „Do's“ and „Dont's“), Informationen zur Anatomie u. v. m. nachzulesen.

Für die technischen Aspekte, wie die Integrierung des Designsystems in eigene Anwendungen lassen sich ebenfalls Hinweise und Tutorials finden, allerdings hat dies aufgrund der Einbindung einer externen Library (siehe 4.1.2) keinen großen Mehrwert für diese Arbeit.

Link: <https://material.io/>

5 Spectrum: <https://spectrum.adobe.com/>

6 Eva Design: <https://eva.design/>

7 Carbon Design System: <https://www.carbondesignsystem.com/>

3.3.4.1 Anpassungen

Angelehnt an das Markenbild von iconstorm wurde orange als Primärton gewählt. Durch verschiedene Graustufen sind die typografischen und sekundären Elemente abgesetzt.

Primary I orange			
primary darken-4	#E65100	primary lighten-1	#FFA726
primary darken-3	#EF6C00	primary lighten-2	#FFB74D
primary darken-2	#F57C00	primary lighten-3	#FFCC80
primary darken-1	#FBBC00	primary lighten-4	#FFE0B2

Secondary I grey			
secondary darken-4	#017374	secondary lighten-1	#00B3A6
secondary darken-3	#018786	secondary lighten-2	#00C4B4
secondary darken-2	#019592	secondary lighten-3	#03DAC5
secondary darken-1	#01A299	secondary lighten-4	#70EFDE

Error	#B00020	Success	#4CAF50		
High Emphasis	On Primary I White	100%	High Emphasis	On Surface I White	87%
Medium Emphasis	On Primary I White	74%	Medium Emphasis	On Surface I White	60%
Disabled	On Primary I Black	38%	Disabled	On Surface I White	38%

Abbildung 3.10 Integrierte Farbpalette

Da der genaue Farbwert des Orangetons für die Entwicklung agenturinterner Projekte variiert werden kann, nutzte ich eine vorgegebene Farbpalette von Material Design. Wird eine vom System empfohlene Primärfarbe gewählt, werden automatisch verschiedene Abstufungen eingebunden, welche sorgfältig ausgewählt und aneinander angepasst wurden. Um dem System eine gewisse Grundlage an Benutzerfreundlichkeit zu gewähren, sind zudem zwei Aktionsfarben für Fehler- und Erfolgsmeldungen definiert.

Bei der Schriftart setze ich in meinem Prototypen mit „Roboto“ auf den erprobten Standard von Material Design.

Headline 1	H1/Roboto/Light/96px		
Headline 2	H2/Roboto/Light/60px		
Headline 3	H3/Roboto/Regular/48px		
Headline 4	H4/Roboto/Regular/34px		
Headline 5	H5/Roboto/Regular/24px		
Headline 6	H6/Roboto/Medium/20px		
Subtitle 1	Subtitle 1/Roboto/Regular/16px	Subtitle 1/Roboto/Regular/16px	Subtitle 1/Roboto/Regular/16px
Subtitle 2	Subtitle 2/Roboto/Medium/14px	Subtitle 2/Roboto/Medium/14px	
Body 1	Body 1/Roboto/Regular/16px	Body 1/Roboto/Regular/16px	Body 1/Roboto/Regular/16px
Body 2	Body 2/Roboto/Regular/14px	Body 2/Roboto/Regular/14px	Body 2/Roboto/Regular/14px
Button	BUTTON/ROBOTO/MEDIUM/14PX		
Caption	Caption/Roboto/Regular/12px	Caption/Roboto/Regular/12px	Caption/Roboto/Regular/12px
Overline	OVERLINE/ROBOTO/REGULAR/10PX		

Abbildung 3.11 Wahl der Typografie

3.3.5 Screen-Flow

Wie in der Anforderungsanalyse (siehe 3.3.1) thematisiert, hat die Anwendung eine gewisse Grundeinteilung in Screens bzw. Panels.⁸

⁸ Innerhalb dieser schriftlichen Arbeit werden nur die wichtigsten Screens behandelt. Da weder die Thesis noch der Anhang genügend Platz für eine Präsentation des gesamten Screen-Flows (inkl. Ablaufdiagramm) bietet, kann dieser lediglich extern übermittelt werden. Für eine interaktive Erfahrung wird das zugehörige Bachelorprojekt empfohlen.

3.3.5.1 Dashboard

Das Dashboard enthält allgemeine Informationen für den Nutzer und bietet die Möglichkeit, alle Kunden und Projektpartner zu verwalten. Dazu gehören die Optionen einen Kunden erstellen, bearbeiten oder entfernen zu können.

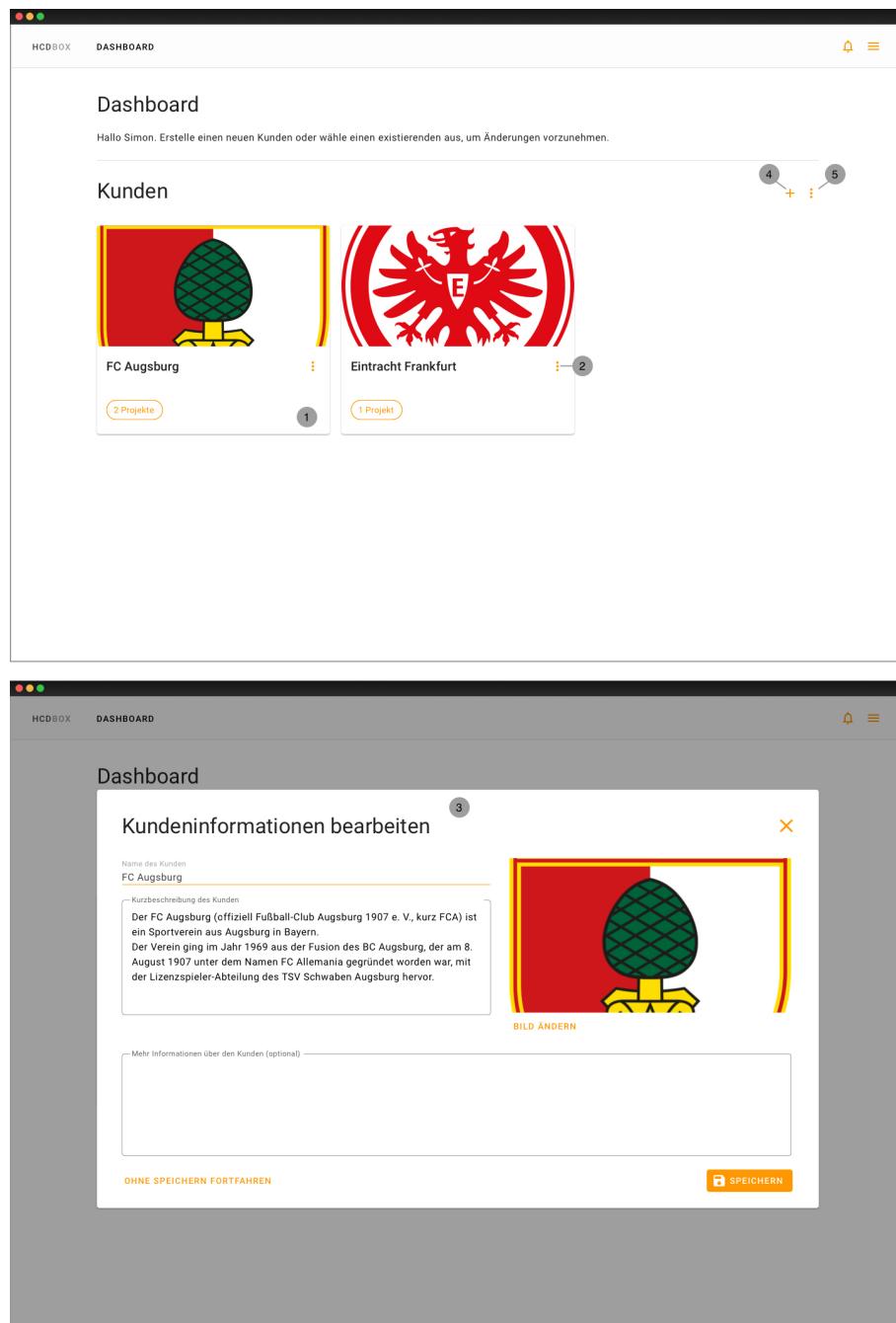


Abbildung 3.12 Dashboard

3 Konzeption

Die Kunden werden standardmäßig als Cards ① aufgelistet. Per Klick auf diese wird das entsprechende Kundenpanel aufgerufen. Über ein Menü ② kann ein Dialogfenster ③ geöffnet werden, welches dem Nutzer ermöglicht, die Informationen des Kunden zu bearbeiten.

Über den „+“-Button ④ können neue Kunden erzeugt werden. Hierfür werden dieselben Informationen durch eine Eingabe benötigt, wie bei einer Informationsänderung. Über das Menü ⑤ kann auf eine Listenansicht gewechselt werden. Dies ermöglicht es, eine Übersicht zu erhalten, sobald eine Vielzahl an Einträgen erstellt wurde. Zusätzlich zur Liste wird eine Sortier- und Filterfunktion geboten, welche den gesuchten Eintrag schneller finden lässt.

Eine weitere Funktionalität ist die manuelle Ordnung. Per Drag-and-Drop kann der User die Cards so anordnen, wie es für ihn am effektivsten ist. Sowohl die Listenansicht, als auch die Filter- und Drag-and-Drop-Funktionalität ist in allen Card-Listen verfügbar.

Über einen Klick auf den entsprechenden Kunden wechselt der Screen auf das Kundenpanel.

3.3.5.2 Kundenpanel

Wie der Name schon sagt, sind im Kundenpanel alle Informationen über den Kunden bzw. Projektpartner nachzulesen und zu bearbeiten. Da oftmals nicht nur ein einziges Projekt eines Kunden verwaltet wird, erhält der Nutzer eine Projektübersicht.

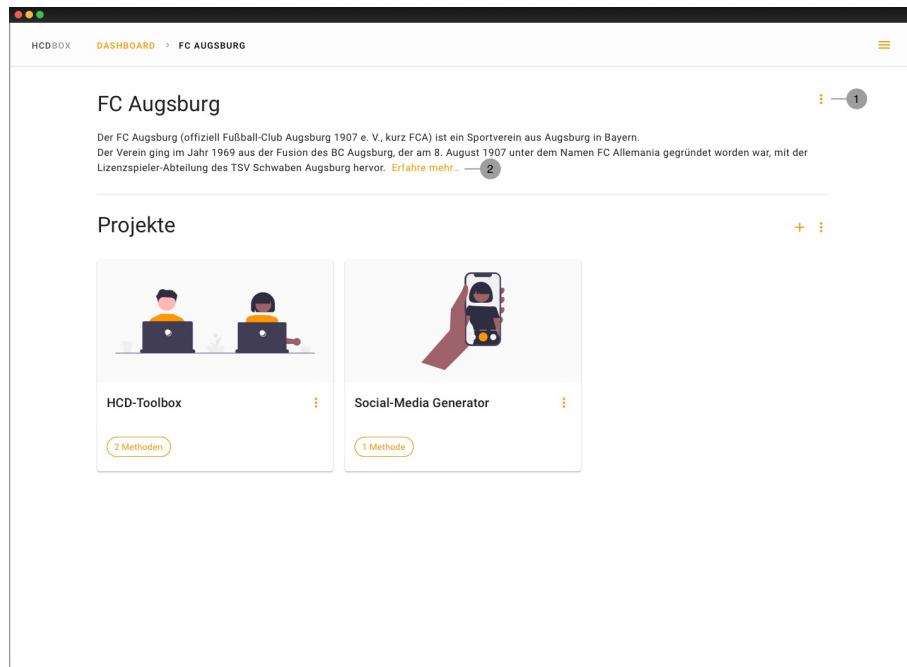


Abbildung 3.13 Kundenpanel

Wie auf dem ersten Blick bereits ersichtlich, ist der Screen ähnlich zum Dashboard aufgebaut. Grund dafür ist, dass die Funktionalitäten äquivalent aufgebaut sind - nur mit Projekten anstelle von Kunden. Der Nutzer kann also bestehende Projekte verwalten (bearbeiten und entfernen) oder neue hinzufügen.

Der einzige Unterschied ist, dass der Anwender hier eine zweite Möglichkeit bekommt, die Kundeninformationen anzupassen ①. Zudem kann er, sofern tiefere Informationen über den Kunden vorhanden sind, diese über den Aufruf eines Dialogfensters einsehen ②.

Klickt der Nutzer auf ein Projekt, erscheint das entsprechende Projektpanel.

3 Konzeption

3.3.5.3 Projektpanel

Im Projektpanel werden die Methoden aufgelistet, welche i. d. R. der Konzepter hinzugefügt hat.

The image shows two screenshots of the HCD-Toolbox application interface.

Methoden Panel (Top Screenshot):

- The title is "HCD-Toolbox".
- The subtitle is "Der Fußballverein FC Augsburg möchte ein Projekt entwickeln, um seine Mitglieder auf dem Laufenden zu halten, Präsenz zu zeigen und gegen die fallenden Mitgliederzahlen vorgehen. Der Verein in Person von Roland (IT-Leiter des Vereins) engagiert die Agentur iconstorm (in Person Jonathan, Jana und Fabian), um eine Strategie zu entwickeln, aus der ein Produkt entstehen soll, welches die Ziele erfüllt. Erfahre mehr...".
- The main section is titled "Methoden". It contains two cards:
 - Szenarios:** Nutzungskontext verstehen und Beschreiben. 10 Ausführungen.
 - Personae:** Nutzungskontext verstehen und Beschreiben. 4 Ausführungen.
- A sidebar on the right shows a context menu with options: "Methodenauswahl bearbeiten" and "Ansicht wechseln".
- An orange button at the bottom left says "+ VERKNÜPFUNGSMATRIX HINZUFÜGEN".

Verknüpfungsmatrizen Panel (Bottom Screenshot):

- The title is "Verknüpfungsmatrizen 2".
- The subtitle is "Wähle zwei Methoden aus um eine Vergleichsmatrix zu generieren. Die Ergebnisse dieser werden sowohl horizontal als auch vertikal markiert. Wähle einen Block einer Verknüpfung an, um den Farbwert zu ändern.".
- The matrix table has "Personae" as Method 1 and "Szenarios" as Method 2.
- The rows and columns are labeled with names: Johannes Dampf, Fabian Kissner, Jana Renk, Roland Speer.
- The matrix cells are colored based on relevance: orange (Sehr wichtig), light orange (Wichtig), and yellow (Unwichtig).
- A context menu is open over the cell for Roland Speer and Szenarios, showing options: "Relevanz bearbeiten", "Sehr wichtig", "Wichtig", "Unwichtig", and "Verknüpfung entfernen".
- An orange button at the bottom left says "+ VERKNÜPFUNGSMATRIX HINZUFÜGEN".
- A legend at the bottom right defines the colors: orange for "Sehr wichtig", light orange for "Wichtig", yellow for "Unwichtig", and grey for "Keine Verknüpfung".

Abbildung 3.14 Projektpanel I

Eine Besonderheit dieser Arbeit kann über den Button „Verknüpfungsmatrix hinzufügen“ ① ergänzt werden. Wie bereits dargelegt, gibt es in konkurrierenden Anwendungen kaum sinnvolle Möglichkeiten, HCD-Methoden zu verknüpfen. Durch die Integration dieser Funktion wird ein neuartiger Mehrwert geschaffen: Verknüpfungsmatrizen ②. Der Nutzer kann dabei zwei Methoden wählen ③, welche dem Projekt zugeordnet wurden und die Verknüpfungen der vorhandenen Ausführungen einsehen. Darüber hinaus kann er die Relevanz einzelner Verknüpfungen manuell anpassen ④.

Der Sachverhalt der präsentierten Verknüpfungsmatrix ② wird anhand eines Beispiels verständlich erläutert:

Die beiden Personae *Jana Renk* und *Roland Speer* haben eine Verknüpfung zum Szenario *Ergebnisübersicht* ④. Nun tritt allerdings der Fall ein, dass *Roland* in diesem Szenario eine untergeordnete Rolle spielt. Für *Jana* ist dieses Szenario aber von signifikanter Bedeutung, da sie in jedem Projekt eine solche Übersicht benötigt. Der Nutzer kann also per Klick die Wichtigkeit der Verknüpfung neu einstufen oder auch entfernen.

Diese Matrix schafft neue Möglichkeiten. Beispielsweise wäre mir mit so einem Tool die Erstellung der Anforderungsanalyse und die MoSCoW-Priorisierung (siehe 3.3.1) leichter gefallen. Je höher die Relevanz ist und je mehr Personae mit einem Szenario verbunden sind, desto wichtiger wird die Implementierung eines Features, welches die Anforderungen erfüllt.

Unabhängig von den Matrizen kann die Methodenauswahl über den vorgeesehenen Menüaufruf ⑤ angepasst werden.

3 Konzeption

The image consists of two vertically stacked screenshots of a software application window titled "HCD-Toolbox".

Screenshot 1 (Top): Methoden auswahl bearbeiten

- Header:** HCD-BOX DASHBOARD > FC AUGSBURG > HCD-TOOLBOX
- Title:** Methoden auswahl bearbeiten
- Section:** Methoden auswahl
 - Icon:** Two user profile icons.
 - Section:** Personae
 - Icon:** A person icon.
 - Description:** Personae sind Steckbriefe von Benutzern und repräsentieren diese mit ihren Wünschen, Zielen und weiteren Eigenschaften im Entwicklungsprozess. Im Grunde stellt eine Persona eine Gruppe von Benutzern in abstrahierter Form dar, welche ähnliche Verhaltensmuster aufweisen. Dies können beispielsweise Motivationen, Präferenzen bei der Produktnutzung oder auch Fähigkeiten sein.
 - Section:** Card Sorting
 - Icon:** Two people standing next to an open book.
 - Description:** Card-Sorting ist eine benutzerorientierte, empirische Usability-Methode zur Generierung von quantitativen (und qualitativen) Aussagen. Sie ermittelt die aus Sicht der User optimale Gruppierung von Inhalten und unterstützt somit bei der nutzerzentrierten Entwicklung von Informationsarchitekturen.
- Buttons:** OHNE SPEICHERN FORTFAHREN (orange), METHODENAUSWAHL SPEICHERN (orange)

Screenshot 2 (Bottom): Methoden auswahl bearbeiten

- Header:** HCD-BOX DASHBOARD > FC AUGSBURG > HCD-TOOLBOX
- Title:** Methoden auswahl bearbeiten
- Section:** Methoden filtern (highlighted with a yellow circle)
- Section:** Projektyp

Wählen die Art des Produkts, an dem gearbeitet wird

 - Physisches Produkt
 - Benutzeroberfläche (Screen Interface)
 - Dienstleistung
- Section:** Ziel

Welches Ziel soll mithilfe der Methoden verfolgt werden?

 - Nutzungskontext verstehen und beschreiben
 - Nutzungsanforderungen festlegen
 - Gestaltungslösungen erarbeiten
 - Gestaltungslösungen evaluieren
- Section:** Teamgröße

Wie viele Personen können bei der Ausführung der Methode zu wesentlichen Teilen (z. B. an der Durchführung) teilnehmen?

 - 1
 - 2
 - 3
 - 4
 - 5 oder mehr
- Section:** Arbeitsstunden

Wie viele Stunden / Mon hours soll die Durchführung der Methode beanspruchen?
- Buttons:** OHNE SPEICHERN FORTFAHREN (orange), METHODENAUSWAHL SPEICHERN (orange)

Abbildung 3.15 Projektpanel II

Hierfür öffnet sich ein Dialogfenster ⑥, welches eine Übersicht aller im System integrierten Methoden bietet. Sowohl ein möglicher Wechsel der Ansicht (wie gewohnt über einen Menüaufruf ⑦) als auch ein manuelles Filtersystem ⑧ erleichtern es dem User, die gewünschten Methoden zu integrieren.

3 Konzeption

3.3.5.4 Methodenpanel Szenarios

The figure consists of two screenshots of a software application titled "HCD-Toolbox".

Szenarios Panel (Top Screenshot):

- Header:** HCD-BOX, DASHBOARD > FC AUGSBURG > HCD-TOOLBOX > SZENARIOS.
- Title:** Szenarios
- Description:** Ein Nutzungsszenario (oder Use Case) ist die narrative Beschreibung der (gedachten) Interaktion eines (potenziellen) Benutzers mit einem interaktiven IT-System. In den Szenarios werden also die Handlungsziele- und -aufgaben der Benutzer definiert. Wie man schon erahnen kann, wird diese Methode oft mit Personae verknüpft. [Erfahren mehr...](#)
- Ausführungen:**
 - Workshop und Zusammenarbeit:** 3, 4, 8, 10, 18, +, 5.
 - Der Fußballverein FC Augsburg möchte ein Projekt entwickeln, um seine Mitglieder auf dem Laufenden zu halten, Präsenz zu zeigen und gegen die fallenden Mitgliederzahlen vorgehen. Der Verein in Person von **Roland** (IT-Leiter des Vereins) engagiert die Agentur iconstorm (in Person **Jonathan, Jana** und **Fabian**), um eine Strategie zu entwickeln, aus der ein Produkt entstehen soll, welches die Ziele erfüllt.**
 - > Verknüpfungen:** 6.
 - Anmelden:** 2, 9.
 - Projekt anlegen:** 11, 12.
 - Kunden anlegen:** 13, 14, 15, 16, 17.
- Buttons:** + GRUPPE HINZUFÜGEN, Alle Verknüpfungen genutzt, 1 ungenutzte Verknüpfung, 2 ungenutzte Verknüpfungen.

Workshop und Zusammenarbeit Panel (Bottom Screenshot):

- Header:** HCD-BOX, DASHBOARD > FC AUGSBURG > HCD-TOOLBOX > SZENARIOS.
- Title:** Workshop und Zusammenarbeit
- Description:** Der Fußballverein FC Augsburg möchte ein Projekt entwickeln, um seine Mitglieder auf dem Laufenden zu halten, Präsenz zu zeigen und gegen die fallenden Mitgliederzahlen vorgehen. Der Verein in Person von **Roland** (IT-Leiter des Vereins) engagiert die Agentur iconstorm (in Person **Jonathan, Jana** und **Fabian**), um eine Strategie zu entwickeln, aus der ein Produkt entstehen soll, welches die Ziele erfüllt.
- Verknüpfungen:** Jonathan Dampf, Fabian Kissner, Jana Renk (7).
- Verknüpfungen bearbeiten:** Search bar.
- Personae:**

Name ↑ 2	Gruppe ↑ 1	Status
<input checked="" type="checkbox"/> Jonathan Dampf	Entwickler und Designer	DETAILED
<input checked="" type="checkbox"/> Fabian Kissner	Entwickler und Designer	RAW
<input checked="" type="checkbox"/> Jana Renk	Entwickler und Designer	WIP
<input type="checkbox"/> Roland Speer	Kunden und weitere Workshopteilnehmer	RAW

Abbildung 3.16 Methodenpanel Szenarios

Im obersten Abschnitt findet der Nutzer eine Kurzbeschreibung der Methode. Auch hier kann er ein Dialogfenster aufrufen ①, welches ihm detaillierte Informationen zur Methode liefert. Dies ist vor allem für Stakeholder sinnvoll, welche eher weniger Erfahrung im Umgang mit HCD-Methoden haben und bei der Ausführung häufiger Unterstützung benötigen. So müssen sie in Workshops seltener die Zeit des Konzepters beanspruchen, wenn Nachfragen zur Anwendung bestehen.

Die Szenarios werden in sogenannte Szenariogruppen ② unterteilt. Jede Szenariogruppe besitzt einen Namen ③ und eine Beschreibung ④, welche per Klick jederzeit bearbeitet werden können. Das Menü ⑤ bietet dem Nutzer die Möglichkeit die Gruppe (in ein Projekt seiner Wahl) zu duplizieren oder sie und ihre Inhalte zu entfernen.

Jede Szenariogruppe kann Verknüpfungen besitzen, welche über ein Erweiterungspanel ⑥ verwaltet werden (rechter Abschnitt des Bildes). Bei einem Klick auf eine hinzugefügte Verknüpfung ⑦ gelangt der Nutzer automatisch zum entsprechenden Ausführungspanel. Außerdem können die vorhandenen Verknüpfungen in der Beschreibung (z. B. mit „@Roland“) getaggt ⑧ werden.

Der Inhalt der Ausführungsgruppen wird durch die Ausführungen selbst ⑨ definiert. Sie werden ebenfalls standardmäßig als Cards repräsentiert, wobei die Ansicht natürlich erneut geändert werden kann ⑩. Die Cards der Ausführungen beinhalten die wichtigsten Informationen der Szenarios: Den Status ⑪ (Detailanzeige erscheint bei Hover), das Titelbild ⑫, den Namen ⑬, die Anzahl der Einträge ⑭, die Beschreibung ⑮ und wie viele Verknüpfungen sich von der Szenariogruppe unterscheiden ⑯. Zu letzterem erfolgt die Erklärung im Ausführungspanel. Möchte man eine Ausführung einzeln duplizieren (in eine gewünschte Szenariogruppe eines beliebigen Projekts) oder entfernen, ist dies über ein Menü ⑰ möglich.

Beim Klick auf eines der vorhandenen Szenarios oder beim Erstellen einer neuen Ausführung ⑯ gelangt der Nutzer zum entsprechenden Ausführungspanel.

3.3.5.5 Ausführungspanel Szenario

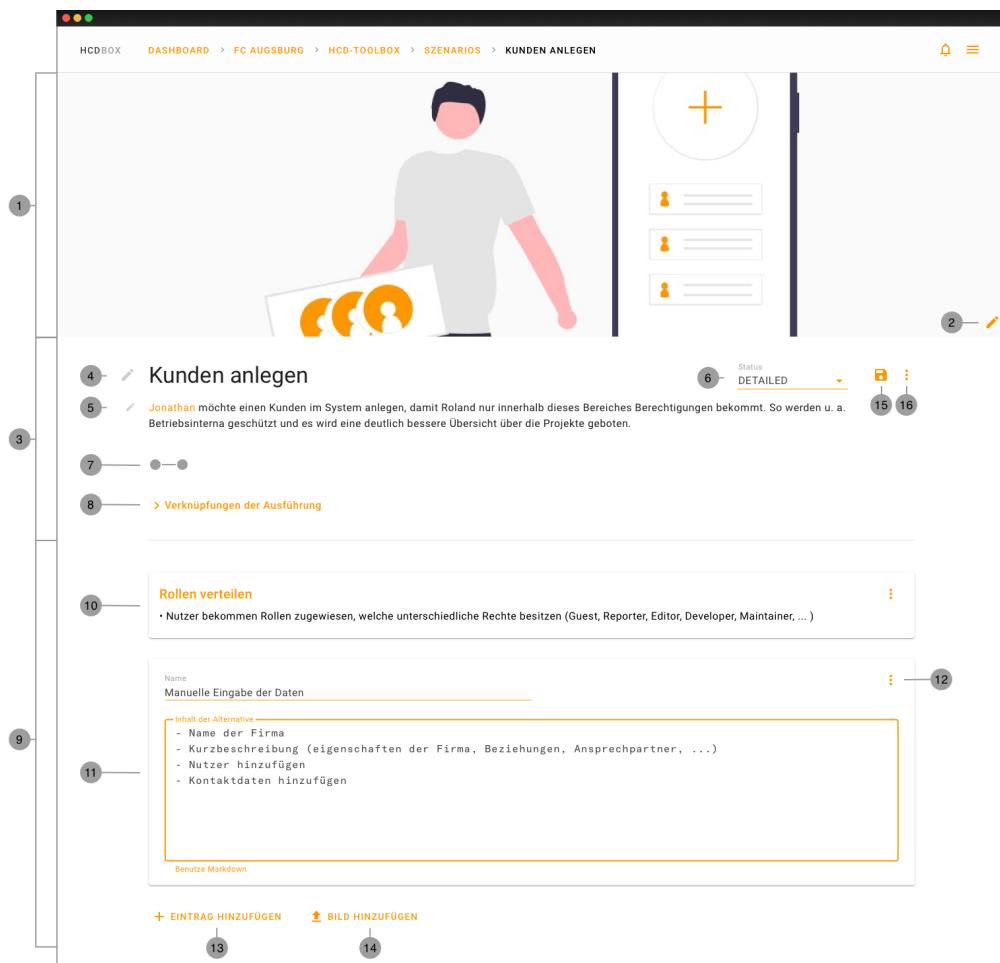


Abbildung 3.17 Ausführungspanel Szenario

Das Ausführungspanel besteht aus drei großen Bereichen:

Titelbild ①

Im obersten Bereich befindet sich das Titelbild des Szenarios. Es kann über einen Klick auf den Button ② oder das Bild selbst geändert bzw. entfernt werden. Für die Änderung öffnet sich ein Dialogfenster, in welchem bereits gespeicherte Fotos auswählbar sind. Auch ein neues Bild kann aus den lokalen Daten des Nutzers hochgeladen werden. Ist beim Speichervorgang kein Titelbild vorhanden, so wird auf einen Platzhalter zurückgegriffen.

Kopfbereich ③

Im Kopfbereich werden die Kerninformationen des Szenarios verwaltet. Diese gleichen den Informationen, welche in den Cards des Methodenpanels sichtbar sind.

Der Name ④ und die Beschreibung ⑤ lassen sich per einfachen Klick in ein Text-Input bzw. Text-Area Element umwandeln und bearbeiten. Für den Status dient ein Dropdown-Menü ⑥. Dort kann sich der Nutzer zwischen den Möglichkeiten „RAW“, „WIP“ und „DETAILED“ entscheiden. Dies hilft dem User dabei schnell zu erkennen, welche Szenarios überarbeitet werden müssen. Auch wird dem Grundsatz der iterativen Bearbeitung (siehe 1.1.1.4) Folge geleistet. Befindet sich das Projekt also wieder in der ersten Phase, so kann jedes Szenario aktualisiert und der Status wenn nötig angepasst werden. Somit kann jeder Stakeholder einsehen, an welchen Ausführungen noch gearbeitet werden muss.

Das einzige Element, welches nicht direkt vom Nutzer beeinflusst werden kann, ist die Perlenkette ⑦. Sie wird automatisch an die Anzahl der Einträge in der Eintragsliste angepasst und macht dem Benutzer des Systems bereits in der Cardansicht deutlich, mit welchem Volumen das Szenario gefüllt ist.

Auch innerhalb des Szenarios können Verknüpfungen abgeändert werden ⑧. Das Erweiterungspanel gleicht jenem der Gruppen im Methodenpanel. Mit einer Ausnahme: Die Auswahl ist begrenzt auf die ausgewählten Ausführungen der Gruppe. Standardmäßig sind alle Verknüpfungen der Gruppe auch in dem Szenario aktiv. Wird jedoch eine Verknüpfung innerhalb des Szenarios gelöst, so wird dies als „ungenutzte Verknüpfung“ gewertet. Hier schließt sich der Bogen zum Methodenpanel: Die Anzahl dieser entfernten Verknüpfungen wird als Tag in der Card im Methodenpanel angezeigt (siehe Abbildung 3.16 ⑯).

Eintragsliste ⑨

Hier werden die Einträge ⑩ des Szenarios gesammelt. Jeder Eintrag besitzt einen Namen und eine Beschreibung. Möchte man diesen Eintrag bearbeiten, reicht ein Klick auf seine Vorschau. Um die Eigenschaften bearbeiten zu können, werden

auch diese in ein Text-Input bzw. Text-Area Feld umgewandelt ¹¹. In der Text-Area ist es möglich Markdown zu verwenden, dessen Inhalt das System wiederum automatisch in HTML-Code übersetzt. Ist die Bearbeitung zufriedenstellend, reicht ein Klick außerhalb des Feldes aus, um die Änderungen zu übernehmen. Soll der Eintrag entfernt werden, kann dies über das gewohnte Menü ¹² erledigt werden.

Zum Hinzufügen von Feldern hat der Anwender zwei Möglichkeiten:
Eintrag hinzufügen ¹³: Fügt eine Card hinzu (wie in der Grafik präsentiert).
Bild hinzufügen ¹⁴: Lässt den Nutzer über das übliche Dialogfenster ein vorhandenes Bild auswählen oder ein neues hochladen.

Wie auch bei den Auflistungen durch Cards, können die Benutzer hier per Drag-and-Drop ihre eigene Sortierung vornehmen.

Die Änderungen im Ausführungspanel werden lokal gespeichert. Dies hat den Vorteil, dass Neuerungen, welche sich bei der Bearbeitung doch als ungewollt herausstellen, verworfen werden können. Dieses Vorgehen erwies sich in Anwendungstests als vorteilhafter. Dem Nutzer wird über einen Speicherbutton ¹⁵ die Möglichkeit gegeben, die Änderungen nach Abschluss seiner Bearbeitung zu sichern. Versucht er die Seite zu verlassen, wird er gewarnt, dass alle ungespeicherten Anpassungen verworfen werden, wenn er diese Aktion ausführt.

Das Szenario über ein Menü ¹⁶ auch innerhalb des Ausführungspanels dupliziert, exportiert oder gelöscht werden.

3 Konzeption

3.3.5.6 Methodenpanel Personae

Das Methodenpanel der Personae gleicht im Grunde dem der Szenarios. Auch hier liefert das System dem Nutzer nützliche Informationen über die Methode und Gruppen können verwaltet werden. Unterschiede lassen sich nur in den Gruppen selbst wiederfinden.

The screenshots illustrate the 'Personae' feature in the HCD-Toolbox. The top screenshot shows the main interface for managing user profiles. The bottom screenshot shows the process of selecting a template for creating or editing a persona.

Top Screenshot: Personae Overview

- Ausführungen:** Shows three profiles:
 - Jonathan Dampf:** Konzepter | 50, 3 leere Felder
 - Fabian Kissner:** Auszubildender | 19
 - Jana Renk:** Visual Designerin | 30
- A context menu is open over the first profile, showing the following options:
 - Gruppentemplate bearbeiten (marked with 2)
 - Gruppentemplate auf andere Gruppen anwenden (marked with 5)
 - Gruppe duplizieren
 - Gruppe entfernen

Bottom Screenshot: Gruppentemplate bearbeiten (Edit Template) Modal

The modal window has the following structure:

- Titel:** Gruppentemplate bearbeiten
- Beschreibung:** Das Template dient als Grundlage für die Personenerstellung. Alle Personae besitzen die gleichen Elemente (z. B. „Profilbild“). Wird ein Element in der Bearbeitung einer Persona verändert (Position, Größe, Name) oder entfernt, werden auch die Elemente alle anderen Personae in der Gruppe angepasst. So wird garantiert, dass alle Ausführungen einheitlich gehalten werden. Es wird also dazu geraten, das Template zu Beginn auszuwählen und anzupassen.
- Aus vorhandenen Templates wählen:** Zwei Vorschläge sind angezeigt:
 - Blanko Persona (Standard)** (marked with 3)
Diese Vorlage enthält alle Abschnitte, die Sie für die Erstellung einer Persona benötigen könnten, und Sie können jederzeit eigene Abschnitte hinzufügen, wenn Sie möchten.
 - Buyer Persona** (marked with 4)
Buyer Personas sind für jedes Unternehmen einzigartig und sind das Ergebnis einer sorgfältigen Marktforschung und der Analyse Ihrer realen Kunden. Diese Vorlage zeigt Ihnen alle Informationen, die Sie aufnehmen müssen, und gibt Ihnen einige nützliche Hinweise, worauf Sie achten sollten.
- Aktionen:** Ein Button 'TEMPLATE VERWENDEN' am unteren rechten Rand.

Abbildung 3.18 Methodenpanel Personae

3 Konzeption

Die Personaegruppen besitzen keine Verknüpfungen, was daran liegt, dass Szenariogruppen direkt mit den Personae selbst verknüpft werden. Die Gründe dafür wurden bereits in der Beschreibung des ER-Diagramms erläutert (siehe 3.3.3).

Auch die Informationen ①, welche innerhalb der Cards dargestellt werden, unterscheiden sich von denen, der Szenarios, da Personae andere Grundinformationen besitzen. Woher diese Informationen stammen wird im Ausführungspanel der Personae erklärt.

Neu ist die Möglichkeit Gruppentemplates zu bearbeiten ②. Jede Persona in einer Personaegruppe enthält dieselben Felder (wie z. B. *Name* oder *Beruf*). Wird ein Feld innerhalb einer Persona hinzugefügt, bearbeitet (z. B. *Beruf* wird zu *Berufsbezeichnung* oder die Positionierung wird geändert) oder entfernt, passen sich die Felder in den anderen Ausführungen an. Es wird also empfohlen das Gruppentemplate zu Beginn der Bearbeitung zu wählen. I. d. R. ist das ohnehin das Blanko-Template ③, weshalb es als Standardtemplate definiert ist. Der Grund dafür ist, dass die Personae von Projekt zu Projekt unterschiedlich aufgebaut sind. Ein passendes Template existiert also normalerweise nicht. In speziellen Fällen können aber auch die gespeicherten Templates (wie z. B. die *Buyer Persona* ④) als Grundlage für die Bearbeitung genutzt werden. Eine Bearbeitung erfolgt, wie bereits beschrieben, in der Persona selbst.

Die Felder und dessen Eigenschaften können auch projektunabhängig auf andere Gruppen übertragen werden ⑤.

3 Konzeption

3.3.5.7 Ausführungspanel Persona

Auch hier ähnelt das Erscheinungsbild dem Ausführungspanel der Szenarios.

The image consists of two vertically stacked screenshots of a software application window titled "HCD-Toolbox".

Screenshot 1 (Top): This shows the main "PERSONAE" view for "JANA RENK". At the top, there is a large portrait of a woman with brown hair. Below the portrait, the name "Jana Renk" is displayed, followed by a link "Verknüpfungen". The interface is organized into several sections:

- Beruf:** Visual Designerin (marked with circle 1)
- Alter:** 35 (marked with circle 2)
- Technologie:** Handy (iOS), Laptop (MacOS)
- Ein typischer Arbeitstag:** A list of tasks for the morning (marked with circle 3).
 - 08:00 - 10:00
 - Tägliches Meeting am Morgen
 - Organisiert die Aufgaben für den Tag
 - Verschafft sich Informationen aus den digitalisierten Überblicken der Methodenergebnisse
 - 10:00 - 17:00
 - Sucht nach Zusammenhängen und Erkenntnissen des Workshops
 - Leitet daraus ab, welche Ergebnisse das Projekt erzielen muss
- In meiner Tasche:** Icons for a backpack, sunglasses, and gum (marked with circle 4).

Screenshot 2 (Bottom): This shows a modal dialog titled "Änderungen speichern" (Save changes) over the main view. The dialog contains the following fields:

- Als neues Template speichern
- Vorschaufelder:** A note stating that preview fields are used as descriptions in cards. It lists "Alter" and "Beruf" as selected fields (marked with circle 6).
- OHNE SPEICHERN FORTFAHREN:** A button to continue without saving.
- Angabe der Felder:** A dropdown set to "innerhalb der Gruppe".
- Status:** A dropdown set to "WIP".
- SPEICHERN:** A button to save.

The background of the modal shows the same persona details and "In meiner Tasche" section as the main view.

Abbildung 3.19 Ausführungspanel Personae

Wie in der Definition der Personae (siehe 1.1.3.1) thematisiert, sind lediglich der Name und das Bild bei der Erstellung verpflichtend. Deshalb finden sich diese Optionen auch im Kopfbereich wieder. Eine Beschreibung an sich gibt es nicht, da sich die Eigenschaften aus den Feldern ergeben sollen.

Das Editieren und Entfernen der Felder erfolgt ähnlich wie die Bearbeitung der Einträge in den Szenarios (je nach Art des Feldes). Des Weiteren kommt die Funktionalität hinzu, dass die Felder innerhalb des Rasters frei in der Größe und Positionierung sind. So lässt sich ein individuelles Design erzeugen. Die Positionierung erfolgt über ein simples Drag-and-Drop System, während die Skalierung an den Rändern des Feldes angepasst werden kann.

Beim Hinzufügen neuer Felder hat der Nutzer derzeit eine Auswahl zwischen *Text Input* ①, *Text Field* ②, *Range Slide* ③ und *Bildupload* ④. Das *Text Field* kann mit Markdown gefüllt werden, was eine Vielfalt unterschiedlichster Formatierungen ermöglicht. Sollten bei der Bearbeitung einer Persona spezielle Feldtypen benötigt werden, so lassen sich diese durch spätere Erweiterungen der Anwendung problemlos ergänzen.

Der Speichervorgang besitzt einige spezielle Funktionen. So ist es dem Nutzer möglich, die aktuellen Einstellungen als neues Template zu speichern ⑤, damit es bei einer künftigen Nutzung (z. B. außerhalb des Projektes) vorhanden ist. Auch werden hier die Vorschaufelder ⑥ definiert, welche die Beschreibung in den Cards des Methodenpanels umfassen. Zur Auswahl stehen dabei alle Felder des Typs *Text Field*. Ebenfalls steht zur Wahl, auf welche Personae sich die Änderungen der Feldeigenschaften auswirken sollen (alle Personae innerhalb der gleichen Gruppe oder auch diejenigen, welche sich in einer anderen Personaegruppe befinden) ⑦.

3 Konzeption

3.3.5.8 Kollaboration

Da meine Anwendung das asynchrone und kollaborative Arbeiten ermöglicht sind einige Elemente vorhanden, die dies fördern oder kennzeichnen.

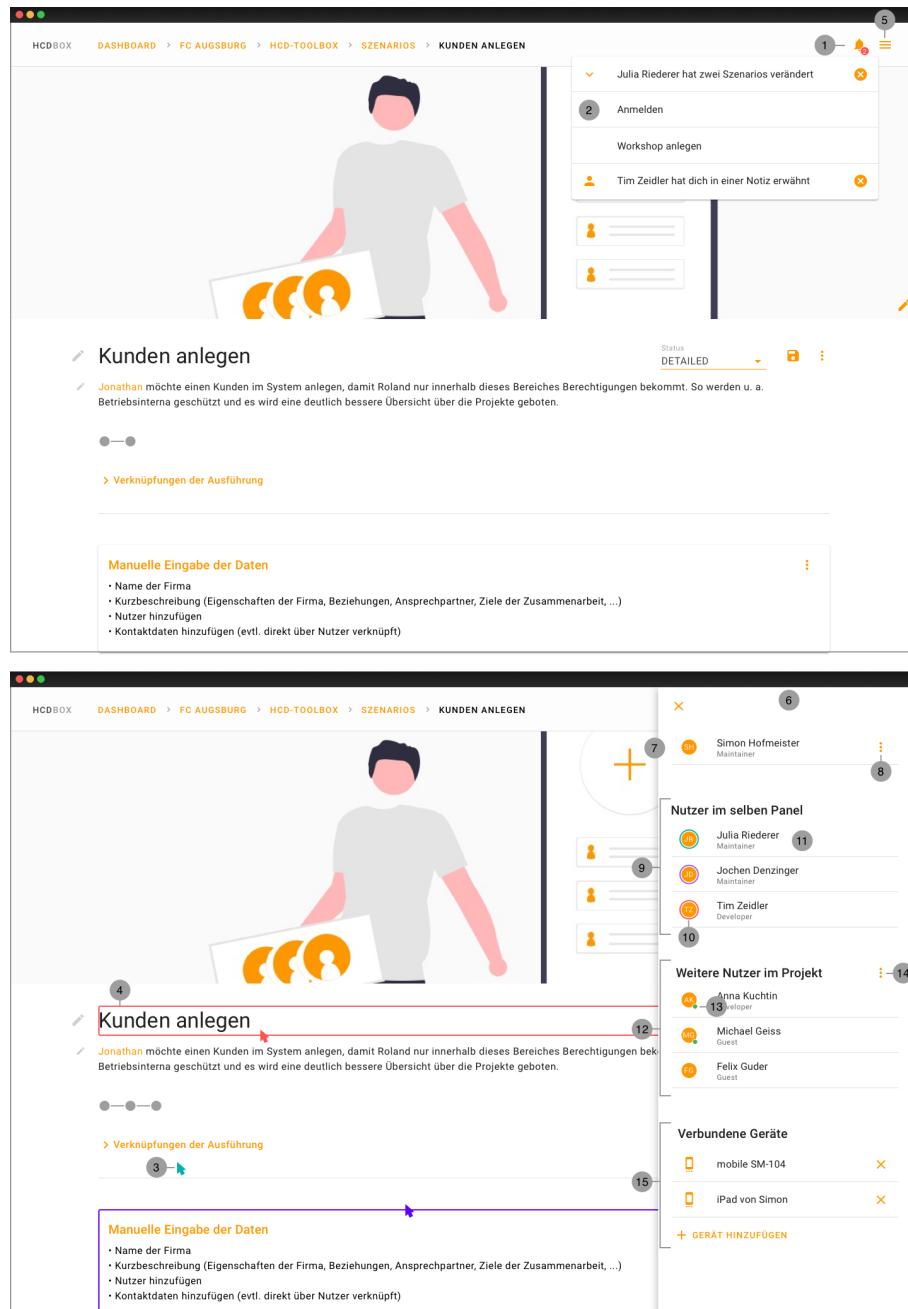


Abbildung 3.20 Features zur Kollaboration

Über das Glocken-Icon ① im Header sind Neuigkeiten einsehbar. Um auf den neuesten Stand zu bleiben, erhält der User beispielsweise Einsicht darüber, ob eine Ausführung in seiner Abwesenheit verändert wurde oder ob er in Notizen erwähnt wurde. Folgt der Nutzer der Meldung ② oder besucht die entsprechende Seite zum ersten Mal nach der Änderung, so werden ihm, ähnlich wie beim System von Miro (siehe 2.1.3.1) die Anpassungen angezeigt.

Befindet sich ein oder mehrere andere User auf dem selben Screen, so wird deren Mausposition mithilfe eines farbigen Mauszeigers ③ dargestellt. Bearbeitet einer dieser Nutzer gerade ein Feld wird dies ebenfalls hervorgehoben ④.

Bei Aufruf des Menüs ⑤ öffnet sich ein Drawer ⑥ an der rechten Bildschirmseite. Dieser wird in verschiedene Bereiche unterteilt:

Profil ⑦

Hier lassen sich Informationen über das eigene Profil einsehen. Über den Menüaufruf ⑧ auf der rechten Seite kann der Nutzer die Profileinstellungen (z. B. Anmeldedaten, Profilbild, Benutzerinformationen) anpassen oder sich vom System abmelden.

Nutzer im selben Panel ⑨

Wie der Name schon sagt, werden hier die aktiven Nutzer aufgelistet, welche sich auf demselben Screen befinden, wie der User selbst. Hier findet man auch eine farbliche Zuordnung der Mauszeiger im Bildschirm zum entsprechenden Profil ⑩. Klickt man auf ein angezeigtes Profil ⑪ kann man diesem Nutzer folgen. Das bedeutet, dass die Ansicht dieses Nutzers auf den eigenen Bildschirm gespiegelt wird. Verlässt er das aktuelle Panel, geschieht das auch bei jedem Nutzer der ihm folgt.

Weitere Nutzer im Projekt ⑫

Hier befinden sich alle Profile von Nutzern, die Berechtigungen in diesem Projekt besitzen, sich allerdings nicht auf dem aktuellen Panel des Users aufhalten. Über ein kleines Symbol ⑬ wird angezeigt ob der zugehörige Nutzer aktiv ist. Wenn das der Fall ist, kann ihm auch hier über einen Klick auf das Profilelement gefolgt

werden. Unter Zuhilfenahme des externen Menüs ¹⁴ ist es möglich alle aktiven Benutzer auf den eigenen Screen zu versammeln. Natürlich werden diese vor Verlassen des Panels gefragt, ob sie dies auch möchten oder ob ungespeicherte Änderungen gesichert werden sollen.

Verbundene Geräte ¹⁵

Dieser Abschnitt ist für künftige Features vorgesehen. Hierfür kann eine mobile Anwendung entwickelt werden, welche mit dem System verbunden werden kann (siehe 5.0.2.4).

4 Technische Umsetzung

Da ich vor Anfertigung dieser Arbeit kaum Erfahrung mit datenbasierter Entwicklung hatte, musste ich mich über die einzelnen Bestandteile der technischen Umsetzung informieren. So lernte ich viele neue Systeme kennen und konnte im Zuge der Recherche einige Vor- und Nachteile abwägen. Letztendlich legte ich mich auf eine Auswahl an Integrationen fest, welche mir dabei halfen, einen zukunftsorientierten Prototypen zu erstellen.

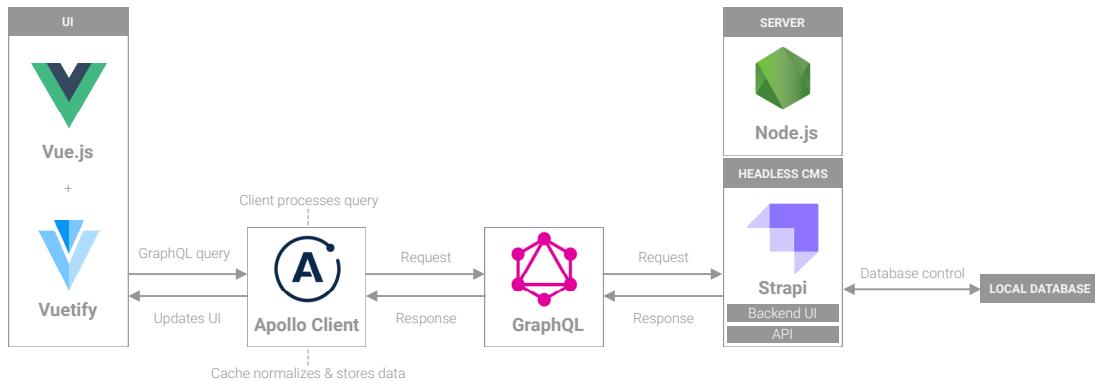


Abbildung 4.1 Integrierter Tech-Stack

4.1 User Interface

4.1.1 Vue

Grundvoraussetzung der Anwendung war ein komponentenbasierter Aufbau des User Interface (UI). Da ich allerdings kaum Erfahrungen in diesem Bereich hatte, musste ich mir die Fähigkeiten eines Frameworks aneignen, weshalb ich einige Alternativen miteinander verglich. Aufgrund der Größe der Community, dem Umfang der Dokumentation und vor allem den Erfahrungswerten von Betreuern, habe ich mich für Vue.js entschieden. Dafür habe ich mir vor allem Meinungen der Agentur iconstorm eingeholt, da die dort beschäftigten Entwickler in den

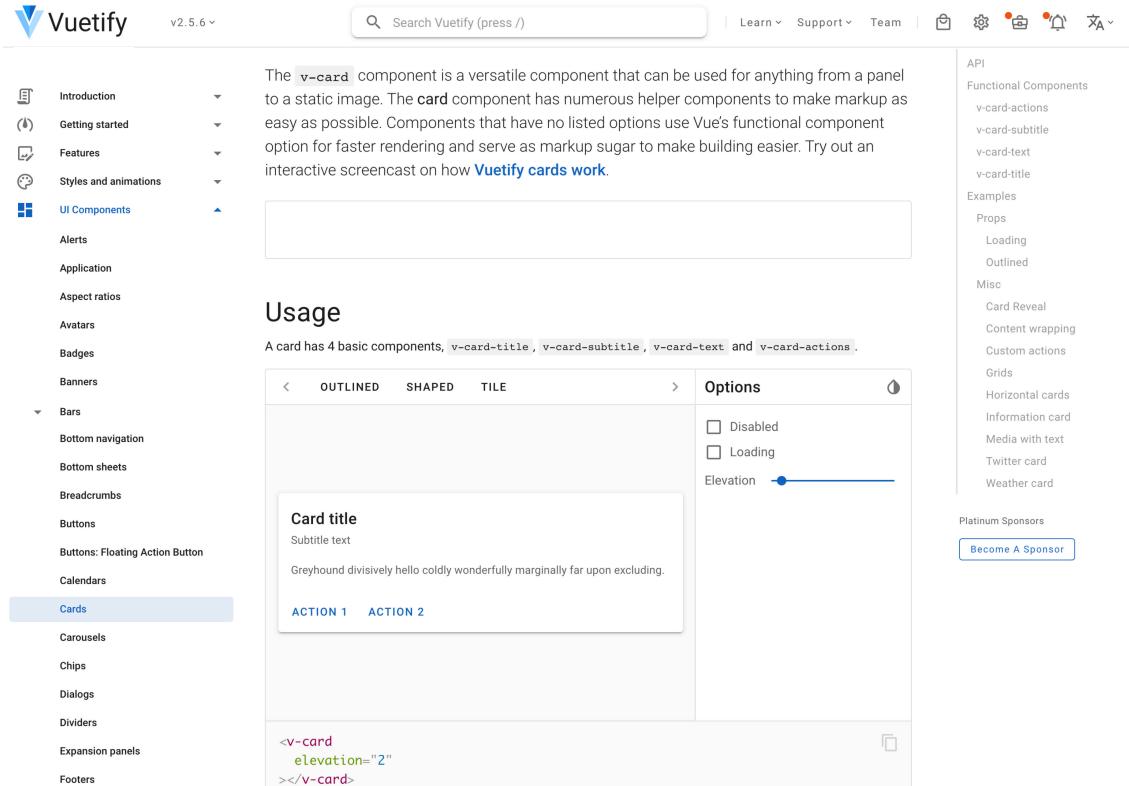
Weiterentwicklungsprozess der Anwendung miteinbezogen werden.

Vue.js fungiert als clientseitiges JavaScript-Webframework und ist dank des eigenen Command Line Interface (CLI) besonders anfängerfreundlich. Mit wenigen Konsolen-Befehlen kann eine erste Anwendung erstellt werden. Auch die Integration von Erweiterungen, wie dem Vue-Router oder Vuetify, ist dank der CLI äußerst simpel und benutzerfreundlich.

Link: <https://vuejs.org/>

4.1.2 Vuetify

Vuetify fungiert als Vue UI Library und macht es möglich, Komponenten in das Framework zu integrieren, welche den Standards von Material Design (siehe 3.3.4) folgen.



The screenshot shows the Vuetify documentation website for version v2.5.6. The left sidebar has a navigation menu with categories like Introduction, Getting started, Features, Styles and animations, and UI Components (which is currently selected). Under UI Components, there are sub-categories such as Alerts, Application, Aspect ratios, Avatars, Badges, Banners, Bars, Bottom navigation, Bottom sheets, Breadcrumbs, Buttons, Buttons: Floating Action Button, Calendars, Cards (which is also selected), Carousels, Chips, Dialogs, Dividers, Expansion panels, and Footers. The main content area has a search bar at the top. Below it, there is a detailed description of the `v-card` component, stating it's a versatile component for anything from a panel to a static image. It mentions numerous helper components to make markup as easy as possible. A callout box highlights the `v-card` component. Below this, there's a section titled "Usage" with examples of different card types: OUTLINED, SHAPED, TILE, and a larger example of a card with a title, subtitle, and actions. To the right of the usage examples is a "Options" panel with checkboxes for Disabled and Loading, and a slider for Elevation. At the bottom, there's a code snippet showing the `<v-card elevation="2"></v-card>` syntax. The right sidebar contains links to the API (Functional Components, v-card-actions, v-card-subtitle, v-card-text, v-card-title), Examples, Props, Loading, Outlined, Misc, Card Reveal, Content wrapping, Custom actions, Grids, Horizontal cards, Information card, Media with text, Twitter card, Weather card, and Platinum Sponsors.

Abbildung 4.2 Screenshot der Dokumentation von Vuetify

Die Library hat, wie auch Material Design, eine sehr ausführliche und interaktive

Dokumentation, welche die Komponenten und deren Integrierungsmöglichkeiten beschreibt. Sowohl die präsentierten Beispiele als auch die detaillierten Informationen zur Application Programming Interface (API) erleichterten mir die praktische Arbeit.

Link: <https://vuetifyjs.com/>

4.2 Datenverwaltung und Kommunikation

4.2.1 Strapi

Strapi ist ein sogenanntes Headless-Content Management System (CMS). Die Idee Strapi zu verwenden, entstand im Laufe des Projekts, da mir das System einige Arbeiten abnahm und die Datenverwaltung zu Beginn ohne Code ermöglichte.

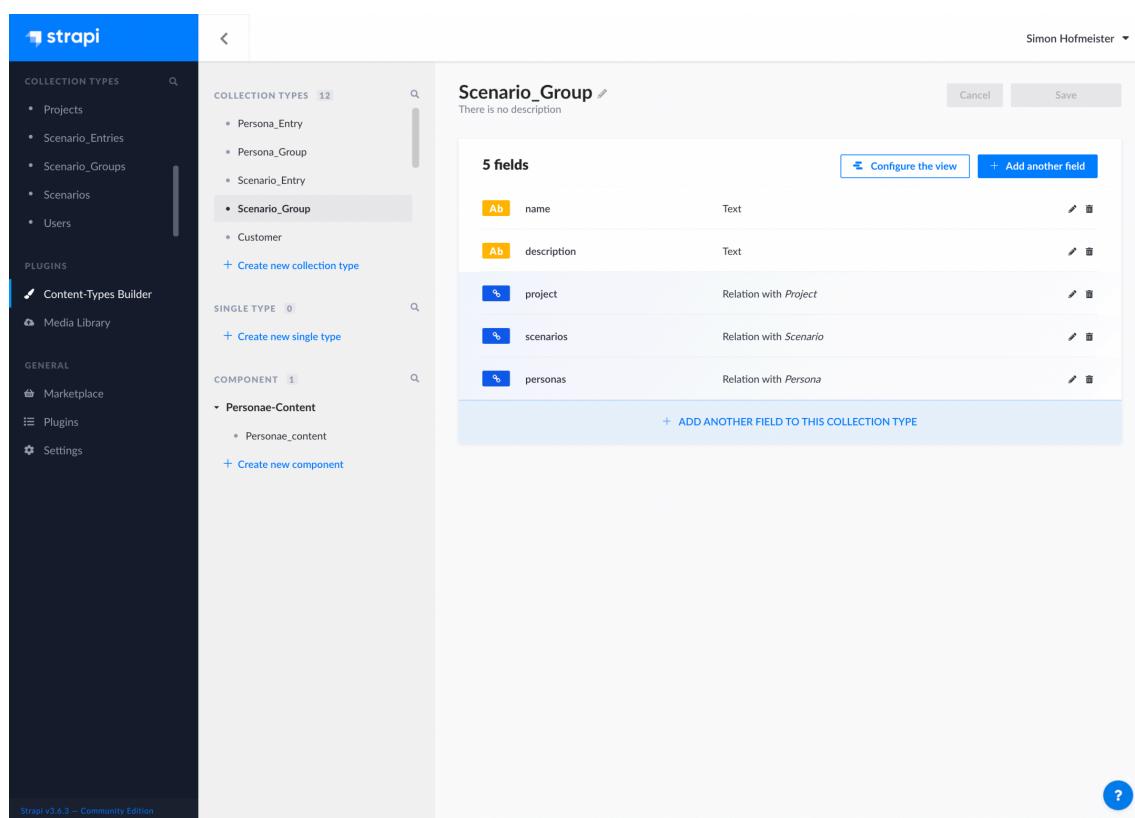


Abbildung 4.3 Screenshot einer Collectionverwaltung in Strapi

So konnte ich meine Collections im Backend UI anlegen und diese mithilfe von Relations miteinander verknüpfen. Diese Arbeitsweise half mir vor allem dabei,

meine Anwendung bereits in einer frühen Phase datenbasiert zu testen, da mit wenigen Aktionen Einträge erstellt, bearbeitet und entfernt werden können.

Die Daten selbst werden im aktuellen Stand des Projekts noch lokal mithilfe von SQLite¹, gespeichert. Eine künftige Entwicklung sieht die Datenverwaltung auf einem Server vor (siehe 5.0.2.3).

Link: <https://strapi.io/>

4.2.2 GraphQL

GraphQL ist eine, von Facebook veröffentlichte[Fac15], Datenabfrage- und Manipulationssprache, sowie ein „Laufzeitsystem zum Beantworten von Abfragen mit vorhandenen Daten“[Gra].

Mithilfe von GraphQL ist es mir also möglich, die Inhalte von Strapi abzurufen und zu verändern.

Ein großer Vorteil von GraphQL ist, dass genau bestimmt werden kann, welche Daten abgefragt werden - nicht mehr und nicht weniger. Somit müssen keine Informationen im System gespeichert werden, welche gar nicht benötigt werden. Eine Beispielabfrage sieht folgendermaßen aus:

```
query ScenarioGroups() {  
  scenarioGroups(where: { project: 3 }) {  
    name  
  }  
}
```

Mit dieser Abfrage wird lediglich der Name einer Szenariogruppe mit der ID „3“ abgefragt. Die weiteren Inhalte der Collection (z. B. die Beschreibung) werden nicht berücksichtigt. Auch die Veränderung der Daten (Mutations) stellt nach kurzer Einarbeitung keine Probleme dar. So können neue Daten erzeugt, bearbei-

¹ SQLite: Eine „C-Sprachbibliothek, die eine kleine, schnelle, in sich geschlossene, hochzuverlässige SQL-Datenbank-Engine mit vollem Funktionsumfang implementiert“[Hip].

tet oder entfernt werden.

Ein weiterer Vorteil ist es, dass GraphQL einen eigenen Editor mitliefert, in welchem die Abfragen ausgeführt werden können. So benötigt es keine Implementierung in den Code, um die Abfragen auf ihre Korrektheit zu überprüfen. Dem Nutzer werden zudem vordefinierte Codeauszüge vorgeschlagen, welche Zeit in der Recherche sparen. Über einfache Tastenkombinationen lässt sich so schnell herausfinden, welche Variablen zur Verfügung stehen.

Link: <https://graphql.org/>

4.2.3 Apollo Client ^(A)

Apollo dient dazu „APIs, Datenbanken und Microservices in einem einzigen Datendiagramm“[Apoa] zu kombinieren. Mithilfe des Apollo Clients, einer Statusverwaltungsbibliothek für JavaScript, können Daten mit GraphQL verwaltet werden. So ist es mir möglich, im Frontend die Daten der Anwendung abzurufen, zwischenzuspeichern und zu verändern, während die Benutzeroberfläche automatisch aktualisiert wird [Apob].

Die zuvor gezeigte Beispielabfrage kann mithilfe von Apollo Client folgendermaßen integriert werden:

```
apollo: {
  scenarioGroups: {
    query: gql`query ScenarioGroups($projectId: ID!) {
      scenarioGroups(where: { project: $projectId }) {
        name
      }
    },
    variables() {
      return {
        projectId: this.projectId,
      };
    },
  },
}
```

Hierbei ist es mir möglich, die Variablen (in diesem Fall `this.projectId`) zu verwenden, welche mir im User-Interface bereitstehen. Die Ergebnisse der Abfrage werden anschließend in einer Variable (`scenarioGroups`) hinterlegt. Der Moment der Abfrage kann frei definiert werden, womit auch erneute Aufrufe möglich sind.

Link: <https://www.apollographql.com/>

4.2.4 Node

Um Strapi in das Projekt zu integrieren, wird ein Webserver benötigt. Dafür nutzte ich Node.js, eine plattformübergreifende Laufzeitumgebung. Über diese Alternative besitze ich die meiste Erfahrung und konnte zudem meine vorhandenen JavaScript-Kenntnisse nutzen.

Link: <https://nodejs.org/>

5 Resümee und Ausblick

5.0.1 Zusammenfassung

Sowohl der konzeptionelle als auch technische Teil meiner Arbeit bieten eine gute Grundlage, um die Anwendung weiterzuentwickeln. Das zugehörige Bachelorprojekt legt zudem die Machbarkeit mit einer prototypischen Umsetzung der erforschten Ideen dar und bietet die Möglichkeit für eine erste vielversprechende Evaluierung.

Die Kombination und Zentralisierung der auf dem Markt verfügbaren Vergleichsprodukte ist ein erheblicher organisatorischer Fortschritt. Einen Mehrwert bietet dann die daraus folgende Möglichkeit beliebige Methoden mithilfe von Verknüpfungen zu kombinieren und somit eine neue Sicht auf die Relevanz der einzelnen Ausführungen zu schaffen.

Zusammenfassend kann durch die Nutzung meiner Anwendung die Arbeit von Kooperationspartnern und diversen Abteilungen der Agentur nicht nur erleichtert, sondern auch stark verbessert werden.

5.0.2 Ausblick

Doch um die Anwendung für den regelmäßigen Gebrauch bereitstellen zu können, müssen noch Verbesserungen vorgenommen werden:

5.0.2.1 Weitere Methoden

Natürlich sind die beiden Methoden Szenarios und Personae nur ein Bruchteil der Methoden, welche im HCD-Prozess Verwendung finden. Allerdings bietet das System eine sehr gute Grundlage, neue Methoden zu integrieren und Mehrwerte daraus zu erkennen.

5.0.2.2 Designsystem

Die Integration von Vuetify bzw. Material Design bietet in der Erstellung des Prototypen zwar Vorteile, weil diese die Arbeit beschleunigen, allerdings kann die Gestaltung noch verbessert und enger auf das Projekt zugeschnitten werden. Eine Möglichkeit wäre es, die eingebundenen Elemente in deren Details zu überarbeiten. Alternativ kann für langfristige Unabhängigkeit ein eigenes Designsystem entwickelt und integriert werden. Dieser Vorgang würde maximale Flexibilität und Performance gewährleisten.

5.0.2.3 Kooperatives Arbeiten

Die Anwendung kann aktuell aufgrund der Datenverwaltung nur lokal verwendet werden. Um kooperatives Arbeiten zu ermöglichen, muss daher eine zentrale Datenbank eingerichtet werden.

Zudem muss für eine reale Veröffentlichung eine ausgereifte Nutzerverwaltung konzipiert und umgesetzt werden, welche beispielsweise Anmeldefenster und die Vergabe von Rechten an diverse Profile umfasst.

5.0.2.4 Weitere Ideen

Nicht alle im Screen-Flow behandelten Ansätze und Features konnten im Prototypen realisiert werden. So steht beispielsweise eine begleitende mobile Anwendung aus, die das schnelle hinzufügen von Skizzen oder Fotos ermöglichen könnte. Auch das versenden von Push-Benachrichtigungen und die Anfertigung von Notizen wären über externe Geräte möglich.

Zudem können einige Einstellungs- oder Exportfunktionen noch in ihrer grafischen Umsetzung durchdacht werden. Allerdings sind diese bereits mithilfe von Platzhaltern angedacht, was eine spätere Integration in die einzelnen Views erleichtert.

5.0.3 Iterativer Prozess

Gerade aus der behandelten Thematik dieser Arbeit geht hervor, dass ein Projekt diesen Ausmaßes im aktuellen Stadium nicht als „fertig“ angesehen werden kann.

5 Resümee und Ausblick

Selbst wenn die eben genannten Ideen in die Tat umgesetzt werden, befindet sich die Anwendung und dessen Konzept am Anfang eines langen Prozesses. Im nächsten Schritt steht eine ausführliche Evaluierung durch die Prüfer an, denen diese Arbeit nun übergeben wurde.

Im weiteren Verlauf wird der Kreislauf fortgesetzt und die gesamte Konzeptionsarbeit wird erneut durchlaufen, bis ein optimales Ergebnis erzielt wird, welches die Arbeit mit HCD-Methoden zukunftsweisend verbessern kann.

6 Literatur

- [Apoa] Apollo Graph Inc. *Apollo GraphQL | Apollo Data Graph Platform— unify APIs, microservices, and databases into a data graph that you can query with GraphQL*. URL: <https://www.apollographql.com/> (besucht am 14.07.2021).
- [Apob] Apollo Graph Inc. *Introduction to Apollo Client - Client (React) - Apollo GraphQL Docs*. URL: <https://www.apollographql.com/docs/react/> (besucht am 14.07.2021).
- [Bau14] Ralf Baumann. *Von der User Story zum Use Case. Agil und langfristig*. informatik aktuell. Okt. 2014. URL: <https://www.informatik-aktuell.de/entwicklung/methoden/von-der-user-story-zum-use-case-agil-und-langfristig.html> (besucht am 06.10.2014).
- [Col] Coley Consulting. *MoSCoW Priorisation method*. URL: <https://www.coleyconsulting.co.uk/moscow.htm> (besucht am 19.07.2021).
- [DIN16] DIN Deutsches Institut für Normung e.V. *Ressourcenschonende Anwendung von Methoden und Werkzeugen zur menschenzentrierten Gestaltung gebrauchstauglicher interaktiver IT-Systeme (DIN SPEC 91328:2016-02)*. Norm. Feb. 2016.
- [DIN20] DIN Deutsches Institut für Normung e.V. *Ergonomie der Mensch-System-Interaktion – Teil 210: Menschzentrierte Gestaltung interaktiver Systeme (ISO 9241-210:2019); Deutsche Fassung EN ISO 9241-210:2019*. Norm. März 2020.
- [Duda] Duden Redaktion. *Duden | Betaversion | Rechtschreibung, Bedeutung, Definition, Herkunft*. Duden online. URL: <https://www.duden.de/rechtschreibung/Betaversion> (besucht am 13.07.2021).
- [Dudb] Duden Redaktion. *Duden | Briefing | Rechtschreibung, Bedeutung, Definition, Herkunft*. Duden online. URL: <https://www.duden.de/rechtschreibung/Briefing> (besucht am 13.07.2021).

6 Literatur

- [Dudc] Duden Redaktion. *Duden | Stakeholder | Rechtschreibung, Bedeutung, Definition, Herkunft*. Duden online. URL: <https://www.duden.de/rechtschreibung/Stakeholder> (besucht am 13.07.2021).
- [Fac15] Facebook Engineering. *GraphQL: A data query language*. Facebook. Sep. 2015. URL: <https://engineering.fb.com/2015/09/14/core-data/graphql-a-data-query-language/> (besucht am 25.07.2021).
- [Gooa] Google Inc. *Homepage - Material Design*. URL: <https://material.io/> (besucht am 14.07.2021).
- [Goob] Google Inc. *States - Material Design*. URL: <https://material.io/design/interaction/states.html#usage> (besucht am 14.07.2021).
- [Gra] GraphQL. *GraphQL | A query language for your API*. URL: <https://graphql.org/> (besucht am 25.07.2021).
- [Hip] Hipp, Wyrick & Company, Inc. *SQLite Home Page*. URL: <https://www.sqlite.org/index.html> (besucht am 28.07.2021).
- [IDE] IDEO. *Frequently Asked Questions | IDEO.org*. IDEO.org. URL: <https://www.ideo.org/perspective/frequently-asked-questions> (besucht am 12.07.2021).
- [Las16] Lasar Liepins. *markdown.de | Markdown Syntax-Dokumentation*. liepins.net. 2013 - 2016. URL: <https://markdown.de/> (besucht am 03.08.2021).
- [Mir] Miro. *Online Whiteboard | Collaborative Free Online Whiteboard | Miro*. URL: <https://miro.com/online-whiteboard/> (besucht am 13.07.2021).
- [Nor13] Don Norman. *The Design of Everyday Things*. Basic Books, 2013. ISBN: 978-3-8006-4809-2.
- [Not] Notion. *Notion – The all-in-one workspace for your notes, tasks, wikis, and databases*. URL: <https://www.notion.so/product> (besucht am 13.07.2021).
- [UXPa] UXPressia. *About UXPressia Customer Experience Mapping Tool*. URL: <https://uxpressia.com/about> (besucht am 13.07.2021).
- [UXPb] UXPressia. *Pricing*. URL: <https://uxpressia.com/pricing> (besucht am 13.07.2021).
- [Weea] Tristan Weevers. *About - UCDtoolbox*. UCDtoolbox. URL: <https://ucdtoolbox.com/about/> (besucht am 12.07.2021).

6 Literatur

- [Weeb] Tristan Weevers. *Creating the UCDtoolbox - UCDtoolbox*. UCDtoolbox. URL: <https://ucdtoolbox.com/about/creating-ucdtoolbox/> (besucht am 12.07.2021).

7 Erstellungserklärung

ERKLÄRUNG ZUR ABSCHLUSSARBEIT

Hiermit versichere ich, die eingereichte Abschlussarbeit selbständig verfasst und keine andere als die von mir angegebenen Quellen und Hilfsmittel benutzt zu haben. Wörtlich oder inhaltlich verwendete Quellen wurden entsprechend den anerkannten Regeln wissenschaftlichen Arbeitens zitiert. Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht anderweitig als Abschlussarbeit eingereicht wurde. Das Merkblatt zum Täuschungsverbot im Prüfungsverfahren der Hochschule Augsburg habe ich gelesen und zur Kenntnis genommen. Ich versichere, dass die von mir abgegebene Arbeit keinerlei Plagiate, Texte oder Bilder umfasst, die durch von mir beauftragte Dritte erstellt wurden.

Augsburg, den 13.09.2021



Simon Hofmeister

A Szenarios

Szenarios

Verknüpfte Personae

			
NAME Jonathan Dampf	NAME Fabian Kissner	NAME Jana Renk	NAME Roland Speer
Demographic Male 50 years Frankfurt Konzepter	Demographic Male 19 years Frankfurt Auszubildender (Design)	Demographic Female 35 years Frankfurt Visual Designer	Demographic Male 45 years Rosenheim IT-Leiter FC Augsburg

Im Grunde wird jeder Anwendungsfall folgendermaßen aufgebaut:

Als **<Person>** möchte ich **<Tätigkeit>**, um **<Grund>**

Anschließend wird dieser Anwendungsfall konkretisiert. Dabei werden beispielsweise Varianten aufgelistet, wie dieses Ziel erreicht werden kann. Zu diesem Zeitpunkt können auch vereinzelte technische Ansätze genannt werden.

Fiktive Grundlage / Vorgeschichte

Der Fußballverein FC Augsburg möchte ein Projekt entwickeln, um seine Mitglieder auf dem Laufenden zu halten, Präsenz zu zeigen und gegen die fallenden Mitgliederzahlen vorzugehen. Der Verein in Person von Roland (IT-Leiter des Vereins) engagiert die Agentur iconstorm (in Person Jonathan, Jana und Fabian), um eine Strategie zu entwickeln, aus der ein Produkt entstehen soll, welches die genannten Ziele erfüllt.

iconstorm möchte einen Workshop abhalten, um sowohl die Interessen der Stakeholder als auch die Zielgruppe besser kennenzulernen. Dafür nutzen Jonathan und Fabian sogenannte Human-Centered Design Methoden, welche bereits über Jahre hinweg optimiert wurden.

Am Workshop nehmen mehrere Personen aus verschiedenen Positionen des Vereins teil. Diese werden künftig in Person von Roland abgebildet, weil sie im Grunde dieselbe Rolle in dem Szenario einnehmen: Workshopteilnehmer. Die Teilnehmer haben zwar einen unterschiedlichen Erfahrungsschatz, was konzeptuelle Vorgänge betrifft, allerdings ist allen das HCD-Prinzip geläufig, da sie i. d. R. in ähnlichen Abteilungen arbeiten

wie Roland. Auch wurde bereits ein Kick-off-Meeting veranstaltet, in welchem geklärt wurde, was die Ziele des Workshops sind.

Der Workshop wird i. d. R. vor Ort abgehalten. Allerdings wird dies in der aktuellen Lage aufgrund der Corona-Pandemie zunehmend erschwert. So ging eine Vision hervor diese Workshops auch remote anbieten zu können, sollte keine andere Möglichkeit bestehen.

Der Ablauf eines Workshops wird definiert vom typischen Vorgehen von Johannes (Konzepter), welcher diese leitet und vorwiegend Daten für Jana sammelt.

Nach Durchführung des Workshops bleibt Roland i. d. R. im engen Kontakt mit Jonathan und Jana, um den Prozess gemeinsam voranzutreiben. Da Roland genug Expertise im Umgang mit Methoden besitzt, kann auch er weiterhin auf der Plattform arbeiten.

Die Erkenntnisse des Workshops bilden keine finalen Ergebnisse ab. Es werden lediglich Informationen gesammelt, welche Johannes und seinem Team dabei helfen, ein detailliertes Bild zu schaffen. (Beispiel: Im Workshop werden sogenannte RAW-Personae erstellt. Für eine detaillierte Ausarbeitung fehlt hierbei die Zeit, da viele Bilder gesucht oder Informationen gesammelt werden müssen. Die Detaillierung findet zu einem späteren Zeitpunkt statt).

Anmerkung: Die folgenden Szenarios fungieren lediglich als Ideensammlung. Im späteren Projekt sollen nicht alle angesprochenen Punkte umgesetzt werden, dienen aber als Stoffsammlung für die Bedürfnisse der einzelnen Stakeholder. Eine Priorisierung findet dabei nicht statt.

Übersicht:

a. Workshop und Zusammenarbeit

1. Vorbereitung
2. Durchführung

b. Weiterverarbeitung

3. Ausarbeitung
4. Designableitung
5. Wiederaufbereitung

c. Recherche

6. Arbeiten vor der Workshopvorbereitung

a. Workshop und Zusammenarbeit

1. Vorbereitung

1.1 Anmelden

Sowohl Jonathan als auch Fabian, Roland und Jana haben einen eigenen Account von der Agentur bekommen. Sie melden sich an, damit das System weiß, um welchen Nutzer es sich handelt.

1.1.1 Eingabe der Daten

- Benutzername / E-Mail
- Kennwort

1.2 Kunden anlegen

Jonathan möchte einen Kunden im System anlegen, damit Roland nur innerhalb dieses Bereiches Berechtigungen erhält. So werden u. a. Betriebsinterna geschützt und es wird eine bessere Übersicht über die Projekte geboten.

1.2.1 Manuelle Eingabe der Daten

- Definition der wichtigsten Informationen
 - Name der Firma
 - Kurzbeschreibung (Eigenschaften der Firma, Beziehungen, Ansprechpartner, Ziele der Zusammenarbeit, ...)
 - Nutzer hinzufügen
 - Kontaktdaten hinzufügen (evtl. direkt über Nutzer verknüpft)

1.2.2 Rollen verteilen

- Nutzer bekommen Rollen zugewiesen, welche unterschiedliche Rechte besitzen (Guest, Reporter, Editor, Developer, Maintainer, ...)

1.2.3 Rollen verwalten

- Definition der Rollen (welche Rechte besitzen diese)
- z. B. Read-Only, Bearbeitung von Methoden innerhalb eines Projektes eines Kunden, Bearbeitung von Methoden in allen Projekten eines Kunden, Bearbeitung von Methoden von allen Kunden, ...

1.3 Projekt anlegen

Jonathan legt im Zuge der Vorbereitung ein Projekt innerhalb des Kunden an, worin die künftig erhobenen Daten gesammelt werden können. Dies verschafft ihm einen Überblick über das weitere Vorgehen.

1.3.1 manuelle Anlage des Projekts

- Definition der wichtigsten Informationen
 - Klar erkennbare Bezeichnung
 - Kurzbeschreibung
 - Weiterführende Links

- Startdatum
- geplanter Produkttyp (falls bekannt)
- Step-by-Step Abfrage

1.4 Workshop anlegen

Jonathan legt einen Workshop an, damit er die Methoden hinzufügen kann und eine Übersicht über diese bekommt.

1.4.1 manuelle Anlage des Workshops

- Definition der Key-Facts des Workshops
 - Welches Ziel soll erreicht werden?
 - Wie viele Teilnehmer (evtl. aus Nutzeranzahl übernehmen)?
 - Art der Teilnehmer (z. B. Vorwissen, ...)?
 - Ort der Ausführung (vor Ort, Remote)?

1.4.2 Workshop definiert sich selbst

- Kein fester Ablauf, da Methoden selbstständig bearbeitet werden können
- Ordnung der Methoden zur Unterstützung

1.5 Methoden hinzufügen

Jonathan sucht nach passenden Methoden für den Workshop, um ideal vorbereitet zu sein. Er hat zwar einen sehr guten Überblick über die Methoden, dennoch übersieht er gerne mal eine, welche ihm in diesem Fall weiter geholfen hätte und bemerkt den "Fehler" erst hinterher.

1.5.1 Empfehlungen der Anwendung

- Empfehlungen auf Grundlage der Projekt- und Workshopinformationen (1.1.1 & 1.4.1)
- Auswahl der für ihn passenden Methoden

1.5.2 Manuell filtern

- Jonathan kann Parameter eingeben, um die passenden Methoden zu finden
- Jonathan kann spezifisch nach einer Methode suchen (Texteingabe)

1.5.3 Automatisches filtern

- Filtermöglichkeit auf Grundlage der Projekt- und Workshopinformationen (1.1.1 & 1.4.1)

1.5.4 Schneller / Übersichtlicher Zugriff

- Die Methoden sind so übersichtlich angeordnet, dass Jonathan sie auch auf Anhieb findet, ohne den umständlichen Filter-Weg zu gehen (z. B. Cards mit passenden Bildern, Aufteilung nach Phasen, ...)
- Häufig gewählte Methoden (z. B. Personae) hervorheben
- Bewertungssystem der Methoden, damit "gute" Methoden hervorgehoben werden können

1.6 Übersicht über ausgewählte Methoden

Jonathan hat nun die passenden Methoden hinzugefügt. Er möchte sich nun einen Überblick über die Methoden verschaffen, um seine Auswahl zu überprüfen.

1.6.1 Übersichtsseite

- Überblick, welche Methoden für den Workshop ausgewählt wurden
- mögliche Probleme bei Workshop kennzeichnen (z. B. Methode für min. 5 Mitglieder sinnvoll, aber der Workshop sieht nur 4 Mitglieder vor)
- Schnell einsehbarer Überblick über Key-Facts (z. B. Dauer der Methoden, Gesamtdauer des Workshops einschätzen)

1.6.2 Ordnung manuell schaffen

- Ablaufreihenfolge der Methoden definieren (z. B. über Drag-and-Drop)
- mögliche Probleme beim Ablauf kennzeichnen (z. B. User Journey vor Persona)
- Schafft dies mögliche Verwirrung bei der Aufbereitung?

1.6.3 Ordnung durch Design definiert

- Grober Ablauf durch Einteilung in Phasen vordefiniert
- Kein detaillierter Ablauf benötigt, da Methoden schnell gefunden werden können (z. B. Cards mit Bildern, Filterfunktionen, Sortierung nach Namen, manuelle Sortierung, ...)
- Möglicher Wechsel der Ansicht (Galerie / Liste)

1.7 Methodendurchführung vorbereiten

Anmerkung: Unterschiedliche Methoden erfordern unterschiedliche Vorgehensweisen. Da im Bachelorprojekt lediglich die Methoden Personae und Szenarios behandelt werden, haben diese im Folgenden einen erhöhten Fokus.

Jonathan möchte für den Workshop die ausgewählten Methoden (Personae und Szenarios) so vorbereiten, dass diese schnellstmöglich präsentiert und ausgeführt werden können. Somit erhält er einen zeitlichen Vorteil gegenüber der manuellen Ausführung auf einem Whiteboard oder Stift und Papier.

1.7.1 Personabearbeitung vorbereiten (Template erstellen)

- Die gängigsten Templates als Auswahl
- Templates anpassbar für das Projekt (nicht immer gleiches Vorgehen)
- Neu erstellte / abgeänderte Templates speicherbar für Zukunft
- Blöcke in Template hinzufügen, ändern und löschen
- Vorauswahl der gängigsten Blöcke
- Neu erstellte / abgeänderte Blöcke speicherbar für Zukunft (innerhalb der Methode)

1.7.2 Methodeninformationen anpassen

Vorgreifend. Siehe 2.2.1 Informationspanel

- Informationen über Methoden variabel gestalten, damit sie für spezifisch auf die Workshopteilnehmer angepasst werden können
-

2. Durchführung

2.1 Workshopziel erklären / definieren

Anmerkung: Wie bereits angemerkt findet die Definition der Wokshopziele normalerweise in einem vorhergehenden Kick-off-Meeting statt, weshalb dieser Punkt in der Umsetzung weniger Beachtung findet. Die gesammelten Ergebnisse bleiben hier dennoch vorhanden, da diese für eine künftige Weiterentwicklung der Anwendung nützlich sein können.

Jonathan und Fabian möchten allen anderen Stakeholdern (in diesem Fall die Teilnehmer des Workshops) die Workshopziele erklären, um auf ein gerichtetes Ergebnis hinzuarbeiten.

2.1.1 Allgemeine Informationen über HCD-Methoden kurz und prägnant zusammengefasst

- Die Informationen, was die HCD-Methoden überhaupt bewirken sollen, zusammengefasst auf einer kurzen Übersichtsseite
- In welcher Phase befindet man sich aktuell
- Erklärungen für Laien, sodass auch Roland die Erklärungen versteht
- Viele Grafiken, um die vokale Erklärung zu unterstützen
- Tipps für Jonathan und Fabian, wie sie die HCD-Prozess für Laien erklären
- Zusammenarbeit mit Laien benötigt (wiederholter Prozess: Erklären - was wurde nicht verstanden - abändern - erneut befragen)
- Roland soll die Informationen schnell auf Abruf parat haben, um noch einmal nachzulesen
- Roland soll die wichtigsten Punkte, die immer beachtet werden müssen, stetig in Erinnerung haben (Tipps / regelmäßige Einblendungen / Separater Bildschirm / ..)
- Weiterführende Links für tiefere Recherche

2.2 Methoden erklären

Jonathan und Fabian erklären Roland den Ablauf der kommenden Methode und die Ergebnisse, die dabei erzielt werden können. So erhält Roland alle Informationen, welche er für die Ausführung der anstehenden Methode benötigt.

2.2.1 Informationspanel

- Jede Methode benötigt ein Informationspanel, welcher während der Bearbeitung schnell aufrufbar ist (z. B. Link zur Seite, Pop-up, Kurzinformation zu Beginn der Bearbeitung, ...)
- Informationen für drei Zielgruppen:
 - Personen mit keinem Vorwissen (z. B. Roland)
 - Personen mit etwas Vorwissen (z. B. Fabian)
 - Personen mit viel Vorwissen (z. B. Jonathan / Jana)
- Informationen müssen auf alle drei Zielgruppen abgestimmt sein:
 - Roland: Kurze prägnante Stichpunkte (z. B. Ziele, Vorgehen, Dauer, ...)
 - Fabian: Weiterführende Informationen für tiefere Erklärung
 - Jonathan/Jana: Fachspezifische Informationen
- Informationen variabel gestalten (abänderbar, siehe 1.7.2)
- Informationen müssen so aufbereitet sein, dass eine Bearbeitung ohne Experten ermöglicht wird

2.2.2 Informationspräsentation

- Einbindung eines Präsentationsmodus speziell für Workshops
- Kurze Präsentation / Slideshow, mit der die wichtigsten Punkte erklärt werden
- Präsentation durchgehend sichtbar (z. B. über Beamer / Second Screen / ...)

2.3 Methoden mit Stakeholdern ausführen (allgemein)

Roland führt eine erklärte Methode aus, um die gewünschten Ergebnisse zu erzielen und der Agentur zu helfen, ein passendes Produkt zu entwickeln.

2.3.1 Bearbeitungspanel für Roland

- Ausfüllen der vordefinierten Vorlagen
- Step-by-Step Guide?
- Tipps einblenden, damit das Ziel nicht verfehlt wird
- schnelle Eingabe (Zeitersparnis zur Bearbeitung mit Stift und Block)

2.3.2 Gemeinsames Ausführungspanel

- Methoden von mehreren Stakeholdern zeitgleich ausführbar
- Jonathan kann alle aktiven Nutzer auf einen Bildschirm holen (vgl. Miro: sammeln der Bildschirme)

- Jonathan kann seinen Bildschirm mit anderen teilen

2.3.3 Notizssammlung

- Notizen innerhalb der Methoden, welche Hinweise für eine spätere erneute Bearbeitung geben können (z. B. in User Journey bemerkt Jonathan, dass die Personae erneut angegangen werden müssen, allerdings will er den Work-Flow nicht unterbrechen)
- Hinweise an andere User (z. B. "@Fabian: überarbeite bitte deine Ergebnisse")
- Erinnerungen für den Nutzer selbst

2.3.4 Kombination der Methoden

- Beim Erstellen eines Szenarios möchte Roland eine Persona verknüpfen
- Andersrum ist dies genauso möglich
- Automatische Anzeige der Verknüpfung in der Persona
- Innerhalb der Szenarios kann Roland schnell eine weitere Persona hinzufügen, welche noch gar nicht existiert.
 - Diese entsteht als Raw-Version mit den wichtigsten Eigenschaften (Name, Alter, ..) und kann später weiter bearbeitet werden
 - Sie entsteht mit den selben Blöcken wie die anderen Personae, allerdings nicht ausgefüllt

2.3.5 Ausführung importieren

- Eine Ausführung aus einem anderen Projekt oder Kunden kann importiert werden

2.4 Methode Szenarios ausführen

Roland möchte die Methode "Szenarios" ausführen, welche von Jonathan vorbereitet wurde. Diese Methode soll dabei helfen mögliche Anwendungsfälle zu bündeln, in denen ein Akteur (z. B. in Form von Personae) mit Hilfe der geplanten Anwendung ein bestimmtes fachliches Ziel erreichen möchte.

2.4.1 Neues Szenario anlegen

- Roland definiert ein Szenario, in der Art "als <Person> möchte ich <Tätigkeit>, um <Grund>" (Genauer Wortlaut nicht erforderlich. Das Vorgehen ist auch Roland bekannt, da ihm Jonathan die Methode bereits erklärt hat)

2.4.2 Szenario gruppieren (Hierarchisches System)

- Sinnvolle Kategorisierung

Erklärung erfolgt über Methabene. Da dies eine Szenariosammlung ist, werden Beispiele aus dem aktuellen Dokument verwendet.

- Mögliche Unterteilung in Kategorien (steigend im Detailgrad):
 - Szenario (hier: "a. Workshop und Zusammenarbeit" in Form einer Ordered List / list-style: lower-latin / Einordnung in Cards/Liste)

- Bezeichnung
- *Überpunkte* (hier: "1. Vorbereitung" in Form einer Ordered List / list-style: decimal / Einordnung in Cards/Liste)
 - Bezeichnung
- *Unterpunkte* (hier: "1.1 Anmelden" in Form einer Ordered List / list-style: decimal / Einordnung in Cards/Liste)
 - Bezeichnung
 - Definition
- *Möglichkeiten* (hier: "1.1.1 Eingabe der Daten" in Form einer Ordered List / list-style: decimal)
 - Bezeichnung
 - *Detailierungen* (hier: "Benutzername/E-Mail" in Form einer Unordered List)
- Sinnvolle Screen- bzw. Paneeleinteilung:
 - Panel 1: Sammlung an Szenarios (Beinhaltet Überpunkte)
 - Panel 2: Sammlung an Überpunkten (evtl. passendere Bezeichnung für Kategorien definieren, Beinhaltet Unterpunkte)
 - Panel 3: Sammlung an Unterpunkten (Beinhaltet Möglichkeiten)
 - Wie viele Unterpanels sind noch benutzerfreundlich?
- Nutzerfreundlich? Viele Sprünge bzw. Seitenwechseln und irreführende Bezeichnungen für Kategorien
- Dem Nutzer Freiheiten gewähren und die Einteilung selbst vornehmen lassen

2.4.3 Szenariogruppen im Methodenpanel

- Szenarios zu Gruppen zusammenfassen
- Gruppe benennen
- (Optionale) Beschreibung zur Gruppe hinzufügen
- Möglichkeit Szenarien schnell in andere Gruppen zu verschieben
- Gruppen zu Übergruppen zuordnen?
 - Nutzerfreundlich?
 - Wirklich nötig?

2.4.4 Szenario bearbeiten

- Einfacher Input (vgl. Notion)
- Automatische Auflistung (vgl. HTML ** bzw. **)
- Evtl. mit simplen Markdown arbeiten
- Möglichkeit bzw. Alternative hinzufügen (hier: "1.1.1 Eingabe der Daten")
- Details zu den Möglichkeiten bzw. Alternativen hinzufügen (hier: Unordered-List)
- Evtl. Galerie (Drag-and-Drop)

2.4.5 Personae verknüpfen

- Vorhandene Personae hinzufügen (vgl. 2.3.4)
- Neue Persona erstellen (nur Key-Facts: z. B. Name, Foto)
- Vorhandene Persona direkt bearbeiten
- Verknüpfte Persona in Szenario erwähnen (Mention-Funktion mit @...)

2.4.6 Status in Sammlung oder deren Unterpunkte definieren

- Siehe 3.1.1
- Kein automatischer Status möglich, da Szenarios keinen Mindestanspruch oder Templates besitzen
- Welche Panels machen Sinn für einen Status?

2.5 Methode Personae ausführen

Roland möchte eine Persona erstellen, um einen Prototypen für eine Gruppe von Nutzern darzustellen.

2.5.1 Persona hinzufügen

- Roland fügt eine neue Ausführung hinzu
- Möglichkeit eine vorhandene Persona als Grundlage zu kopieren

2.5.2 Template ausfüllen

- Roland füllt das Template der Raw-Persona aus
- Falsch ausgefüllte oder leere Felder werden gekennzeichnet
- Mindestanforderungen kennzeichnen (Name, Foto sind immer Pflichtangaben, vgl. Formular. Wichtig auch für Key-Facts)

2.5.3 Automatische WIP-Deklarierung

- Personae mit fehlerhaften Feldern werden gekennzeichnet
- Status wird automatisch geändert in "WIP" oder "RAW"

2.5.4 Szenarios verknüpfen

- Sind Personae bereits in einem Szenario (oder einer Szenariogruppe) verknüpft wird diese Verknüpfung automatisch in den entsprechenden Personae angezeigt
- Da die Reihenfolge der Bearbeitung keine Rolle spielt können auch Szenarios (oder Szenariogruppen) in den Personae verknüpft werden

2.6 Synchrone Arbeiten

Johannes und Fabian können aufgrund der Coronabeschränkungen nicht vor Ort sein und möchten dennoch den Workshop abhalten. Dabei gehen viele physische Eigenschaften verloren.

2.6.1 Kennzeichnung von allen Usern

- Pfeile (mit Namen) mit Live-Position innerhalb der Methode (vgl. Miro)
- Anzeige von Teilnehmern, welche sich im selben Panel befinden (vgl. Miro)
- Kennzeichnung der Felder, welche gerade bearbeitet werden

2.6.2 Kameraübertragung

- Bild-Einblendung von Teilnehmern
- Kooperation bzw. Integrierung von Zoom, Skype, Microsoft Teams, ...
- Anzahl der eingeblendeten Personen begrenzen? Nur sprechender?
- Umgehbar mit zweiten Monitor

2.6.3 Gesten-Integrierung

- Emotion-Reaktion (vgl. Zoom)
- VR-Integrierung?

2.7 Asynchrones Arbeiten

Johannes detailliert eine Persona in der Woche nach dem Workshop. Roland hat in diesem Zeitraum allerdings andere Arbeiten zu erledigen und kann ihn deshalb nicht unterstützen.

2.7.1 Manuelle Notification

- Johannes kann Anmerkungen an Rolands Persona anheften, damit er diese noch einmal überarbeitet
- Notification in Dashboard und an Persona sichtbar
- Nachrichtenbereich bzw. Chatfunktion sinnvoll?

2.8 Methodenergebnisse sammeln

Jonathan und Fabian gehen noch einmal die Ergebnisse des Workshops durch und überprüfen, ob sie noch Änderungen vornehmen oder Methoden erneut angehen müssen.

2.8.1 Übersicht für Jonathan und Fabian

- Ergebnisse der Methoden in einem Schnellüberblick
- Keine störenden Details
- Schneller Zugriff auf Ausführungen
- Übersicht über verknüpfte Ergebnisse
 - Visualisieren (z. B. mit einer Tabelle)
 - Relevanz der Verknüpfung miteinfließen lassen

2.8.2 Übersicht für Roland

- Was wurde im Workshop erreicht
- Das Erfüllen der zuvor definierten Ziele überprüfen

2.8.3 Notizsammlung

- Überblick über gemachte Notizen aus 2.3.4 um sicher zu gehen, dass nichts vergessen wurde

2.9 Physische Arbeiten werden erforderlich

Als Johannes und Fabian den Workshop zusammen mit Roland abhalten, bemerken sie, dass sie eine Methode über Scribbles deutlich einfacher ausführen könnten, da das Zeichnen am Rechner unhandlich ist.

2.9.1 Import von physischen Ergebnissen

- Möglichkeit in Bilder oder Zeichnungen zu integrieren
- Import-Button (lokal vom Anwendungssystem)
- Import via App (Kamera, Google-Bildersuche, Umwandlung von Bildern in Formularfelder)

2.9.2 Implementierung von externen Geräten

- Zeichnungen über Touchpad ermöglichen
- Zeichnungen über Handyapp ermöglichen
- (Handy-)Mikrofone nutzen für Audiokommentare oder Sprachaufnahmen

2.9.3 Import von externen Plattformen

- Externe Plattformen nutzen, um physische Anfertigungen zu umgehen
- Bildimport von externen APIs (Unsplash / Shutterstock / ...)
- Integrierung von Whiteboards (Miro)

2.9.4 Nutzung von Code-Editoren

- Feld mit integriertem Code-Editor verfügbar machen
- Angefertigte Code-Schnipsel als Vorlage speichern

b. Weiterverarbeitung

3. Ausarbeitung

3.1 Detaillierung

Johannes hat einige Raw-Personae vorliegen und möchte diese detaillieren, um tiefere Informationen für die weitere Bearbeitung zu erhalten.

3.1.1 Artefaktprinzip (allgemein)

- Jedes Feld kann bearbeitet werden
- Ältere Version bleibt (als Kopie) vorhanden
- Status hinzufügen möglich (z. B.: RAW - WIP - DETAILED)
- Asynchrones Arbeiten wird dabei ermöglicht (Roland kann seine oder auch andere Personae ebenfalls überarbeiten)

3.1.2 Artefaktprinzip (Persona)

- Felder können zur aktuellen Vorlage hinzugefügt werden (z. B. "Typischer Arbeitstag")
- Die Felder werden bei anderen Personae automatisch hinzugefügt - bleiben aber leer
- Personae mit leeren Feldern werden gekennzeichnet
 - Meldung für zugehörigen Nutzer
 - Notifications benötigt (Zusatz zu 2.7)

3.1.3 Globalisierung

- Import aus anderen Projekten ermöglichen
 - Beim Import müssen die Felder abgeglichen werden
 - Möglichkeit, die Personae in Kundenpanel zu verschieben (Verfügbarkeit für jedes Projekt)
-

4. Designableitung

4.1 Ergebnisübersicht

Jana verschafft sich einen Überblick über die Ergebnisse des Workshops, um die gesammelten Daten in ihrem Designsystem verarbeiten zu können.

4.1.1 Übersicht für Jana

- Übersicht über die wichtigsten Ergebnisse und Verknüpfungen (ähnlich wie für Jonathan und Fabian - siehe 2.8.1)
- Keine speziell aufbereitete Übersicht benötigt, da diese ohnehin gegeben ist
- Schnelle Übersicht wie viele Ausführungen eine Methode besitzt (Relevanz)

4.1.2 Export der Übersicht(en)

- Exportmöglichkeit für einzelne Methoden und Übersicht um Ergebnisse bei der Designentwicklung schneller griffbereit zu haben (z. B. Import in Figma)

- Auswahlmöglichkeit, welche Informationen für Jana im aktuellen Designprozess hilfreich sind

4.1.3 Vergleichsansicht

- Ergebnisse gleicher Methoden können gegenübergestellt werden

4.1.4 Verknüpfungen visualisieren

- Matrixansicht
 - Auswahl zweier Methoden
 - Visualisierung der verknüpften Ergebnisse (und deren Relevanz)
-

5. Wiederaufbereitung

5.1 Erneute Verwendung der Ergebnisse

Zwei Jahre, nachdem das Projekt vollendet wurde, merkt der FC Augsburg, dass das Projekt erfolgreich umgesetzt und gewinnversprechend entwickelt wurde. Sie haben Ideen für ein neues Projekt. Um Zeit zu sparen, möchten sie die alten Ergebnisse wieder neu aufleben lassen. Johannes merkt allerdings, dass sich in der Zeit einiges getan hat, sodass die alten Ergebnisse überarbeitet werden müssen.

5.1.1 Überarbeitung innerhalb des Projektes

- Ergebnisse können jederzeit überarbeitet werden
- Vorlagen erneut aufrufbar machen

5.1.2 Wiederverwendung in neuem Projekt

- Ergebnisse können für den Kunden global definiert werden, sodass diese entweder im neuen Projekte übernommen oder auch überarbeitet werden können
- Duplizierungs- / Importierungsmöglichkeit von Ausführungen

5.1.3 Globale Wiederverwendbarkeit erweitern

Beispiel: FSV Frankfurt startet ein ähnliches Projekt bei iconstorm

- Persona aus den Projekten des FC Augsburg kann als Grundlage dienen und importiert werden.
- Auswahl, ob Ursprungspersona bei Bearbeitung verändert werden soll oder nur innerhalb des Projektes (Globale Personä vs. Duplizierte Persona)
- Globale Persona kann "Kinder" haben
 - z. B. Globale Persona "Timo"
 - Kind 1 (von "Timo"): FC Augsburg
 - Kind 2 (von "Timo"): FSV Frankfurt
- Globale Felder und Felder zur Bearbeitung müssen definiert werden

c. Recherche

6. Arbeiten vor der Workshopvorbereitung

6.1 Ideensammlung

Im Zuge der Recherche ist es möglich, dass Johannes Ideen sammelt, indem er Roland eine Woche lang begleitet. Dabei möchte er seinen Workshop ideal vorbereiten und Ideen für die Anwendung sammeln.

6.1.1 Integrierte Methode

- Eigene Methode "Ideensammlung" (in eigener Phase)

6.1.2 Gesonderter Bereich

- Eigener Bereich in den Projektinformationen, um Informationen zu sammeln
- Ähnlich Notizbereich (vgl. Notion)

6.2 Einstieg in HCD

Fabian informiert sich in der Plattform über das HCD-Prinzip und dessen Methoden, um seine Konzeptarbeiten und grafischen Ausarbeitungen auf ein höheres Level zu heben und sich auf den Workshop vorzubereiten.

6.2.1 Informationen über das HCD-Prinzip und dessen Methoden kurz und prägnant zusammengefasst

- Informationen über HCD-Prinzip (siehe 2.1)

6.2.2 Informationen über HCD-Methoden kurz und prägnant zusammengefasst

- Informationen über HCD-Methoden (siehe 2.2)

6.2.3 Erklärungen in Videoform

- Videos mit Visualisierungen, um komplexere Vorgänge besser zu erklären
- Innerhalb der Agentur durchführbar? / Kooperation mit YouTube-Kanälen? / Zuständige Agenturen beauftragen?
- Animiert? Schauspieler? Sprecher?

6.2.3 Lernbasis für Azubis

- Quiz für angehende Konzepter
- Vorlesungen

B Anforderungsanalyse und MoSCoW-Prioriserung

Anforderungsanalyse & MoSCoW-Priorisierung

Universell

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung BP, bei "Won't have"	≡ Kommentar
<u>UNIVERSELL</u>						Da die universellen Anforderungen stetig parat sein müssen, ist der Header die eleganteste Lösung für die Einbindung
<u>Anmelden</u>	Formular		Must Have	Won't Have	Won't Have	Durch die Anmeldung erkennt das System, welche Rechte der Nutzer besitzt
<u>Hinweise / Notifications einblenden</u>	Dropdown	Header	Should Have	Won't Have	Could Have	z. B. Änderungen in Abwesenheit oder Erwähnungen
<u>Menüführung</u>	Breadcrumbs, Buttons	Header	Must Have	Should Have		
<u>Weitere Nutzer einblenden / Profil aufrufen</u>	Text, Profilbilder	Sidebar, Header	Must Have	Won't Have	Could Have	Ermöglicht paralleles Arbeiten, Unterscheidung zwischen "Im gleichen Panel", "Online", "Offline"
<u>Rollen der Nutzer einblenden</u>	Text unter Profilbild, Rahmen um Profilbild, Tooltip bei Hover auf Profilbild	Sidebar	Should Have	Won't Have	Could Have	
<u>Verbindung mit externen Geräten verwalten</u>	Button, Dropdown, Pop-up	Sidebar	Could Have	Won't Have	Could Have	Erst relevant, wenn mobile Anwendungen entwickelt wurden
<u>Chat</u>	Dropdown	Sidebar, Header	Won't Have	Won't Have	Won't Have	Vgl. Facebook, Kombination mit Kundenpanel "Mitglieder anzeigen". Obsolet, da Kommunikation über Notizen oder externe Anwendungen stattfinden wird.
<u>Paralleles Arbeiten kennzeichnen</u>	Pfeile, Markierung von Feldern		Should Have	Won't Have	Won't Have	vgl. Miro
<u>Nutzer auf eigenen Screen holen</u>	Button	Sidebar, Header	Could Have	Won't Have	Won't Have	vgl. Miro
<u>Nutzer folgen</u>	Button, Rahmen um Screen	Sidebar (Aktive Nutzer), Header	Could Have	Won't Have	Won't Have	vgl. Miro, Screen muss gekennzeichnet werden, z. B. mit Rahmen
<u>Profil- / Kunden- / Nutzer- / Projekteinstellungen ändern</u>	Link zu externer Seite	Sidebar, Header	Must Have	Won't Have	Won't Have	evtl. auslagern, da nur für bestimmte Nutzer relevant (Nutzereinstellungen: z. B. Rechte bearbeiten)
<u>Abschließen</u>	Button	Sidebar, Header	Must Have	Won't Have	Won't Have	Im BP nicht relevant, da kein Nutzersystem integriert wird

Dashboard (Kundenübersicht)

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
----------------	------------------------	------------------------------	------------------------	----------------	--	-------------

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
<u>DASHBOARD</u>						Dashboard wird erst in BP eingebunden, wenn Kundenpanel eingebunden wurde. Umsetzung allerdings sehr nah zum Kundenpanel
<u>Anleitung für Anwendung</u>	Text, Pop-up, Link zu externer Seite		Could Have	Won't Have	Won't Have	
<u>Kunden anlegen</u>	Pop-up (Formular), Link zu externer Seite, Transparente Card		Must Have	Could Have		
<u>Kunden auflisten</u>	Card mit Key-Facts, Liste		Must Have	Could Have		Mögliche Key-Facts: Anzahl der Projekte
<u>Kundeninformationen verwalten</u>	Pop-up, Link zu externer Seite		Must Have	Could Have		
<u>Kunden entfernen</u>	Button	Card / Liste (Kunden auflisten)	Must Have	Could Have		
<u>Ansicht wechseln</u>	Button, Dropdown		Should Have	Won't Have	Could Have	Ansichten: Galerie oder Liste
<u>Kunden sortieren</u>	Dropdown, Drag-and-Drop	Card / Liste (Kunden auflisten)	Could Have	Could Have		
<u>Kunden filtern</u>	Text-Input	Card / Liste (Kunden auflisten)	Could Have	Could Have		
<u>Einstieg in HCD-Prinzip</u>	Text, Pop-up, Link zu externer Seite		Could Have	Won't Have	Won't Have	

Kundenpanel (Projektübersicht)

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
<u>KUNDENPANEL</u>						
<u>Kundeninformationen verwalten</u>	Pop-up, Link zu externer Seite		Must Have	Could Have		inkl. tiefere Informationen über Kunden
<u>Projekt anlegen</u>	Button mit Pop-up (Formular), Link zu externer Seite, Transparente Card		Must Have	Could Have		
<u>Projekte auflisten</u>	Card mit Key-Facts, Liste		Must Have	Could Have		Mögliche Key-Facts: Anzahl der hinzugefügten Methoden
<u>Projektinformationen verwalten</u>	Pop-up, Link zu externer Seite		Must Have	Could Have		
<u>Projekt entfernen</u>	Button	Card / Liste (Projekte auflisten)	Must Have	Could Have		
<u>Ansicht wechseln</u>	Button, Dropdown		Should Have	Won't Have	Could Have	Ansichten: Galerie oder Liste
<u>Projekte sortieren</u>	Dropdown, Drag-and-Drop	Card / Liste (Projekte auflisten)	Could Have	Could Have		
<u>Projekte filtern</u>	Text-Input	Card / Liste (Projekte auflisten)	Could Have	Could Have		

Projektpanel (Methodenübersicht)

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
<u>PROJEKTPANEL</u>						
<u>Projektinformationen verwalten</u>	Pop-up, Link zu externer Seite		Must Have	Could Have		inkl. tiefere Informationen über Kunden
<u>Methode hinzufügen</u>	Button mit Pop-up, Link zu externer Seite, Transparente Card		Must Have	Won't Have	Should Have	Methoden Personae und Szenarios sind standardmäßig im BP hinzugefügt
<u>Methoden filtern</u>	Empfehlung der Anwendung, Filter durch Formular mit Radiobutton	Pop-up, externe Seite (Methode hinzufügen)	Should Have	Won't Have	Won't Have	vgl. UCDtoolbox
<u>Methode auswählen</u>	Übersichtliche Darstellung (Vorsortierung nach Phasen)	Pop-up, externe Seite (Methode hinzufügen)	Must Have	Won't Have	Won't Have	
<u>Hinzugefügte Methoden auflisten</u>	Card mit Key-Facts, Liste		Must Have	Must Have		Mögliche Key-Facts: Phase, Anzahl der Ausführungen
<u>Ansicht wechseln</u>	Dropdown		Should Have	Won't Have	Should Have	Ansichten: Galerie oder Liste
<u>Methoden sortieren</u>	Dropdown, Drag-and-Drop	Card / Liste (Methoden auflisten)	Should Have	Won't Have	Should Have	
<u>Methoden filtern</u>	Text-Input	Card / Liste (Methoden auflisten)	Should Have	Won't Have	Should Have	
<u>Methode entfernen</u>	Button	Card / Liste (Methoden auflisten)	Must Have	Won't Have	Won't Have	
<u>Methodeninformationen (Details) anzeigen</u>	Pop-up	Card / Liste (Methoden auflisten)	Must Have	Could Have		Möglichlicherweise im Methodenpanel sinnvoll
<u>Warnungen über Ausführungen anzeigen</u>	Text, Icon (Notification)	Card / Liste (Methoden auflisten)	Won't Have	Won't Have		Obsolet, da Funktionalität der "fehlerhaften Ausführungen" entfernt wurde
<u>Verknüpfungsmatrix hinzufügen</u>	Button		Must Have	Won't Have	Must Have	Die Matrix soll dabei helfen anzugeben, welche Ausführungen miteinander verknüpft sind. So kann der Nutzer den "Stage of Character" ableiten
<u>Definition der Verknüpfungsmatrizen</u>	Text	Verknüpfungsmatrix	Must Have	Won't Have	Won't Have	Erklärung, welche Vorteile die Matrix bietet
<u>Verknüpfungsmatrix verwalten</u>	Button, Dropdowns	Verknüpfungsmatrix	Must Have	Won't Have	Won't Have	Methoden auswählen, Relevanz einer Verknüpfung bearbeiten, Verknüpfung entfernen
<u>Legende für Verknüpfungsmatrix</u>	Text	Verknüpfungsmatrix	Should Have	Won't Have	Won't Have	Definitionen der Farbeinstellungen

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⌚ MoSCoW-Priorisierung	⌚ Umsetzung BP	⌚ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
<u>Verknüpfungsmatrix / -matrizen exportieren</u>	Button, Dropdown	Verknüpfungsmatrix	Could Have	Won't Have	Won't Have	
<u>Verknüpfungsmatrix entfernen</u>	Button, Dropdown	Verknüpfungsmatrix	Must Have	Won't Have	Won't Have	

Methodenpanel (Ausführungsübersicht)

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⌚ MoSCoW-Priorisierung	⌚ Umsetzung BP	⌚ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
<u>METHODENPANEL ALLGEMEIN</u>						Hinweis: Diese Tabelle wird unterteilt in allgemeine und methodenbezogene Anforderungen
<u>Methodeninformationen anzeigen</u>	Text		Must Have	Must Have		Kurzinformationen über die Methode
<u>Methodeninformationen (Details) anzeigen</u>	Text, Bilder, Pop-up, Link zu externer Seite		Must Have	Must Have		
<u>Gruppe hinzufügen</u>	Pop-up		Must Have	Should Have		
<u>Gruppe verwalten</u>	Button, Klick auf Textfeld		Must Have	Should Have		Gruppenname und -beschreibung
<u>Gruppe duplizieren</u>	Pop-up, Dropdown		Should Have	Won't Have	Could Have	Duplizierungsmöglichkeiten können mithilfe eines Formulars und einer Abfrage nach gewünschtem Ablageort (Kunde / Projekt) und Inhalt (Auswahl der Ausführungen) dupliziert werden
<u>Gruppe entfernen</u>	Button		Must Have	Must Have		
<u>Ausführung hinzufügen</u>	Button, Pop-up, Link zu externer Seite, Transparente Card	Gruppe	Must Have	Must Have		
<u>Ausführungen auflisten</u>	Cards mit Key-Facts, Liste	Gruppe	Must Have	Must Have		Mögliche Key-Facts: Beschreibung bei Szenarios oder Alter / Beruf bei Personae. Immer: Status
<u>Ausführung anderer Gruppe zuordnen</u>	Drag-and-Drop, Dropdown	Gruppe, Card / Liste (Ausführung auflisten)	Should Have	Should Have		
<u>Ausführung duplizieren</u>	Pop-up, Dropdown	Card / Liste (Ausführung auflisten)	Should Have	Won't Have	Could Have	Duplizierungsmöglichkeiten können mithilfe eines Formulars und einer Abfrage nach gewünschtem Ablageort (Kunde / Projekt / Gruppe) dupliziert werden
<u>Ausführung entfernen</u>	Button	Card / Liste (Ausführung auflisten)	Must Have	Must Have		
<u>Ausführungen filtern</u>	Text-Input	Card / Liste (Ausführung auflisten)	Should Have	Could Have		evtl. nur bei Listenansicht
<u>Ausführungen sortieren</u>	Drag-and-Drop, Dropdown	Card / Liste (Ausführung auflisten)	Should Have	Could Have		

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
<u>Ansicht wechseln</u>	Dropdown	Card / Liste (Ausführung auflisten)	Could Have	Won't Have	Could Have	Ansichten: Galerie, Liste
<u>Warnungen / Key-Facts anzeigen</u>	Text, Icon (Notification), Chip	Card / Liste (Ausführung auflisten)	Should Have	Should Have		z. B. Anzahl ungenutzter Verknüpfungen bei Szenarios oder leerer Felder in Personae
<u>Ausführung aus anderem Projekt/ anderen Kunden importieren</u>	Pop-up	Card / Liste (Ausführung auflisten)	Won't Have	Won't Have		Duplizierungsmöglichkeit erfüllt die Anforderung bereits.
METHODENPANEL SZENARIOS						
<u>Verknüpfungen anzeigen</u>	Card, Liste mit Key-Facts	Gruppe	Must Have	Must Have		
<u>Verknüpfungen verwalten</u>	Liste, Checkboxes	Gruppe	Must Have	Must Have		Liste beinhaltet alle Personae (inkl. Name, Gruppe und Status)
METHODENPANEL PERSONAE						
<u>Template wählen/ bearbeiten</u>	Pop-up, Link zu externer Seite	Gruppe	Should Have	Won't Have	Could Have	Hinweis: Im BP ist die Ausführung der Personae nur in Teilen geplant und dient lediglich zur Präsentation der Verknüpfungsmöglichkeiten.
<u>Template auf andere Gruppen anwenden</u>	Dropdown, Pop-up	Gruppe	Could Have	Won't Have	Could Have	

Ausführungspanel

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
AUSFÜHRUNGSPANEL ALLGEMEIN						
<u>Step-by-step Guide zur Ausführung</u>	Text, Präsentation, Pop-up		Won't Have	Won't Have		Bereits im Methodenpanel aufzufinden
<u>Name bearbeiten</u>	Text-Input		Must Have	Must Have		
<u>Titelbild verwalten</u>	Pop-up		Must Have	Must Have		Vorauswahl an Bildern, Möglichkeit zum Bildupload
<u>Ausführung speichern</u>	Button, Pop-up		Must Have	Must Have		Feedback an User ist wichtig
<u>Status wählen</u>	Dropdown		Must Have	Must Have		Raw - WIP - Detailed. Bei Persona evtl. automatisch "WIP" definieren, wenn Felder leer sind.
<u>Ausführung exportieren</u>	Pop-up		Must Have	Won't Have	Could Have	

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
<u>Ausführung taggen</u>	Chips		Could Have	Could Have		z. B. "Medizintechnik", "Dyslexia", usw. Sinnvoll für Wiederverwendbarkeit und zum Filtern der Ausführungen
<u>Notizen hinzufügen</u>	Textfield	Sidebar (1)	Should Have	Won't Have	Could Have	Zwischen Öffentlich / Privat unterscheiden, Markieren von anderen Nutzern ermöglichen
<u>Verknüpfungen anzeigen</u>	Card, Liste mit Key-Facts	Sidebar (2), Methodenausführung	Must Have	Must Have		Key-Facts müssen vorher in der zu verknüpfenden Methode bestimmt werden (z. B. Persona: Name, Bild, ...)
<u>Verknüpfungen verwalten</u>	Liste, Checkboxes	Gruppe, Tabelle	Must Have	Must Have		Sinnvolle Verknüpfungsmöglichkeit zwischen Personae, Szenariogruppen und Szenarios integrieren
<u>Verknüpfungen anzeigen (Detailansicht)</u>	Button, Link zu externer Seite, Pop-up	Sidebar (2), Methodenausführung	Won't Have	Won't Have		Kein Extrapanel. Übergänge sollen derart flüssig verlaufen, dass ein schneller Wechsel mittels Miniaturansicht ausreicht
<u>Verknüpfungsmöglichkeiten filtern</u>	Text-Input	Verknüpfungstabelle	Could Have	Could Have		
<u>Verknüpfungsmöglichkeiten sortieren</u>	Dropdown	Verknüpfungstabelle	Could Have	Could Have		
<u>Ansicht wechseln</u>	Dropdown	Verknüpfungstabelle	Won't Have	Won't Have		Galerieansicht für Verknüpfungen und Listenansicht für Bearbeitung erscheint als Ideallösung
AUSFÜHRUNGSPANEL SZENARIO						
<u>Beschreibung verwalten</u>	Text-Input		Must Have	Must Have		z. B. "als <Person> möchte ich <Tätigkeit>, um <Grund>"
<u>Verknüpfte Ausführung (z. B. Persona) erwähnen</u>	Text-Input, Mention	Detaillierung, Beschreibung, Alternative / Möglichkeit	Could Have	Won't Have	Could Have	"@...", vgl. Notion
<u>Eintrag hinzufügen</u>	Button		Must Have	Must Have		
<u>Eintrag bearbeiten</u>	Klick auf Eintrag, Textinput, Textblock (Markdown)		Must Have	Must Have		Text-Inputs für Name und Beschreibung
<u>Eintrag entfernen</u>	Button		Must Have	Must Have		
<u>Eintrag verschieben</u>	Drag-and-Drop		Must Have	Could Have		vgl. Notion
<u>Bild (als Eintrag) hinzufügen</u>	Pop-up		Should Have	Could Have		Upload und Auswahl: vgl. "Titelbild verwalten"

Aa Anforderung	≡ mögliche Umsetzungen	≡ möglicher Parent Component	⦿ MoSCoW-Priorisierung	⦿ Umsetzung BP	⦿ Andeutung in Anwendung, bei "Won't have"	≡ Kommentar
<u>AUSFÜHRUNGSPANEL PERSONA</u>						Hinweis: Im BP ist die Ausführung der Persona nur in Teilen geplant. Der Fokus bei der Umsetzung liegt auf der Methode Szenarios.
<u>Template ausfüllen</u>			Must Have	Won't Have	Could Have	Art des Inputs in Template definiert
<u>Block hinzufügen</u>	Pop-up		Must Have	Won't Have	Could Have	Blöcke werden bei allen Personae innerhalb der Gruppe automatisch mit gleichem Titel hinzugefügt
<u>Blockart auswählen</u>	Button	Pop-up	Must Have	Won't Have	Could Have	
<u>Blockinhalt bearbeiten</u>	Klick auf Block			Won't Have	Could Have	Art des Inputs variiert von Blockart (Textinput, Textblock, Bild, ...)
<u>Block entfernen</u>	Button		Must Have	Won't Have	Could Have	Blöcke werden bei allen Personae innerhalb der Gruppe automatisch entfernt
<u>Block verschieben</u>	Drag-and-Drop		Must Have	Won't Have	Could Have	Blöcke werden bei allen Personae innerhalb der Gruppe automatisch verschoben
<u>Speicheroptionen vornehmen</u>	Formular	Pop-up (Speichern)	Should Have	Won't Have	Could Have	z. B. Key-Facts definieren, Blockanordnung als Template speichern, Auf welche Ausführungen soll die Blockanordnung übertragen werden (innerhalb der Gruppe / Projekt)