



TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Konzeption und Implementierung einer Android Applikation zur Raumbuchung via Conversational User Interface

Masterarbeit

vorgelegt von: **Michael Wagner**

Fakultät: Elektrotechnik Feinwerktechnik Informationstechnik

Studiengang: Elektronische und Mechatronische Systeme

Vertiefungsrichtung: Informationstechnik

Erstgutachter: Prof. Dr. Oliver Hofmann

Zweitgutachter: Prof. Dr. Hans-Georg Hopf

Betreuer: Dipl.-Inf. Steffen Blümm

WiSe 2018/2019

Schlagworte: Conversational User Interface, Chatbot, Android, Dialogflow, Node.js



Prüfungsrechtliche Erklärung der/des Studierenden

Angaben des bzw. der Studierenden:

Name: Wagner

Vorname: Michael

Matrikel-Nr: 2560795

Fakultät: Elektro-, Feinwerk-, Informationstechnik

Studiengang: Elektronische und Mechatronische Systeme

Semester: Wintersemester 2018/2019

Titel der Abschlussarbeit:

Konzeption und Implementierung einer Android Applikation zur Raumbuchung via Conversational User Interface

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

Ort, Datum, Unterschrift Studierende/Studierender

Erklärung zur Veröffentlichung der vorstehend bezeichneten Abschlussarbeit

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

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

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

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

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

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

Ort, Datum, Unterschrift Studierende/Studierender



Abstract

Bei der Verwendung eines Conversational User Interface (CUI) bedient der Nutzer ein System über geschriebene und gesprochene Sprache. Anders als bei einem Graphical User Interface (GUI) muss die Antwort auf eine Frage nicht mehr gesucht, sondern kann direkt erfragt werden. Nachdem das System auf natürliche Art und Weise mit dem Nutzer interagiert, versteht es idealerweise alle Variationen einer Anfrage. Unabhängig davon, wie diese im Einzelnen formuliert sind.

Im Rahmen dieser Arbeit liegt der Fokus auf der Entwicklung eines Chatbots, also eines textbasierten CUIs. Die Arbeit umfasst die Konzeption und Implementierung eines solchen Systems im Kontext einer Raumbuchung. Dazu werden zunächst die Anforderungen an ein CUI im Allgemeinen und anschließend in Bezug auf ein Raumbuchungssystem betrachtet. In diesem Zusammenhang werden Methoden und Techniken aus dem Bereich Usability Engineering angewendet. Die Idee ist hierbei, multimedial mit dem CUI zu interagieren. So können GUI-Elemente wie beispielsweise Buttons oder Slider in die Konversation mit einfließen. Der Nutzer soll dabei jederzeit die Möglichkeit haben, auch textuelle Nachrichten an den Chatbot zu senden. Diese Ansätze werden mit Hilfe eines prototypischen Systems umgesetzt, überprüft und evaluiert. Durch die Umsetzung einiger der prototypisch erprobten Ansätze ist der Nutzer schließlich in der Lage, mit Hilfe eines Chatbots auf Basis einer Android Applikation nach freien Räumen zu suchen und einen Besprechungsraum zu reservieren.



Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	VI
Tabellenverzeichnis	IX
Quellcodeverzeichnis	X
1 Einleitung	1
1.1 Ausgangssituation	1
1.2 Ziel der Arbeit	2
1.3 Aufbau der Arbeit	2
1.4 Projektträger	2
1.5 Aufbau des Dokuments	4
2 Grundlagen	5
2.1 Conversational User Interface	5
2.1.1 Entwicklung	5
2.1.2 Grundbegriffe	6
2.1.3 Chatbot	8
2.2 Android Applikation	11
2.2.1 Manifest	11
2.2.2 Layout	12
2.2.3 Activity	13
3 Konzeption	16
3.1 Requirements Engineering	19
3.1.1 Fragebogen	19
3.1.2 Interviews	22
3.1.3 Personae	27



Inhaltsverzeichnis

3.2	Prototyping	30
3.2.1	Anforderungen	31
3.2.2	Prototyping Tool	34
3.2.3	Usability Testessen	38
3.2.4	Szenarien	41
3.2.5	Workflow	44
3.2.6	User Tests	48
4	Implementierung	52
4.1	Auswahl der Komponenten	52
4.1.1	Frontend	53
4.1.2	NLP-Plattform	53
4.1.3	Webservice	55
4.2	Kommunikation	56
4.2.1	Authentifizierung und Autorisierung	56
4.2.2	Schnittstellen	59
4.3	Android Applikation	61
4.3.1	Manifest	61
4.3.2	Anmelden	62
4.3.3	Kalenderzugriff	63
4.3.4	Chat Interface	64
4.4	Dialogflow Agent	67
4.4.1	Intents	68
4.4.2	Entitäten	69
4.5	Node.js Webservice	70
5	Schlussbetrachtung	74
5.1	Fazit	74
5.2	Ausblick	76
Literaturverzeichnis		78
A	Anhang	i
A.1	Ergebnisse des Fragebogens	i
A.2	Interviewleitfaden	x
A.3	Einverständniserklärung zum Interview	xiv



Inhaltsverzeichnis

A.4	Ergebnisse der Interviews	xv
A.4.1	Interviewter I1	xv
A.4.2	Interviewter I2	xxii
A.4.3	Interviewter I3	xxix
A.4.4	Interviewter I4	xxxv
A.5	Personae	xliii
A.6	UI-Elemente aus dem Prototyping Tool	lvii
A.7	Einverständniserklärung zum User Test	liii
A.8	Feedback zum Usability Testessen	lv
A.9	Feedback zum User Test	lxvii
A.10	CD Verzeichnisstruktur	lxxix



Abkürzungsverzeichnis

AI Artificial Intelligence

ANN Artificial Neural Network

API Application Programming Interface

ASR Automatic Speech Recognition

CLI Command Line Interface

cpt conversation prototyping tool

CUI Conversational User Interface

DL Deep Learning

DM Dialog Management

dp density-independent pixel

GUI Graphical User Interface

HCI Human Computer Interaction

HR Human Resources

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

ID Identifier

JS JavaScript

JSON JavaScript Object Notation

JVM Java Virtual Machine



Abkürzungsverzeichnis

KI Künstliche Intelligenz

ML Machine Learning

MS-DOS Microsoft Disk Operating System

NLP Natural Language Processing

NLU Natural Language Understanding

NoSQL Not only Structured Query Language

npm Node Package Manager

REST Representational State Transfer

SDK Software Development Kit

TTS Text To Speech

UI User Interface

URL Uniform Resource Locator

UCD User Centered Design

UX User Experience

WOz Wizard of Oz

XML Extensible Markup Language



Abbildungsverzeichnis

1.1	Logo der adorsys GmbH & Co. KG [ado18]	3
2.1	Chronologische Entwicklung von User Interfaces [Tec17]	6
2.2	Aufbau eines Chatbots [Fou17]	10
2.3	Lebenszyklus einer Aktivität [Pro18]	15
3.1	Usability und User Experience	17
3.2	User Centered Design - Menschzentrierter Gestaltungsprozess [Gei10]	18
3.3	Einordnung der Kriterien zu den Personae	28
3.4	Einfaches Prototyping mit der Wizard of Oz Methode	32
3.5	Erweitertes Prototyping mit der Wizard of Oz Methode	33
3.6	Kompletaufbau des Prototypeings mit der Wizard of Oz Methode . . .	34
3.7	Grafische Benutzeroberfläche des Tester-Panels	35
3.8	Grafische Benutzeroberfläche des Simulator-Panels	36
3.9	Struktur eines Szenarios im JSON-Format	37
3.10	Ergebnisse des Usability Testessens	40
3.11	Entscheidungsbaum zum Szenario <i>Vertrieb</i>	44
3.12	Benutzeroberfläche des Operators zum Szenario <i>Vertrieb</i>	46
3.13	Benutzeroberfläche des Probanden zum Szenario <i>Vertrieb</i>	47
3.14	Ergebnisse des User Tests	49
4.1	Übersicht der drei Hauptkomponenten	52
4.2	Übersicht aller Komponenten ohne Kommunikation	56
4.3	Übersicht aller Komponenten mit Kommunikation	59
4.4	Icon der Android Applikation	61
4.5	Benutzeroberfläche zum Anmelden in der Android Applikation . . .	62
4.6	Sequenzdiagramm zur Autorisierung	63
4.7	Benutzeroberfläche zum Autorisieren in der Android Applikation . . .	64
4.8	Chat Interface in der Android Applikation	66
4.9	Prinzipieller Aufbau eines Dialogflow Agents [Goo18a]	67



Abbildungsverzeichnis

4.10 Struktur der Intents im Dialogflow Agent	68
4.11 Sequenzdiagramm zur Interaktion der Systeme	73
5.1 Kosten der Fehlerbehebung in der Softwareentwicklung [Cla16]	75
A.1 Persona Heiko Neumann aus dem Bereich <i>Development</i>	xliii
A.2 Persona Sandra Böhm aus dem Bereich <i>Coaching</i>	xliv
A.3 Persona Michelle Färber aus dem Bereich <i>Human Resources</i>	xlv
A.4 Persona Eric Schweizer aus dem Bereich <i>Vertrieb</i>	xlvi
A.5 UI-Element <i>TextMessage</i> aus dem Prototyping Tool	xlvii
A.6 UI-Element <i>TypeOfMeeting</i> aus dem Prototyping Tool	xlvii
A.7 UI-Element <i>SliderParticipants</i> aus dem Prototyping Tool	xlvii
A.8 UI-Element <i>SliderStartTime</i> aus dem Prototyping Tool	xlviii
A.9 UI-Element <i>SliderTimeBuffer</i> aus dem Prototyping Tool	xlviii
A.10 UI-Element <i>CalendarView</i> aus dem Prototyping Tool	xlix
A.11 UI-Element <i>CollectionViewDates</i> aus dem Prototyping Tool	xlix
A.12 UI-Element <i>CollectionViewRooms</i> aus dem Prototyping Tool	1
A.13 UI-Element <i>DetailView</i> aus dem Prototyping Tool	1
A.14 UI-Element <i>ConfirmBooking</i> aus dem Prototyping Tool	li
A.15 UI-Element <i>Certificate</i> aus dem Prototyping Tool	li
A.16 UI-Element <i>RoomMap</i> aus dem Prototyping Tool	lii
A.17 Ergebnisse zum Fragebogen von user0028	lv
A.18 Notizen zum Fragebogen von user0028	lvii
A.19 Ergebnisse zum Fragebogen von user0082	lvii
A.20 Notizen zum Fragebogen von user0082	lviii
A.21 Ergebnisse zum Fragebogen von user0382	lix
A.22 Notizen zum Fragebogen von user0382	lx
A.23 Ergebnisse zum Fragebogen von user3729	lxii
A.24 Notizen zum Fragebogen von user3729	lxii
A.25 Ergebnisse zum Fragebogen von user5013	lxiii
A.26 Notizen zum Fragebogen von user5013	lxiv
A.27 Ergebnisse zum Fragebogen von user6271	lxv
A.28 Notizen zum Fragebogen von user6271	lxvi
A.29 Ergebnisse zum Fragebogen von user1723	lxvii
A.30 Notizen zum Fragebogen von user1723	lxviii
A.31 Ergebnisse zum Fragebogen von user2626	lxix
A.32 Notizen zum Fragebogen von user2626	lx



Abbildungsverzeichnis

A.33 Ergebnisse zum Fragebogen von user5156	lxxi
A.34 Notizen zum Fragebogen von user5156	lxxii
A.35 Ergebnisse zum Fragebogen von user7027	lxxiii
A.36 Notizen zum Fragebogen von user7027	lxxiv
A.37 Ergebnisse zum Fragebogen von user9337	lxxv
A.38 Notizen zum Fragebogen von user9337	lxxvi
A.39 Ergebnisse zum Fragebogen von user9716	lxxvii
A.40 Notizen zum Fragebogen von user9716	lxxviii



Tabellenverzeichnis

2.1	Attribute von Layout-Elementen	13
3.1	Auswahl einer Anwendung für den Fragebogen	20
3.2	Haupttätigkeitsfelder der Interviewten	23
3.3	Priorisierung der Räume	26
3.4	Randomisierung der Szenarien	48
4.1	Vergleich verschiedener NLP-Plattformen	54
4.2	Arten von Nachrichten im Chat Interface	65
4.3	Übersicht der Entitäten des Dialogflow Agents	69
4.4	Relevante Daten aus Dialogflow JSON-Request	70
4.5	Parameter aus der Datenbank	71



Quellcodeverzeichnis

2.1	Beispielhafte AndroidManifest.xml Datei	11
3.1	Beispielhafte Response im JSON-Format	38
4.1	Darstellung des UI-Elements im Backend im JSON-Format	72



1 Einleitung

Dieses Kapitel spiegelt im Wesentlichen die Inhalte des im Vorfeld der Arbeit entstandenen Exposees wider. Wird bei bestimmten Begriffen, die sich auf Personengruppen beziehen, nur die männliche Form gewählt, so ist dies nicht geschlechtsspezifisch gemeint, sondern geschah ausschließlich aus Gründen der besseren Lesbarkeit.

1.1 Ausgangssituation

Bei der Verwendung eines Conversational User Interfaces (CUIs) bedient der Nutzer ein System durch schriftliche und mündliche Konversation [MCG16, S. 11]. Dabei beschreiben CUIs die Sprachverarbeitung und Kommunikation als Schnittstelle, unabhängig davon in welcher Modalität diese vorliegt. Es kann sich deshalb also sowohl um geschriebene als auch um gesprochene Sprache handeln.

Eine wesentliche Herausforderung dabei ist es, den Nutzungskontext zu erschließen, diesen richtig zu interpretieren und eine passende Antwort zu geben. Die Textversionen der CUIs werden meist als Chatbots bezeichnet [KD18, S. 1]. Diese interaktiven Systeme mit teilweise künstlicher Intelligenz (KI) bieten Funktionen an, welche unseren Alltag erleichtern sollen. Während also über ein Graphical User Interface (GUI) die Antwort auf eine Frage gesucht werden muss, kann diese über ein Conversational User Interface einfach gestellt und direkt erfragt werden.

Mit diesem Hintergrund soll in dieser Masterarbeit die Möglichkeit einer Raumbuchung via Conversational User Interface umgesetzt werden. Heutzutage werden Raumreservierungen in nahezu allen Unternehmen mit Hilfe von Softwaresystemen wie Microsoft Outlook oder IBM Notes getätigkt [Foi17]. Dabei lässt sich über ein GUI zu meist eine Kalenderanwendung öffnen. Über diese kann dann nach verfügbaren Räumen gesucht, Einladungen verschickt oder die freien Zeiten der Teilnehmer geprüft werden.



1 Einleitung

1.2 Ziel der Arbeit

Das Ziel dieser Arbeit besteht darin, eine Android Applikation zur Raumreservierung via Conversational User Interface zu entwerfen. Dem Nutzer soll es möglich sein, einen Besprechungsraum mittels eines Chatbots zu reservieren.

Das CUI wird dabei durch Visualisierungen erweitert. So können GUI-Elemente beispielsweise in Form von Buttons oder Slidern in die Konversation einfließen und diese erweitern. Der Nutzer soll dabei jederzeit die Möglichkeit haben, auch textuelle Nachrichten an den Chatbot zu senden. Im Gegensatz zur Bedienung einer rein grafischen Benutzeroberfläche sparen die Nutzer dabei Zeit [Wed16], da das Suchen nach einem verfügbaren Raum vereinfacht wird. Anhand eines Prototypen kann dabei überprüft werden, wie schnell und effektiv eine Raumbuchung via CUI stattfinden kann.

1.3 Aufbau der Arbeit

Um dieses Ziel zu erreichen, wird die Vorgehensweise anhand der nachfolgenden Schritte genauer erläutert. Die grundsätzliche Anforderung dieser Arbeit ist bereits durch das Ziel beschrieben. Es geht prinzipiell darum ein Konzept zu erarbeiten, die Kommunikation zwischen den Komponenten zu klären und Nutzungskonzepte zu evaluieren.

Durch das konkrete Thema sind bereits viele der Softwarekomponenten klar beschrieben. So wird es sich beispielsweise in jedem Fall um eine Android Applikation handeln. Da die adorsys GmbH & Co. KG den Google-Kalender nutzt, würde sich hier die Verwendung des Chatbot-Builders Dialogflow anbieten. Zudem müssen Überlegungen hinsichtlich des Aufbaus der Konversationsstruktur und der Konversationsstrategie des Chatbots angestellt und evaluiert werden. Sämtliche Ergebnisse werden in dieser Arbeit zusammengefasst und bewertet.

1.4 Projektträger

Diese Masterarbeit wurde in Zusammenarbeit mit dem Unternehmen adorsys GmbH & Co. KG durchgeführt. Um deren fachliche Kompetenzen und die Hintergründe, die



1 Einleitung

zu dieser Arbeit geführt haben, besser einordnen zu können, wird das Unternehmen im Folgenden kurz vorgestellt.

Die adorsys GmbH & Co. KG ist ein im Jahre 2006 in Nürnberg gegründetes, mittelständisches IT-Unternehmen. Es wird aktuell von dem Geschäftsgründer Francis Pouatcha Nouyeuve und Stefan Hamm geleitet und beschäftigt derzeit 112 Mitarbeiter¹ an den Standorten Nürnberg und Frankfurt am Main.

Fachlich hat sich die adorsys GmbH & Co. KG auf individuelle Softwarelösungen für Kunden aus der Banken- und Versicherungsbranche spezialisiert. Mit dem Motto „Wir entwickeln Software für eine digitale Zukunft“ begleiten die eingesetzten Projektteams die Kunden von der ersten Idee bis hin zum fertigen Produkt. Dabei ist es unabhängig davon, ob eine bestehende Software modernisiert werden soll oder die Lösung zu einem neuen Geschäftsmodell fehlt.

Selbst beschreibt sich das Unternehmen wie folgt:

„Die adorsys ist ein seit 2006 bestehendes innovatives IT-Unternehmen für zielgenaue, individuelle und exklusive IT-Lösungen. Wir decken eine Vielzahl fachlicher und technologischer Themen ab und bieten die komplette Projektrealisierung aus einer Hand. Von Projektmanagement, Business-analyse und Anforderungsentwicklung, Softwarearchitektur und -entwicklung über Development Services bis zur Betriebsvorbereitung.“ [ado18]

Das Unternehmen beschäftigt sich seit dem Aufkommen der neuen Generation von Sprachassistenten und Chatbots mit dem Thema Conversational User Interfaces. Im Rahmen von wissenschaftlichen Arbeiten werden derzeit Kompetenzen in diesem Bereich aufgebaut. In Abbildung 1.1 ist das aktuelle Firmenlogo der adorsys GmbH & Co. KG dargestellt.



Abbildung 1.1: Logo der adorsys GmbH & Co. KG [ado18]

¹Stand: 15. November 2018



1 Einleitung

1.5 Aufbau des Dokuments

Zur besseren Übersicht und Orientierung soll hier ein Überblick über die nachfolgenden Kapitel und deren Inhalte gegeben werden.

1. In der Einleitung wird in die Thematik eingeführt sowie auf die Ziele und den Aufbau der Arbeit eingegangen. Des Weiteren wird der Projektträger adorsys GmbH & Co. KG kurz vorgestellt.
2. Im Kapitel Grundlagen werden einige für die Masterarbeit notwendigen Begrifflichkeiten, Technologien und Funktionsweisen beschrieben. Dabei wird verstärkt auf Conversational User Interfaces und Android Applikationen eingegangen.
3. Das Kapitel Konzeption befasst sich mit dem Requirements Engineering und der prototypischen Umsetzung des Systems. Dabei werden Nutzeranforderungen im Kontext des Usability Engineering analysiert und bewertet.
4. Im Kapitel Implementierung werden die für die Umsetzung benötigten Komponenten ausgewählt, deren Kommunikation und Schnittstellen geklärt sowie einige Anmerkungen zur Implementierung gegeben.
5. In der Schlussbetrachtung werden die Ergebnisse der Arbeit in Form eines Fazits und Ausblicks zusammengefasst.
6. Im Anhang sind alle Ausarbeitungen in vollständiger Form dargestellt. Zusätzlich ist die Struktur der beiliegenden CD beschrieben.



2 Grundlagen

In diesem Kapitel werden verschiedene in der Masterarbeit verwendete Begriffe, Technologien und Funktionsweisen näher erläutert. Dabei werden die grundlegenden Inhalte erklärt, welche für das Verständnis der Masterarbeit notwendig sind.

2.1 Conversational User Interface

Conversational User Interfaces erlauben es Menschen, durch natürliche Sprache mit Maschinen zu kommunizieren. Diese Idee stellt einen Paradigmenwechsel gegenüber der heute meist verbreitetsten Mensch-Maschine-Interaktion, englisch auch Human Computer Interaction (HCI), in Form von Graphical User Interfaces dar [Tec17]. Die Kommunikation erfolgt dabei meist verbal oder textuell. Prinzipiell gehören aber auch Steuerungen via Gestik oder Mimik zu den CUIs. Im Rahmen dieser Arbeit liegt der Fokus aufgrund der Entwicklung eines Chatbots allerdings auf textbasierten Conversational User Interfaces.

2.1.1 Entwicklung

Conversational User Interfaces haben eine lange Geschichte hinter sich [MCG16, S. 51]. Bereits in den 1960er Jahren wurden textbasierte Dialogsysteme zum Beantworten von Fragen entwickelt. In den späten 1980er Jahren folgten die ersten sprachbasierten Systeme. In [Hei18, S. 167] wird *Karl Klammer* (oder englisch auch *Clippy*) aus Microsoft Office 1997 sogar als erster Chatbot bezeichnet. Dieser ist mit der Intelligenz von heutige bereits verfügbaren Chatbots natürlich nicht zu vergleichen. Grundsätzlich verfolgt er jedoch den verwandten Ansatz, den Nutzer auf natürliche Art und Weise bei der Bedienung des Programms zu unterstützen.

2 Grundlagen

Aufgrund der in den früheren Jahren unausgereiften Technik im Bereich der CUIs wurde sich zunächst anderen User Interfaces (UIs) bedient. So stand zu Beginn der Computergeschichte ausschließlich eine Tastatur zur Eingabe von Befehlen in Command Line Interfaces (CLIs), wie zum Beispiel MS-DOS, zur Verfügung. Im Laufe der Jahre entwickelten sich daraus die ersten Graphical User Interfaces, welche zusätzlich mit einer Maus bedient werden konnten. Spätestens mit dem Verkaufsstart der ersten Generation des *iPhones* im Juni 2007 [Mol17] hat der revolutionäre Touchscreen einen festen Platz im Portfolio der Graphical User Interfaces eingenommen. In den vergangenen Jahren haben die CUIs derartige Fortschritte gemacht, dass sie sich nahtlos in die Entwicklung der User Interfaces einreihen. In Abbildung 2.1 ist die Entwicklung der User Interfaces nochmals in einem Bild festgehalten. Die Bildüberschrift *The Evolution of UI* bezieht sich in diesem Zusammenhang nicht auf die Verbesserung, sondern lediglich auf die chronologische Entwicklung der User Interfaces. Dabei steht die sprachbasierte Lösung ganz rechts stellvertretend für alle Conversational User Interfaces.

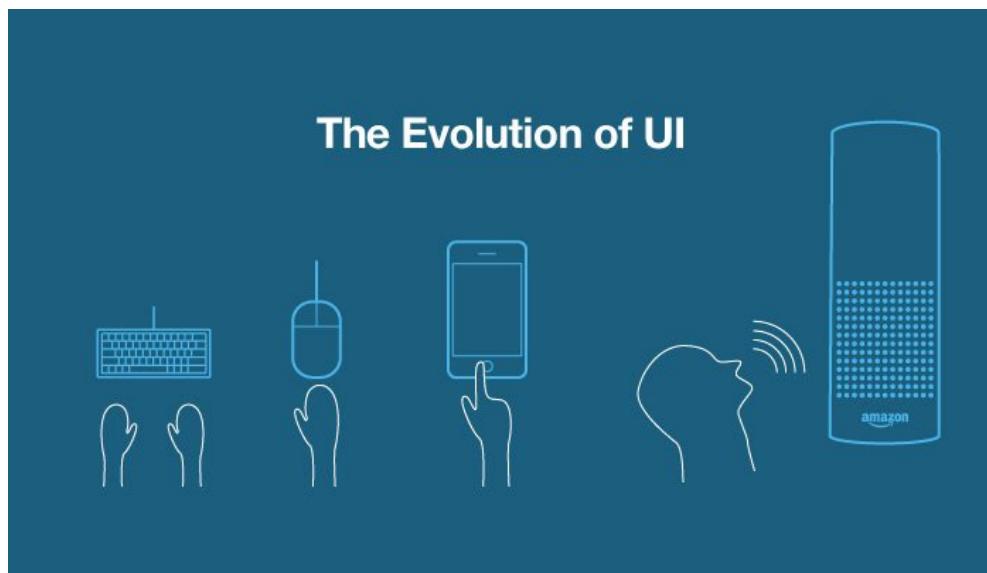


Abbildung 2.1: Chronologische Entwicklung von User Interfaces [Tec17]

2.1.2 Grundbegriffe

Im Folgenden werden einige Grundbegriffe, die immer wieder in Kombination mit CUIs vorkommen, erläutert.



2 Grundlagen

Für den Begriff **Künstliche Intelligenz (KI)** oder **Artificial Intelligence (AI)** existiert eine Vielzahl von Definitionen und Interpretationen. Grundsätzlich zielt er darauf ab, einer Maschine intelligentes Verhalten anzueignen. Nach einer der ersten Definitionen für KI von Alan M. Turing ist ein System dann intelligent, wenn es in seinen Antworten und Reaktionen nicht von einem Menschen zu unterscheiden ist [Nah12, S. 237]. Der Begriff wird im Zusammenhang mit vielen anderen Forschungsbereichen verwendet, wie zum Beispiel intelligentem menschlichen Verhalten oder Entscheidungsfindung. Bezogen auf Conversational User Interfaces bedeutet Künstliche Intelligenz, die Bedeutung eines Satzes herauszufinden und zuzuordnen. Dabei darf es keine Rolle spielen, wie der Nutzer seine Intention ausdrückt. [Nis17, S. 52]

Natural Language Processing (NLP) beschreibt Techniken und Methoden zur maschinellen Verarbeitung natürlicher Sprache. Das Ziel ist es dabei, mit der Entwicklung neuartiger Applikationen die Interaktion zwischen Mensch und Maschine auf Basis von natürlicher Sprache zu verbessern [DL18, S. 1]. Häufig ist in diesem Zusammenhang auch von **Natural Language Understanding (NLU)** die Rede, was einen Teilbereich von NLP darstellt. Bei NLU geht es dabei explizit darum, einen eingegebenen Text zu interpretieren und Informationen oder Intentionen daraus zu gewinnen [Ovc12, S. 15].

Machine Learning (ML) bezeichnet ein fundamentales Konzept zum Realisieren von KI. Dadurch kann ein System auf Basis von Datenbeständen und Algorithmen dazulernen, Vorhersagen treffen, Muster erkennen oder Lösungen entwickeln [Nis17, S. 9]. ML ermöglicht einem System somit Entscheidungen zu treffen, ohne dass diese zuvor explizit programmiert oder vordefiniert werden [Mit18, S. 187].

Deep Learning (DL) ist eine Technik aus dem Bereich Machine Learning. In diesem Bereich sind so genannte Artificial Neural Networks (ANNs), quasi Modelle nach dem Vorbild von biologischen neuronalen Netzen, weit verbreitet. Mit Hilfe von DL und der heute zur Verfügung stehenden Rechenleistung ist es nun möglich, diese sehr rechenintensiven Netzwerke schneller und zuverlässiger abzuarbeiten als herkömmliche Methoden des Machine Learnings. Für ein möglichst intelligent und zuverlässig arbeitendes System sind dazu riesige Datenmengen, in diesem Zusammenhang oft als **Big Data** bezeichnet, nötig. [Nis17, S. 9][VR18, S. 9-10]

Der Begriff **Dialog Management (DM)** bezeichnet das Systemverhalten in Abhängigkeit von Nutzeranfragen. Dabei können neben der Intention des Nutzers auch weitere



2 Grundlagen

Informationsquellen, wie der Gesprächsverlauf oder Ergebnisse aus Datenbankabfragen relevant sein. Die Komplexität des DM kann dabei je nach Umfang und Eigenschaften des Systems variieren. [MCG16, S. 209-210]

Zwei der wichtigsten Begriffe im Bereich CUI sind **Intents** und **Entities**. Der *Intent* gibt an, welches Vorhaben der Nutzer umsetzen will. Dazu wird die textuelle Nutzeranfrage mit Hilfe von NLU verarbeitet und daraus der Intent extrahiert. Ist die Intentiion erkannt, kann eine entsprechende Aktion ausgeführt werden. Die Begriffe Intent, Intention und Vorhaben können im Verlauf dieser Arbeit als gleichwertig betrachtet werden. *Entities* sind wichtige Parameter oder Schlüsselwörter in einer Nutzeranfrage. Entitäten geben wichtige Zusatzinformationen zu einem Intent an und können helfen, diesen überhaupt erst zu identifizieren. Dabei gibt es sowohl Systementitäten (z.B. Wochentag) als auch userspezifische Entitäten (z.B. Flugzeugtyp). [KD18, S. 27, 44-45][Mit18, S. 12-13]

Dies soll anhand eines Beispiels verdeutlicht werden. Die verwendeten Bezeichnungen für Intents, Entitäten und Parameter sind fiktiv und dienen lediglich dem besseren Verständnis.

Nutzer: „Kannst du mir einen Flug von **Nürnberg** nach **Berlin** buchen?“

[Intent erkannt: **BucheFlugIntent**]

Chatbot: „Wann möchtest du fliegen?“

Nutzer: „Am **Freitag**“

Chatbot: „Ich habe folgende Flüge gefunden.“

Aus der ersten Frage des Nutzers kann der Intent **BucheFlugIntent** sowie die beiden Entitäten **Abflughafen** und **Zielflughafen** ermittelt werden. Es fehlt allerdings noch die Auskunft nach dem Tag des Abflugs. Das System arbeitet dabei über eine Rückfrage und erwartet als Antwort einen Wert der Systementität *Wochentag*. Gibt der Nutzer eine entsprechende Antwort (z.B. **Freitag**), hat das System alle notwendigen Daten erhalten und kann über ein entsprechendes Dialog Management eine Antwort generieren.

2.1.3 Chatbot

Der Begriff Chatbot oder auch Chatterbot setzt sich aus den beiden Begriffen *chat* (= Gespräch) und *bot* (= Kurzform für Roboter) zusammen. In der Informatik wird ein



2 Grundlagen

Bot als weitgehend selbstständig laufendes Programm bezeichnet. Ein Chatbot ist somit eine Softwareanwendung zur Realisierung einer Unterhaltung zwischen Mensch und Maschine [SGSB17, S. 232]. Die meisten Chatbots sind derzeit textbasiert, einige bieten allerdings auch die Möglichkeit der sprachlichen Ein- und Ausgabe. Das Ziel ist es dabei, dem Chatbot intelligentes Verhalten anzueignen und ihn dadurch für unterschiedlichste Aufgaben verwenden zu können. Um dies zu ermöglichen, werden die in 2.1.2 beschriebenen Techniken angewendet.

Einer der größten Vorteile von Chatbots im Vergleich zu herkömmlichen mobilen Applikationen ist, dass der Nutzer dem System sein Vorhaben auf natürliche Art und Weise mitteilt. Nach einer Statistik aus [McG18] haben immer noch 70 % der Menschen in reichen Ländern mangelnde oder gar keine Computerkenntnisse. Conversational User Interfaces und damit auch Chatbots umgehen diese Probleme, da alle Menschen den natürlichen Gesprächsdialog kennen und verstehen. Aus Entwicklersicht ist die Verarbeitung von natürlicher Sprache jedoch schwieriger zu handhaben als beispielsweise ein User Interface mit verschiedenen Buttons. Zwar funktioniert das mit modernsten NLP-Plattformen wie *Dialogflow* bereits erstaunlich gut, allerdings gibt es bei komplizierteren oder ungewöhnlicheren Formulierungen noch Probleme. Ein anderer Ansatz ist es dabei, das Conversational User Interface mit Visualisierungen zu erweitern. So kann beispielsweise eine Rückfrage seitens des Chatbots schnell und einfach durch eine Auswahl an Popup-Buttons beantwortet werden. Die Textbox am unteren Ende wird dabei zu jeder Zeit angezeigt und bietet dem Nutzer die Möglichkeit zur textbasierten Kommunikation. Mittels einer *Web View* können dabei ganze HTML-Seiten geladen und angezeigt werden. Verwendet werden sie für Aufgaben, die für ein chatbasiertes User Interface nur schwer oder gar zu kompliziert umzusetzen sind. [KD18, S. 7-10]

Hinsichtlich der Architektur gibt es eine Reihe verschiedener Technologien, Frameworks und Plattformen um einen Chatbot zu realisieren. Im Folgenden soll daher eine Möglichkeit vorgestellt werden, die sich im Rahmen einer Recherche als plausibel und auch für die weitere Arbeit nutzbar herausgestellt hat. Das Schaubild dazu zeigt Abbildung 2.2.

Ein *User* schreibt zu Beginn der Konversation eine Nachricht über eine *Messaging Platform* an das *Natural Language Processing Tool*. Es gibt einige Nachrichtenplattformen, in denen Chatbots entwickelt und integriert werden können. Vorreiter in diesem Gebiet war vor allem der chinesische Messenger *WeChat*, ehe in den letzten Jahren weitere Nachrichtenplattformen wie beispielsweise der *Facebook Messenger* oder

2 Grundlagen

Slack nachzogen [Roc17]. Nach [Riy18] sind im Bereich Natural Language Processing derzeit die Plattformen *Dialogflow*, *Wit.ai*, *Amazon lex*, *Luis* und *IBM Watson* marktführend. Diese Plattformen kümmern sich um das Natural Language Understanding und ordnen den Anfragen die Intents und Entities zu.

Die *Bot-Logic* interpretiert die Nachricht und leitet sie an das Backend, in der Abbildung 2.2 als *Information Sources* dargestellt, weiter. Durch immer mehr Nutzeranfragen wird der Chatbot trainiert und kann sich dadurch stetig verbessern. Dahinter steckt das in 2.1.2 erklärte Prinzip des *Machine Learning*. Im Backend findet dann üblicherweise das Dialog Management statt. Hier werden die Informationen verarbeitet und Entscheidungen in Form von *Actions* getroffen.

Zudem können gegebenenfalls weitere Informationsquellen, beispielsweise aus *Datenbanken*, angebunden werden. Außerdem ist es möglich *Application Programming Interfaces (APIs)* wie zum Beispiel die *Google Calendar API* zum Verwalten von Terminen in das Backend einzubinden. Gibt es Probleme mit der Dialogführung oder kommt der Chatbot an seine Grenzen, kann die Nachricht an einen echten Menschen, in der Darstellung als *Human Intervention* bezeichnet, weitergeleitet werden. [Fou17]

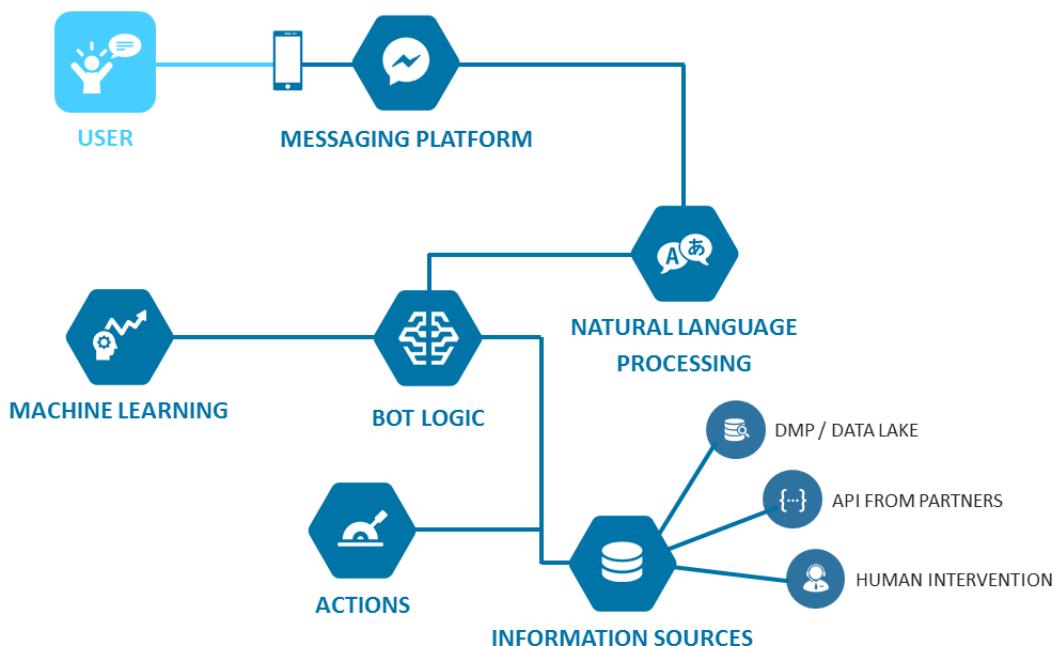


Abbildung 2.2: Aufbau eines Chatbots [Fou17]



2 Grundlagen

2.2 Android Applikation

Das Smartphone-Betriebssystem *Android* hat derzeit in Deutschland einen Marktanteil von etwa 83 %¹ [sta18]. Der Standard für die Entwicklung von Android Applikationen ist *Java*, allerdings wird seit einiger Zeit auch *Kotlin* vollständig unterstützt. *Kotlin* ist eine statisch typisierte Programmiersprache, die auf Java Virtual Machine (JVM) basiert. Einer der größten Vorteile ist dabei, dass *Java* und *Kotlin* gemeinsam innerhalb eines Projekts genutzt und sogar automatisch in die jeweils andere Sprache übersetzt werden können. Entwickelt wurde *Kotlin* von der Firma *JetBrains*, die auch die Entwicklungsumgebung *Android Studio* zur Verfügung stellt. [PR17, S. 272]

2.2.1 Manifest

Zu Beginn jeder Android Applikation steht das *Manifest*, repräsentiert durch die Datei *AndroidManifest.xml*. Es nutzt Extensible Markup Language (XML) und dient als zentraler Punkt zur Deklaration von wichtigen Informationen über die App und ihre Komponenten. Hier werden unter anderem Aktivitäten, Services und Berechtigungen eingetragen [All15, S. 41]. Im Folgenden ist eine beispielhafte Manifest Datei dargestellt, welche einen grundlegenden Eindruck vom Aufbau vermitteln soll.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   package="com.example.michaelwagner.beispiel">
4
5   <uses-permission android:name="android.permission.INTERNET"/>
6
7   <application
8     android:allowBackup="true"
9     android:icon="@mipmap/ic_launcher"
10    android:label="@string/app_name"
11    android:roundIcon="@mipmap/ic_launcher_round"
12    android:supportsRtl="true"
13    android:theme="@style/AppTheme">
14      <activity android:name=".Activities.MainActivity">
15        <intent-filter>
16          <action android:name="android.intent.action.MAIN" />
17          <category android:name="android.intent.category.LAUNCHER" />
18        </intent-filter>
```

¹Stand: September 2018



2 Grundlagen

```
19    </activity>
20    <activity android:name=".Activities.SecondActivity"></activity>
21  </application>
22</manifest>
```

Listing 2.1: Beispielhafte AndroidManifest.xml Datei

Der Name hinter dem Tag *package* dient der Identifikation der App auf dem Smartphone oder auch im *Google Play Store*. Durch die Angabe von *uses-permission* können diverse Berechtigungen, wie hier etwa *INTERNET* zur Herstellung von Internetverbindungen, erteilt werden. Einige häufig verwendete Permissions sind *CAMERA* für Kamerazugriff, *ACCESS_FINE_LOCATION* für den genauen Standort und *RECORD_AUDIO* zur Sprachaufnahme via Mikrofon. Das Element *application* definiert das Anfangsverhalten der App. Die Attribute für *icon*, *label* oder *theme* definieren Basisinformationen wie das Icon oder den Namen für die App. Außerdem muss jede *activity* dort eingetragen sein. Dabei muss auch eine *Launcher Activity* deklariert werden, die der Nutzer als Erstes beim Start der App sieht.. [All15, S. 42-43][And18a]

2.2.2 Layout

Wie schon beim Manifest wird auch für das Layout einer Android App die erweiterbare Auszeichnungssprache XML genutzt. Das Layout definiert den Aufbau des User Interfaces. Unterschieden werden dabei *View*- und *ViewGroup*-Objekte. Eine *ViewGroup* ist eine Art unsichtbarer Container, der die Struktur für weitere Elemente vorgibt. Jede Layout-Datei besitzt dabei genau eine übergeordnete *ViewGroup*, die für das gesamte UI gilt. Nach [All15, S. 79-80] sind die am häufigsten verwendeten Layout-Container:

- **RelativeLayout** - Verschiedene Regeln zur relativen Anordnung der Elemente unter-, über- oder nebeneinander
- **LinearLayout** - Alle Elemente werden in einer Richtung, horizontal oder vertikal, angeordnet
- **ConstraintLayout** - Keine verschachtelten *ViewGroups*, alle Elemente werden relativ angeordnet, über so genannte Constraints (Regeln)



2 Grundlagen

Eine View ist die Basisklasse eines *Widgets*, wie *Button* oder *TextView*. Widgets werden genutzt, um ein interaktives User Interface zu erstellen [Jac13, S. 137]. Die Views und ViewGroups unterstützen verschiedene Attribute. Dabei gibt es sowohl Attribute für alle Elemente als auch spezifische Attribute, die nur bestimmte Elemente implementieren können. Generell sollten bei der Angabe von Höhe, Breite oder Schriftgröße konstante Werte vermieden werden. Besser ist die Angabe von relativen Maßen wie *density-independent pixel (dp)*, *wrap_content* oder *match_parent*, weil dadurch sichergestellt wird, dass die App auch auf unterschiedlichen Bildschirmgrößen korrekt dargestellt wird. Tabelle 2.1 zeigt eine kleine Auswahl an Attributen inklusive Beispielwert und Bedeutung. [And18b]

Attribut	Beispielwert	Bedeutung
android:id	@+id/my_button	Eindeutiger Identifier des Elements
android:layout_width	wrap_content	Breite des Elements an Inhalt anpassen
android:layout_height	match_parent	Höhe des Elements wie übergeordnetes Element
android:text	Das ist ein Text	Textinhalt des Elements
android:textSize	12dp	Schriftgröße des Elements
android:background	#f9aa20	Hintergrundfarbe des Elements

Tabelle 2.1: Attribute von Layout-Elementen

2.2.3 Activity

Eine Aktivität (engl. *Activity*) repräsentiert einen Bildschirm, mit dem der Nutzer interagieren kann. Die Aktivität selbst stellt dabei in der Regel keine Benutzeroberfläche bereit, sondern ist mit einem entsprechen XML-Layout verknüpft. Ein äußerst wichtiges Prinzip im Zusammenhang mit Aktivitäten ist der Lebenszyklus. Dabei werden Aktivitäten via *Callback*-Methoden über Zustandsänderungen informiert. Jede dieser Methoden kann überschrieben werden. [GC15, S. 187-188][Pro18]



2 Grundlagen

Die verschiedenen Zustände und dazugehörigen Methoden werden im Folgenden anhand von Abbildung 2.3 ausführlich erläutert.

- **onCreate()**: Wird beim erstmaligen Erzeugen der Aktivität aufgerufen. Hier sollten Initialisierungen vorgenommen, Views erzeugt oder Daten geladen werden. Anschließend wird *onStart()* aufgerufen.
- **onRestart()**: Die Aktivität muss zuvor gestoppt worden sein. Der Aufruf erfolgt, wenn User zu der Activity zurück navigiert. Der Methode folgt immer *onStart()*.
- **onStart()**: Wird aufgerufen, kurz bevor die Aktivität auf dem Bildschirm sichtbar wird. Anschließend wird *onResume()* aufgerufen.
- **onResume()**: Diese Methode wird ausgeführt, wenn die Aktivität bereit für Benutzereingaben ist. Nach Beendigung dieser Methode wird die Aktivität ausgeführt.
- **onPause()**: Wird kurz vor dem Start einer anderen Aktivität aufgerufen. Geänderte Daten oder Zustände sollten an dieser Stelle gespeichert werden. Anschließend wird *onResume()* aufgerufen, wenn die Activity wieder in den Vordergrund rückt oder *onStop()*, wenn sie verdeckt wird.
- **onStop()**: Wird aufgerufen, wenn die Aktivität nicht mehr sichtbar ist. Nach dieser Methode kann das Android-System den App-Prozess aufgrund von Speicherplatzmangel direkt beenden. Anschließend folgt die *onRestart()* Methode, wenn die Activity wieder fortgesetzt wird oder die *onDestroy()* Methode, wenn die Activity endgültig zerstört wird.
- **onDestroy()**: Aufruf kurz vor dem Zerstören der Aktivität. Erfolgen kann dies planmäßig innerhalb der Applikation oder durch das Android-System. Nachdem dies der letzte Aufruf ist, folgt keine weitere Methode.

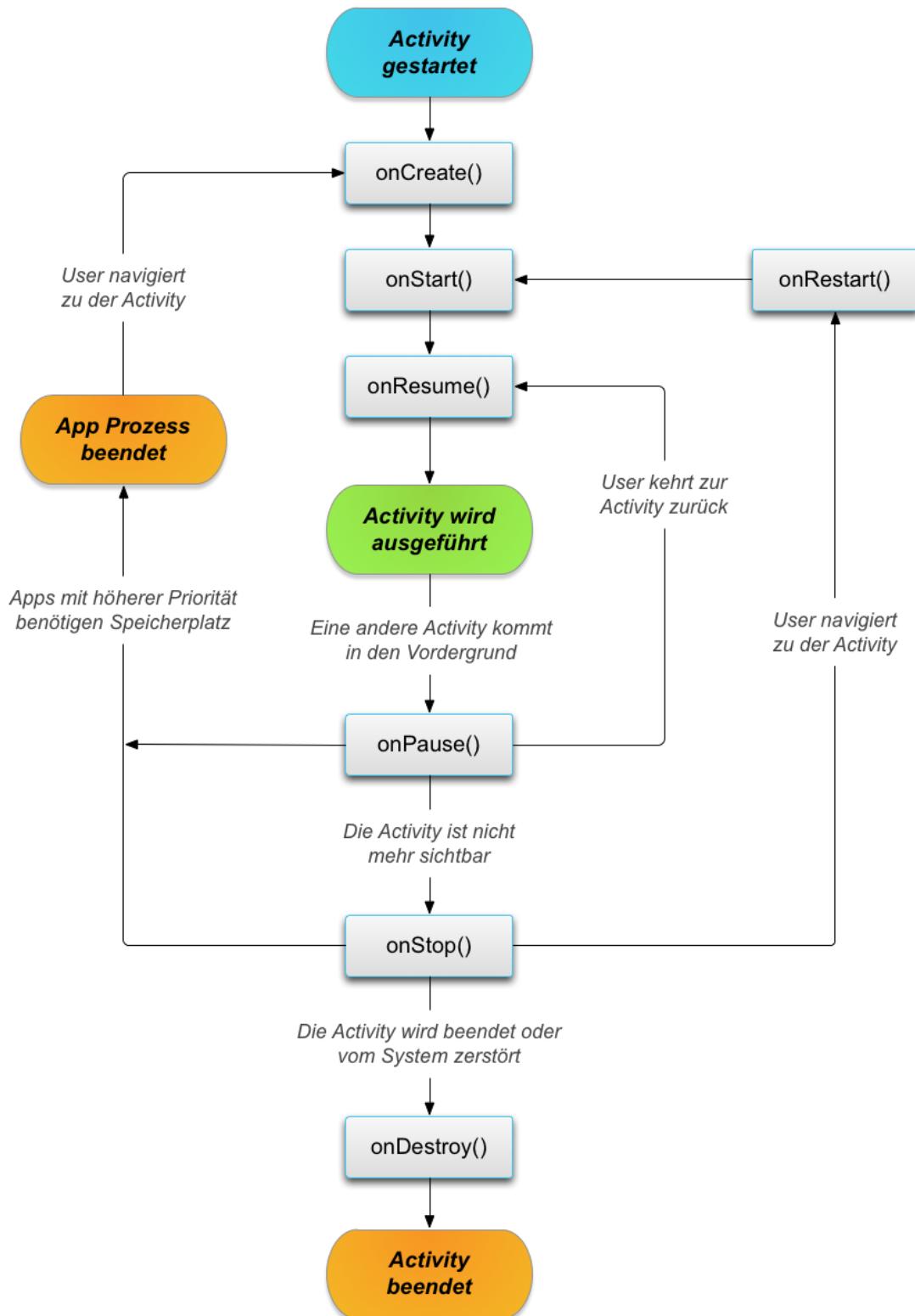


Abbildung 2.3: Lebenszyklus einer Aktivität [Pro18]



3 Konzeption

In diesem Kapitel werden die Anforderungen an den Chatbot und ein Konzept für die Implementierung erarbeitet. Dafür werden Ansichten und Methoden aus dem Bereich *Usability Engineering* verwendet. Usability wird auch als Gebrauchstauglichkeit bezeichnet und definiert sich wie folgt:

Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext dazu genutzt werden kann, bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen [RF16, S. 11].

Der *Nutzungskontext* umfasst dabei die Benutzer, deren Aufgaben, soziale und physische Umgebung sowie die aufgewendeten Ressourcen. Die *Effektivität* misst dabei die Vollständigkeit und Genauigkeit der Zielerreichung. Das Kriterium *Effizienz* bezeichnet den Aufwand und die Ressourcen zur Zielerreichung. Die *Zufriedenstellung* gibt Auskunft über die Freiheit von Beeinträchtigungen und positive Einstellung des Benutzers gegenüber dem System.

Die Definition von Usability impliziert dabei jedoch immer eine sehr funktionsbezogene Betrachtungsweise. Um auch Aspekte wie Emotionen, Ästhetik und Wahrnehmungen zu betrachten, kann das Konzept der *User Experience (UX)* angewendet werden. Die User Experience erweitert die Usability und definiert sich wie folgt:

Wahrnehmungen und Reaktionen einer Person, die aus der tatsächlichen und/oder der erwarteten Benutzung eines Produktes, eines Systems oder einer Dienstleistung resultieren [Gas18, S. 14].

UX umfasst demnach auch Effekte auf den Nutzer, die ein System bereits vor oder auch nach seiner Verwendung hat. Diese Empfindungen haben wiederum Einfluss auf den nachfolgenden Aspekt („*Nach der Nutzung ist vor der Nutzung*“). Die Usability bleibt dennoch ein wichtiger Faktor und Teil der User Experience. Aufgrund der umfassenderen Betrachtungsweise wird der Begriff UX immer öfter auch anstelle der

3 Konzeption

Bezeichnung *Usability* verwendet. Abbildung 3.1 erläutert die beiden Begriffe anhand einer grafischen Darstellung.

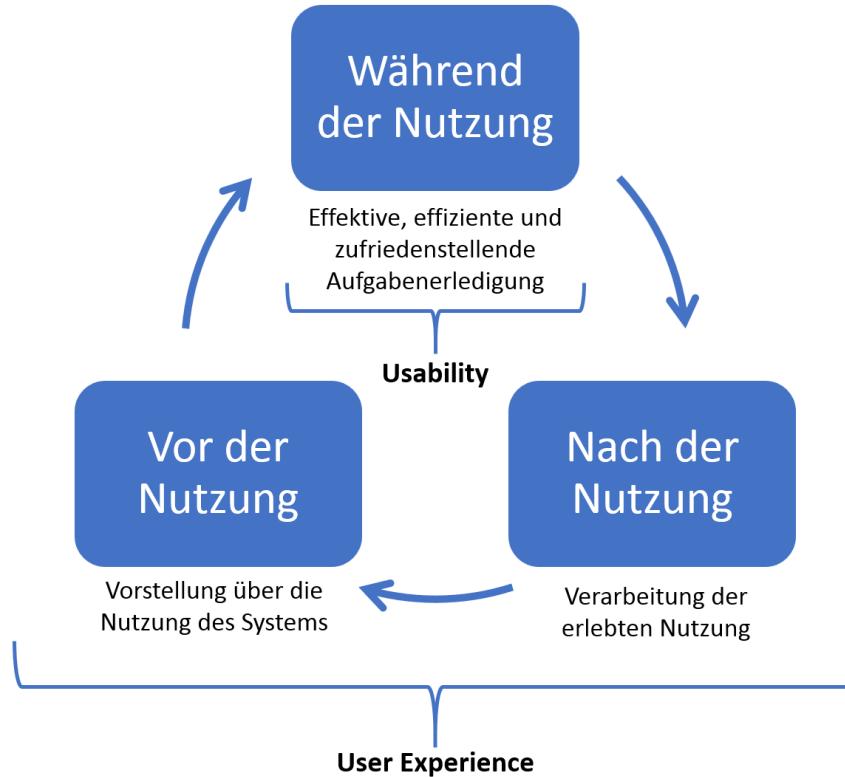


Abbildung 3.1: Usability und User Experience

Sowohl Usability als auch User Experience stellen einen wichtigen Faktor in der Softwareentwicklung dar. Gerade im Bereich der Conversational User Interfaces ist die Anwendung dieser Konzepte und Methodiken äußerst wichtig. Nachdem der Benutzer beim CUI auf natürliche Art und Weise mit dem System interagiert, müssen Formulierungen und Ausdrucksweisen fehlerfrei erkannt und interpretiert werden. Zudem müssen auch eventuelle Rückfragen, Statusmeldungen oder Antworten korrekt und ohne Spielraum für Missinterpretationen formuliert werden.

Ein weiterer wichtiger Aspekt in diesem Zusammenhang ist das *User Centered Design (UCD)*, im Deutschen oft als *menschzentrierter Gestaltungsprozess* übersetzt. Diese nutzerzentrierte Gestaltung zielt darauf ab, interaktive Produkte so zu gestalten, dass sie über eine hohe Gebrauchstauglichkeit verfügen. Der Begriff beschreibt Gestaltungsprozesse, die den zukünftigen Benutzer mit seinen Aufgaben, Zielen und Eigenschaften ins Zentrum des Entwicklungsprozesses stellen. Dabei spielt die frü-

3 Konzeption

he Fokussierung auf den Nutzer und dessen Anforderungen eine entscheidende Rolle. Der Prozess sieht außerdem Iterationen vor, sobald Evaluierungsergebnisse Bedarf dafür aufzeigen. [Try16, S. 67-68]

Diese Vorgehensweise hat sich in den letzten Jahren bei der Entwicklung von Mobile- und Webapplikationen bewährt. Die grundlegenden Nutzeranforderungen ändern sich nicht durch einen Wechsel des Interfaces von GUI zu CUI. Es muss jedoch verstanden werden, welche Methoden und Techniken in diesem Zusammenhang sinnvoll sind und verwendet werden sollten. Werden diese Aspekte bei der Entwicklung eines Chatbots angewendet, so können auch hier wertvolle Daten gesammelt werden, woraus dann hochwertige User Interfaces entstehen. [Cha18]

Die Inhalte des *User Centered Design* sind in der ISO 9241-210 festgehalten und können anhand von Abbildung 3.2 grafisch dargestellt werden.

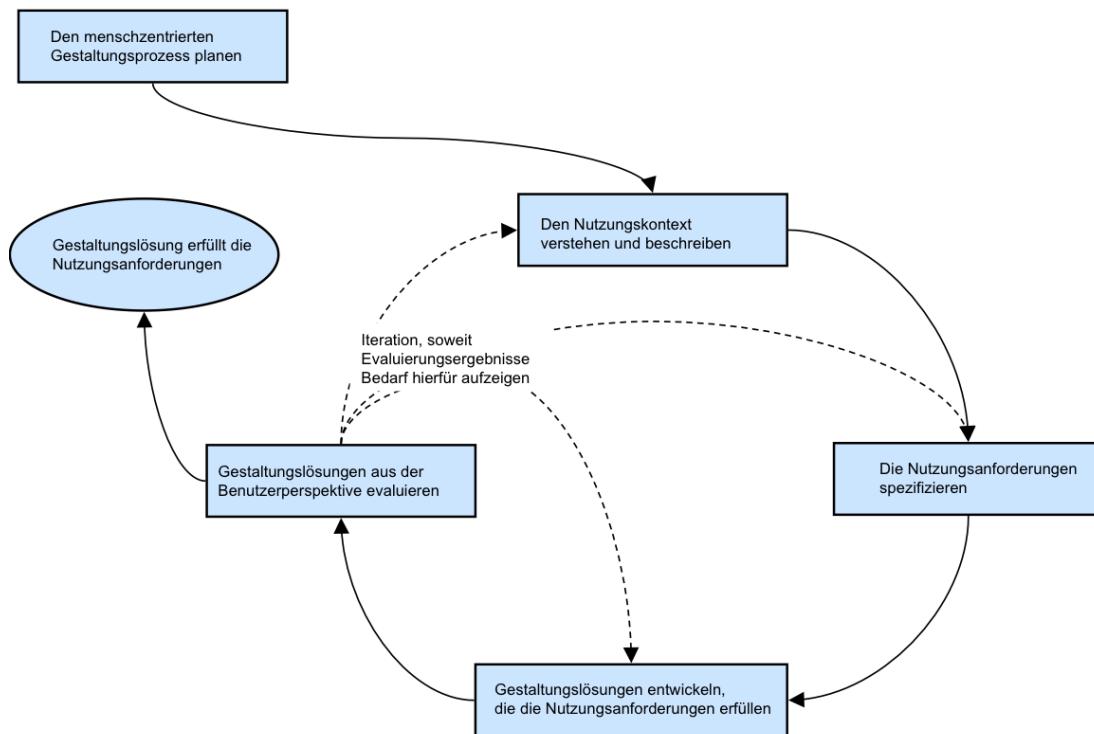


Abbildung 3.2: User Centered Design - Menschzentrierter Gestaltungsprozess [Gei10]

Um all diese Ziele zu erreichen, werden zunächst Überlegungen hinsichtlich des Requirements Engineering angestellt. Dazu können verschiedene Techniken aus dem Bereich des Usability- und Requirements Engineering eingesetzt werden. Mit Hilfe einer



3 Konzeption

prototypischen Umsetzung können diese Überlegungen anschließend überprüft und evaluiert werden.

3.1 Requirements Engineering

Requirements Engineering befasst sich damit, die Bedürfnisse der Benutzer, Auftraggeber und weiterer Interessengruppen, im Allgemeinen auch als Stakeholder bezeichnet, hinreichend aufzuarbeiten [RF16, S. 23]. Darauf basierend können dann Rückschlüsse auf die Rahmenbedingungen, Qualitätsanforderungen sowie vorgesehene Funktionen und Systemverhalten geschlossen werden.

3.1.1 Fragebogen

Um einen groben Überblick über die Thematik zu erhalten, sollen zunächst das Verhalten und die Präferenzen bei der Raumbuchung im Office der adorsys GmbH & Co. KG erfasst werden. Um eine mögliche Voreingenommenheit gegenüber Chatbots zu vermeiden, wird der Hintergrund und das explizite Thema der Masterarbeit in diesem Zusammenhang nicht erwähnt. Die Zielgruppe sind dabei alle Mitarbeiter der adorsys GmbH & Co. KG unabhängig davon, wie oft diese in der Regel einen Raum buchen.

Um möglichst alle relevanten Informationen einzufangen, soll dazu ein Online-Fragebogen erstellt werden. Mit dessen Hilfe lassen sich quantitative Daten sammeln und direkt elektronisch auswerten. Zur Erstellung und Durchführung des Fragebogens werden im Folgenden die Anwendungen *Survey Monkey*, *Typeform* und *Google Forms* genauer betrachtet. Dazu wird, wenn es unterschiedliche Varianten gibt, jeweils die kostenlose Basisvariante der Anwendungen untersucht. Die kostenlosen Varianten der Anwendungen werden verwendet, um möglichst keine Mehrkosten zu generieren sowie den Aufwand und die damit verbundenen Verzögerungen minimal zu halten. Die Anforderungen resultieren aus den Rahmenbedingungen des zuvor erstellten Fragebogens. So besteht dieser aus 18 Fragen, bei denen es sich hauptsächlich um Basis-Fragetypen wie Multiple-Choice, Textfelder oder linearen Skalen handelt. Relevant ist dabei auch die Anonymisierung der Daten, um möglichst ehrliche und ungefilterte Antworten zu erhalten. Das Unterbinden einer Mehrfachteilnahme ist ebenso wünschenswert.



3 Konzeption

Der vollständige Fragebogen ist aus Gründen der besseren Lesbarkeit im Anhang A.1 zu finden. Tabelle 3.1 zeigt eine Übersicht der Anforderungen und betrachteten Anwendungen. Erfüllte Anforderungen sind dabei mit einem *grünen Haken*, nicht erfüllte mit einem *roten Kreuz* versehen. Ein *gelbes Warndreieck* sagt aus, dass die Anforderung nur unter einer bestimmten Bedingung erfüllt werden kann. Die Inhalte der Tabelle stützen sich auf die Informationen aus [Sur18], [Typ18] und [Goo18f].

Anforderungen	Survey Monkey	Typeform	Google Forms
18 Fragen	✗	✗	✓
Basis-Fragetypen	✓	✓	✓
Anonymisierung	✓	✓	✓
Mehrfachteilnahme unterbinden	✓	⚠	✓

Tabelle 3.1: Auswahl einer Anwendung für den Fragebogen

Die beiden Anwendungen Survey Monkey und Typeform ermöglichen in der Basisvariante nur maximal zehn Fragen. Bei Google Forms gibt es eine solche Beschränkung dagegen nicht. Die für den Fragebogen relevanten Basis-Fragetypen wie Multiple-Choice, Textfelder oder lineare Skalen unterstützen dagegen alle Anwendungen. Gleiches gilt für die Möglichkeit, die erfassten Daten zu anonymisieren. Die Unterbindung einer Mehrfachteilnahme ist mit Typeform nur über Umwege möglich, wird von Survey Monkey und Google Forms dagegen standardmäßig unterstützt. Aufgrund der erläuterten Kriterien fällt die Wahl der Anwendung auf Google Forms. Die Anwendung unterstützt alle Anforderungen und kann ohne zusätzliche Kosten genutzt werden.

Im Folgenden werden einige markante Ergebnisse aus dem Fragebogen herausgegriffen und genauer betrachtet. Die Fragen sind dabei oft nach Schlagwörtern wie *gewöhnlich* oder *in der Regel* gestellt und müssen daher mit Vorsicht betrachtet werden, um keine falschen Rückschlüsse auf die Ergebnisse zu ziehen.

Knapp die Hälfte der Befragten buchen täglich oder alle zwei bis drei Tage einen Raum. Dabei ist erwähnenswert, dass sich der Fragebogen auf die Raumbuchung im Office der adorsys GmbH & Co. KG bezieht und einige der Befragten durch Ihre Tätigkeit bei Kunden und in Projekten oft außerhalb des Büros arbeiten. Daraus lässt sich schließen, dass das Buchen von Räumen bei vielen Mitarbeitern ein fester Bestandteil des Arbeitsalltags ist.



3 Konzeption

Über 60 % der Buchungen werden dabei mindestens einen Tag im Voraus getätigt, mit knapp 2 % gibt es vergleichsweise wenige Buchungen über Wochen oder Monate im Voraus. Erklären lässt sich dies dadurch, dass weit im Voraus geplante Termine oft einen Workshop- oder Schulungs-Charakter haben, über mehrere Stunden gehen und eine größere Anzahl an Teilnehmern besitzen. Diese müssen entsprechend frühzeitig geplant werden, kommen aber vergleichsweise selten vor.

Erwähnenswert ist außerdem das Ergebnis auf die Frage nach der gewöhnlichen Besprechungsdauer. Über 90 % der Teilnehmer geben an, dass ihre Besprechungen in der Regel weniger als zwei Stunden dauern. Dies ist möglicherweise auf die Vielzahl an relativ häufigen und gleichermaßen kürzeren Termine wie beispielsweise Code-Reviews, Team-Meetings aber auch Vorstellungs- oder Mitarbeitergesprächen zurückzuführen.

Einige Fragen beziehen sich auf das *E-Paper Display* am Eingang der Besprechungsräume, welches den aktuellen Status des Raumes sowie das nächste anstehende Meeting anzeigt. Außerdem bietet es durch Bedienung die Möglichkeit, den Belegungsplan des Raumes einzusehen und den Raum direkt oder zu einem späteren Zeitpunkt zu reservieren. Nach den Ergebnissen aus dem Fragebogen nutzen etwa zwei Drittel das E-Paper Display, wenn sie sofort einen Raum benötigen. Um einen Raum zu einem späteren Zeitpunkt zu verwenden, nutzen es dagegen nur knapp 8 %. Als Grund hierfür könnte die Bedienung eine Rolle spielen. Wenn sofort ein freier Raum benötigt wird, zeigt das Display den Status des Raumes an und bietet die Möglichkeit, den Raum über einen Klick zu reservieren. Für eine Reservierung zu einem späteren Zeitpunkt ist eine aufwendigere Bedienung des E-Paper Displays nötig.

Mehr als die Hälfte der Teilnehmer finden eine Bestätigung der Raumbuchung sinnvoll. Als häufigste Gründe dafür werden das Vermeiden von Leerbuchungen und eine gesteigerte Konsistenz von Buchung zum tatsächlichen Status genannt. Auf der Gegenseite stehen Bedenken in Form von Vergessen der Bestätigung sowie Mehrarbeit und zusätzlicher Aufwand bei der Buchung.

Auf die Frage nach bevorzugten Räumen bei der Buchung gibt es relativ eindeutige Ergebnisse. Knapp 60 % der Befragten nannten in diesem Zusammenhang die Räume R1, R2 und R4. Häufig genannte Gründe hierfür sind die angenehme Raumgröße für bis zu acht Personen und die Helligkeit des Raumes durch die großen Fenster (Tageslicht). Den Raum R3 bevorzugen nur etwa 28 % der Teilnehmer und führen dies meist auf die Nutzung des Videokonferenzsystems zurück. Da dieser Raum keine Fenster



3 Konzeption

für direktes Tageslicht hat, ist er auch weniger beliebt. Die Räume R3 und R4 sind durch ein Milchglas bzw. einen Vorhang zusätzlich diskreter, was für die Befragten bei gewissen Terminen relevant ist. Außerdem stimmten etwa 12 % für den Raum R6 und begründeten dies mit der großen Anzahl an Teilnehmern.

Es besteht außerdem die Möglichkeit, Besprechungsräume über einen *Slackbot* zu reservieren. Ein Slackbot ist sozusagen ein Chatbot innerhalb der Nachrichtenplattform Slack. Dieser kann über eine API relativ einfach eingerichtet werden. Damit ist es möglich, Räume kommandobasiert abzufragen oder zu reservieren. Nach den Ergebnissen des Fragebogens wird dieser Slackbot allerdings bisher kaum bis gar nicht genutzt. Dies ist möglicherweise auf die Bedienung zurückzuführen. Auf die Frage, wie die Befragten am liebsten ihre Raumbuchung durchführen würden, antworteten aber immerhin 6,5 % mit der Antwort: *Slackbot*. Daraus lässt sich schließen, dass der Wunsch nach der Durchführung einer Raumbuchung über ein Conversational User Interface bei einigen Nutzern vorhanden ist, bisher aber noch nicht zufriedenstellend erfüllt wurde.

3.1.2 Interviews

Durch den Fragebogen ist es möglich, einen allgemeinen Einblick in das Verhalten und die Präferenzen bei der Raumbuchung zu erhalten. Diese Methode liefert allerdings hauptsächlich quantitative Daten. Um einen tieferen Einblick in das Nutzungsverhalten zu erlangen, sollen nun qualitative Daten gesammelt werden. Diese Daten können im Rahmen eines Interviews erhoben werden. Diese Methode bietet den Vorteil, direkt auf die Antworten des Interviewten eingehen zu können und mögliche Unklarheiten sofort zu beseitigen. Zudem liefert ein Interview durch die direkte Kommunikation sehr komplexe und detaillierte Daten. Damit ist gemeint, dass zusätzlich zu den verbalen Aussagen auch die Mimik und Gestik des Befragten wertvolle Hinweise liefern kann.

Bei der Auswahl der Probanden wird Wert darauf gelegt, unterschiedlichste Charaktere und Rollen zu interviewen. Diese sollen sich sowohl im Tätigkeitsfeld, als auch in ihren technischen Fähigkeiten oder Präferenzen unterscheiden. Die Haupttätigkeitsfelder der vier Interviewten sind in Tabelle 3.2 dargestellt.



3 Konzeption

Interviewter	Haupttätigkeitsfeld	Kürzel
I1	Development	D
I2	Coaching	C
I3	Human Resources	HR
I4	Vertrieb	V

Tabelle 3.2: Haupttätigkeitsfelder der Interviewten

Im Verlauf des Interviews wird dabei auf verschiedene Zusammenhänge und Hintergründe im Kontext einer Raumbuchung eingegangen. Die Kernpunkte und Überbegriffe des Interviewleitfadens sind nachfolgend aufgelistet.

- Vorgehen/Ablauf bei der Raumbuchung
- Fragen zum E-Paper Display
- Fragen zum Zeitmanagement
- Ergänzende Fragen

Ein vollständiger Interviewleitfaden mit allen Fragen ist in Anhang A.2 zu finden. Um sicherzustellen, dass keine Informationen verloren gehen, wird das Interview mit einem Aufnahmegerät aufgezeichnet. Der Interviewer kann während des Interviews Notizen anfertigen und eine ausführliche schriftliche Auswertung nachträglich durchführen. Eine entsprechende Einverständniserklärung wird jedem Teilnehmer vor Beginn des Interviews zur Unterschrift vorgelegt. Diese ist im Anhang A.3 zu finden.

In Bezug auf die Durchführung eines Interviews gibt es einige wichtige Verhaltensregeln. So ist es zunächst wichtig, dem Teilnehmer einen neutralen und vertrauenswürdigen Eindruck zu vermitteln. Dazu können so genannte *Warming-Up Fragen* als Eisbrecher genutzt werden. Die Fragen im Verlauf des Interviews sollten neutral formuliert werden, sodass sie den Befragten nicht in eine Richtung lenken. Um optimale Ergebnisse zu erzielen, kann das *Meister-Schüler-Modell* angewendet werden. Der Interviewte ist dabei der Meister, der Interviewer der Schüler. Alles, was der Meister sagt, ist richtig. Der Interviewer passt sich also dem Interviewten an, widerspricht und verbessert ihn nicht. Der Teilnehmer kann keine falschen Antworten geben. Ziel des Interviews ist es, die Gedanken und Meinungen des Nutzers zu erfassen und zu verstehen.



3 Konzeption

Nachfolgend sollen einige Auffälligkeiten, Übereinstimmungen oder auch Diskrepanzen im Rahmen der vier Interviews beschrieben werden. Eine ausführliche, schriftliche Auswertung ist im Anhang A.4 zu finden.

Auffällige Übereinstimmungen gibt es im Raumbuchungsverhalten, sowohl um einen Raum jetzt sofort, als auch zu einem späteren Zeitpunkt zu buchen. Wie bereits im Fragebogen erkennbar war, werden die E-Paper Displays nur genutzt um einen Raum für jetzt sofort zu buchen. Dieses Vorgehen ist bei allen Interviewten nahezu identisch. Es wird zunächst physikalisch in den Raum und am E-Paper nachgeschaut, ob der Raum frei ist und dieser wird anschließend direkt per Knopfdruck gebucht. Wird ein Raum zu einem bestimmten Zeitpunkt in der Zukunft benötigt, bevorzugen die Interviewten eine Raumbuchung am Rechner über das *Web Interface* vom Google-Kalender. Folgende Stärken und Schwächen wurden in diesem Zusammenhang erwähnt. In den nachfolgenden Klammern steht jeweils, wie oft der Aspekt erwähnt wurde.

- + Freie Zeiten von Räumen und Personen nebeneinander darstellen (3)
- + Überblick und Übersichtlichkeit (3)
- + Längere Termine oder Serientermine (1)
- + Feedback falls Raum nicht frei (1)
- + Andere Leute können Termin bearbeiten (1)
- + Möglichkeit zum Einplanen von Pufferzeiten (1)
- + Überall verfügbar, sowohl Office als auch privater Rechner (1)
- Anfrage zum Verschieben bzw. Verlegen von anderen Terminen (2)
- Keine genaue Information bei Konflikt in wiederkehrendem Termin (1)
- Räume fälschlicherweise als verfügbar angezeigt (1)
- Raumnamen irritierend (1)
- Hangout-Link standardmäßig im Termin (1)
- Kürzel der Teilnehmer unpersönlich (1)
- Teams bzw. Gruppen definieren (1)



3 Konzeption

In Bezug auf die Verwendung des Slackbots äußerte I1, dass bei der Verwendung unklar ist was geschrieben werden muss, um einen Raum zu buchen. Außerdem sind die Antworten bzw. das Feedback des Slackbots nicht hilfreich. Dies deckt sich mit der Interpretation der Ergebnisse des Fragebogens aus 3.1.1.

Auch die Frage nach der Bestätigung der Raumbuchung spiegelt in etwa die Eindrücke aus dem Fragebogen wider. Die prinzipiellen Vorteile dieser Idee, wie die Vermeidung von Leerbuchungen oder die bessere Konsistenz des Raumstatus, erscheinen allen Interviewten als sinnvoll. Allerdings gibt es auch hier Bedenken hinsichtlich der Umsetzung und Art der Bestätigung. Außerdem sind dabei unterschiedliche Betrachtungsweisen aufgefallen. Der einen Ansicht nach ist es besser, wenn Besprechungsräume einen freien Raumstatus anzeigen, aber dennoch belegt sind. Der anderen Ansicht nach ist es besser, wenn die Räume im Zweifel belegt anzeigen und dann doch frei sind. Dies soll durch ein Zitat von I2 verdeutlicht werden: „*Ich find's blöder wenn belegt ist und frei drin steht als wenn's andersrum der Fall ist*“

Der Wunsch nach Diskretion tauchte mehrfach im Rahmen der Interviews auf. So äußerte I3 den Wunsch nach einer Art *Privatisierung* des E-Paper Displays bei gewissen Terminen, wie Mitarbeiter- oder Vorstellungsgesprächen. Es soll von außen nicht erkennbar sein, welchen Titel der Termin hat. Nachdem das E-Paper allerdings nur den Titel aus dem Kalender anzeigt, ist dies eher als Problem auf den Google-Kalender zurückzuführen. Weiterhin spielte in diesem Zusammenhang die Auswahl des Besprechungsraums eine Rolle. Wirklich diskret, begünstigt durch Milchglas bzw. einen Vorhang, sind nur die beiden Besprechungsräume R3 und R4. Diesen Aspekt äußerte I3 ebenfalls bei der Frage nach bevorzugten Räumen.

Generell decken sich die Aussagen bezüglich der bevorzugten Räume mit den Ergebnissen des Fragebogens. Die Räume R1, R2 und R4 sind aufgrund der Helligkeit durch Tageslicht und ihre angenehme Größe sehr beliebt. Der Raum R4 hat durch seine Hochstühle außerdem einen gewissen Workshop-Charakter. Wie bereits erwähnt, bieten die Räume R3 und R4 eine gewisse Diskretion und werden daher für bestimmte Arten von Meetings bevorzugt. Zudem ist der Raum R3 der Einzige, der über ein Videokonferenzsystem verfügt. I4 merkte allerdings an, dass die Räume R3 und R4 durch die Lüftung und Klimaanlage teilweise sehr laut werden können. Zudem hat der Raum R3 leider keine Fenster mit direktem Tageslicht und wirkt deshalb sehr dunkel. Auf den Raum R6 greifen die Interviewten in der Regel nur zu, wenn es die Teilnehmerzahl erfordert. Generell gilt dabei der Gedanke, den Kollegen keine *besonderen*



3 Konzeption

Räume (Videokonferenzraum R3, großer Raum R6) wegzunehmen, wenn diese nicht unbedingt benötigt werden.

In Anlehnung an die Ergebnisse aus dem Fragebogen und den Interviews ergibt sich die in Tabelle 3.3 dargestellte, standardmäßige Priorisierung der Räume.

Prio	Raum	Plätze	Tageslicht	Diskretion	Besonderheiten
1	R1	8	✓	✗	-
2	R2	8	✓	✗	-
3	R4	8	✓	✓	Hochstühle
4	R3	9	✗	✓	Videokonferenz
5	R6	20	⚠	✗	-

Tabelle 3.3: Priorisierung der Räume

I3 erwähnte den interessanten Ansatz, aus der Formulierung der Buchungsanfrage den Kontext zu erschließen. Somit könnten hierbei bereits Informationen über die Art des Termins gewonnen und eine entsprechende Priorisierung der Räume getroffen werden. Formuliert der Nutzer seine Anfrage beispielsweise nach einem Mitarbeiter- oder Vorstellungsgespräch, schlägt das System zuerst einen Raum mit Diskretion (R3, R4) vor. Möchte der User hingegen einen Raum für einen Workshop oder eine Schulung buchen, könnte dazu implizit eine große Teilnehmerzahl angenommen werden und der Raum R6 vorgeschlagen werden.

In diesem Zusammenhang könnte auch eine Verbesserung des Workflows beim Buchen von ganztägigen Terminen getroffen werden. I3 merkte an, dass hierzu derzeit oft drei Buchungen durchführt werden. Einen Slot zum Vorbereiten des Termins, eine Kernzeit mit allen Teilnehmern und einen Slot zum Nachbereiten des Termins. Wünschenswert wäre hier ein Workflow, der nur eine Kernbuchung beinhaltet und im Buchungsprozess eine Vor- und Nachbereitung vorsieht, welche dann automatisch *um den Kerntermin herum* gelegt wird.

Auch wenn einige der Erkenntnisse aus den konkreten Interviewfragen weniger hilfreich für die nachfolgende Bearbeitung des Themas sind, können dennoch wichtige Gesichtspunkte in die Entwicklung des Systems fließen. Vor allem die Priorisierung der Räume und Ideen zur Verbesserung des Workflows erscheinen äußerst sinnvoll und geben gute Ansatzpunkte für den weiteren Verlauf der Arbeit. Außerdem konnten



3 Konzeption

dadurch viele wertvolle Informationen über das Verhalten und die Präferenzen bei der Raumbuchung und -nutzung gewonnen werden.

3.1.3 Personae

Die Persona ist eine häufig eingesetzte Methode aus dem Bereich Human Computer Interaction. Sie stellt eine Art *Prototyp* für eine bestimmte Gruppe von Nutzern dar und verkörpert deren Bedürfnisse, Fähigkeiten, Verhaltensweisen und Ziele. Eine Persona ist also eine fiktive Person, mit deren Hilfe Entwickler und Designer eine klare Vorstellung über die Empathien und Verhaltensweisen der Nutzer erhalten.

Nicht zu unterschätzen sind in diesem Zusammenhang die Charaktereigenschaften einer Persona. Der erstellte Charakter soll einprägsam und einfach zu verinnerlichen sein. Dazu wird der Persona mit zusätzlichen Informationen wie Name, Alter, einem Bild oder persönlichem Zitat mehr Leben eingehaucht. [RF16, S. 57-58]

Der Ausgangspunkt für die Erarbeitung von Personae sind Informationen über die zukünftigen Benutzer eines Systems. Dazu dienen die Ergebnisse aus dem Fragebogen und den Interviews, die in den vorherigen Kapiteln beschrieben wurden. Da eine Persona stellvertretend für eine Gruppe von Nutzern steht, werden nur einige fiktive Personen erdacht. Die Auswahl der Personae ist angelehnt an die vier Interviewten aus 3.1.2 und wird im Folgenden nochmals kurz vorgestellt.

- **D:** Mitarbeiter in der Projektumsetzung (Development, Design, DevOps)
- **HR:** Mitarbeiter im Bereich Human Resources
- **C:** Mitarbeiter im Bereich Coaching
- **V:** Vertriebsmitarbeiter

Die vier Personae sollen bestimmte Verhaltensweisen darstellen und gegebenenfalls etwas überzeichnen. Zur genaueren Definition der Verhaltensweisen werden Gegensatzpaare gebildet und die Personae entsprechend eingeordnet. Die Gegensatzpaare beziehen sich auf die verschiedenen Arten der Termine sowie das Verhalten und die Präferenzen bei der Raumbuchung. Die Einordnung geht aus den Informationen des Fragebogens und der Interviews hervor und ist in Abbildung 3.3 dargestellt.

3 Konzeption

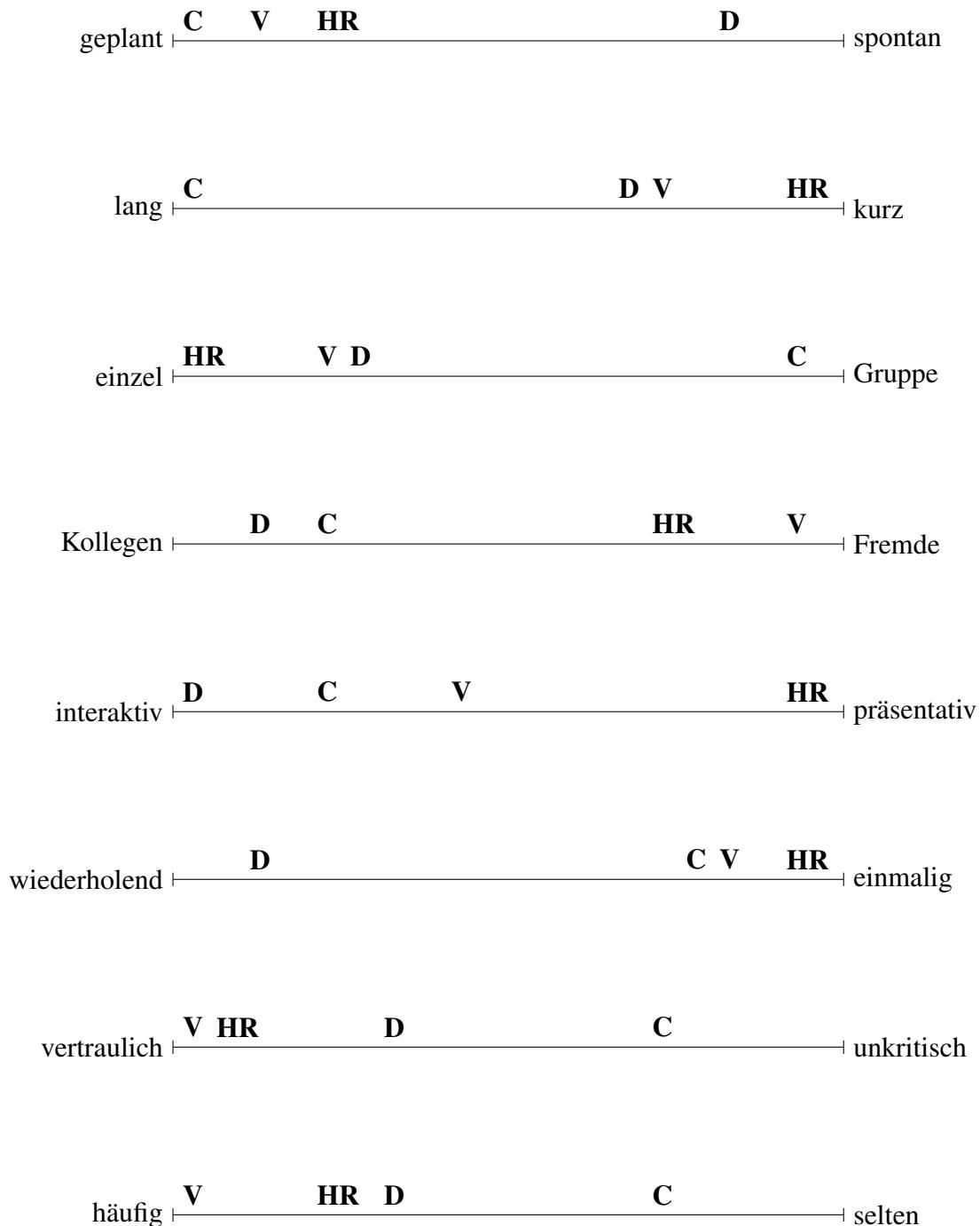


Abbildung 3.3: Einordnung der Kriterien zu den Personae

Alle erdachten Personae werden im Folgenden kurz beschrieben. Eine ausführliche Darstellung befindet sich aus Gründen der besseren Lesbarkeit im Anhang A.5.



3 Konzeption

Persona Development

In Abbildung A.1 ist Heiko Neumann dargestellt. Er ist Softwareentwickler und stellt die Persona aus dem Bereich *Development* dar. In seiner täglichen Arbeit möchte Heiko jederzeit spontane Code-Reviews durchführen können. Außerdem organisiert und verwaltet er ein wöchentliches Team-Meeting zum fachlichen Austausch mit seinen Kollegen.

Persona Coaching

Die Abbildung A.2 zeigt die Persona Sandra Böhm aus dem Bereich *Coaching*. Sie veranstaltet in der Regel Workshops und Schulungen. In dem Zusammenhang ist sie im ständigen Austausch mit verschiedenen Kollegen, um einen passenden Termin für den nächsten Workshop zu finden. In diesem Zusammenhang möchte sie nicht ewig nach freien Räumen suchen müssen.

Persona Human Resources

Michelle Färber aus Abbildung A.3 ist an den Bereich *Human Resources* angelehnt. Vorstellungs- und Mitarbeitergespräche stehen bei ihr an der Tagesordnung. Um ihrem Gegenüber eine angenehme Gesprächsatmosphäre bieten zu können, legt sie großen Wert auf Privatsphäre und Diskretion. Sie ist der Meinung, dass nur unter harmonischen Bedingungen gute Ergebnisse erzielt werden können.

Persona Vertrieb

Die Persona Eric Schweizer aus Abbildung A.4 repräsentiert den Bereich *Vertrieb*. Zwischen seinen eng getakteten Terminen pendelt er oft zwischen unterschiedlichen Orten. Der Vertriebsmitarbeiter ist nur selten im Office und nutzt sein Smartphone, um den Überblick in seinem vollen Terminkalender zu behalten und sich auf die zeitliche Verfügbarkeit seiner Kunden einzustellen.

Durch das Modellieren der Personae auf Grundlage des Fragebogens und der Interviews können wichtige Erkenntnisse gewonnen werden. Die Personae liefern dabei wertvolle Informationen im Zusammenhang mit dem Nutzungskontext und die damit verbundenen Anforderungen an das System. Abschließend konnte hierbei ein guter Eindruck über die Ziele und Wünsche der Nutzer aus den verschiedenen Tätigkeitsfeldern vermittelt werden.



3.2 Prototyping

Prototyping ist eine Methode, bei der so früh wie möglich eine erste, vereinfachte Version (Prototyp) der angestrebten Lösung realisiert wird [FFH02, S. 25]. Diese wird genutzt, um das System zu evaluieren und zu verbessern, noch bevor ein lauffähiges System vorhanden ist. Abhängig vom verfolgten Ziel ist der Prototyp in seinen Eigenschaften mehr oder weniger stark ausgeprägt. Ein Prototyp lässt sich dabei durch die Betrachtung der folgenden Dimensionen charakterisieren. [RF16, S. 72-73]

- Der *Funktionsumfang* beschreibt, welche und wie viele Funktionen, bezogen auf den gesamten Umfang, im Prototyp gezeigt werden sollen
- Als *Funktionstiefe* wird bezeichnet, wie detailliert die einzelnen Funktionen wiedergegeben werden sollen
- Die *Darstellungstreue* gibt an, wie ähnlich der Prototyp dem Endprodukt in Bezug auf Aussehen und Benutzeroberfläche sein soll
- Mit der *Interaktivität* wird beschrieben, wie interaktiv der Prototyp sein soll
- Mit *Datengehalt* ist gemeint, ob reale Daten, Beispiele oder nur Platzhalter für Bezeichnungen zum Einsatz kommen
- Die *technische Reife* beschreibt, wie viel der endgültigen Technologie im Prototyp verwendet wird

In diesem Zusammenhang wird auch oft von *low-fidelity*, *mid-fidelity* und *high-fidelity* Prototypen gesprochen. Diese Begriffe und deren Unterschiede werden daher im Folgenden kurz erläutert.

- *low-fidelity*: Dies ist die am einfachsten gehaltene und kostengünstigste Ausprägung eines Prototyps. In den ersten Schritten werden oft einfache Sketches mittels Stift und Papier entworfen. Die Ideen der Entwickler können dabei schnell in eine begreifbare Form gebracht werden. Durch die Visualisierung und Kommunikation ergibt sich ein besseres Verständnis des Produkts.
- *mid-fidelity*: Oft wird nur zwischen low-fidelity und high-fidelity gesprochen. Ein mid-fidelity Prototyp ist eine Art Zwischenschritt und entspricht einem erweiterten low-fidelity Prototyp. Seine Darstellung ist dabei weiterhin zurückhaltend und größtenteils in schwarz-weiß gehalten. Der Vorteil ist, dass hierbei



3 Konzeption

bereits digitale und interaktive Benutzeroberflächen zum Einsatz kommen. Benötigte Daten oder komplexe Funktionen werden dabei meist vereinfacht oder nur vorgetäuscht.

- *high-fidelity*: Dieser Prototyp bildet ein interaktives System so ab, wie es tatsächlich sein soll. Das Design ist dabei meist größtenteils ausformuliert, Funktionsumfang oder -tiefe können dagegen noch eingeschränkt sein. Diese Art ist dabei sehr zeit- und ressourcenintensiv, weshalb dabei oft Technologien zum Einsatz kommen, die später auch im Produkt genutzt werden können.

Der Prototyp im Rahmen der Konzeption dieser Arbeit wird dabei zwischen low- und mid-fidelity umgesetzt. Der Funktionsumfang soll dabei relativ hoch sein und möglichst viele Szenarien des endgültigen Produkts anbieten. Die Funktionstiefe beschränkt sich dabei eher auf die visuelle Darstellung ohne Anbindung an ein Backend oder ähnliches. Die Darstellung soll bereits in vielen Teilen in die Richtung des Produkts gehen, dabei aber trotzdem dezent wirken. Zudem soll der Prototyp sehr interaktiv sein, um auch tatsächlich den realen Kommunikationsfluss bei der Interaktion zu simulieren. Als Datenmodell werden dabei statische, fiktive Werte zum Einsatz kommen. Technisch gesehen lässt sich für das Endprodukt dabei wenig bis gar nichts nutzen, da der Prototyp in einer komplett anderen Entwicklungsumgebung und Zielplattform entwickelt wird.

3.2.1 Anforderungen

Beim prototypischen Aufbau des Chatbots geht es darum zu erkennen, ob eine gewisse Konversationsstruktur oder UI-Erweiterung für die User funktioniert und verständlich ist oder nicht. Der Prototyp soll daher die Möglichkeit bieten, Ideen relativ schnell und unkompliziert umsetzen und evaluieren zu können. Ist diese Möglichkeit gegeben, können die erarbeiteten Konzepte sehr früh getestet werden (*test early - fail early*). Der Vorteil ist dabei, dass Probleme frühzeitig erkannt und beseitigt werden können. Kommen diese Fehler erst im späteren Verlauf zum Vorschein, verursachen sie deutlich mehr Aufwand. In Bezug auf einen komplexen Chatbot, der in der Realität von Trainingsdaten und Machine Learning profitiert, muss eine vereinfachte Testumgebung verwendet werden.

Eine erste Möglichkeit ist dabei das *Paper Prototyping*. Bei dieser low-fidelity Methode wird mit Hilfe von gezeichneten oder gedruckten Komponenten das Verhalten eines

3 Konzeption

zu entwickelnden Systems simuliert. Dabei können vor allem erste, prinzipielle Ideen wie der Aufbau eines User Interfaces und die Anordnung von Labels oder Buttons evaluiert werden. In Bezug auf die Entwicklung eines Chatbots müssen dafür vorgefertigte Textpassagen oder UI-Elemente erstellt werden, die dann bei Bedarf genutzt werden können. Ein Vorteil dieser Methode ist sicherlich, dass Skizzen sehr schnell erstellt und angepasst werden können. Außerdem können alle Personen intuitiv damit umgehen. Diese Vorgehensweise erweckt beim Probanden allerdings nicht den Eindruck, dass er ein interaktives oder gar intelligentes System bedient. Daher soll zunächst ein anderer Ansatz verfolgt werden.

Nachdem als Prototyp kein komplett funktionales System aufgesetzt wird, soll dies mit Hilfe der *Wizard of Oz (WOz)* Methode umgesetzt werden. Mit WOz-Prototypen werden meist komplexe Systemreaktionen durch einen menschlichen Assistenten simuliert [Web08, S. 342]. So können komplexe Dialoge evaluiert werden, bevor überhaupt ein spracherkennendes System implementiert wird. Der menschliche Agent (*Wizard of Oz*) steuert die Reaktionen des Prototyps auf Basis von Texteingaben des Users. Es muss dazu also keine aufwendige NLP-Plattform verwendet oder gar implementiert werden. Dabei geht der WOz nach konkreten Regeln vor, die zuvor für das System festgelegt wurden. Ein WOz-Test kann gemäß Abbildung 3.4 aufgebaut werden.

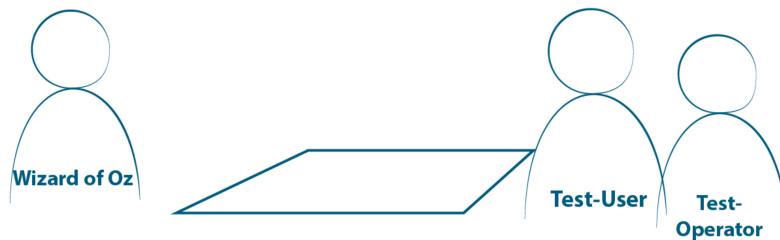


Abbildung 3.4: Einfaches Prototyping mit der Wizard of Oz Methode

Dabei sitzt der Wizard of Oz auf der einen, der Test-User und Test-Operator auf der anderen Seite eines Tisches. Der Operator moderiert das Szenario und führt die Testperson durch die Aufgaben. Der User bearbeitet die Aufgaben und sendet seine Anfragen an den WOz, der diese nach einem zuvor definierten Ablaufszenario beantwortet. Der Operator kann währenddessen Notizen über die Testdurchführung machen.

Der große Nachteil bei einer Durchführung an diesem Aufbau ist, dass sich alle Beteiligten gegenseitig sehen und hören können. Der Wizard of Oz bekommt also sowohl

3 Konzeption

die Kommunikation zwischen Test-User und Test-Operator, als auch deren Gestiken und Mimiken mit. Er könnte das System durch diese weiteren Informationskanäle daher anders bedienen, als er es bei einer reinen Textanfrage tun würde. Dieses Problem gilt gleichermaßen auch umgekehrt. Wenn die Testperson den WOz sieht und realisiert, dass die Antworten von ihm kommen ist die Illusion von der Interaktion mit einem intelligenten System zerstört. [Blu17]

Diese Probleme lassen sich beheben, wenn der Wizard of Oz in einen Raum, Test-User und Test-Operator in einen anderen Raum platziert werden (siehe Abbildung 3.5).

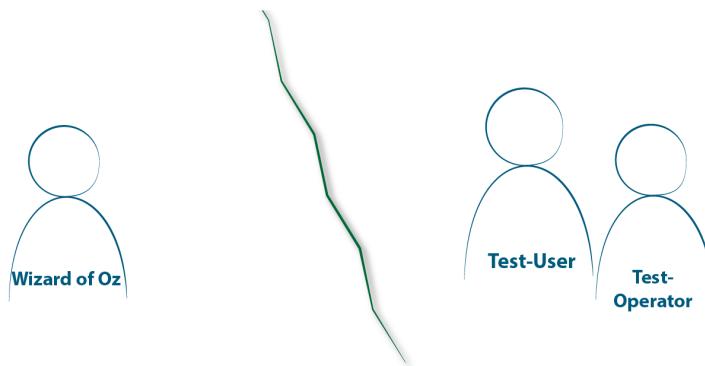


Abbildung 3.5: Erweitertes Prototyping mit der Wizard of Oz Methode

Im einfachsten Fall könnte dies mit Hilfe eines *Chat Messengers*, wie z.B. Slack umgesetzt werden. User und Operator sitzen dazu in einem Raum, der WOz in einem anderen. Beide haben jeweils ein Fenster des Chat Messengers geöffnet, wodurch sie ihre Nachrichten austauschen können. Die prinzipiellen Anforderungen an getrennte Räume und der Möglichkeit zum Textnachrichten-Austausch sind damit gegeben. Problematisch wird in diesem Zusammenhang aber die Chat-Umgebung an sich. Zunächst könnte der User erkennen, dass es sich um eine standardmäßige Messaging Platform handelt und die Illusion vom intelligenten Chatbot zerstören. Weiterhin sind die Möglichkeiten zur UI-Erweiterung begrenzt, wodurch nur ein Bruchteil des Funktionsumfangs getestet werden könnte. Diese Nachteile sollen im nächsten Schritt egalisiert werden.

Der Test-User soll nun eine prototypische Benutzeroberfläche an einem Rechner bedienen. Seine Eingaben sendet er dabei über eine Netzwerkverbindung direkt an den WOz. Nachdem dieser ein Mensch ist, versteht er alle Intentionen des Nutzers und kann diese entsprechend zuordnen. Dabei geht er nach festgelegten Regeln vor und

3 Konzeption

sendet die Antworten zurück an den User. Diese Antworten tippt er nicht selbst, sondern bedient sich dabei an vorgefertigten Phrasen, Textpassagen oder UI-Elementen. Diese unterscheiden sich dabei je nach Szenario oder Intention des Nutzers. Bei einer übersichtlichen Gestaltung der Benutzeroberfläche findet sich der Wizard of Oz auch bei komplexen Tests zurecht.

3.2.2 Prototyping Tool

Um all diese Anforderungen umsetzen zu können, wird ein *conversation prototyping tool (cpt)* verwendet¹. Umgesetzt wird dies mit *Max* von *Cycling74*. Max ist eine grafische Entwicklungsumgebung, in der Module in einer Art *Baukastensystem* verbunden werden. Die Entwicklung und Implementierung in diesem Zusammenhang ist nicht Bestandteil dieser Arbeit und soll daher nicht weiter beschrieben werden. Das Prototyping-System wird im Rahmen der Arbeit zur konzeptionellen Überprüfung der Konversationsstruktur und der UI-Elemente genutzt. Wichtig zu erwähnen ist in diesem Zusammenhang, dass es gemäß der vorherigen Überlegungen zwei Benutzeroberflächen (Panels) gibt. Eine der Benutzeroberflächen bedient der Wizard of Oz, die andere der Test-User. Wie bereits zuvor festgelegt, befinden sich die jeweiligen Rechner in verschiedenen Räumen und tauschen ihre Daten über eine Netzwerkverbindung aus.

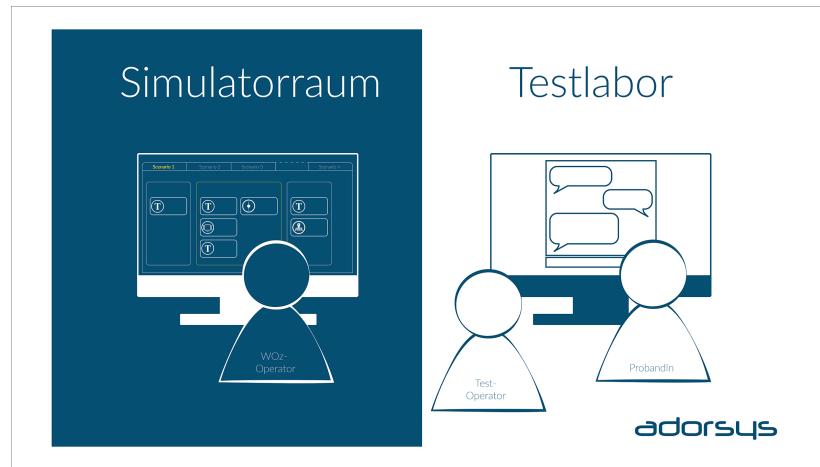


Abbildung 3.6: Kompletaufbau des Prototypings mit der Wizard of Oz Methode

¹Die Arbeit im Prototyping Tool stammt vom CUI-Team der adorsys GmbH & Co. KG unter der Leitung von Steffen Blümm

3 Konzeption

Mit dem *Tester-Panel* interagiert in der Regel nur der Test-User. Es ist dafür zuständig, dass die vom Nutzer eingegebenen Textanfragen übertragen werden. Seine Textnachrichten kann der User über die dunkelgraue Leiste am unteren Rand eingeben und abschicken. Außerdem gibt dieses Panel die vom Wizard of Oz generierten Antworten aus. Diese können, wie schon im vorherigen Teil der Arbeit beschrieben, sowohl Textnachrichten als auch erweiterte UI-Elemente sein. Ziel dieser prototypischen User-Tests ist herauszufinden, welche erweiterten User Interfaces für den Nutzer verständlich sind und ihn bei der Raumbuchung unterstützen. Die entsprechenden UI-Elemente können dann interaktiv bedient werden. Ein beispielhafter Ausschnitt des Tester-Panels ist in Abbildung 3.7 dargestellt. Eine ausführliche Darstellung aller UI-Elemente befindet sich aus Gründen der besseren Lesbarkeit im Anhang A.6.

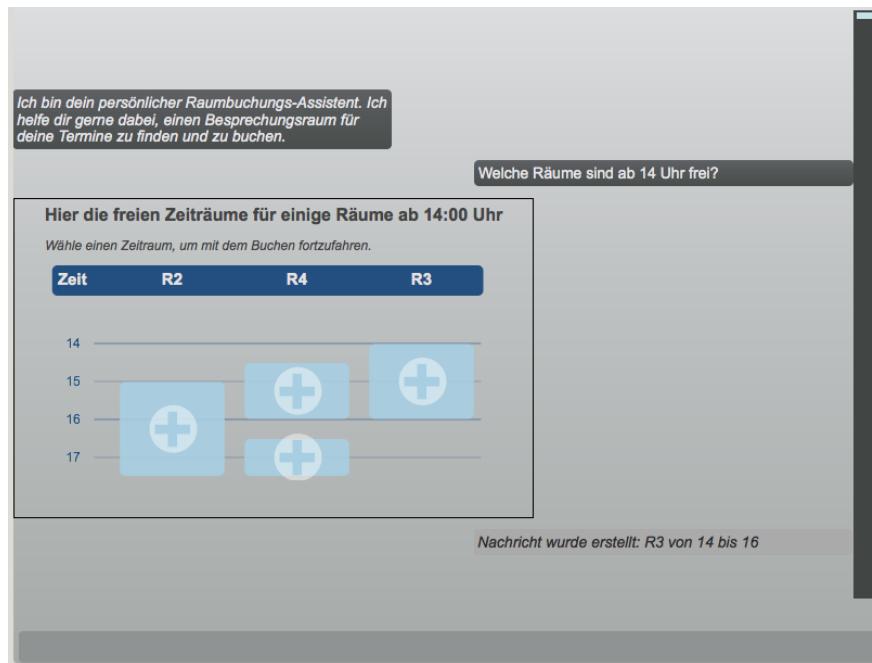


Abbildung 3.7: Grafische Benutzeroberfläche des Tester-Panels

Der WOz sitzt während der Durchführung vor dem *Simulator-Panel*. Diese Benutzeroberfläche kann gemäß eines vorher definierten Szenarios dynamisch aufgebaut werden. Dabei kann auf vordefinierte Phrasen oder UI-Elemente zurückgegriffen werden. Zudem blinkt ein kleiner Kreis oben links, sobald der User auf der Tastatur tippt. Sendet er eine Nachricht ab, erscheint diese ebenfalls in der oberen linken Ecke. Durch das Triggern der Nachrichten wird Zeit gespart, da der WOz nicht jede Antwort selbst generieren und formulieren muss. Außerdem ist so die Möglichkeit gegeben, dass jeder



3 Konzeption

User innerhalb eines bestimmten Szenarios die gleichen Antworten und Formulierungen erhält, wodurch eine Verfälschung der Testergebnisse verhindert wird. Die Benutzeroberfläche des Simulator-Panels ist in Abbildung 3.8 dargestellt.



Abbildung 3.8: Grafische Benutzeroberfläche des Simulator-Panels

Um ein *Szenario* mit den verschiedenen Intents und Responses zu erstellen, wird eine Art *Skript* verwendet. Dieses Skript wird im Format *JavaScript Object Notation (JSON)* geschrieben und enthält die entsprechenden Elemente in strukturierter Form. Für jedes Szenario gibt es dabei einen eigenen Tab (Tabs sind nicht in der Abbildung 3.8 ersichtlich).

Innerhalb jedes Szenarios können dann *Intents* (blaue Rahmen) angelegt werden. Die ersten beiden Intents jedes Szenarios heißen *helpIntent* und *errorIntent*. Diese sind zum Schnellzugriff in der oberen Leiste (gelbes und orangefarbene Felder) und triggern einfache Textnachrichten. Jeder Intent ist gleichermaßen aufgebaut und enthält die Parameter *name*, *utterances* und *responses* (siehe Abbildung 3.9).

Mit dem Parameter *name* kann dem Intent ein Überbegriff zugeordnet werden. Unter *utterances* können einige Beispielsätze des Users aufgezählt werden, die dem Operator das Bedienen erleichtern. Der letzte Parameter *responses* kann je nach Art der Nachricht sehr umfangreich werden.

Jede Response hat einen Parameter *subject*, mit dem ihr ein Name zugeordnet werden kann. Mit Hilfe einer *phrase* wird angegeben, welche Textnachricht an den User gesendet wird. Mit einem *hint* kann dieser Text darunter zusätzlich in kleinerer Schrift



3 Konzeption

ergänzt werden. Die Werte für *type* und *tags* ordnen der Response ein entsprechendes UI-Element zu. Dabei gibt es verschiedene Elemente, wie z.B. eine *CalendarView*, eine *CollectionView* oder unterschiedliche *Slidebars*. Mit Hilfe des Parameters *payload* können dann zusätzliche Informationen über Startzeit, Dauer, Datum oder Raumname mitgegeben werden. An dieser Stelle soll aus Gründen der Übersichtlichkeit auf eine umfangreiche Darstellung verzichtet werden. Stattdessen ist nachfolgend zur beispielhaften Darstellung der Struktur einer Response die Implementierung einer kurzen *Textmessage* in Listing 3.1 dargestellt.

```
▼ object {2}
  projectName : Usability Testessen
  ▼ scenarios [1]
    ▼ 0 {2}
      name : Szenario1
      ▼ intents [6]
        ▼ 0 {3}
          name : helpIntent
          ► utterances [5]
          ► responses [1]
        ▼ 1 {3}
          name : errorIntent
          ► utterances [5]
          ► responses [1]
        ▼ 2 {4}
          name : AnyRoomIntent
          ► priority {1}
          ► utterances [2]
          ▼ responses [2]
            ▼ 0 {6}
              subject : presentRoomsWithFreeSlots
              phrase : Hier einige Räume, die morgen vormittag frei sind.
              hint : Wähle einen Zeitraum, um mit dem Buchen fortzufahren.
              type : calendar
              ► tags [1]
              ► payload [1]
```

Abbildung 3.9: Struktur eines Szenarios im JSON-Format



3 Konzeption

```
1 {
2     "responses": [
3         {
4             "subject": "Help",
5             "phrase": "Ich bin dein persoenlicher Raumbuchungs Assistent. Ich
6                 helfe dir gerne dabei, einen Besprechungsraum fuer deine Termine
7                 zu finden und zu buchen.",
8             "type": "textmessage"
9         }
10    ]
11 }
```

Listing 3.1: Beispielhafte Response im JSON-Format

3.2.3 Usability Testessen

Erste User Tests mit dem *Prototyping Tool* konnten mit externen Nutzern im Rahmen eines so genannten *Usability Testessens*² durchgeführt werden. Bei dieser Veranstaltung bringen Entwickler ihre Prototypen, Websites oder Apps mit. Freiwillige, vom Produkt unabhängige Nutzer testen diese dann auf ihre Gebrauchstauglichkeit. Dies hat den Vorteil, dass die Nutzer in der Regel keinen Bezug zum Produkt haben und somit ein unvoreingenommenes und ehrliches Feedback abgeben.

Die Veranstalter selbst bezeichnen das als *Speedtesting*. Die Probanden testen die verschiedenen Produkte in sechs Runden à zwölf Minuten. Nach Ablauf einer Runde wandern die Tester zum nächsten Stand. Getestet wird dabei mit Hilfe der *Thinking-Aloud-Methode*. Dabei wird der Teilnehmer während der Testsitzung angehalten zu äußern, was er gerade denkt oder fühlt. Die Methode liefert somit ein schnelles Feedback und hilft, Usability-Probleme zu identifizieren. Um sicherzustellen, dass keine Informationen verloren gehen, wird der User Test mit einem Aufnahmegerät aufgezeichnet. Eine entsprechende Einverständniserklärung wird jedem Teilnehmer vor Beginn des Tests zur Unterschrift vorgelegt. Diese ist im Anhang A.7 zu finden.

Der Testdurchlauf besteht im Wesentlichen aus zwei Teilen. Zunächst wird der Proband gebeten, ein bestimmtes Szenario mit Hilfe des Protoyping Tools abzuarbeiten. Als Benutzeroberfläche steht ihm dabei das Tester-Panel (vgl. Abbildung 3.7) zur Verfügung. Alle seine Anfragen werden über eine Netzwerkverbindung an den Wizard

²www.usability-testessen.org



3 Konzeption

of Oz gesendet, der vordefinierte Antworten über das Simulator-Panel (vgl. Abbildung 3.8) auswählen kann. Dem Proband wird somit der Eindruck einer Interaktion mit einem intelligenten Chatbot vorgetäuscht, hinter dem in Wirklichkeit ein Mensch steckt. Während der Nutzer das Szenario bearbeitet, können dank der Thinking-Aloud-Methode wertvolle Notizen zu Wahrnehmung und Verhalten des Nutzers gemacht werden.

Szenario Usability Testessen

Beschreibung:

„Du willst morgen Vormittag um 10 Uhr an einem Meeting (Skype) teilnehmen, das bis zu 2 Stunden dauern kann. Die Videokonferenz findet an deinem eigenen Rechner statt. Um deine Kollegen nicht zu stören, willst du nicht vom Arbeitsplatz, sondern von einem Besprechungsraum aus teilnehmen.“

Aufgabe:

„Lasse dir die verfügbaren Räume anzeigen und reserviere anschließend den Raum R2 von 10 bis 12 Uhr.“

Ist das Szenario erfolgreich abgeschlossen, wird der User noch gebeten, einen *Feedback-Fragebogen* auszufüllen. Dieser besteht aus elf Gegensatzpaaren hinsichtlich der Wahrnehmung und Einstellung des Nutzers zum Produkt. Die Fragen sind dabei durch eine Skala mit fünf Auswahlmöglichkeiten zu beantworten. Eine Übersicht der Kriterien des Feedback-Fragebogens ist in der nachfolgenden Liste dargestellt.

- verständlich ↔ unverständlich
- leicht zu erlernen ↔ schwer zu erlernen
- schnell ↔ langsam
- aufgeräumt ↔ überladen
- unterstützend ↔ behindernd
- angenehm ↔ unangenehm
- einfach ↔ kompliziert
- sympathisch ↔ unsympathisch
- erwartungskonform ↔ nicht erwartungskonform

3 Konzeption

- effizient ↔ ineffizient
- übersichtlich ↔ verwirrend

Eine ausführliche Auflistung der Fragebögen mit Notizen zu den jeweiligen Teilnehmern befindet sich im Anhang A.8. Die Inhalte und deren durchschnittliche Bewertung sind in Abbildung 3.10 anhand eines Diagramms dargestellt. Die gelbe Referenzlinie zeigt die durchschnittliche Bewertung des Systems durch alle Probanden.

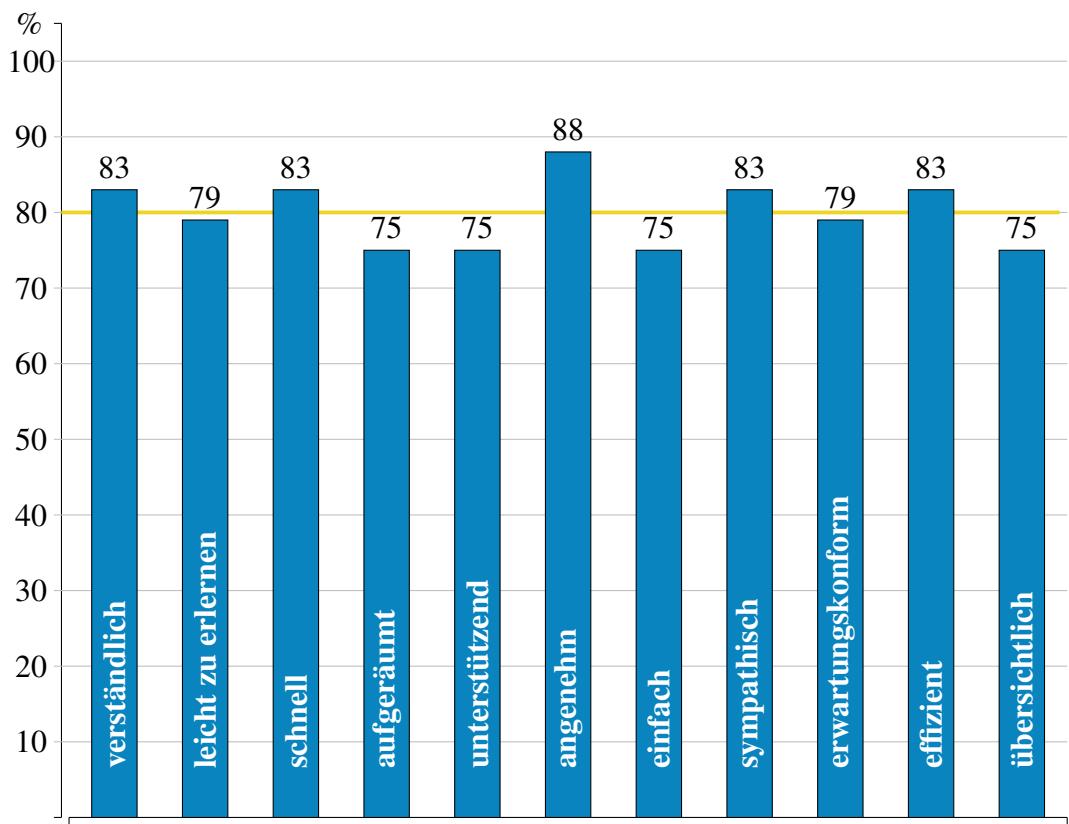


Abbildung 3.10: Ergebnisse des Usability Testessens

Im Großen und Ganzen gab die Probanden positives Feedback. Im Durchschnitt ergibt sich eine Zustimmung zum System von ca. 80 %. Insgesamt gibt es dabei keine allzu großen Diskrepanzen. Anzumerken ist, dass ein Großteil der Nutzer den Prototyp als *angenehm* (88 %), *verständlich* und *schnell* (je 83 %) empfand. Etwas weniger, aber dennoch 75 % Zustimmung bekamen Kriterien wie *aufgeräumt* oder *übersichtlich*, was jedoch auch auf die prototypische Umgebung zurückzuführen sein kann. Anzumerken ist hierbei, dass aufgrund von begrenzter Zeit und Umfang dieser Testveran-



3 Konzeption

staltung keine fundierte Aussage über qualitative oder funktionale Eigenschaften des Systems getroffen werden kann. Die Bewertung ist jedoch äußerst hilfreich, um ein erstes, schnelles Feedback der geplanten Applikation zu erhalten und hilft damit bei der weiteren Konzeption des Systems.

Die ersten Ergebnisse spiegeln sich auch mit einigen Zitaten der Probanden wider. „*Wenn das bei uns so einfach funktionieren würde, wäre das schon toll.*“ oder „*Das hätte ich auch gerne, ich kenne das komplizierter.*“ sind nur einige Zitate in diesem Zusammenhang. Daraus lässt sich ableiten, dass die Nutzer dem System grundsätzlich positiv gegenüberstehen und sowohl die Raumbuchung via Chatbot als auch die UI-Erweiterungen verständlich sind.

Von beinahe allen Probanden wurde außerdem erwähnt, dass das Feature *Chatbot is thinking* hilfreich ist und den Chatbot obendrein sympathisch macht. Mit dieser Funktion wird die Zeit überbrückt, die der Chatbot für die Verarbeitung einer Nutzereingabe benötigt. Als negatives Feedback erwähnten einige User die farblichen Markierungen im Lageplan. Hier geht die farbliche Markierung des gebuchten Raumes gegenüber den anderen Symbolen unter. Abgesehen davon war der Lageplan jedoch hilfreich. Als Anmerkung nannten die Nutzer immer wieder die Frage nach der Ausstattung der Räume. Hier spielten vor allem die Fragen nach WLAN, Anzahl der Plätze, Whiteboard, Flipchart, Beamer, Getränke oder Catering eine Rolle. Zudem äußerten einige User den Wunsch, dass der gebuchte Termin anschließend automatisch in ihren elektronischen Kalender eingetragen werden soll.

3.2.4 Szenarien

Im Rahmen von ausführlichen User Tests werden verschiedene Szenarien, basierend auf die erdachten Personae aus 3.1.3, entwickelt. Entsprechend den jeweiligen Bereichen Development, Coaching, Human Resources und Vertrieb wird jeweils ein spezifisches Szenario erstellt. Die Szenarien werden im Folgenden kurz vorgestellt und deren Hintergedanken beschrieben.

Das Szenario aus dem Bereich *Development* ist an die Persona Heiko Neumann angelehnt. Heiko ist Softwareentwickler und möchte zusammen mit einem Kollegen ein spontanes Code-Review durchführen. Er hat dabei keine besonderen Ansprüche an einen Raum. Für seine Buchung wird die entsprechende Raumhierarchie aus 3.3 angewendet.



3 Konzeption

Szenario Development

Beschreibung:

Du willst jetzt sofort (14:00 Uhr) ein Meeting (Code-Review) machen. Der Termin dauert mindestens eine halbe Stunde und findet mit einem weiteren Kollegen statt. Um ungestört arbeiten zu können, willst du einen beliebigen Besprechungsraum dafür buchen.

Aufgabe:

Lasse dir die verfügbaren Räume anzeigen und reserviere anschließend den Raum R3 so, dass du ihn ab jetzt eine Stunde nutzen kannst.

Weitere Informationen:

Titel für den Termin: Keinen Titel angeben

Sandra Böhm ist die Persona aus dem Bereich *Coaching*. Sie möchte einen interaktiven Scrum Workshop veranstalten. Aufgrund der Teilnehmerzahl von zwölf Personen benötigt sie dazu den Raum R6. Das System soll dabei erkennen, dass es sich um einen ganztägigen Termin handelt und bei der Buchung, entsprechend der Idee aus 3.1.2, nach Vor- und Nachbereitungszeit für den Termin fragen. Wählt der User eine entsprechende Vor- und Nachbereitungszeit aus, nimmt das System direkt alle drei Buchungen vor und erleichtert dem Nutzer somit die Buchung von drei einzelnen Terminen.

Szenario Coaching

Beschreibung:

Du willst einen interaktiven Scrum Workshop halten. Daran werden etwa zwölf Leute teilnehmen, weshalb du den Raum R6 benötigst. Der Termin soll als ganztägiges Ereignis stattfinden. Die Kernzeit für alle Teilnehmer ist dabei von 9:00 Uhr bis 17:00 Uhr. Für dich selbst möchtest du jedoch zusätzlich je eine Stunde zur Vor- und Nachbereitung des Workshops buchen.

Aufgabe:

Lasse dir die nächsten Termine anzeigen, an denen der Raum R6 ganztägig frei ist und reserviere ihn anschließend für den nächstmöglichen Zeitpunkt zu den oben stehenden Bedingungen.

Weitere Informationen:

Titel für den Termin: „Scrum Workshop“



3 Konzeption

Die Persona Michelle Färber repräsentiert den Bereich *Human Resources*. In dem spezifischen Szenario möchte sie einen Raum für ein Vorstellungsgespräch buchen. Wird dem System entsprechend mitgeteilt, dass es sich um ein Vorstellungsgespräch handelt, ändert sich die Priorisierung der Räume. Räume, die Diskretion bieten, werden in diesem Fall höher priorisiert und zuerst vorgeschlagen.

Szenario Human Resources

Beschreibung:

Du arbeitest im Bereich HR und möchtest einen Raum für ein Vorstellungsgespräch buchen. Du hast mit dem Bewerber einen Termin am nächsten Montag von 10:00 Uhr bis 11:30 Uhr vereinbart. Da du als HR-Mitarbeiter möchtest, dass der Bewerber sich wohlfühlt und möglichst ein gewisses Maß an Diskretion gegeben ist, ergibt sich für dich folgende Priorisierung der Räume: R4 (Diskretion, hell), R3 (Diskretion, dunkel), R1 (keine Diskretion, hell), R2 (keine Diskretion, hell).

Aufgabe:

Reserviere den bestmöglichen Raum zu den oben genannten Bedingungen.

Weitere Informationen:

Titel für den Termin: „VG Java Entwickler“

Eric Schweizer steht für die Persona aus dem *Vertrieb*. Er möchte für die kommende Woche ein Projektanbahnungsgespräch organisieren. Die Uhrzeit ist durch den Kunden fest vorgegeben. Er benötigt dazu keinen besonderen Raum. Für seine Buchung wird die entsprechende Raumhierarchie aus Tabelle 3.3 angewendet.

Szenario Vertrieb

Beschreibung:

Du willst nächste Woche am Dienstag ein Projektanbahnungsgespräch organisieren. Zu dem Termin kommen neben dir noch ein Designer, der vorgesehene Projektmanager sowie drei Mitarbeiter des Kunden. Das Meeting soll um 9:30 Uhr beginnen und etwa 2-3 Stunden dauern.

Aufgabe:

Lasse dir die verfügbaren Räume anzeigen und reserviere anschließend den Raum R1 von 9:30 Uhr bis 13:00 Uhr.

Weitere Informationen:

Titel für den Termin: „Projektanbahnung VR Bank“

3 Konzeption

3.2.5 Workflow

Zu den beschriebenen Szenarien werden User Tests durchgeführt. Der damit verbundene Workflow und die Grundüberlegungen sollen im Folgenden anhand eines spezifischen Szenarios erläutert werden. Für diesen Fall wird das Szenario aus dem Bereich *Vertrieb* betrachtet.

Um diese Aufgabe zu erledigen und den Raum R1 von 9:30 Uhr bis 13:00 Uhr zu reservieren, hat der Nutzer verschiedene Möglichkeiten. Diese sind in Abbildung 3.11 in Form eines Entscheidungsbaums (engl. *decision tree*) dargestellt. Die einzelnen Zweige werden anschließend genauer erläutert.

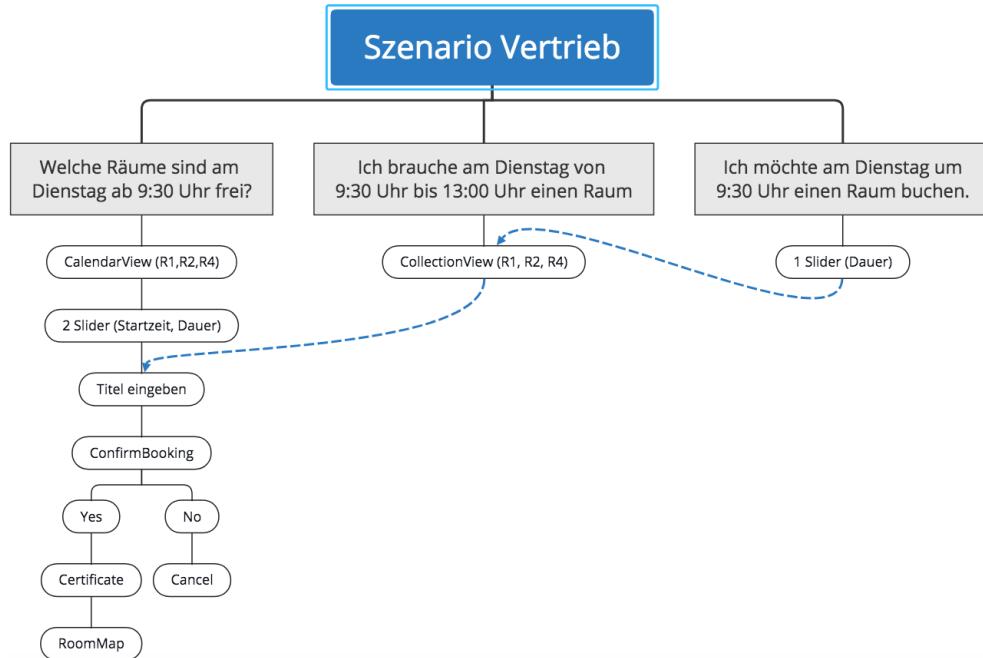


Abbildung 3.11: Entscheidungsbaum zum Szenario *Vertrieb*

Zunächst kann sich der Nutzer die ab 9:30 Uhr verfügbaren Räume anzeigen lassen (linker Zweig). Eine entsprechende Formulierung könnte so aussehen:

Welche Räume sind am Dienstag ab 9:30 Uhr frei?

Das System zeigt daraufhin eine *CalendarView* mit den verfügbaren Räumen. Der Nutzer wählt einen für seinen Zeitraum passenden Zeitslot aus und wird anschließend mit



3 Konzeption

Hilfe von zwei *Slidern* nach der Startzeit und Dauer seines Termins gefragt. Danach hat das System alle notwendigen Parameter und kann die Buchung nach Eingabe eines Titels abschließen.

Bei anderer Formulierung der Intention kann das System direkt erkennen, dass der Nutzer einen Raum von 9:30 Uhr bis 13:00 Uhr benötigt (mittlerer Zweig). Diese Formulierung könnte beispielsweise so lauten:

Ich brauche am Dienstag von 9:30 Uhr bis 13:00 Uhr einen Raum.

Das System erkennt direkt den Start- und Endzeitpunkt des Termins und kann die verfügbaren Räume mit Hilfe einer *CollectionView* anzeigen. Wählt der Nutzer einen Raum aus, hat das System alle notwendigen Parameter und kann mit der Eingabe des Titels fortfahren.

Ist die Formulierung ähnlich, beinhaltet allerdings nur den Startzeitpunkt von 9:30 Uhr für den Termin, kann mit einem kleinen Umweg genauso vorgegangen werden (rechter Zweig). Eine Möglichkeit der Formulierung wäre:

Ich möchte am Dienstag um 9:30 Uhr einen Raum buchen.

Das System fragt den Nutzer über einen *Slider* nach der Dauer des Termins. Hat der User diese eingegeben, kann auf gleiche Art und Weise vorgegangen werden. Der Raum wird über eine *CollectionView* ausgewählt und die Buchung anschließend abgeschlossen.

Der Abschluss einer Buchung trifft sich für alle möglichen Zweige an einem Punkt wieder. In diesem Szenario ist das die Stelle, an der die *Eingabe des Titels* erfolgt. Im Rahmen der Arbeit wird eine Festlegung zur Angabe von Titeln bei Terminen getroffen. Ab einer Dauer des Termins von 1,5 Stunden muss ein Termin einen Titel haben. Dauert der Termin weniger als 1,5 Stunden, kann der Nutzer optional und auf eigenen Wunsch einen Titel hinzufügen. Tut er dies nicht, wird ein automatisch generierter Titel mit dem Namen des Organisators eingetragen. Anschließend wird die Buchung noch über das UI-Element *ConfirmBooking* abgeschlossen.

3 Konzeption

Der Wizard of Oz hat die in 3.2.2 beschriebene Operator-Ansicht. Die Zweige des oben beschriebenen Entscheidungsbaums sind in Abbildung 3.12 als blaue Kästen wieder zu finden.

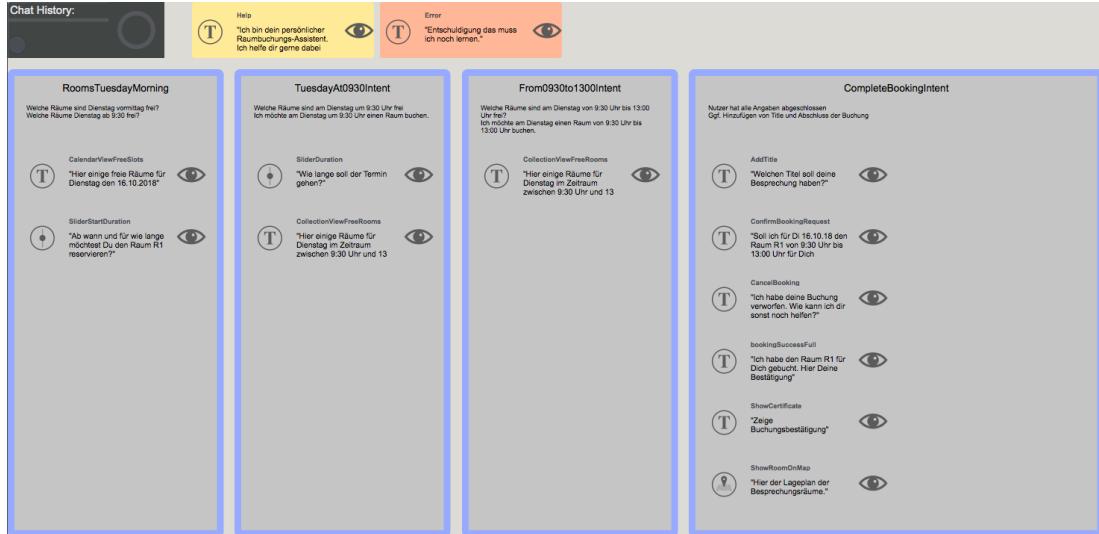


Abbildung 3.12: Benutzeroberfläche des Operators zum Szenario *Vertrieb*

Die linken drei Kästen können entsprechend von oben nach unten abgearbeitet werden. Danach befindet sich der Knotenpunkt zur Eingabe des Titels. Ab hier ist der Ablauf für alle Fälle gleich, wonach der rechte blaue Kasten von oben nach unten durchlaufen wird.

Der Test-User bekommt von all dem nicht viel mit. Er bearbeitet das Szenario und wählt eine für ihn passende Formulierung aus. Über die Benutzeroberfläche des Tester-Panels interagiert er mit dem System und erhält entsprechende Rückmeldungen in Form von Text oder UI-Elementen. Ein möglicher Durchlauf aus der Sicht des Probanden ist in Abbildung 3.13 dargestellt.



3 Konzeption

Ich bin dein persönlicher Raumbuchungs-Assistent. Ich helfe dir gerne dabei, einen Besprechungsraum für deine Termine zu finden und zu buchen.

Ich brauche am Dienstag um 9:30 Uhr einen Raum

Wie lange soll der Termin gehen?

bis 13 Uhr

Nachricht wurde erstellt: Zeitraum von 9 Uhr 30 bis 13 Uhr

Hier einige Räume für Dienstag im Zeitraum zwischen 9:30 Uhr und 13 Uhr. Wähle einen Raum, um mit dem Buchen fortzufahren.

R1 (available)

R2 (available)

R4 (available)

Nachricht wurde erstellt: Der Raum R1 wurde ausgewählt.

Welchen Titel soll deine Besprechung haben?

Projektanbahnung VR Bank

Soll ich für Di 16.10.18 den Raum R1 von 9:30 Uhr bis 13:00 Uhr für Dich buchen?

X ✓

Nachricht wurde erstellt: Raumbuchung für R1 bestätigt.

Ich habe Dienstag den Raum R1 für Dich gebucht.
Hier, Deine Bestätigung.

Buchungsbestätigung

R1

Projektanbahnung VR Bank

Di 16.10.2018

9 Uhr 30 – 13 Uhr

Abbildung 3.13: Benutzeroberfläche des Probanden zum Szenario *Vertrieb*

3 Konzeption

3.2.6 User Tests

Zur Evaluierung der konzeptionellen Überlegungen werden abschließend User Tests durchgeführt. Grundsätzlich sollen dabei Parallelen zu den ersten Tests im Rahmen des Usability Testessens (siehe 3.2.3) gezogen werden können. Daher wird der Prototyp auch in diesem Fall von sechs Probanden getestet. Aufgrund der intensiveren Be trachtungen im Kontext des Requirements Engineering in Form von Fragebögen, Interviews oder der Entwicklung von Personae fällt der Test in diesem Fall umfangreicher aus. Die Probanden bearbeiten jedes der in 3.2.4 definierten Szenarien als eigenständige Aufgabe. Auch hier wird wieder die *Thinking-Aloud-Methode* angewendet und der User Test mit einem Aufnahmegerät aufgezeichnet. Dazu wird die entsprechende Einverständniserklärung aus Anhang A.7 verwendet.

Nach der Begrüßung und Unterschrift der Einverständniserklärung bearbeitet der Proband die vier Szenarien in einer zuvor festgelegten Reihenfolge. Nachdem die Nutzer bereits nach wenigen Arbeitsschritten die Bedienung des Systems erlernen, werden die Szenarien unterschiedlich angeordnet, um so eine Verfälschung der Ergebnisse auszuschließen. Dazu werden die vier Szenarien für die User mit Hilfe eines Online-Tools randomisiert [Jum18]. In Tabelle 3.4 ist die unterschiedliche Reihenfolge der Szenarien und die jeweils dazugehörige, anonymisierte User-ID dargestellt.

User-ID	1	2	3	4
user1723	HR01	V01	D01	C01
user2626	D01	V01	C01	HR01
user5156	C01	V01	HR01	D01
user7027	V01	C01	HR01	D01
user9337	V01	D01	HR01	C01
user9716	C01	D01	V01	HR01

Tabelle 3.4: Randomisierung der Szenarien

Nach erfolgreicher Durchführung aller Szenarien füllt der Proband den bereits in 3.2.3 beschriebenen Feedback-Fragebogen aus. Eine ausführliche Auflistung der Fragebögen mit Notizen zu den jeweiligen Teilnehmern befindet sich im Anhang A.9.

3 Konzeption

Im Folgenden werden die Ergebnisse des Feedback-Fragebogens in Abbildung 3.14 dargestellt und bewertet. Die gelbe Referenzlinie zeigt die durchschnittliche Bewertung des Systems durch alle User.

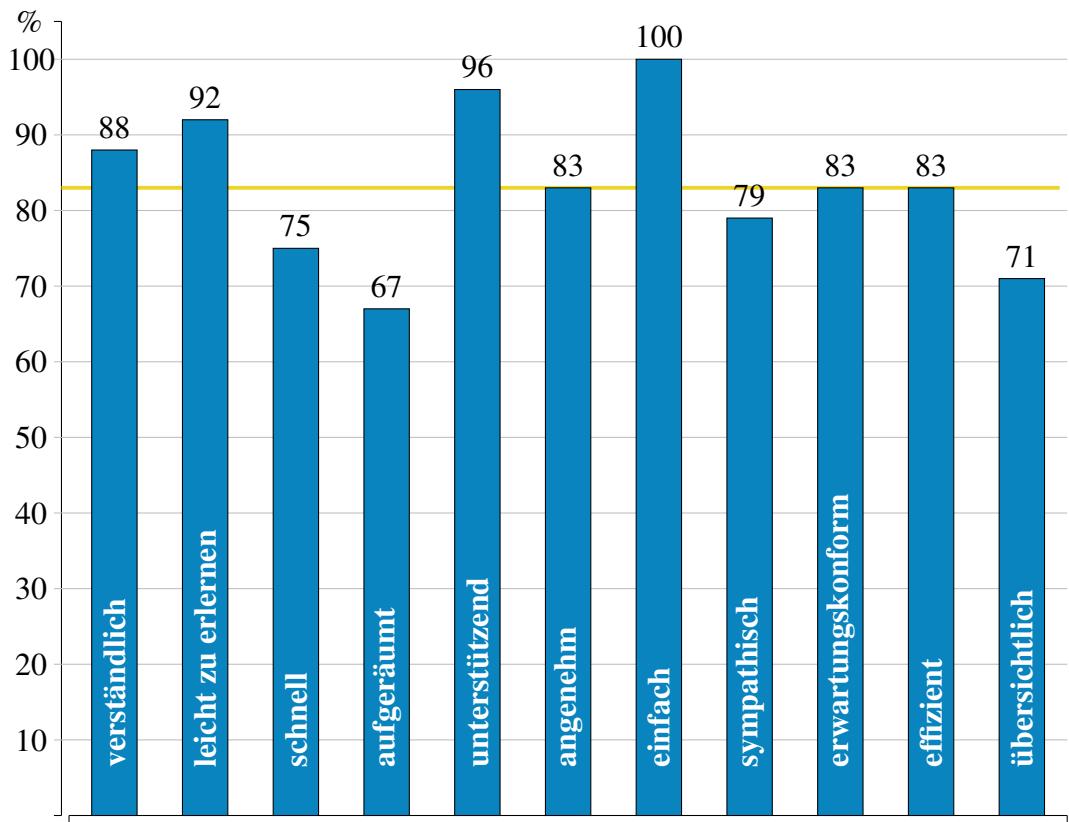


Abbildung 3.14: Ergebnisse des User Tests

Im Vergleich zu den User Tests beim Usability Testessen steigerte sich die Zustimmung zum System um etwa drei Prozent auf ca. 83 %. In dieser zweiten Testreihe werden auch die Diskrepanzen zwischen den unterschiedlichen Kriterien deutlicher. Dies könnte daran liegen, dass sich die Nutzer in diesem Fall deutlich länger, nämlich über vier statt nur einem Szenario, mit dem System auseinandersetzten.

Heraus stechen die Bewertungen für *verständlich* mit 88 %, *leicht zu erlernen* mit 92 %, *unterstützend* mit 96 % und *einfach* mit gar 100 %. All diese Kriterien beziehen sich auf die direkte Interaktion mit dem System und bestätigen den Eindruck, dass der Umgang mit dem konzipierten Chatbot intuitiv und einfach zu erlernen ist. Grund für diese auffallend positive Bewertung sind vermutlich das Verständnis der natürlichen



3 Konzeption

Sprache und die Unterstützung des Users durch die Möglichkeit der Eingabe über UI-Elemente.

Grundsätzlich empfand die Mehrzahl der Probanden den Chatbot als *schnell* (75 %). Hierbei ist anzumerken, dass sich einige der User bei der Bewertung nicht auf den Workflow beziehen, sondern auf die teilweise etwas längere Antwortzeit durch den Wizard of Oz. Diese sind natürlich durch die Art des Prototypings begründet und im Endprodukt nicht mehr in dieser Form vorhanden.

Wie bereits beim Usability Testessen, schneiden die Bewertungen *aufgeräumt* (67 %) und *übersichtlich* (71 %) am schlechtesten ab. Schon bei dieser ersten Testreihe kam die Vermutung auf, dass dies auf die prototypische Umgebung zurückzuführen sein kann. Zusätzlich kann nach dem zweiten Test gesagt werden, dass einigen Nutzern die UI-Elemente, wie beispielsweise die Buchungsbestätigung, zu groß und wuchtig sind und sie sich diese etwas kleiner und kompakter gewünscht hätten.

Mit einer Bewertung von 79 % empfanden die Probanden den Chatbot als durchaus *sympathisch*. Einigen Nutzern fehlte in diesem Zusammenhang noch etwas der Charakter des Chatbots. Hierbei könnte ein Charakterdesign in Form eines Namens oder Avatar hilfreich sein. Diese Hypothese wird im Kapitel Ausblick nochmals aufgegriffen und erläutert.

Eine solide Bewertung von 83 % steht bei den Attributen *angenehm*, *erwartungskonform* und *effizient*. Genau diese Kriterien spiegeln im Prinzip das Gesamtergebnis wieder und zeigen, dass die Probanden sich gerne mit dem System auseinandergesetzt haben und das Gefühl bekamen, vom Chatbot verstanden zu werden. Abschließend lässt sich zur Auswertung der Kriterien also sagen, dass der Workflow und die Kombination aus natürlicher Sprache und UI-Elementen prinzipiell verständlich und leicht zu erlernen ist. Die Nutzer bekommen das Gefühl, vom Chatbot verstanden und unterstützt zu werden. Optimieren lässt sich in diesem Zusammenhang sicherlich noch die Übersichtlichkeit und Performance des Systems, sowie einige Feinheiten an Workflow oder UI-Elementen.

Des Weiteren liefern die User Tests interessante Eindrücke in das mentale Modell der Nutzer. Ein mentales Modell beschreibt die Annahme von Nutzern über die Funktionsweise eines Systems [Rit15]. In diesem Zusammenhang beschreibt das Modell, welche Vorstellung ein individueller Nutzer zum Ablauf einer Raumbuchung hat. Dabei hat jeder Nutzer sein eigenes mentales Modell. Ziel ist es nun, im Verlauf der User Tests



3 Konzeption

herauszufinden welche Gemeinsamkeiten und Unterschiede in den mentalen Modellen der Nutzer herrschen. Am Ende kann sich dann auf eine Vorgehensweise geeinigt werden, die für einen Großteil der Nutzer zutrifft. Anzumerken ist hierbei, dass im Rahmen dieses User Tests die sechs Probanden nicht ausreichen, um eine fundierte Aussage zu treffen. Die Tests geben allerdings erste interessante Einblicke in das mentale Modell der Nutzer.

Im Szenario *Coaching* kommt es zu Unterschieden im mentalen Modell der Nutzer. Ein entscheidender Arbeitsschritt ist hierbei, wann die Angabe der Vor- und Nachbereitungszeit für den Termin gemacht werden soll. Einige User würden dies gerne vor der Auswahl der Kernzeit des Termins, andere lieber danach oder gar ganz am Ende tun. Weiterhin ergeben sich Unterschiede, ob bei der Angabe der Kernzeit bereits die Vor- und Nachbereitungszeit mit eingerechnet wird oder nicht. Auch hier hatten die User unterschiedliche Vorstellungen. Nach Erläuterung der Hintergründe und Gedankengänge zu diesem Workflow erachteten diesen jedoch alle Probanden als sinnvoll. Dabei wurde angemerkt, dass dies nach einmaligem Durchführen oder einer initialen Einführung problemlos angewendet werden kann.

4 Implementierung

In diesem Kapitel wird die Auswahl der für die Umsetzung benötigten Komponenten beschrieben und deren Kommunikation und Schnittstellen geklärt. Anschließend werden Anmerkungen zur Implementierung gegeben und der Workflow anhand von Beispielen verdeutlicht. Den Kern des Kapitels bilden die drei Hauptkomponenten bestehend aus *Android Applikation*, *Dialogflow Agent* und *Node.js Webservice*.

4.1 Auswahl der Komponenten

In diesem Kapitel werden die verwendeten Plattformen und Komponenten beschrieben. Als Vorlage dient dabei der in 2.1.3 vorgestellte Aufbau eines Chatbots. Einige der Komponenten sind dabei von Beginn an festgelegt, bei anderen bedarf es einer technischen Einordnung oder Entscheidungsfindung. Abbildung 4.1 zeigt die drei Hauptkomponenten und die Interaktion des Users.

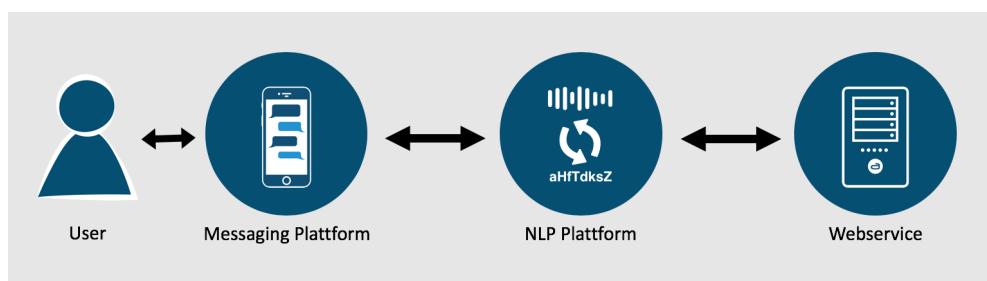


Abbildung 4.1: Übersicht der drei Hauptkomponenten



4 Implementierung

4.1.1 Frontend

Zu Beginn jeder Interaktion mit einem Chatbot steht ein Frontend oder, wie in 2.1.3 bezeichnet, eine Messaging Platform. Diese stellt gewissermaßen das Interface dar, über das der User seine Nachrichten eingeben und abschicken kann.

Wie bereits eingangs erwähnt, gibt es einige Nachrichtenplattformen, in denen Chatbots entwickelt und integriert werden können. Häufig verwendet werden hierbei der Facebook Messenger, WeChat oder Slack. Teilweise bieten diese Plattformen bereits die Möglichkeit zur Darstellung und Integration von UI-Elementen in Form von Buttons oder Auswahllisten. Die Vorgehensweise und Umsetzung ist dabei allerdings gleichermaßen einfach wie begrenzt. Während es relativ unkompliziert möglich ist, Antwortmöglichkeiten auf die Frage eines Chatbots via Popup-Buttons zu visualisieren, sind komplexere oder modifizierte UI-Elemente nur schwierig bis gar nicht umsetzbar.

Nachdem die adorsys GmbH & Co. KG den Google-Kalender nutzt gibt es von Beginn an den Wunsch bzw. die Idee, dass die Nutzer des Chatbots sich mit dem gewöhnlichen Firmenaccount und der Endung @adorsys.de anmelden und authentifizieren können. Die Integration des Firmenkontos in eine andere Nachrichtenplattform erscheint dabei weder besonders sinnvoll noch risikofrei.

All diese Kriterien führen dazu eine eigene, unabhängige Lösung zu nutzen. Wie bereits dem Titel der Arbeit zu entnehmen ist, handelt es sich dabei um eine Android Applikation.

4.1.2 NLP-Plattform

Ein wesentlicher Bestandteil des Chatbots ist die NLP-Plattform. Nach der Entscheidung aus 4.1.1 sollte dabei nach Möglichkeit ein Android *Software Development Kit (SDK)* zur Verfügung stehen. Ein SDK ist typischerweise eine Ansammlung von Tools und bereits geschriebenem Programmcode, die bei der Entwicklung von Software-Anwendungen unterstützen. Wichtig ist zudem, dass an die Plattform ein *Server* angebunden werden kann. Die Applikation soll zunächst auf *Deutsch*, später möglicherweise auch noch auf *Englisch* umgesetzt werden. Daher werden auch die beiden zu unterstützenden Sprachen betrachtet.

4 Implementierung

Nachdem der Chatbot eine Raumreservierung durchführen soll, muss die ausgewählte NLP-Plattform entsprechende Systementitäten unterstützen und erkennen. Wichtig sind in diesem Zusammenhang vor allem Entitäten für *Datum*, *Uhrzeit*, *Dauer* und *Anzahl*. Während sich *Datum*, *Uhrzeit* und *Dauer* auf die Start- und Endzeit des Termins beziehen, können durch die Entität *Anzahl* Informationen über die Anzahl der teilnehmenden Personen gewonnen werden. Zudem wird eine möglichst unbeschränkte und kostenfreie *Nutzung* in Bezug auf die Anzahl der Anfragen (engl. *queries*) betrachtet.

Auf Basis dieser Anforderungen werden im Folgenden einige der bekanntesten Plattformen für *Natural Language Processing* miteinander verglichen. Nach [Riy18] sind dies derzeit die Plattformen *Dialogflow* von Google, *Wit.ai* von Facebook, Amazon *Lex*, Microsoft *Luis* und IBM *Watson*.

	Dialogflow	Wit.ai	Lex	Luis	Watson
Android SDK	✓	✗	✓	✓	⚠
Serveranbindung	✓	✓	✓	✓	✓
Sprache: Deutsch	✓	✓	✗	✓	✗
Sprache: Englisch	✓	✓	✓	✓	✓
Entität: Datum	✓	✓	✓	✓	✓
Entität: Uhrzeit	✓	✓	✓	✓	✓
Entität: Dauer	✓	✓	✗	✓	✓
Entität: Anzahl	✓	✓	✓	✓	✓
Freie Nutzung	✓	✓	⚠	⚠	⚠

Tabelle 4.1: Vergleich verschiedener NLP-Plattformen

Die Informationen aus Tabelle 4.1 stützen sich auf die Inhalte in [Cab18], [Wit18] und [Ama18]. Erfüllte Anforderungen sind mit einem *grünen Haken* und nicht erfüllte mit einem *roten Kreuz* versehen. Die *gelben Warndreiecke* bedeuten, dass die Anforderungen nur teilweise oder unter bestimmten Bedingungen erfüllt werden. Im Fall von IBM Watson gibt es beispielsweise kein offizielles *Android SDK*, es unterstützt allerdings die Programmiersprache Java. Im Zusammenhang mit der Anforderungen *Freie Nutzung* bedeutet ein gelbes Dreieck, dass die Plattform nur einen bestimmten Zeitraum oder bis zu einer bestimmten Anzahl an Anfragen kostenfrei ist.



4 Implementierung

Aufgrund der Erkenntnisse aus Tabelle 4.1 fällt die Wahl auf die Plattform Dialogflow. Diese erfüllt alle Anforderungen und kann ohne zusätzliche Kosten genutzt werden.

4.1.3 Webservice

In Anlehnung an den Abschnitt 2.1.3 wird bei der Umsetzung eines intelligenten Chatbots, der mehr als nur *short answers* gibt, immer ein Backend für das Dialog Management genutzt. Der Server muss also die von der NLP-Plattform Dialogflow bereitgestellten JSON-Objekte verarbeiten. In diesem Fall kommen auf den Webservice weitere Aufgaben in Form von Authentifizierung, Kalenderfreigabe oder die Abfrage von freien Räumen zu.

Bei der Implementierung des Backends wird *Node.js* verwendet. Node.js ist eine serverseitige *JavaScript (JS)* Plattform und zudem kompatibel zu Dialogflow. Mit Hilfe von Node.js lässt sich ein Webserver realisieren. Zusätzlich lassen sich vorkompilierte Module oder einfache JavaScript-Dateien einbinden. Die Verwaltung der Module übernimmt der *Node Package Manager (npm)*. Dieser hilft bei der Installation und Aktualisierung der Module und berücksichtigt dabei auch deren Abhängigkeiten untereinander. Der npm macht es somit einfach, den Webserver mit beliebigen Modulen zu erweitern.

Der Server muss also Zugriff auf weitere Daten, Module und APIs haben, welche im Folgenden kurz beschrieben werden.

- Zur Verwaltung von Terminen kann die Schnittstelle ***Goolge Calendar API*** verwendet werden. Damit können unter anderem Termine für einen Nutzer erstellt, geändert oder gelesen werden.
- An die Datenbank bestehen in diesem Status noch keine speziellen Anforderungen. Es sollen gegebenenfalls Einträge der User mit Profilinformationen und Zugriffstoken für die Kalender-Authentifizierung abgespeichert werden. Dazu reicht ein einfaches *Not only Structured Query Language (NoSQL)*-Datenbanksystem wie ***MongoDB***.
- Zudem müssen die im Office verfügbaren Büroräume nach ihrer Verfügbarkeit abgefragt und an das Backend angebunden werden. Die Besprechungsräume sind dabei als Ressourcen in der Domäne der adorsys GmbH & Co. KG angelegt und können ebenfalls über die Google Calendar API angesprochen werden.

4 Implementierung

Eine Übersicht aller Komponenten ohne Kommunikation zeigt die Abbildung 4.2.

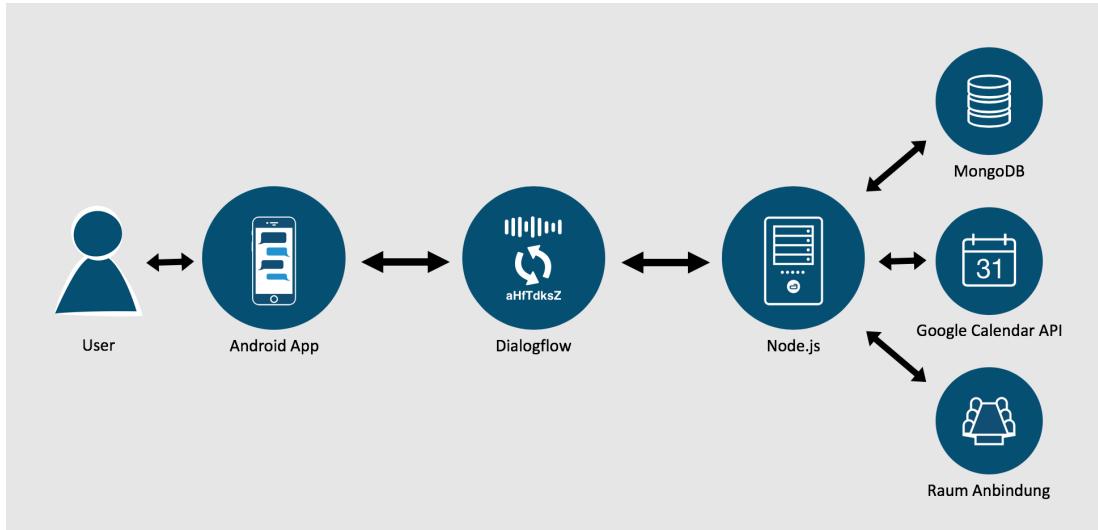


Abbildung 4.2: Übersicht aller Komponenten ohne Kommunikation

4.2 Kommunikation

Nachdem im vorangegangenen Kapitel 4.1 die verschiedenen Komponenten beschrieben wurden, stellt sich nun die Frage wie all diese unterschiedlichen Plattformen miteinander kommunizieren. Um dies zu beantworten, werden einige in der Arbeit erprobten Ansätze erläutert und bewertet.

4.2.1 Authentifizierung und Autorisierung

Wie bereits eingangs erwähnt, nutzt die adorsys GmbH & Co. KG den Google-Kalender. Für den Nutzer ist es demnach wünschenswert und komfortabel, wenn er sich in der App auf einer Art *Loginpage* anmelden kann. Genutzt werden können dazu die gewöhnliche E-Mail-Adresse mit der @adorsys.de-Endung, welche auf dem Google-Kalender basiert, und das entsprechende Passwort. Der Login innerhalb der App kann für beliebige Anwendungszwecke konfiguriert werden. So lassen sich dabei in der Basisvariante nach erfolgreicher Anmeldung beispielsweise Profilinformationen wie Vor- und Nachname oder der individuelle Identifier (ID) eines Nutzers (auch *Person-ID*)



4 Implementierung

gewinnen. Je nach Anwendungsfall kann zudem die E-Mail-Adresse oder auch ein *Android-ID-Token* erhalten werden. Zu dem Android-ID-Token können zusätzlich so genannte *Scopes* hinzugefügt werden, die beispielsweise Lese- oder Schreibzugriff auf den Kalender gewähren. [Goo18d]

Die erste Idee ist hierbei, bei Anmeldung in der App einen Android-ID-Token zu erhalten, der durch alle Komponenten durchgeschleift wird. Dieser würde also über die Textanbindung der App an Dialogflow und deren Verarbeitung im Backend mitgeliefert werden. Damit wäre in jedem Schritt klar, von wem die aktuelle Textnachricht kommt. Außerdem kann jederzeit auf Profilinformationen zugegriffen werden. Bei genauerer Betrachtung treten hierbei zwei wesentliche Probleme auf. Zum einen gibt es keine Möglichkeit, über das Android SDK weitere Daten in die Textanfrage an Dialogflow zu stecken. Das JSON-File wird anhand des eingegebenen Textes und weiterer Konfigurationen automatisch erzeugt und kann somit nicht modifiziert werden. Ein weiteres Problem stellt sich bei der Authentifizierung mittels Android-ID-Token im Backend dar. Aus Sicherheitsgründen ist es nicht möglich, mit einem Token aus einer Android Applikation im Backend auf Kalenderereignisse zuzugreifen [Goo18b].

Für den Zugriff auf Kalenderereignisse im Backend ist nämlich ein eigener, serverseitiger *access token* notwendig. Dieser kann nach dem Aufruf eines Authentifizierungslinks und der Generierung eines Authentifizierungscodes schließlich im Backend erzeugt werden. Mitgeliefert wird dabei noch ein dazugehöriger *refresh token*, der bei Ablauf zur Generierung eines neuen *access token* genutzt werden kann. Außerdem gibt es die Möglichkeit der Verifizierung des Android-ID-Tokens im Backend. Über entsprechende Bibliotheken und Methoden kann dabei der Android-ID-Token verfiziert und Basis-Profilinformationen wie Name, E-Mail-Adresse oder Person-ID herausgefunden werden.

Zunächst wird noch ein weiterer Ansatz verfolgt. Dabei muss überlegt werden, woran eine Nachricht von bzw. an Dialogflow identifiziert werden kann. Im JSON-File von Dialogflow steht neben den Informationen zum geschriebenen Text, dem Intent und verschiedenen Entitäten auch eine ID für die aktuelle Session. Diese *Session-ID* bleibt dabei im Laufe einer Konversation unverändert und kann somit als Identifier im Backend verwendet werden. Um im Backend zu wissen, zu wem diese Session-ID gehört, muss ihr ein weiterer Identifier zugeordnet werden. Denkbar wäre hierbei eine universelle *Device-ID* des Geräts. Es gibt dazu mehrere Möglichkeiten, wie beispielswei-



4 Implementierung

se eine sogenannte *Unique Telephony Number*, die *MAC-Adresse*, eine *Seriennummer* oder eine *Secure Android-ID* [Sau18].

Eine dieser Device-IDs würde dann zusammen mit der Session-ID in einer Datenbank liegen, wo sie als gemeinsamer Eintrag miteinander verknüpft werden. Verarbeitet der Server nun im Rahmen des Dialog Managements eine Nutzeranfrage, kann anhand der Session-ID die dazugehörige Device-ID aus der Datenbank geholt und somit die Anfrage einem konkreten Nutzer bzw. Gerät zugeordnet werden. Außerdem können zu diesen Einträgen passende Token für den Zugriff auf den Google-Kalender in der Datenbank hinterlegt werden.

Nachteilig an der Lösung mit der Verknüpfung von Session-ID und Device-ID ist, dass diese beiden Werte keinerlei Informationen über die Identität des Nutzers geben. Hier kann nun wieder der Google-Login innerhalb der App genutzt werden. Nachdem der User sich erfolgreich angemeldet hat, stehen seine persönlichen Informationen wie Name, E-Mail-Adresse und Person-ID zur Verfügung. Nachdem die Person-ID universell und für jedes Google-Konto eindeutig ist, kann diese problemlos für den Datenbank-eintrag und die damit verbundene Verknüpfung zur Session-ID und den Token aus dem Backend verwendet werden. Die Device-ID wird damit überflüssig. Im Backend kann die Session-ID dann jederzeit über die Abfrage einer Datenbank zugeordnet und entsprechende Daten für die Identität des Users oder seine Token für den Kalenderzugriff gelesen werden.

Nach ausführlicher Betrachtung stellt sich die Möglichkeit einer Zuordnung des Users durch seine aktuelle Session-ID und der Person-ID seines Google-Accounts als beste Möglichkeit heraus. Wird der Google-Login innerhalb einer App genutzt, um sich im Backend zu authentifizieren, sollten die Informationen in jedem Fall verschlüsselt übertragen werden [Goo18b]. Dazu kann der innerhalb der Android App generierte ID-Token an den Server gesendet und dort authentifiziert werden. Anschließend können daraus alle notwendigen Profilinformationen exportiert und in einer Datenbank zusammen mit der entsprechenden Session-ID gespeichert werden. Möchte der User auf seinen Kalender zugreifen, muss er dies innerhalb der App einmalig mittels einer im Backend generierten Uniform Resource Locator (URL) bestätigen. Anschließend werden sein *access token* und *refresh token* in der Datenbank hinterlegt. Aus den vorhergehenden Überlegungen ergibt sich für die Struktur zur Kommunikation zwischen den Komponenten das Schaubild aus Abbildung 4.3.

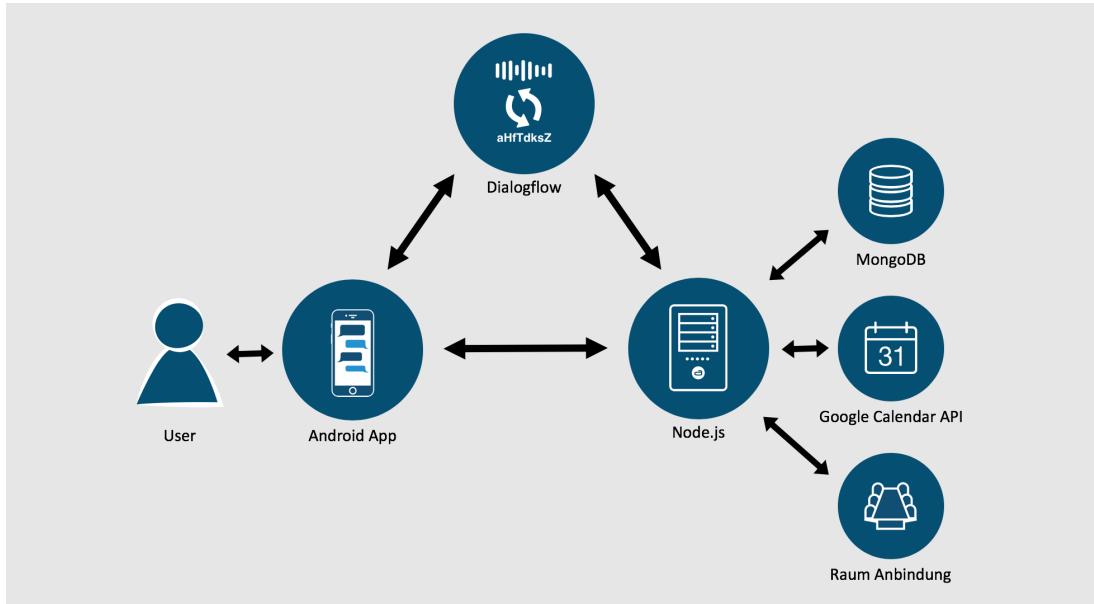


Abbildung 4.3: Übersicht aller Komponenten mit Kommunikation

4.2.2 Schnittstellen

Als Schnittstellen zwischen den Komponenten kommen verschiedene Techniken zum Einsatz. Die verwendeten Techniken sollen daher im Folgenden kurz erläutert werden.

1. Von einer Android Applikation aus kann das *Android SDK für api.ai*¹ genutzt werden, um Anfragen an Dialogflow zu senden [Ska18]. Damit können sowohl sprach- als auch textbasierte Anfragen an Dialogflow gesendet und die Antworten ausgewertet werden.
2. Zur Kommunikation zwischen der Android App und dem Node.js Webservice kann *Retrofit* verwendet werden. Retrofit ist ein Representational State Transfer (REST) Client für Android und Java und basiert auf der *OkHttp* Bibliothek. Damit ist es möglich, strukturierte Daten (z.B. JSON) zu einem REST-basierten Webservice zu schicken oder abzurufen. [VSW18]
3. Mit Hilfe von *Bespoken* kann eine Verbindung vom Dialogflow Agent auf den lokalen Rechner hergestellt werden. Über einen Proxy-Dienst gibt das Tool eine URL aus, welche im Agent als Webhook hinterlegt wird [Bes18]. Dies ist

¹api.ai heißt jetzt Dialogflow



4 Implementierung

allerdings nur so lange relevant, wie der Webservice auf einem lokalen Rechner läuft. Zum einem späteren Zeitpunkt kann hier auf eine REST-Schnittstelle zurückgegriffen werden.

4. Um vom Node.js Webserver aus mit einer MongoDB kommunizieren zu können, kann über npm ein *MongoDB Driver* installiert werden. Anschließend kann das entsprechende Modul in den Server eingebunden und über eine API darauf zugegriffen werden. [npm18]
5. Der Webservice kann Kalenderereignisse über die *Google Calendar API* abfragen oder erstellen. Auch diese wird über ein Modul in Node.js eingebunden. Die Google Calendar API basiert wiederum auf REST und Hypertext Transfer Protocol (HTTP). [Goo18e]
6. Die Raumanbindung im Backend geschieht ebenfalls über die *Google Calendar API*. Die Besprechungsräume sind dabei als Ressourcen in der Domäne der adorsys GmbH & Co. KG angelegt und können nach ihrer Verfügbarkeit abgefragt werden.

4.3 Android Applikation

Wie bereits dem Titel der Arbeit zu entnehmen ist und unter 4.1.1 nochmals ausführlich erläutert wird, handelt es sich bei dem Frontend bzw. der Messaging Platform des Systems um eine Android App. Über diese Applikation ist es dem Nutzer möglich, sich im Rahmen des Raumbuchungssystems einzuloggen und anschließend durch die Nutzung eines Chatbots Räume zu reservieren. Im Folgenden soll hierzu ein Überblick gegeben, allerdings nicht detaillierter auf den Quelltext eingegangen werden. Es wird dabei versucht, die Bedienung und den Workflow der App beispielhaft zu verdeutlichen. Dabei ist es leider nicht immer möglich, jeden Arbeitsschritt auf seine kleinste Ebene zu zerlegen. Vielmehr werden die prinzipiellen Abläufe anhand einiger Abbildungen erläutert.

4.3.1 Manifest

Im Android Manifest werden einige grundlegende Informationen deklariert. Nachdem die App über Dialogflow mit dem Internet verbunden sein muss, wird eine entsprechende *Permission* im Manifest verwaltet. Des Weiteren werden hier die beiden Aktivitäten *LoginActivity* und *ChatActivity* eingetragen, wobei die *LoginActivity* als *Main*-bzw. *Launcher*-Aktivität festgelegt und beim Start der App als erstes aufgerufen wird. Zudem kann hier das Icon der Applikation geändert werden. Das Icon wird vom Nutzer angewählt, wenn er die App auf seinem Smartphone starten möchte und hat eine dementsprechend wichtige Bedeutung. Das Icon² ist in Abbildung 4.4 dargestellt.



Abbildung 4.4: Icon der Android Applikation

²Das Icon wurde von Mona Kögel entworfen

4 Implementierung

4.3.2 Anmelden

Nach erstmaligem Start der Applikation findet sich der Nutzer auf einer *Loginpage* wieder. Der Login wurde gemäß der offiziellen Dokumentation zum *Google Sign-In für Android Applikationen* integriert [Goo18d]. Abbildung 4.5 zeigt die relevanten Anmeldebildschirme.

Auf der linken Seite ist der Startbildschirm zu sehen. Betätigt der Nutzer den Button *Sign in*, kann er aus seinen mit dem Android-Device verknüpften Google-Accounts wählen oder einen weiteren beliebigen Account verknüpfen. Nach erfolgreicher Anmeldung steht anschließend ein *Android-ID-Token* zur Verfügung, der wie bereits unter 4.2.1 beschrieben, im Backend verifiziert werden kann. Erwähnenswert ist in diesem Zusammenhang, dass dieser Login mit allen Google-Konten funktioniert, eine tatsächliche Raumbuchung im späteren Verlauf allerdings nur mit Accounts innerhalb der Domäne der adorsys GmbH & Co. KG möglich ist. Ist der Nutzer erfolgreich eingeloggt, ändert sich der *Login Status* auf dem Screen und der User wird automatisch zum Chatbot weitergeleitet.

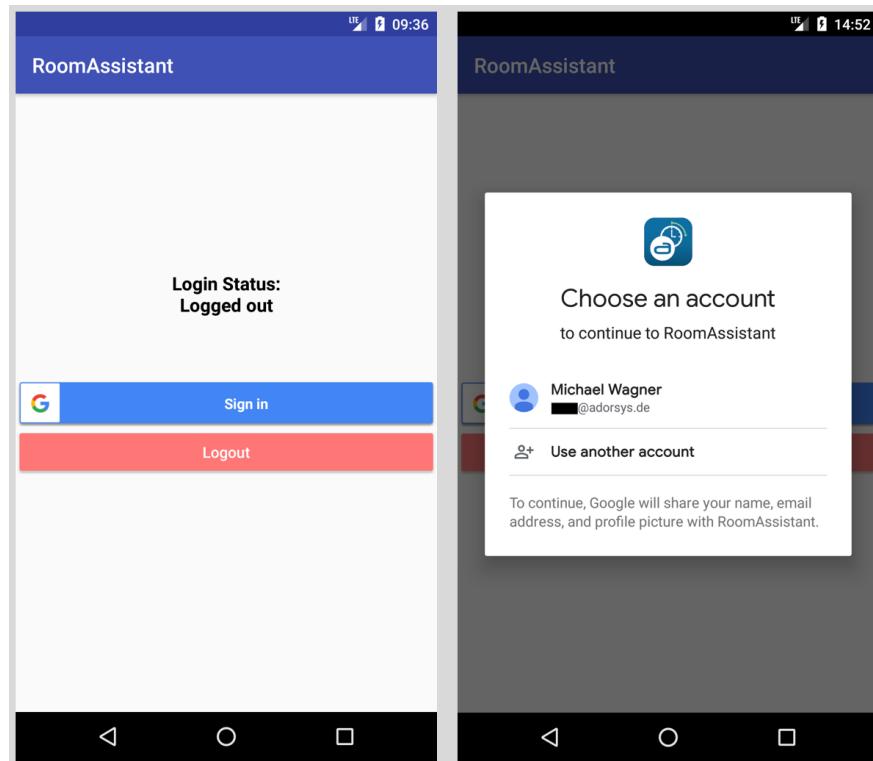


Abbildung 4.5: Benutzeroberfläche zum Anmelden in der Android Applikation

4.3.3 Kalenderzugriff

Beim Start der *ChatActivity* wird zunächst, vom Nutzer unbemerkt, eine Art *Handshake* durchgeführt. Die Applikation sendet einen Request mit der Query *Hallo* an Dialogflow. Über den Webhook kann diese Anfrage dann im Node.js Webservice ausgewertet werden. Anhand der mitgelieferten Daten können die dazugehörigen Nutzereinträge aus der Datenbank gelesen werden. In diesem Zuge wird auch direkt der Eintrag für die aktuelle Session-ID aktualisiert. Anschließend wird geprüft, ob bereits ein gültiger *access token* in der Datenbank vorhanden ist. Eine entsprechende JSON-Response wird zunächst zurück an Dialogflow und von dort aus zurück zur Android App gesendet. Abbildung 4.6 verdeutlicht diesen Prozess durch die Darstellung eines entsprechenden Sequenzdiagramms.

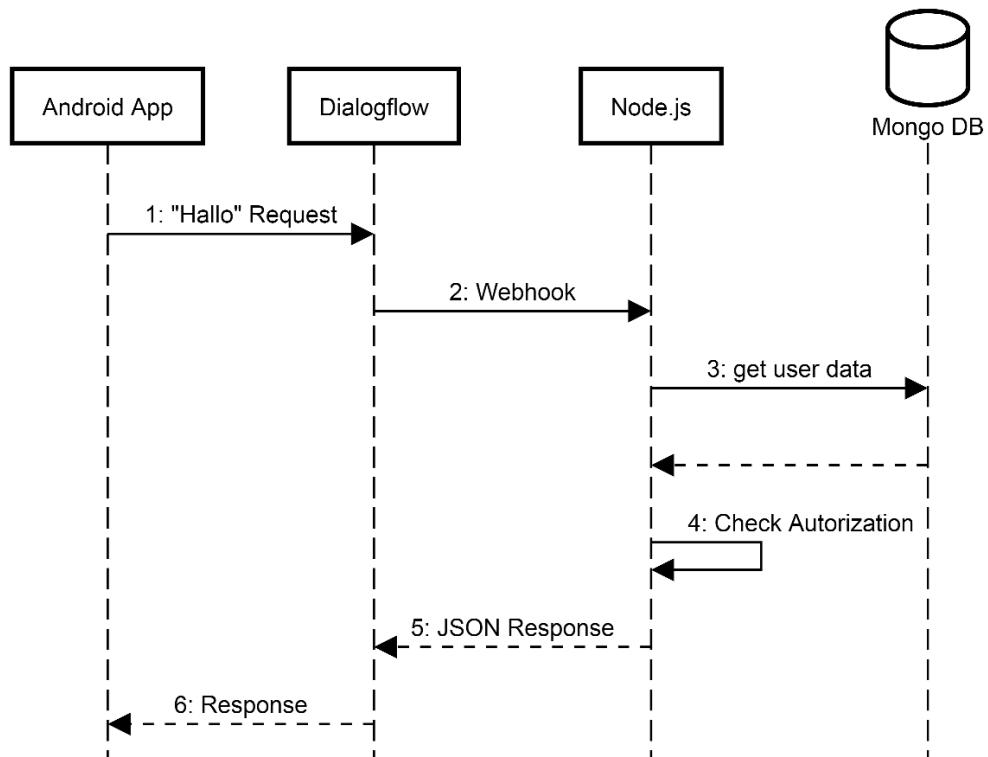


Abbildung 4.6: Sequenzdiagramm zur Autorisierung

Ist der User noch nicht autorisiert, wird er durch eine entsprechende Response dazu aufgefordert. Diese Abfolge ist in Abbildung 4.7 festgehalten. Innerhalb der *ChatActivity* öffnet sich in diesem Zusammenhang ein Browserfenster (linker Screen), in dem



4 Implementierung

sich der Nutzer anmelden und Zugriff auf seinen Kalender gewähren kann (mittlerer Screen). Hat er dies getan, erhält er einen *Authorization Code*, den er dem Chatbot anschließend zuschickt (rechter Screen). Im Backend wird der entsprechende Code geprüft, der *access token* bzw. *refresh token* generiert und in der Datenbank hinterlegt. Der Chatbot vermeldet dem Nutzer die Nachricht, dass die Autorisierung erfolgreich war. Beim nächsten Konversationsstart kennt der Chatbot den Nutzer und begrüßt ihn mit Phrasen wie „Hey, willkommen zurück“ oder „Schön dich zu sehen. Was kann ich für dich tun?“.

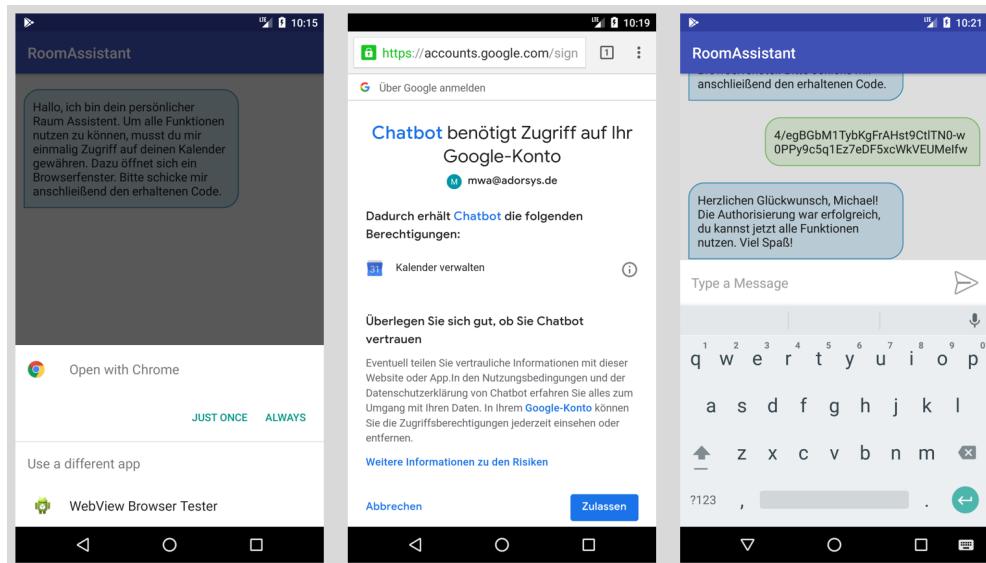


Abbildung 4.7: Benutzeroberfläche zum Autorisieren in der Android Applikation

4.3.4 Chat Interface

Die *ChatActivity* bzw. das *Chat Interface* stellt das Kernelement der Android Applikation dar. Es besteht im Prinzip aus einer *RecyclerView*, einem *EditText* Feld und einem *Button*. Diese sind durch mehrere, verschachtelte *RelativeLayouts* so angeordnet, dass sich Eingabefeld und Sendebutton immer am unteren Rand befinden. Bei Eingabe einer Nachricht durch den Nutzer klappt sich eine Tastatur auf und die *RecyclerView* passt sich entsprechend an. In die *RecyclerView* können verschiedene Elemente nach und nach eingeschoben werden, sodass für den Nutzer am Ende ein chronologischer und scrollbarer Nachrichtenverlauf sichtbar wird. Dieses Prinzip wurde zuvor bereits im rechten Screen der Abbildung 4.7 dargestellt.



4 Implementierung

Die UI-Erweiterungen sind dabei in einer eigenen XML-Datei definiert und werden codeseitig von einer Art *Adapter* verwaltet. Dieser Adapter beinhaltet eine Liste mit Instanzen einer eigenen Klasse *ChatMessage*. Eine Chatmessage hat üblicherweise je ein Attribut für den *Content* und den *Type*. Der Content beschreibt den anzuzeigenden Text, während der Type angibt um welche Art von Nachricht es sich handelt. Aufgrund des Umfangs und der begrenzten Zeit wurden in der Arbeit nur einige im Rahmen der Konzeption entstandenen UI-Erweiterungen umgesetzt. Diese werden in Tabelle 4.2 genauer erläutert.

Type	Art	Beschreibung	Orientierung
MSG_SENT	Text	Textnachricht User	rechts
MSG RECEIVED	Text	Textnachricht Chatbot	links
BUTTON_YES_NO	Buttons	Textnachricht mit Buttons ja/nein	links
SLIDER_DURATION	Slider	Textnachricht mit Slider für Dauer	links

Tabelle 4.2: Arten von Nachrichten im Chat Interface

Diese Unterscheidung macht es möglich, der RecyclerView je nach Ausgangspunkt unterschiedliche UI-Elemente hinzuzufügen. Ein beispielhafter Konversationsablauf mit allen implementierten UI-Elementen ist in Abbildung 4.8 dargestellt.

Sendet der User eine Textnachricht an den Chatbot, erscheint seine rechtsbündige Textnachricht mit entsprechender farblicher Kennzeichnung. Entsprechendes gilt analog für die Textnachrichten des Chatbots. Auch das Feature *Chatbot is thinking* kann somit implementiert werden. Dazu wird nach Abschicken der Nutzeranfrage an Dialogflow eine Textnachricht vom Typ *MSG RECEIVED* in die RecyclerView geschoben. Diese bleibt dort so lange, bis die Antwort vom Backend und über Dialogflow zurück kommt.

Dies dauert je nach Komplexität in der Regel wenige Sekunden. Der Nutzer erhält also direkt Feedback, dass seine Anfrage bearbeitet wird. Wurde die Response erhalten, kann das Element wieder entfernt werden und je nach Nachrichtenart die tatsächliche Rückmeldung des Systems angezeigt werden. Slider oder Buttons lassen sich nun wie gewohnt bedienen. Wird eine Auswahl über einen Button bestätigt, senden die *onClick* Ereignisse eine Nachricht in textueller Form an Dialogflow und der Ablauf beginnt wieder von vorne.

4 Implementierung

Da alle Eingaben des Nutzers über Dialogflow laufen und dort in Form einer Textquery ankommen müssen, lässt sich das System auch jederzeit über das Texteingabefeld bedienen. Die UI-Erweiterungen dienen lediglich als Vorschlag und Angebot durch das System und sollen gegebenenfalls die Bedienung des Chatbots erleichtern.

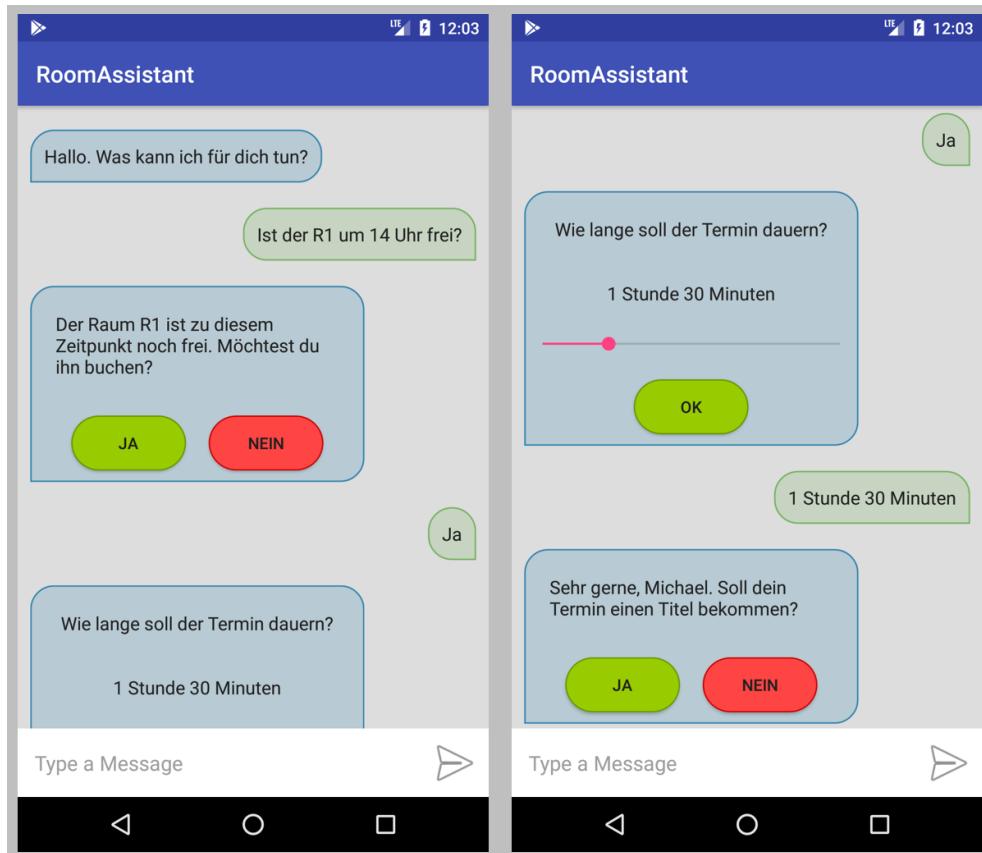


Abbildung 4.8: Chat Interface in der Android Applikation

Es ist nicht immer leicht, das Prinzip und den Umgang mit einer mobilen Applikation nur durch textuelle Beschreibung und Abbildungen darzustellen. Um die Bedienung und den Workflow der Android Applikation greifbarer zu machen, befinden sich auf der CD im Anhang einige *Videos* zum beispielhaften Ablauf einer *Autorisierung*, *Raumbuchung* oder -*abfrage*.

4.4 Dialogflow Agent

Nach dem ausführlichen Vergleich verschiedener NLP-Plattformen in 4.1.2 fiel die Wahl auf Dialogflow. Die prinzipielle Aufgabe der NLP-Plattform ist es, die Intents und Entitäten der Nutzeranfrage zu erkennen und richtig zuzuordnen. Außerdem ist Dialogflow für einen möglichst flüssigen Konversationsflow zuständig. All diese Aufgaben kombiniert in diesem Zusammenhang ein so genannter *Agent*.

Abbildung 4.9 zeigt den prinzipiellen Aufbau eines *Dialogflow Agents* (orange Komponenten). Im Grunde kann dieses Strukturbild analog zu der Abbildung 4.2 betrachtet werden.

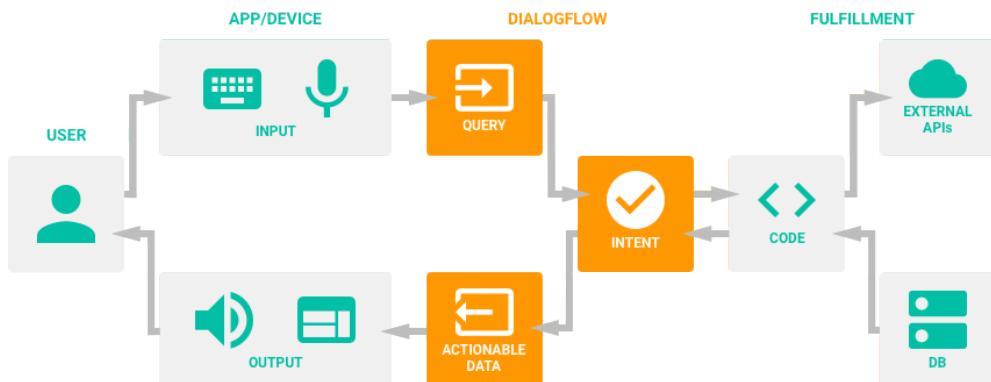


Abbildung 4.9: Prinzipieller Aufbau eines Dialogflow Agents [Goo18a]

Die linke Seite stellt den User inklusive seiner Ein- und Ausgabemöglichkeiten dar. Im Rahmen der Arbeit ist dies die zuvor beschriebene Android Applikation. Direkt danach ist der Dialogflow Agent anzusehen. Er erhält eine Query, also die Nutzeranfrage in Textform, ordnet diese einem Intent zu und zieht die relevanten Entitäten heraus. Das hier als *Fulfillment* bezeichnete stellt im Rahmen der Arbeit das Backend dar. Hier werden die Intents und Entitäten ausgewertet und externe APIs oder Datenbanken angebunden. Eine entsprechende Response gelangt dann auf dem Rückweg wiederum über Dialogflow an den User.

4 Implementierung

4.4.1 Intents

Aufgrund der begrenzten Zeit liegt der Fokus in der Implementierung auf der grundlegenden Funktion. Dazu soll es möglich sein, über das Conversational User Interface nach freien Räumen zu suchen und diese zu reservieren. Auch die Umsetzung der Intents wird daher auf diese beiden Fälle beschränkt. Zudem gibt es standardmäßig bei der Erstellung eines *Agents* je einen *Default Welcome Intent* zur Begrüßung und einen *Default Fallback Intent* als Absicherung, wenn der Chatbot den Nutzer nicht verstanden hat. Um die verschiedenen Intents und Abläufe etwas konkreter darstellen zu können, sind diese in Abbildung 4.10 zusammengefasst. Die Abbildung dient der Verdeutlichung des Workflows und stimmt daher nicht eins zu eins mit dem tatsächlichen Agent überein.

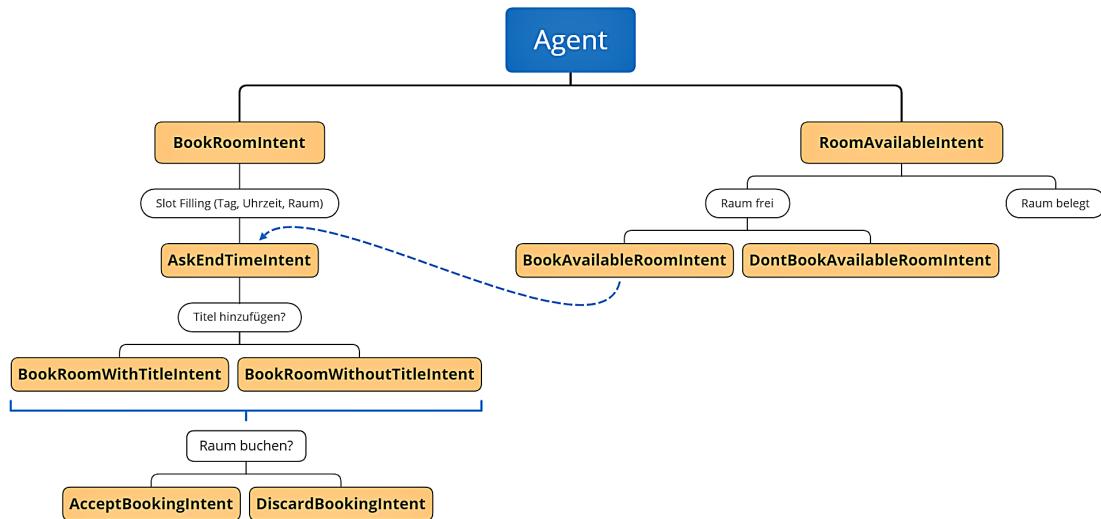


Abbildung 4.10: Struktur der Intents im Dialogflow Agent

Zur Realisierung der Raumbuchung gibt es die beiden Intents *BookRoomIntent* und *RoomAvailableIntent*. Erkennt der Chatbot, dass der Nutzer einen Raum buchen möchte, findet zunächst ein so genanntes *Slot Filling* statt. Dabei werden nach und nach alle für die Buchung notwendigen Parameter abgefragt. Ohne diese Parameter kann im Backend keine Aussage getroffen werden, ob ein oder mehrere Räume zu diesem Zeitpunkt noch frei sind. Ist eine Anfrage in sofern verifiziert, dass alle nötigen Parameter vorhanden sind und der Raum frei ist, springt der Agent in den *AskEndTimeIntent*. Hier wird der Nutzer dann gebeten, eine Endzeit oder Dauer des Termins anzugeben.



4 Implementierung

Anschließend wird noch eine Festlegung zum Titel des Termins getroffen. Wie bereits vorher festgelegt, müssen alle Termine ab 1,5 Stunden einen spezifischen Titel haben. Für kürzere Meetings ist die Angabe eines Titels optional. Möchte der User keinen Titel angeben, wird vom System ein Standardtitel mit der E-Mail-Adresse des Organisators eingetragen. Vor der endgültigen Buchung gibt es noch eine Art *Confirm Message*, mit der der Nutzer die Buchung final bestätigt.

Möchte der Nutzer dagegen zunächst nur nach einem oder mehreren freien Räumen suchen, wird dies über den *RoomAvailableIntent* realisiert. Wird explizit nach einem Raum gefragt oder ist zu dem gewählten Zeitpunkt nur ein Raum frei, besteht für den User die Möglichkeit, durch einen einfachen Buttonklick direkt in die Buchung des Raums zu springen. Ab hier wiederholt sich dann der zuvor beschriebene Ablauf nach dem *AskEndTimeIntent*.

4.4.2 Entitäten

Im Kontext der Raumbuchung kommen viele, verschiedene Entitäten vor. Wie in Tabelle 4.1 untersucht, werden die relevanten Parameter schon als so genannte *System Entities* von Dialogflow mitgeliefert [Dia18]. Eine Ausnahme stellt hierbei die Entität für den Raum dar. In Dialogflow ist es allerdings problemlos möglich, eigene Entitäten, so genannte *Developer Entities* zu definieren. In der Tabelle 4.3 sind alle im Rahmen des Agents verwendeten Entitäten aufgeführt.

Name	Art	Beispiel	Beschreibung
@sys.date	System	2018-10-17T12:00:00+02:00	Datum (ISO-8601)
@sys.time	System	2018-10-17T16:30:00+02:00	Uhrzeit (ISO-8601)
@sys.duration	System	{“amount”:1,“unit”:“stunde”}	Dauer als Objekt
@sys.any	System	Vorstellungsgespräch	Beliebiger Terminname
@RoomEntity	Developer	R1	Name des Raumes

Tabelle 4.3: Übersicht der Entitäten des Dialogflow Agents

Die Systementitäten *date* und *time* müssen getrennt betrachtet werden. Beide liefern einen Zeitstempel im *ISO-8601 Format*. Im Backend wird dann aus der Information



4 Implementierung

für Datum und Uhrzeit ein kombinierter Zeitstempel errechnet. Für die Dauer des Termins kann der Typ *duration* verwendet werden. Dieser liefert ein Objekt mit einem Integer für den Wert und einem String für die Einheit. Als Titel kann über den Typ *any* ein beliebiger Text eingegeben werden. Die selbst erstellte Entität für die verschiedenen Räume kann den Namen jedes Besprechungsraums beinhalten. Weiterhin gibt es hierbei den Wert *RX*, falls der Nutzer keinen bestimmten Raum buchen möchte.

4.5 Node.js Webservice

Wie bereits in 4.1.3 beschrieben, wird für die Implementierung des Backends Node.js verwendet. Auf das Backend kommen in diesem Fall mehrere Aufgaben zu. Es erhält über einen so genannten Webhook ein von Dialogflow generiertes JSON-Objekt. In diesem Objekt sind alle für die aktuelle Session relevanten Daten aufgelistet. Da nicht alle von Dialogflow mitgelieferten Daten relevant sind, wird zunächst über einen *Callback* ein Objekt generiert, in dem alle relevanten Daten zur Useranfrage abgelegt sind. Die relevanten Daten sind in Tabelle 4.4 aufgelistet.

Name	Art	Beschreibung
session_id	String	Identifier zur Zuordnung zu User aus Datenbank
intent_id	String	Eindeutiger Identifier für den Intent
intent_name	String	Name des Intents
parameters	Object	Sämtliche relevante Parameter, z.B. time, date, etc.

Tabelle 4.4: Relevante Daten aus Dialogflow JSON-Request

Weiterhin wird eine Datenbankanbindung benötigt, in der die Daten der User gespeichert und verwaltet werden. Die Zuordnung im Rahmen der Anfragen von Dialogflow geschieht durch die Session-ID. Jeder Datenbankeintrag besitzt eine universelle Person-ID, über die der Eintrag einem User zugeordnet werden kann. Zudem sind die E-Mail-Adresse und der Vorname des Nutzers hinterlegt. Über die abgespeicherten Token kann anschließend ein Zugriff auf die Kalenderereignisse stattfinden. Der Wert für das *expiry date* dient dem internen Handling zur Überprüfung der Gültigkeit und Erneuerung des *access token*. Eine Übersicht aller Parameter zu einem Nutzereintrag



4 Implementierung

in der Datenbank ist in Tabelle 4.5 dargestellt.

Name	Art	Beschreibung
session_id	String	Identifier zur Zuordnung des Users
person_id	String	Eindeutiger Identifier des Google-Accounts
email	String	E-Mail-Adresse des Users
given_name	String	Vorname des Users
access_token	String	Zugriffs Token des Users
refresh_token	String	Refresh Token des Users
expiry_date	Number	Ablaufdatum des Tokens

Tabelle 4.5: Parameter aus der Datenbank

Die Anbindung der *Google Calendar API* zum Abfragen und Erstellen von Terminen ist ebenso Aufgabe des Webservices. Die Google Calendar API kann dabei sowohl für den Eintrag in den Kalender des Users, als auch für die Abfrage der Besprechungsräume im Office der adorsys GmbH & Co. KG verwendet werden. Die Räume sind dabei als Ressourcen innerhalb der Domäne der Firma angelegt und über eine kryptische E-Mail-Adresse ansprechbar. Abfragen von freien Zeiträumen und Einträge in die Terminkalender können dann gemäß [Goo18e] und [Goo18c] getätigert werden.

Um die Einträge in der Datenbank auf dem aktuellen Stand zu halten, müssen die aktuelle Session-ID und der zum Nutzer gehörige Android-ID-Token über eine HTTP-Nachricht von der Android App an das Backend geschickt werden. Auf die gleiche Weise wird auch der Authentifizierungscode versendet, aus dem anschließend im Backend die Token für den Kalenderzugriff generiert werden. Für diese Funktionalitäten werden im Backend so genannte Endpunkte definiert. Um dies zu realisieren wird das Node.js-Framework *Express* verwendet. Damit kann relativ einfach und minimalistisch eine REST-Schnittstelle umgesetzt werden. Es wird dazu jeweils eine *POST-Methode* zum Verifizieren von Session-ID und Android-ID-Token, sowie der Auswertung des Authentifizierungscodes implementiert.

Im Backend findet mit den von Dialogflow gelieferten Daten sozusagen das Dialog Management statt. Welches UI-Element die Android App am Ende als Antwort für den User anzeigen soll, ist durch ein JSON-Objekt definiert. Dabei gibt es die Möglichkeit, Daten über einen so genannten *payload* durch Dialogflow durchzuschleifen.



4 Implementierung

Ein Beispiel der Darstellung des UI-Elements im Backend zeigt Listing 4.1.

```
1 "payload":  
2 {  
3     "android":  
4     {  
5         "type": "button",  
6         "text": "Soll dein Termin einen Titel haben?"  
7     }  
8 }
```

Listing 4.1: Darstellung des UI-Elements im Backend im JSON-Format

Zur Unterscheidung der unterschiedlichen UI-Elemente dient hierbei der *type*. In Listing 4.1 ist dies ein *button*, der im Frontend eine Phrase und die Auswahlbuttons *Ja/Nein* darstellt. Der *text* repräsentiert an dieser Stelle die Phrase, die der Nutzer als Antwort erhält und ist für alle UI-Elemente notwendig. Anstelle von *button* kann an dieser Stelle auch *textmessage* oder *slider* angegeben werden. Die Zuordnung ist analog zu der im Unterkapitel Chat Interface angegebenen Tabelle 4.2.

Auf diese Art und Weise können nun beliebig viele, weitere UI-Elemente definiert und umgesetzt werden. Dabei muss natürlich das entsprechende UI-Element auch in der Android App bekannt und implementiert sein. Als erstes Modell und zum Datenaustausch funktioniert dies auch problemlos. Jedoch erscheinen zukünftige Probleme nicht unvermeidbar. Über die gewählte Architektur und den Komponentenaufbau aus 4.1 müssen die entsprechenden Daten immer durch Dialogflow hindurch geschleust werden. Ändert sich hierbei der Aufbau des JSON-Objekts oder wird gar das Durchschleifen von Daten unterbunden, so muss die gewählte Vorgehensweise geändert werden oder ist im schlimmsten Fall gar nicht mehr realisierbar. Als Anregung sei in diesem Zusammenhang erwähnt, dass die Komponenten *Webservice* und *NLP-Platform* in ihrer Anordnung auch vertauscht werden könnten. Dies hätte den Vorteil, dass Dialogflow nur noch für das *Natural Language Understanding* zuständig wäre und keine Daten darüber mitgeschleift werden müssen. Diese Ansätze werden allerdings nicht weiter beleuchtet, sondern dienen lediglich als kritische Hinterfragung der eigenen Struktur.

Abschließend sollen alle für das System relevanten Komponenten und der damit verbundene Ablauf vom Senden einer Nachricht durch den User bis zur Rückmeldung

4 Implementierung

einer Nachricht in der App anhand eines Sequenzdiagramms dargestellt werden. Anzumerken ist hierbei, dass dieses Diagramm natürlich nicht für alle Anwendungsfälle gleich ist. Es soll vielmehr die Hintergründe anhand eines typischen und häufig vorkommenden Use Cases, wie beispielsweise der Abfrage eines freien Raumes, verdeutlichen. Das entsprechende Sequenzdiagramm ist in Abbildung 4.11 dargestellt.

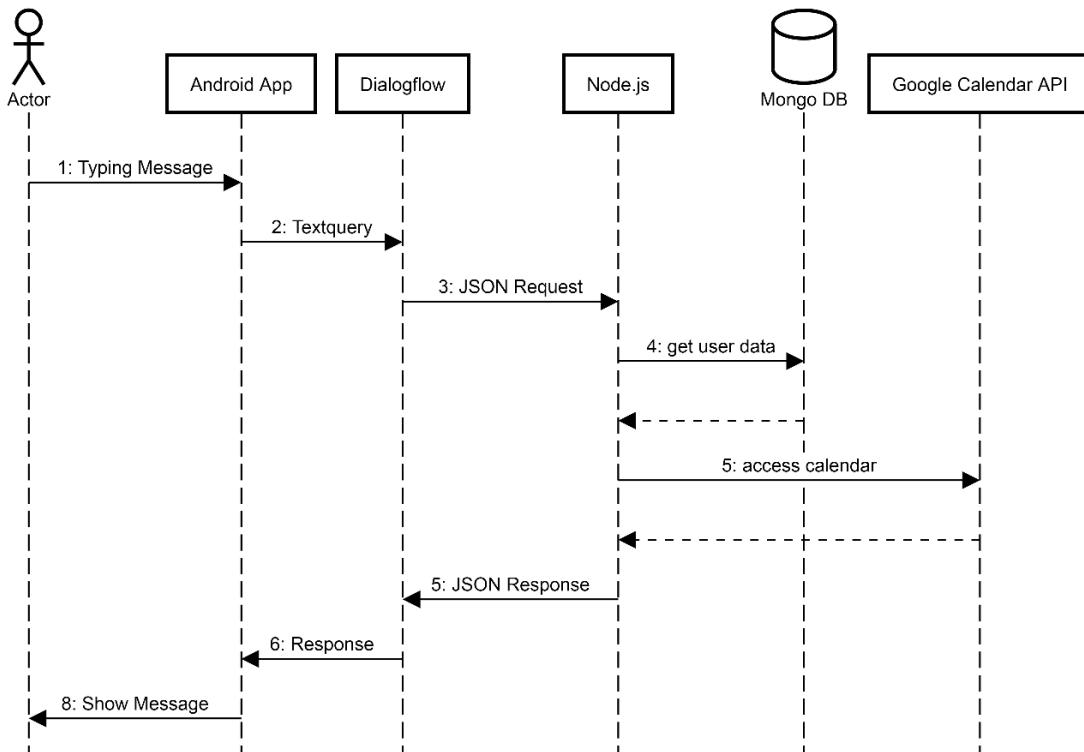


Abbildung 4.11: Sequenzdiagramm zur Interaktion der Systeme

In der Android Applikation tippt der User zunächst eine Nachricht ein oder bedient sich eines bereits vorhandenen UI-Elements und sendet so eine textuelle Nachricht an Dialogflow. Dort wird die Nachricht mit Hilfe von Natural Language Understanding interpretiert und gelangt über den Webhook an den Node.js Server. Anhand der hinterlegten Session-ID können die relevanten Informationen und Zugriffstoken aus einer MongoDB Datenbank gelesen werden. Sind alle Zugriffsrechte vorhanden, können über die Google Calendar API freie Räume abgefragt oder gebucht werden. Eine entsprechende Rückmeldung sendet der Webservice mit Hilfe einer JSON-Response an Dialogflow. Über die in der Android App eingebundene Bibliothek gelangt die Antwort letztendlich zurück zum User und wird ihm entsprechend als Text oder UI-Element angezeigt.



5 Schlussbetrachtung

Im folgenden Kapitel wird die gesamte Arbeit rekapituliert. Dabei wird anhand der Vorgehensweise nochmals auf die wichtigsten Erkenntnisse eingegangen und diese erläutert. Anhand eines kurzen Ausblicks werden Anregungen für weitere Untersuchungen gegeben.

5.1 Fazit

Gegenstand der Arbeit war die Konzeptionierung und Implementierung einer Android Applikation zur Raumbuchung via Conversational User Interface. Dazu wurden zunächst die Anforderungen an ein CUI im Allgemeinen und auch im Kontext der Raumbuchung mit Hilfe von Methoden aus dem Bereich Usability Engineering betrachtet.

In diesem Zusammenhang konnten mit Hilfe eines Fragebogens, Interviews und der Entwicklung von Personae ausführliche Einblicke in das Buchungsverhalten der Nutzer gewonnen werden. Hierbei wurde klar, dass die Ziele und Absichten der User sehr kontextspezifisch sind. So hat beispielsweise ein Mitarbeiter aus dem Bereich *Human Resources* ein ganz anderes Verhalten und Präferenzen bei der Buchung als jemand aus dem Bereich *Coaching*. Schnell wurde deutlich, dass diese Unterschiede erkannt und verinnerlicht werden müssen, bevor ein sinnvolles und zielgerichtetes Prototyping überhaupt stattfinden kann.

Anhand von Papier-Prototypen und deren anschließender Umsetzung anhand eines Klickdummmys konnten die ersten Ansätze auf ihre Plausibilität hin untersucht werden, bevor sie dann im Prototyping Tool umgesetzt wurden. Diese verschiedenen Ebenen des Prototypings sind eine gute Hilfe, um am Anfang uneingeschränkt und kreativ Ideen aufbringen zu können, diese anschließend möglichst schnell einem *plausibility check* zu unterziehen und abschließend zu entscheiden, ob diese weiterverfolgt oder verworfen werden.

5 Schlussbetrachtung

Anschließend stand der Workflow, also die Vorgehensweise vom Anfang bis zum Abschluss der Buchung, im Vordergrund. Hierfür wurden auf Grundlage der Ergebnisse aus dem Requirements Engineering und in Anlehnung an die entwickelten Personae verschiedene Szenarien entworfen. Für jedes Szenario wurden dabei die unterschiedlichen Möglichkeiten und Wege bis zum Ende der Buchung betrachtet und anhand von Entscheidungsbäumen dargestellt. Dabei wurde deutlich, dass es je nach Art des Szenarios große Unterschiede in Bezug auf die Variation der Nutzeranfrage gibt. Im Kapitel 3.2.5 wurde dabei zum einfacheren Verständnis das weniger umfangreiche Szenario aus dem Bereich *Vertrieb* gewählt. Für ein anderes Szenario steigt der Umfang der möglichen Eingangsformulierungen des Users allerdings schnell auf bis zu zwölf Zweige an. Dies verdeutlicht die enorme Variabilität des Nutzers bei der Verwendung eines Conversational User Interfaces.

Durch die Umsetzung der Ansätze mit Hilfe des Prototyping Tools werden schnell Probleme beim Workflow oder im mentalen Modell des Nutzers erkannt. Der Vorteil ist hierbei, dass diese Probleme noch im Status der Konzeption bzw. im Prototyping behoben werden können. In Abbildung 5.1 sind die Kosten der Fehlerbehebungen in der Softwareentwicklung über die verschiedenen Entwicklungsphasen dargestellt. Die Vorgehensweise nach dem Motto „*test early - fail early*“ spart hierbei also sowohl Zeit als auch Kosten.

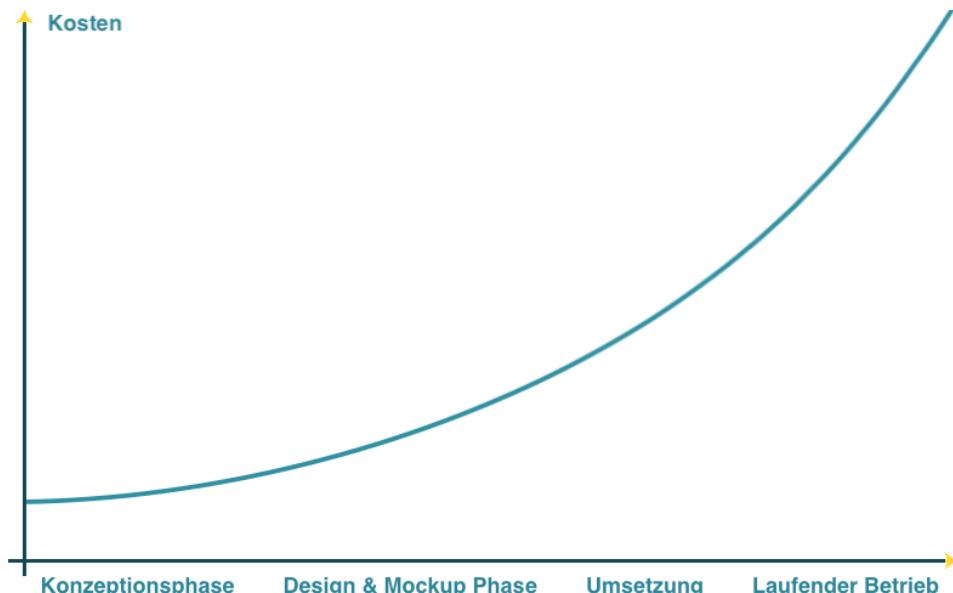


Abbildung 5.1: Kosten der Fehlerbehebung in der Softwareentwicklung [Cla16]



5 Schlussbetrachtung

Die prinzipielle Idee eines Conversational User Interface mit der Kombination aus Text und UI-Elementen im Kontext einer Raumbuchung ist für die Nutzer verständlich. Dies bestätigen die Ergebnisse und Rückmeldungen aus den User Tests. Zudem bieten diese gute Ansatzpunkte für eine weitere Bearbeitung der Thematik. Darauf soll explizit noch im Folgekapitel *Ausblick* eingegangen werden.

Wie auch dem Umfang der entsprechenden Kapitel *Konzeption* und *Implementierung* zu entnehmen ist, stellt die konzeptionelle Betrachtung einen großen Teil der Arbeit dar. Wie zuvor erläutert ist dies gleichermaßen wichtig wie notwendig, um ein fundiertes Verständnis über die Thematik zu erhalten. Nichtsdestotrotz ist auch die Implementierung ein fester Bestandteil der Arbeit und wurde entsprechend umgesetzt. Durch die Auswahl der Komponenten und deren Aufbau sowie die Klärung ihrer Kommunikation und Schnittstellen konnte hierbei ein solides Grundgerüst für eine weitere Implementierung aufgebaut werden. Einige der im Prototyping erprobten Ansätze in Bezug auf den Konversationsfluss und der UI-Elemente konnten hierbei bereits einfließen. Dem Nutzer ist es möglich, mit Hilfe der Android Applikation nach freien Räumen im Office der adorsys GmbH & Co. KG zu suchen und einen Besprechungsraum zu reservieren.

5.2 Ausblick

Aufgrund des begrenzten Zeitrahmens in Abschlussarbeiten können meist leider nicht alle Inhalte einer Thematik ausreichend beleuchtet oder umgesetzt werden. Daher dient dieses Unterkapitel dazu, einige Anmerkungen und Ausblicke für weitere Betrachtungen aufzugreifen und zu erläutern.

Wie bereits eingangs erwähnt, dient Sprache sowohl der schriftlichen als auch der mündlichen Konversation. In dieser Arbeit lag der Fokus auf der Entwicklung eines textuellen CUIs in Form eines Chatbots. Denkbar wäre hier also eine Erweiterung des Systems um eine sprachbasierte Eingabemöglichkeit. Dabei wird die gesprochene Sprache über ein Mikrofon aufgenommen und über *Automatic Speech Recognition (ASR)* in Textform konvertiert. Die textuelle Sprache kann anschließend weiter verarbeitet werden. Der Rückweg zur Ausgabe des Audiosignals wird dann mit Hilfe von *Text To Speech (TTS)* realisiert.



5 Schlussbetrachtung

Während der Durchführung der User Tests kam häufig die Frage nach einem Namen oder dem Aussehen des Chatbots auf. Viele Nutzer haben einen Charakter des Chatbots erwartet oder gar vermisst. In diesem Zusammenhang erscheint ein Charakterdesign für den Chatbot als sinnvoll. Dabei erhält der Chatbot einen Namen und einen Avatar, was ihn für den User transparenter und sympathischer macht.

Nahezu unverzichtbar ist für den Kontext einer Raumbuchung natürlich die Einladung von weiteren Personen. Dieser Teil wurde im Rahmen der Arbeit bewusst zunächst außen vor gelassen, muss aber früher oder später fester Bestandteil des Systems sein. Großen Mehrwert hat das System hier bei der Terminfindung. So können beispielweise alle Teilnehmer eines Meetings angegeben und der nächstmögliche freie Termin inklusive passendem Raum vorgeschlagen werden. Eine ergänzende Idee ist hierbei, im System gewisse Teams zu hinterlegen. So könnte der User beispielsweise nur angeben, er möchte alle Mitglieder aus dem *iOS-Team* zu einem Termin einladen und das System schlägt ihm mögliche Zeiträume vor, bei dem alle oder zumindest ein Großteil der Teilnehmer verfügbar ist.

Enormes Potenzial für den Anwendungsfall der Raumbuchung bietet auch die Technologie des Machine Learnings. Bis zum derzeitigen Stand der Arbeit befindet sich Machine Learning nur im Bereich der NLP-Plattform. Dies passiert jedoch unter der Oberfläche von Dialogflow und stellt somit keinen direkten Berührungspunkt dar. Eine eigene Umsetzung aus dem Bereich ML im Rahmen des Dialog Managements könnte die Intelligenz des Systems wesentlich verbessern. Ein mögliches Anwendungsgebiet ist hierbei das Erkennen und Erlernen von Verhaltensmustern eines Nutzers bei der Buchung. Damit wäre das System in der Lage, für einen Nutzer spezifische Vorhersagen über die Raumpräferenzen oder die Auswahl von Vor- und Nachbereitungszeit für einen Termin zu treffen.



Literaturverzeichnis

- [ado18] adorsys GmbH & Co. KG. Company, June 2018.
- [All15] Grant Allen. *Beginning Android*. Apress, New York, 2015. ISBN: 978-1-4302-4686-2.
- [Ama18] Amazon Lex. Built-in Slot Types - Amazon Lex, June 2018. Abgerufen am 29.06.2018. URL: <https://docs.aws.amazon.com/lex/latest/dg/howitworks-builtins-slots.html>.
- [And18a] Android Developers. App Manifest Overview, June 2018. Abgerufen am 26.06.2018. URL: <https://developer.android.com/guide/topics/manifest/manifest-intro>.
- [And18b] Android Developers. Layouts, June 2018. Abgerufen am 27.06.2018. URL: <https://developer.android.com/guide/topics/ui/declaring-layout>.
- [Bes18] Bespoken. With API.AI - bespoken.io, July 2018. Abgerufen am 10.07.2018. URL: http://docs.bespoken.io/en/latest/tutorials/tutorial_configuring_api_ai/.
- [Blu17] Steffen Bluemm. Designing a low-fidelity prototyping tool for Voice-UI, December 2017. Abgerufen am 27.07.2018. URL: <https://medium.com/@cui.lab/designing-a-low-fidelity-prototyping-tool-for-voice-ui-f33e2192b151>.
- [Cab18] Cabot Technology Solution. An Analysis of the Best NLP Tool to Build a Conversational Bot, February 2018. Abgerufen am 29.06.2018. URL: <https://hackernoon.com/an-analysis-of-the-best-nlp-tool-to-build-a-conversational-bot-18c4de0016c>.
- [Cha18] Nivedita Chandra. How To Conduct User Research for Chatbots ?, January 2018. Abgerufen am 08.08.2018. URL: <https://chatbotslife.com/how-to-conduct-user-research-for-chatbots-c7da114236d8>.



Literaturverzeichnis

- [Cla16] Claus Degendorfer. Wie macht man seine Kunden glücklich?, August 2016. Abgerufen am 15.11.2018. URL: <https://codefluegel.com/de/it-communication/>.
- [Dia18] Dialogflow. System Entities, October 2018. Abgerufen am 17.10.2018. URL: <https://dialogflow.com/docs/reference/system-entities>.
- [DL18] Li Deng and Yang Liu. *Deep learning in natural language processing*. Springer, Singapore, 2018. ISBN: 978-981-10-5209-5.
- [FFH02] Emmerich Fuchs, Karl Hermann Fuchs, and Christian H. Hauri. *Requirements-Engineering in IT effizient und verständlich: praxisrelevantes Wissen in 24 Schritten*. Know-how für das Management. Vieweg, Braunschweig Wiesbaden, 2002. ISBN: 978-3-322-89885-2 978-3-528-05756-5 978-3-322-89886-9.
- [Foi17] Oliver Foitzik. Wie Sie richtig mit Zeitfressern umgehen – Interview mit Thomas Kiefer, September 2017. Abgerufen am 30.05.2018. URL: <https://www.agitano.com/wie-sie-richtig-mit-zeitfressern-umgehen/100422>.
- [Fou17] Sébastien Fourault. The Ultimate Guide To Designing A Chatbot Tech Stack, August 2017. Abgerufen am 19.06.2018. URL: <https://chatbotsmagazine.com/the-ultimate-guide-to-designing-a-chatbot-tech-stack-333eceb431da>.
- [Gas18] Oliver Gast. *User experience im E-commerce: messung von Emotionen bei der Nutzung interaktiver Anwendungen*. Springer Gabler, Wiesbaden, Germany, 2018. ISBN: 978-3-658-22484-4. URL: <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1825473>.
- [GC15] Adam Gerber and Clifton Craig. *Learn Android Studio: build Android apps quickly and effectively*. Books for professionals by professionals. Apress; Springer, New York, NY, 2015. ISBN: 978-1-4302-6602-0 978-1-4302-6601-3.
- [Gei10] Thomas Geis. Neue ISO 9241-210 „Prozess zur Gestaltung gebrauchstauglicher Systeme“ ersetzt die ISO 13407 – ProContext Consulting GmbH, March 2010. Abgerufen am 08.08.2018. URL: <http://www.procontext.com/aktuelles/2010/03/iso-9241210-prozess-zur-entwicklung-gebrauchstauglicher-interaktiver-systeme-veroeffentlicht.html>.



- [Goo18a] Google Cloud. Agents | Dialogflow Enterprise Edition Documentation, October 2018. Abgerufen am 17.10.2018. URL: <https://cloud.google.com/dialogflow-enterprise/docs/concepts/agents>.
- [Goo18b] Google Developers. Authenticate with a backend server | Google Sign-In for Android, July 2018. Abgerufen am 05.07.2018. URL: <https://developers.google.com/identity/sign-in/android/backend-auth>.
- [Goo18c] Google Developers. Create Events | Calendar API | Google Developers, October 2018. Abgerufen am 17.10.2018. URL: <https://developers.google.com/calendar/create-events>.
- [Goo18d] Google Developers. Integrating Google Sign-In into Your Android App | Google Sign-In for Android, July 2018. Abgerufen am 05.07.2018. URL: <https://developers.google.com/identity/sign-in/android/sign-in>.
- [Goo18e] Google Developers. Node.js Quickstart | Calendar API, July 2018. Abgerufen am 10.07.2018. URL: <https://developers.google.com/calendar/quickstart/nodejs>.
- [Goo18f] Google Forms. Google Formulare: Kostenlos Umfragen erstellen und analysieren, August 2018. Abgerufen am 15.08.2018. URL: <https://www.google.de/intl/de/forms/about/>.
- [Hei18] Gerrit Heinemann. *Die Neuausrichtung des App- und Smartphone-Shopping: Mobile Commerce, Mobile Payment, LBS, Social Apps und Chatbots im Handel*. Springer Gabler, Wiesbaden, Germany, 2018. ISBN: 978-3-658-19135-1 978-3-658-19134-4.
- [Jac13] Wallace Jackson. *Learn Android App development: quickly build your first Android apps*. Books for professionals by professionals. Apress / Springer, New York, NY, 2013. ISBN: 978-1-4302-5747-9 978-1-4302-5746-2.
- [Jum18] Jumk.de. Generator für Zufallszahlen, November 2018. Abgerufen am 06.11.2018. URL: <https://rechneronline.de/zufallszahlen/>.
- [KD18] Rashid Khan and Anik Das. *Build Better Chatbots: A Complete Guide to Getting Started with Chatbots*. Apress, 2018. ISBN: 978-1-4842-3111-1.
- [MCG16] Michael McTear, Zoraida Callejas, and David Griol. *The Conversational Interface Talking to Smart Devices*. Springer International Publishing, Cham, 2016. ISBN: 978-3-319-32967-3.



Literaturverzeichnis

- [McG18] Chris McGrath. When a chatbot is better than an intranet - and when it's not, June 2018. Abgerufen am 26.06.2018. URL: <https://chatbotsmagazine.com/when-a-chatbot-is-better-than-an-intranet-and-when-its-not-2d5a7aff2423>.
- [Mit18] Martin Mitrevski. *Developing Conversational Interfaces for iOS: Add Responsive Voice Control to Your Apps*. Apress Imprint : Apress, Berkeley, CA, 2018. ISBN: 978-1-4842-3396-2.
- [Mol17] Rani Molla. How Apple's iPhone changed the world: 10 years in 10 charts, June 2017. Abgerufen am 16.06.2018. URL: <https://www.recode.net/2017/6/26/15821652/iphone-apple-10-year-anniversary-launch-mobile-stats-smart-phone-steve-jobs>.
- [Nah12] Harald Nahrstedt. *Algorithmen für Ingenieure: technische Realisierung mit Excel und VBA ; mit 65 Tabellen*. Studium. Springer Vieweg, Wiesbaden, 2., überarb. aufl edition, 2012. ISBN: 978-3-8348-1980-2 978-3-8348-1692-4.
- [Nis17] Nishith Pathak. *Artificial Intelligence for .NET: Speech, Language, and Search Building Smart Applications with Microsoft Cognitive Services APIs*. Apress, 2017. ISBN: 978-1-4842-2949-1.
- [npm18] npmjs. mongodb, July 2018. Abgerufen am 10.07.2018. URL: <https://www.npmjs.com/package/mongodb>.
- [Ovc12] Ekaterina Ovchinnikova. *Integration of world knowledge for natural language understanding*. Number 3 in Atlantis thinking machines. Atlantis Press, Amsterdam, 2012. ISBN: 978-94-91216-53-4 978-94-91216-52-7.
- [PR17] K. Siva Prasad Reddy. *Beginning Spring Boot 2: applications and microservices with the Spring framework*. Apress, New York, NY, 2017. ISBN: 978-1-4842-2931-6 978-1-4842-2930-9.
- [Pro18] Programmieren Lernen. Programmier Tutorial: Apps für Android entwickeln – Teil 15: Der Activity und Fragment Lifecycle in Android, June 2018. Abgerufen am 27.06.2018. URL: <http://www.programmierenlernenhq.de/tutorial-android-activity-und-fragment-lifecycle/>.
- [RF16] Michael Richter and Markus Flückiger. *Usability und UX kompakt: Produkte für Menschen*. IT kompakt. Springer Vieweg, Berlin Heidelberg, 2016. ISBN: 978-3-662-49828-6 978-3-662-49827-9.



Literaturverzeichnis

- [Rit15] Rita Strebe. Mental Models: Teil 1 - Die Welt im Kopf des Nutzers, November 2015. Abgerufen am 06.11.2018. URL: <https://www.usabilityblog.de/mental-models-teil-1-die-welt-im-kopf-des-nutzers/>.
- [Riy18] Jose Riya. A Study On The Best NLP Tool For Conversational Botmakers, June 2018. Abgerufen am 20.06.2018. URL: <https://www.cabotsolutions.com/2017/12/a-study-on-the-best-nlp-tool-for-conversational-botmakers>.
- [Roc17] Jerry Rocketbots.io. China, WeChat, and the Origins of Chatbots, March 2017. Abgerufen am 17.07.2018. URL: <https://chatbotsmagazine.com/china-wechat-and-the-origins-of-chatbots-89c481f15a44>.
- [Sau18] Saurel S. How to retrieve an Unique ID to identify Android devices ? – All for Android, Android for All, July 2018. Abgerufen am 05.07.2018. URL: <https://www.ssaurel.com/blog/how-to-retrieve-an-unique-id-to-identify-android-devices/>.
- [SGSB17] Remigiusz Smolinski, Moritz Gerdes, Martin Siejka, and Mariusz C. Bodek, editors. *Innovationen und Innovationsmanagement in der Finanzbranche*. Edition Bankmagazin. Springer Gabler, Wiesbaden, Germany, 2017. ISBN: 978-3-658-15648-0 978-3-658-15647-3.
- [Ska18] Danil Skachkov. dialogflow-android-client: Android SDK for Dialogflow, July 2018. Abgerufen am 10.07.2018. URL: <https://github.com/dialogflow/dialogflow-android-client>.
- [sta18] statistica.com. Marktanteile von Android und iOS am Absatz in Deutschland, November 2018. Abgerufen am 15.11.2018. URL: <https://de.statista.com/statistik/daten/studie/256790/umfrage/marktanteile-von-android-und-ios-am-smartphone-absatz-in-deutschland/>.
- [Sur18] SurveyMonkey. SurveyMonkey Angebote und Preise, August 2018. Abgerufen am 15.08.2018. URL: <https://www.surveymonkey.de/pricing/details/>.
- [Tec17] Maruti Techlabs. What is a Conversational UI and why does it matter?, October 2017. Abgerufen am 16.06.2018. URL: <https://chatbotsmagazine.com/what-is-a-conversational-ui-and-why-it-matters-de358507b9a2>.



- [Try16] Theo Tryfonas, editor. *Human aspects of information security, privacy, and trust: 4th International Conference, HAS 2016, held as part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016: proceedings.* Number 9750 in Lecture notes in computer science Information systems and applications, incl. Internet/Web, and HCI. Springer, [Cham] Heidelberg, 2016. ISBN: 978-3-319-39381-0 978-3-319-39380-3.
- [Typ18] Typeform. Plans & Pricing | Typeform, August 2018. Abgerufen am 15.08.2018. URL: <https://www.typeform.com/pricing/>.
- [VR18] Armando Vieira and Bernardete Ribeiro. *Introduction to Deep Learning Business Applications for Developers: From Conversational Bots in Customer Service to Medical Image Processing.* Apress, 2018. ISBN: 978-1-4842-3453-2.
- [VSW18] Lars Vogel, Simon Scholz, and David Weiser. Using Retrofit 2.x as REST client - Tutorial, July 2018. Abgerufen am 10.07.2018. URL: <http://www.vogella.com/tutorials/Retrofit/article.html>.
- [Web08] Wibke Weber, editor. *Kompendium Informationsdesign.* X.media.press. Springer, Berlin, 2008. ISBN: 978-3-540-69818-0 978-3-540-69817-3.
- [Wed16] Jan Weddehage. 10 Gründe, warum Conversational UIs die Zukunft gehört, June 2016. Abgerufen am 30.05.2018. URL: <https://entwickler.de/online/ux/conversational-user-interfaces-10-gruende-250714.html>.
- [Wit18] Wit. Wit — Built-in Entities, June 2018. Abgerufen am 29.06.2018. URL: <https://wit.ai/docs/built-in-entities/20180601>.



A Anhang

A.1 Ergebnisse des Fragebogens

Im Folgenden werden die Fragen und Antworten des Online Fragebogens ausführlich dargestellt. Die Anzahl der Teilnehmer kann dabei variieren, da manche Fragen als Pflichtfragen deklariert wurden und andere nicht. Ob die Frage eine Pflichtfrage ist, Single- oder Multiple-Choice und wie viele der Teilnehmer die Frage beantwortet haben, ist jeweils nachstehend kurz angegeben.

1. Wie häufig buchst du einen Raum?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

6 (13,0 %): täglich

14 (30,4 %): alle 2-3 Tage

10 (21,7 %): wöchentlich

9 (19,6 %): monatlich

7 (15,2 %): nie

2. Wie lange im Voraus buchst du in der Regel einen Raum?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

4 (8,7 %): unmittelbar

4 (8,7 %): am selben Tag

11 (23,9 %): ein Tag

17 (37,0 %): mehrere Tage

1 (2,2 %): Wochen/Monate



A Anhang

9 (19,6 %): Ich habe noch nie einen Raum gebucht

3. Wie lange dauern deine Besprechungen in der Regel?

Keine Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

13 (28,3 %): < 1 Std.

30 (65,2 %): 1 - 2 Std.

3 (6,5 %): 2 - 3 Std.

0 (0,0 %): 3 - 4 Std.

0 (0,0 %): > 4 Std.

4. Wie oft stimmt die gebuchte Dauer eines Termins mit der tatsächlichen überein?

Keine Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 38

9 (23,7 %): sehr oft

21 (55,3 %): oft

3 (7,9 %): ausgeglichen

5 (13,2 %): selten

0 (0,0 %): sehr selten

5. Wie oft wird ein Meeting beendet, weil die Buchung des Raums abgelaufen ist, obwohl noch Gesprächsbedarf bestehen würde?

Keine Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 38

1 (2,6 %): sehr oft

7 (18,4 %): oft

10 (26,3 %): ausgeglichen

15 (39,5 %): selten

5 (13,2 %): sehr selten

6. Hast du beim Buchen von Räumen einen festen oder flexiblen Zeitpunkt?

Keine Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 44



A Anhang

- 4 (9,1 %): fest
- 8 (18,2 %): eher fest
- 19 (43,2 %): ausgeglichen
- 7 (15,9 %): eher flexibel
- 6 (13,6 %): flexibel

7. Wie oft benutzt du den Slackbot Joan?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

- 1 (2,2 %): jede Buchung
- 2 (4,3 %): eher fest
- 2 (4,3 %): ausgeglichen
- 0 (0,0 %): Selten
- 41 (89,1 %): nie

8. Nutzt du das E-Paper Display an den Räumen, wenn du sofort einen Raum benötigst?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

- 16 (34,8 %): immer
- 15 (32,6 %): oft
- 5 (10,9 %): ausgeglichen
- 5 (10,9 %): selten
- 5 (10,9 %): nie

9. Nutzt du das E-Paper Display an den Räumen, wenn zu einem späteren Zeitpunkt den Raum verwenden möchtest

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

- 1 (2,2 %): immer
- 3 (6,5 %): oft
- 5 (10,9 %): ausgeglichen



A Anhang

9 (19,6 %): selten

28 (60,9 %): nie

10. Wie oft stimmt deiner Erfahrung nach der Belegungsstatus des Raumes mit den Angaben auf dem E-Paper Display überein?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

3 (6,5 %): immer

11 (23,9 %): oft

21 (45,7 %): ausgeglichen

11 (23,9 %): selten

0 (0,0 %): nie

11. Wie oft benötigst du bei deinen Meetings bestimmtes Equipment?

Gemeint ist damit Equipment, das nicht standardmäßig in den Räumen vorhanden ist, also etwas anderes als AppleTV, Fernseher, Whiteboard.

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

1 (2,2 %): immer

19 (41,3 %): oft

5 (10,9 %): ausgeglichen

15 (32,6 %): selten

6 (13,0 %): nie

12. Wie führst du deine Raumbuchungen aktuell am häufigsten durch?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

39 (84,8 %): Computer - Web Interface

7 (15,2 %): Ich habe noch nie einen Raum gebucht

0 (0,0 %): Smartphone - Web Interface

0 (0,0 %): Slackbot Joan



A Anhang

13. Wie würdest du deine Raumbuchung am liebsten durchführen?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

34 (73,9 %): Computer - Web Interface

5 (10,9 %): Smartphone - Applikation

3 (6,5 %): Computer - Anwendung

3 (6,5 %): Slackbot Joan

1 (2,2 %): Smartphone - Web Interface

14. Wie fändest du es, wenn du die Raumbuchung bestätigen müsstest?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

11 (23,9 %): gut

15 (32,6 %): eher gut

6 (13,0 %): neutral

6 (13,0 %): eher schlecht

8 (17,4 %): schlecht

Begründungen:

- ich nicht weiß ob immer daran denke [Auswahl: neutral]
- ich sicherlich nicht daran denke, denn nach der Raumbuchung (die nicht abgelehnt wird) ist die Sache bei mir abgehakt [Auswahl: schlecht]
- Dann Räume auch wieder frei werden, allerdings sollte das Zeitintervall schon auch ein paar Minuten -> 15/20 tolerieren [Auswahl: eher gut]
- der Raum bei nicht Benutzung als frei markiert ist somit weiter verwendbar [Auswahl: eher gut]
- weil dann sicher ist welches Meeting in dem Raum ist, falls man zb zu spät kommt, wissen will das genau das meeting um zb 13 Uhr zu ende ist,... [Auswahl: eher gut]
- dadurch der Raum nicht leerstehen würde [Auswahl: eher gut]



A Anhang

- Räume sind beschränkte Ressource und es kann immer etwas dazwischen kommen [Auswahl: gut]
- Räume die nicht bestätigt werden, auch wieder freigegeben werden können [Auswahl: gut]
- so vermieden wird, das unnötig Räume gebucht werden, die danach doch nicht gebraucht werden und leer stehen [Auswahl: eher gut]
- Bei nicht Bestätigung wird der Raum sofort wieder freigegeben [Auswahl: gut]
- nicht bestätigte Räume wieder freigegeben werden und somit wieder nutzbar für andere Kollegen sind [Auswahl: gut]
- sichergestellt wird, dass der Raum auch gebraucht wird und keine Leerbuchung stattfindet [Auswahl: eher gut]
- gebuchte Räume, die nicht verwendet werden Verschwendungen sind [Auswahl: eher gut]
- der Raum dann bei kurzfristigen Änderungen nicht unnötig vergeben bleibt [Auswahl: gut]
- genug zu tun und eh zuviel spam [Auswahl: eher schlecht]
- es mir vorkommt, als müsste ich die Buchung 2x durchführen [Auswahl: schlecht]
- Zeitraubend, überflüssig [Auswahl: eher schlecht]
- Außenstehende dann sofort wissen, ob das Meeting wirklich stattfindet oder nicht [Auswahl: eher gut]
- Sonst gebuchte Räume leer stehen [Auswahl: eher gut]
- Verstehe die Frage nicht das muss man jetzt ja auch? [Auswahl: gut]
- leer stehende Räume vermieden werden können [Auswahl: neutral]
- zusaetzlicher Aufwand + wozu? [Auswahl: schlecht]
- der Raum dann wieder freigegeben wird, wenn nicht genutzt [Auswahl: eher gut]
- zu aufwendig [Auswahl: schlecht]



A Anhang

- Overhead [Auswahl: eher schlecht]
- kurzer Aufwand für mehr Klarheit [Auswahl: gut]
- dann sicher festgestellt werden kann, dass der Raum auch tatsächlich benutzt wird und die Raum ansonsten für andere Personen wieder zur Verfügung steht [Auswahl: gut]
- man dann die genaue Anzahl an Teilnehmer weiß [Auswahl: gut]
- doppelte Arbeit [Auswahl: schlecht]
- ich finde, dass das niemand hilft. Das größte Problem bei uns, dass man ohne Buchung Räume besitzt [Auswahl: schlecht]
- weniger Probleme mit Fehlbuchungen und ungenutzten Räumen [Auswahl: gut]
- mehr darüber nachgedacht werden würde, ob der Raum tatsächlich benötigt wird oder ob sich das Meeting erledigt hat [Auswahl: eher gut]

15. Gibt es bevorzugte Räume bei deiner Buchung?

Keine Pflichtfrage, Multiple-Choice, Anzahl Teilnehmer: 25

14 (56,0 %): Raum R1

15 (60,0 %): Raum R2

7 (28,0 %): Raum R3

14 (56,0 %): Raum R4

3 (12,0 %): Raum R6

Begründungen:

- wir oft Konferenzen mit Mitarbeitern haben, die nicht im Office vorort sind [Auswahl: R3]
- R4 - nicht einfach einsehbar -> privater [Auswahl: R2;R4]
- nicht zu großer Raum [Auswahl: R1;R2;R4]
- Perfekte Anzahl von Sitzplätzen und Raumgröße [Auswahl: R1;R2]



A Anhang

- meine Meetings nicht so groß sind, und ich meistens keine ViCo brauche (Auswahl: R1;R2;R4)
- ausreichend Platz um sich mit Leuten darin zu bewegen und ausreichend Schreifläche [Auswahl: R4]
- Whiteboard, TV, Stühle [Auswahl: R1;R2]
- am angenehmsten , R6 nur für große Meetings, R1,R2 komische Tische [Auswahl: R3]
- Gute Atmosphäre, Klimanalge perfekt, Tische und Ausstattung am besten, Gutes Licht! (negativ beispiel R3 Videoraum...) [Auswahl: R1; R4]
- passen gut zu meinen Meetings [Auswahl: R1;R2;R4]
- Kleinere Räume [Auswahl: R1;R2;R3;R6]
- großer Kreis an Leuten, häufig auch Workshops [Auswahl: R4;R6]
- es mit Videokonferenzsystem aufgebaut ist [Auswahl: R3]
- es dort hell ist, es gibt Fenster und man wird von außen nicht so beobachtet [R1;R2;R4]

16. Welche Kriterien sind für dich bei der Raumbuchung relevant?

Pflichtfrage, Multiple-Choice, Anzahl Teilnehmer: 46

41 (89,1 %): Anzahl der Personen

25 (54,3 %): Art der Besprechung

19 (41,3 %): Ausstattung

18 (39,1 %): Dauer der Besprechung

[Anmerkung]

Für die folgenden Punkte stimmte je ein Teilnehmer (2,2 %)

- Diskretion (dass man in den Raum nicht reinschauen kann + den Bildschirm nicht lesen kann)
- Platz zum bewegen, Schreifläche und Sichtbarkeit der Schreifläche für alle (nichts gedrängtes)



A Anhang

- Teilnehmer außerhalb des Offices?

17. Worin besteht dein Haupttätigkeitsfeld?

Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 46

20 (43,5 %): Entwicklung

8 (17,4 %): Projektleitung

4 (8,7 %): HR

3 (6,5 %): DevOps

2 (4,3 %): Vertrieb

4 (8,7 %): Design

1 (2,2 %): Finanzen

1 (2,2 %): Workshops, Interviews, Coaching

1 (2,2 %): Geschäftsleitung

1 (2,2 %): PSD2

1 (2,2 %): Mehrere der o.g. Themen also nicht eingrenzbar

18. Bitte wähle zum Abschluss noch dein Alter aus.

Keine Pflichtfrage, Single-Choice, Anzahl Teilnehmer: 45

6 (13,3 %): unter 25

28 (62,2 %): 25 - 34

10 (22,2 %): 35 - 44

1 (2,2 %): über 45



A Anhang

A.2 Interviewleitfaden

Wir beschäftigen uns mit der Raumbuchung im Office der adorsys GmbH & Co. KG. Konkret geht es dabei um das Design, die Bedienung und die Funktionalität eines Chatbots, mit dessen Hilfe Mitarbeiter und Mitarbeiterinnen einen Besprechungsraum reservieren können. Hierfür führen wir ein Interview durch, in dem wir Ihnen einige Fragen zu dem Thema stellen werden. Ihre Informationen werden von uns anonymisiert, vertraulich behandelt und ausschließlich für Forschungszwecke verwendet.

Das Interview wird voraussichtlich ca. 45 Minuten dauern.

[Vorgehen]

1. Für welche Arten von Besprechungen buchen Sie Besprechungsräume? [Prompts: Bewerbungsgespräch, Mitarbeitergespräch, Team Meeting, etc.] → Ergänzend fragen: Sind diese Besprechungen wiederkehrende Ereignisse?
2. Wenn bei [1] wiederkehrendes Meeting → Wie oft wird das Meeting angepasst? [Prompts: Uhrzeit oder Tag ändert sich, Teilnehmer ändern sich, etc.]
3. Was sind Ihre Arbeitsschritte vor, während und nach dem Meeting? [Prompts: Unterschied zwischen Repeated und Single Meeting]
4. Wenn Sie zu einem bestimmten Zeitpunkt in der Zukunft einen freien Raum benötigen, wie gehen Sie vor? [Prompts: Web Interface, Slack, etc.]
5. Wenn Sie jetzt sofort einen freien Raum benötigen würden, wie gehen Sie vor? [Prompts: Web Interface, Slack, etc.]
6. Um die Buchung eines Raumes durchzuführen, was sind für Sie wichtige Informationen? [Prompts: Ausstattung, Anzahl der Plätze]

[Raumbuchung]

1. Verwenden Sie momentan das Web Interface von Google, um einen Raum zu buchen? [Prompts: Je nach Antwort schauen, ob alle weiteren Fragen zu diesem Thema gestellt werden]
2. Wozu verwenden Sie das Web Interface von Google? [Prompts: Raum buchen, Termin anlegen, Raumbelegung abfragen]



A Anhang

3. Was sind aus Ihrer Sicht die Stärken des Web Interfaces von Google? [Prompts: Die Auswertung des Fragebogens ergab, dass über 80% der Befragten die Raumbuchung derzeit mit dem Web Interface von Google durchführen.]
4. Was funktioniert aus Ihrer Sicht mit dem Web Interface von Google nicht so gut?
5. Was könnte man am Web Interface von Google verbessern bzw. welches Feature vermissen Sie?
6. Mit welchem Device verwenden Sie das Web Interface von Google? Und warum? [Prompts: Computer, Smartphone]
7. Verwenden Sie momentan den Slackbot Joan, um einen Raum zu buchen? [Prompts: Je nach Antwort schauen, ob alle weiteren Fragen zu diesem Thema gestellt werden.]
8. Wozu verwenden Sie den Slackbot Joan? [Prompts: Raum buchen, Termin anlegen, Raumbelegung abfragen]
9. Was sind aus Ihrer Sicht die Stärken des Slackbot Joan? [Prompts: Die Auswertung des Fragebogens ergab, dass etwa 8% der Befragten die Raumbuchung am liebsten mit dem Slackbot Joan durchführen würden.]
10. Was funktioniert aus Ihrer Sicht mit dem Slackbot Joan nicht so gut?
11. Was könnte man am Slackbot Joan verbessern bzw. welches Feature vermissen Sie?
12. Mit welchem Device verwenden Sie den Slackbot Joan? Und warum? [Prompts: Computer, Smartphone]
13. Halten Sie es für sinnvoll, wenn Sie die Raumbuchung bestätigen müssten? Und warum? Wenn JA → Frage 15; wenn NEIN → Frage 14
14. Wenn 13 → NEIN: Wäre es für Sie in Ordnung, wenn die Bestätigung der Raumbuchung intuitiv funktionieren würde? [Prompts: Ein häufiger Kritikpunkt ist dabei der Mehraufwand für den Nutzer. Die Auswertung des Fragebogens ergab, dass ca. 50% der Befragten eine Bestätigung sinnvoll finden würden, um beispielsweise Leerbuchungen zu vermeiden]
15. Wenn 13 → JA: Kennen Sie die manuelle Bestätigung der Raumbuchung aus dem alten Office? Wenn JA → Frage 16; Wenn NEIN → [E-Paper Display]



A Anhang

16. Wenn 15 → JA: Hat die manuelle Bestätigung der Raumbuchung im alten Office für Sie funktioniert?

[E-Paper Display]

1. Wird das E-Paper Display an den Räumen von Ihnen genutzt, um einen Raum für sofort zu suchen? Und warum? [Prompts: Die Auswertung des Fragebogens ergab, dass etwa 2/3 der Befragten das E-Paper Display nutzen, um einen Raum für sofort zu suchen.]
2. Wird das E-Paper Display an den Räumen von Ihnen genutzt, um einen Raum zu einem späteren Zeitpunkt zu buchen? Und warum? [Prompts: Die Auswertung des Fragebogens ergab, dass etwa 80% der Befragten das E-Paper Display nicht nutzen, wenn sie einen Raum zu einem späteren Zeitpunkt verwenden möchten.]
3. Wie oft stimmt Ihrer Erfahrung nach der Belegungsstatus des Raumes mit den Angaben auf dem E-Paper Display überein? [Prompts: Die Auswertung des Fragebogens ergab, dass nur etwa 1/3 der Befragten das Gefühl haben, dass die Angaben auf dem E-Paper Displays in der Regel stimmen.]
4. Welche Informationen können Sie aus der E-Paper Display herauslesen? [Prompts: Aktueller Status des Raums, Nächster Termin]
5. Was wären sinnvolle Informationen, welche das E-Paper anzeigen könnte?

[Zeitmanagement]

1. Wie oft wird aus Ihrer Sicht die Länge eines Termins eingehalten?
2. Was tun Sie, wenn ein Termin länger dauert als gedacht und der Nachfolgetermin bereits auf den Raum wartet?
3. Wie bekommen Sie Informationen über die Raumbelegung nach Ihrem eigenen Termin?
4. Wie häufig muss ein Termin aufgrund von Zeitüberschreitung umkoordiniert werden? [Prompts: Termin vertagt, verschoben, unterbrochen]
5. Wie häufig müssen Sie trotz eines gebuchten Raumes auf einen anderen ausweichen, weil dieser dann doch belegt ist?



A Anhang

[Ergänzende Fragen]

1. Wie häufig sind Sie in einem Termin mit externen Leuten?
2. Wie häufig kommen Sie in einen Besprechungsraum, der nicht in akzeptablen Zustand ist? [Prompts: Ausstattung fehlt, Tische oder Stühle verschoben, Whiteboards nicht gesäubert, etc.]
3. Wenn bei [2] Antwort: häufig → Wie fänden Sie eine Benachrichtigung (z.B. 10 Minuten vor dem Termin), der Sie darauf hinweist den Zustand des Raums zu überprüfen?
4. Wie häufig benötigen Sie zusätzliche Ausstattung in Ihren Meetings, also mehr als Whiteboard, AppleTV und Fernseher? [Prompts: Die Auswertung des Fragebogens ergab, dass fast die Hälfte der Befragten häufig zusätzliche Ausstattung benötigen]
5. Wäre es hilfreich, wenn das System Fragen nach zusätzlicher Ausstattung beantworten könnte? [Prompts: Moderatorenkoffer, Flipchart, etc.]
6. Gibt es Räume, die Sie bevorzugt buchen? Und warum?
7. Gibt es Räume, die für Ihre Besprechungen unbrauchbar sind? Für alle Arten Ihrer Besprechungen und warum?
8. Gibt es beim Vorgang „Buchen und Nutzen von Besprechungsräumen“ Aspekte, die den Prozess schwierig gestalten? [Prompts: Unterschied zwischen Repeated und Single Meeting]
9. In welchen Punkten könnte man aus Ihrer Sicht das Buchen und das Nutzen von Besprechungsräumen besser gestalten? [Prompts: Unterschied zwischen Repeated und Single Meeting]
10. Haben Sie sonst noch offene Punkte oder Anmerkungen, die wir vielleicht übersehen haben?



A.3 Einverständniserklärung zum Interview

- Projekt: Masterarbeit Conversational Room Booking
- Unternehmen: adorsys GmbH & Co. KG
- Projektleitung: Steffen Blümm, Technical Lead iOS / CUI
- Interviewerin/Interviewer: Michael Wagner

Ich erkläre mich dazu bereit, im Rahmen des oben beschriebenen Projekts an einen Interview teilzunehmen. Ich wurde über die Ziele des Projekts informiert. Ich kann das Interview jederzeit abbrechen, weitere Interviews ablehnen und meine Einwilligung in eine Aufzeichnung und Niederschrift des Interviews jederzeit zurückziehen, ohne dass mir dadurch irgendwelche Nachteile entstehen.

Ich bin damit einverstanden, dass das Interview mit einem Aufnahmegerät aufgezeichnet und sodann von den Projektmitgliedern ausgewertet wird. Für die Auswertung des Interviews werden alle Angaben zu meiner Person aus dem Text entfernt und/oder anonymisiert. Mir wurde außerdem versichert, dass das Interview in Veröffentlichungen nur in Ausschnitten zitiert wird, um sicherzustellen, dass ich auch durch die Reihenfolge von im Interview erwähnten Ereignissen nicht für Dritte erkennbar sein werde.

Datum, Unterschrift des Interviewten



A.4 Ergebnisse der Interviews

A.4.1 Interviewter I1

[Vorgehen]

1. Arten von Besprechungen [00:45]

- Projekt Meetings [Repeated]
- Regelmäßige Termine mit Fachbereich [Repeated]
- Spontane Meetings [Single]

2. Anpassung des Meetings [01:45]

- Selten (zwei Mal im Jahr)

3. Arbeitsschritte vor, während und nach dem Meeting [02:08]

[Repeated]

- Vorher nichts zu tun → Raum ist schon gebucht
- Während und danach nichts zu tun

[Single]

- Vorher Raum organisieren mit Google Kalender → Suchen von freien Besprechungsräumen

4. Raum zu bestimmtem Zeitpunkt in der Zukunft buchen [03:30]

- Schauen im Google Kalender, ob Raum zu bestimmten Zeitpunkt frei ist
- Abhängig wie viele Leute teilnehmen (großer/kleiner Raum)
- Beachten, ob Videokonferenz nötig

5. Raum jetzt sofort buchen [04:09]

- [1] Physikalisches nachschauen, ob jemand im Raum
- [2] Wenn frei → E-Paper Display Joan anschauen, ob für jetzt gerade ein Termin eingetragen ist



A Anhang

- [3] Wenn frei → Drücken „jetzt den Raum buchen“ direkt am E-Paper Display

6. Wichtige Informationen zur Buchung [05:19]

- Anzahl der Personen
- Größe des Raums
- Ausstattung (Fernseher, AppleTV)
- Stühle (Hochstühle für längeres Meeting besser)

[Raumbuchung]

1. Nutzung des Web Interfaces von Google [06:32]

- Ja

2. Wozu Web Interface von Google [06:44]

- Schauen, ob Raum frei ist
- Raum reservieren

3. Stärken des Web Interfaces von Google [07:03]

- Alle Räume nebeneinander → Schnellerer Überblick
- Sinnvoll beim Buchen von längeren Terminen, man sieht schneller die Lücken
- Vorgehen: Alle Räume anhaken → nebeneinander darstellen → Schauen für welchen Timeslot frei
- Auch für wiederkehrende Termine gut → Bei wöchentlichem Meeting die nächsten Wochen schauen, ob Raum immer frei ist [Anmerkung: Terminserien sind oft alle zwei Wochen]
- Bekommt auch Feedback, wenn Raum an einem Tag der Terminserie nicht frei („Raum nimmt Termin an oder nicht“), [Anmerkung: Feedback kommt nicht gleich, sondern erst nach Abschluss der Buchung]

4. Schwächen des Web Interfaces von Google [10:25]

- Konflikt bei wiederkehrendem Termin → Besprechungsraum lehnt Termin ab, man weiß aber nicht genau zu welchem Termin der Konflikt auftritt



A Anhang

5. Verbesserungen bzw. vermisste Features beim Web Interface von Google [11:05]
 - Bei Buchen von längerem Termin → Möglichkeit, andere Termine in anderen Raum zu verlegen [In Diskussion: Anfrage zum Verlegen]
6. Verwendetes Device beim Web Interface von Google [12:17]
 - MacBook → Browser
 - Smartphone Browser nie ausprobiert
7. Nutzung des Slackbots Joan [13:06]
 - Nein [Einmal versucht zu verwenden → hat nicht funktioniert]
8. Wozu Slackbot Joan [13:42]
 - Raum für jetzt sofort buchen (hat allerdings nicht funktioniert)
9. Stärken des Slackbots Joan

[Anmerkung: Die Frage wurde I1 nicht gestellt, da der Slackbot Joan kaum genutzt wurde]
10. Schwächen des Slackbots Joan [14:23]
 - Wusste nicht was man schreiben soll
 - Wusste nicht was ich mit Antwort/Feedback des Slackbots anfangen soll
11. Verbesserungen bzw. vermisste Features beim Slackbot Joan

[Anmerkung: Die Frage wurde I1 nicht gestellt, da der Slackbot Joan bisher kaum genutzt wurde]
12. Verwendetes Device beim Slackbot Joan [14:46]
 - MacBook → Slack App
13. Bestätigung der Raumbuchung sinnvoll [15:10]
 - Ja
 - Vorteil, um Leerbuchungen zu vermeiden (Termine eingetragen aber werden nicht abgesagt)



A Anhang

- Aktualisierung der Termine durch Bestätigung (sollte simpel funktionieren, direkt am Raum, I1 möchte nicht extra den Rechner aufklappen, um Buchung zu bestätigen)

14. Bestätigung der Raumbuchung in Ordnung, wenn intuitiv

[Anmerkung: Die Frage wurde I1 nicht gestellt, da die Bestätigung der Raumbuchung generell für sinnvoll gehalten wird]

15. Ist manuelle Bestätigung aus dem alten Office bekannt [16:29]

- Nein

16. Hat manuelle Bestätigung aus dem alten Office funktioniert

[Anmerkung: Die Frage wurde I1 nicht gestellt, da die manuelle Bestätigung der Raumbuchung aus dem alten Office nicht bekannt war]

[E-Paper Display]

1. Nutzung des E-Paper Displays, um Raum sofort zu suchen [16:57]

- Ja → Erst physikalisches schauen ob Raum frei ist → schauen am E-Paper, ob Raum frei ist → Raum direkt über E-Paper buchen
- Zitat I1: "*Wenn ich sehe da ist jemand in dem Raum drin und das E-Paper sagt frei, dann ist er für mich trotzdem nicht frei.*"

2. Nutzung des E-Paper Displays, um Raum zu späteren Zeitpunkt zu buchen [17:59]

- Nein (weiß nicht das es überhaupt geht)
- Für späteren Termin nicht extra zum Raum gehen → dann lieber Web Interface Google Kalender
- Nur erste View des E-Paper wird genutzt

3. Übereinstimmung Belegungsstatus des Raumes mit E-Paper Display [19:03]

- Bei eigener Buchung: Immer
- Generell, wenn Raum leer steht aber eigentlich Termin gebucht ist: Häufiger (10% der Fälle)



A Anhang

[Anmerkung: I1 ist im Durchschnitt nur 1,5 Tage/Woche im Office und hat trotzdem diesen Eindruck.]

4. Derzeitige Informationen auf E-Paper Display [20:19]

- Wann der nächste Termin ist
- Name des Termins
- Ob Raum gerade frei ist

5. Sinnvolle Informationen auf E-Paper Display [20:49]

- Name der Person, die Raum gebucht hat (Kürzel steht bereits da und würde ausreichen)

[Zeitmanagement]

1. Einhaltung Terminlänge [21:59]

- Länger (5 %)
- Kürzer (20 %)

2. Reaktion bei Terminüberschreitung [23:08]

- Bekommt man meistens gar nicht mit (außer Kollegen unterbrechen)
- Anderen Raum suchen (physikalisches schauen, ob Raum frei)

3. Information über Raumbelegung nach eigenem Termin [23:39]

- Idealfall: Weiß man vorher aus Google Kalender
- Kollegen von Nachfolgetermin weisen persönlich darauf hin
- In der Praxis schaut man nicht vorher im Google Kalender oder auf dem E-Paper, ob und wann ein Nachfolgetermin stattfindet (außer man sieht es zufällig bei der Buchung und hat es noch im Hinterkopf)

4. Termin aufgrund von Zeitüberschreitung umkoordinieren [24:45]

- 5%

5. Trotz gebuchten Raumes auf anderen ausweichen [25:12]



A Anhang

- Selten bis nie

[Ergänzende Fragen]

1. Termine mit externen Leuten [25:53]

- Generell: 80%
- Hier im Office: 10%

[Anmerkung: I1 ist die meiste Zeit der Woche vor Ort bei Kunden mit Partnerunternehmen zusammen tätig.]

2. Besprechungsräume nicht in akzeptablen Zustand [26:51]

- Selten (Whiteboards immer gewischt)

3. Benachrichtigung zum Überprüfen des Raumzustands

[Anmerkung: Die Frage wurde I1 nicht gestellt, da die vorherige Frage nicht mit „häufig“ beantwortet wurde]

4. Zusätzliche Ausstattung in Meeting [27:30]

- Nie (wird durch 3. Punkt widerlegt)
- Moderatorenkoffer/Flipchart bringen Moderatoren mit
- Post-It's und Stifte in jedem Raum wären gut

5. Beantwortung von Fragen nach zusätzlicher Ausstattung durch System hilfreich [29:13]

- Ja, generell schon

6. Bevorzugte Räume [29:45]

- Passend für den Rahmen des Meetings
- Nicht den Größten (R6) wenn ich ihn nicht brauche
- Nicht den Videokonferenzraum (R3) wenn ich ihn nicht brauche
- Hintergedanke: Den Kollegen keinen „besonderen“ Raum wegnehmen, den ich eigentlich nicht brauche



A Anhang

7. Unbrauchbare Räume [30:36]

- R6 vermeiden → immer sehr kühl und zu groß → Klimaanlage nicht gut steuerbar
- Sonst keine Präferenzen

8. Aspekte, die „Buchen und Nutzen von Besprechungsräumen“ schwierig gestalten [31:18]

- Bei längerem Termin (über 2 Std.) → Problem mit kurzen Terminen, die die Buchung eines Raumes für einen längeren Zeitraum verhindern

9. Buchen und Nutzen von Besprechungsräumen besser gestalten [33:08]

- Bestätigen der Buchung
 - Um Leerbuchungen zu vermeiden (sollte einfach und schnell funktionieren)
 - Am E-Paper direkt bestätigen oder Push-Benachrichtigung zum Bestätigen
 - Benachrichtigung zum Terminstart
 - Zeitintervall zum Bestätigen je nach Länge des Terms

10. Offene Punkte oder Anmerkungen [36:10]

- [Anmerkung: I1 hatte keine weiteren Anmerkungen]



A Anhang

A.4.2 Interviewter I2

[Vorgehen]

1. Arten von Besprechungen [00:26]

- Trainings
- Interviews
- Workshops
- Vorstellungsgespräche
- Kundentermine
- Spontane Meetings

2. Anpassung des Meetings [02:45]

- Gering (Uhrzeit wird geändert)

3. Arbeitsschritte vor, während und nach dem Meeting [04:00]

- Teilweise Raum 30 min vorher buchen um Vorbereitungen zu treffen
- Raum bei Training den ganzen Tag blockieren
- Teilweise auch danach noch 1 Stunde buchen für Nachbereitungen

4. Raum zu bestimmtem Zeitpunkt in der Zukunft buchen [05:10]

- Im Google Kalender
 - Ungefährer Zeitraum
 - Wunschtermin in Kalender
 - Welche Räume sind frei (passen Bedürfnisse zum Raum)
 - Wenn Raum nicht frei? → andere Zeit suchen, vorher und nachher schauen
 - Wunsch: Bestimmten Raum für zwei Stunden nächste Woche mit Angabe der Teilnehmer (vor allem bei Trainings mit bestimmtem Raum und größerem Zeitintervall bzw. Dauer)



A Anhang

- Problem mit kleinen Terminen, die einen längeren Tag „blockieren“ (andere Termine verschieben)
- Idee: Angabe: Möchte 2-Tages-Schulung machen (2 Tage, je 8 Std, ca. 9-17 Uhr, vorher eine Stunde vorbereiten, danach eine Stunde nachbereiten → 3 Buchungen oder eine Buchung von 8-18 Uhr)

5. Raum jetzt sofort buchen [12:22]

- [1] An Räumen vorbeilaufen (physikalisches nachschauen)
- [2] Wenn physikalisch frei → E-Paper Display Joan anschauen, ob für jetzt gerade ein Termin eingetragen ist
- [3] Wenn physikalisch nicht frei → Schauen wann Termin zu Ende ist
- [4] Wenn E-Paper frei → Drücken „jetzt den Raum buchen“ direkt am E-Paper Display (schön und cool, nicht extra in Hosentasche greifen müssen)

[Anmerkung: Gut, dass nicht alles gebucht werden kann (Boxen, Ecke bei Bibliothek, etc.)]

6. Wichtige Informationen zur Buchung [16:05]

- „3 Kategorien“
 - Großer Raum (R6)
 - Raum zum Aktivieren von Leuten, aber nicht mehr als 10 Leute (R4)
 - Räume zum angenehmen Arbeiten (R1, R2)

[Raumbuchung]

1. Nutzung des Web Interfaces von Google [17:10]

- Ja

[Anmerkung: I2 verwendet auch noch das E-Paper Joan, nicht aber den Slack-bot]

2. Wozu Web Interface von Google [18:00]

- Raum buchen



A Anhang

- Raum suchen
- Leute zu Terminen einladen (auch ohne Räume)
- Blocker für sich selbst setzen (seine Zeit)

3. Stärken des Web Interfaces von Google [18:43]

- Verfügbare Räume werden angezeigt (gefilterte Liste, nebeneinander)
- Möglichkeit, dass andere Leute den Termin bearbeiten können (Leute einladen, etc.)
- Übersichtlich, einfach, aufgeräumt

4. Schwächen des Web Interfaces von Google [20:26]

- Manche Räume werden als verfügbar angezeigt, sind es aber nicht
- Raumnamen sind teilweise komisch (adorsys – 4. Stock – R1)
- Hangoutlink standardmäßig drin
- Teilnehmer mit Kürzel angegeben (unpersönlich)

5. Verbesserungen bzw. vermisste Features beim Web Interface von Google [23:51]

- Kurzer Termin blockiert das Anlegen eines langen Termins → Anfrage, den kurzen Termin zu verschieben

6. Verwendetes Device beim Web Interface von Google [26:47]

- MacBook → Browser
- Smartphone nicht performant genug dafür (Schreiben von langer Agenda lieber mit Tastatur)

7. Nutzung des Slackbots Joan [28:21]

- Nein

[Anmerkung: I2 kennt den Slackbot Joan nicht, nur das E-Paper Display]

8. Wozu Slackbot Joan



A Anhang

[Anmerkung: Die Frage wurde I2 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

9. Stärken des Slackbots Joan

[Anmerkung: Die Frage wurde I2 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

10. Schwächen des Slackbots Joan

[Anmerkung: Die Frage wurde I2 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

11. Verbesserungen bzw. vermisste Features beim Slackbot Joan

[Anmerkung: Die Frage wurde I2 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

12. Verwendetes Device beim Slackbot Joan

[Anmerkung: Die Frage wurde I2 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

13. Bestätigung der Raumbuchung sinnvoll [28:53]

- Nein

14. Bestätigung der Raumbuchung in Ordnung, wenn intuitiv [28:53]

- Am Display: kritisch (möglicherweise vergessen, Display Akku leer?)
- Benachrichtigung: Hilft nicht, 10-15min vor und nach Termin beschäftigt, Kopf manchmal nicht frei dazu

[Anmerkung: I2 findet eine Bestätigung der Raumbuchung nur sinnvoll, wenn auch Bedarf (im Sinne von Raumknappheit) da ist]

15. Ist manuelle Bestätigung aus dem alten Office bekannt

- Nein

16. Hat manuelle Bestätigung aus dem alten Office funktioniert

[Anmerkung: Die Frage wurde I2 nicht gestellt, da die manuelle Bestätigung der Raumbuchung aus dem alten Office nicht bekannt war]



A Anhang

[E-Paper Display]

1. Nutzung des E-Paper Displays, um Raum sofort zu suchen [34:24]
 - Ja → allerdings nur sofort Raum zu buchen (Erklärung siehe [Vorgehen]/Frage 5)
2. Nutzung des E-Paper Displays, um Raum zu späteren Zeitpunkt zu buchen [34:55]
 - Nein, zu behäbig zu bedienen
 - Zum Planen lieber Rechner verwenden (mehr Zeit und schneller, Leute einladen, Agenda schreiben, weitere Vorbereitungen)
3. Übereinstimmung Belegungsstatus des Raumes mit E-Paper Display [37:40]
 - Eindruck, dass es eher stimmt
 - Zitat I2: „*Ich find's blöder wenn belegt ist und frei drin steht als wenn's andersrum der Fall ist*“
4. Derzeitige Informationen auf E-Paper Display [39:35]
 - Status des Raums (frei/belegt)
 - Nächster Termin
 - Wie lange belegt
 - Name des Termins
 - Name der Person
5. Sinnvolle Informationen auf E-Paper Display [40:13]
 - Mehr Informationen als in (4) nicht nötig
 - Keine Teilnehmerliste, keine Agenda

[Zeitmanagement]

1. Einhaltung Terminlänge [41:41]
 - Länge meist eingehalten (mit Ausnahmen)
 - Geringe Über- und Unterschreitungen (15-30min)



A Anhang

2. Reaktion bei Terminüberschreitung [43:05]

- Raum verlassen (Beenden und neuer Termin, Ausweichen auf Bibliothek)
- Fragen, ob noch wenige Minuten überzogen werden darf

3. Information über Raumbelegung nach eigenem Termin [45:18]

- Beim Buchen sehen das direkt danach Termin stattfindet
- Vor dem Reingehen auf das E-Paper schauen

[Anmerkung: I2 schaut nicht vor jeden Termin nach, ob es direkt einen Anschlusstermin gibt]

4. Termin aufgrund von Zeitüberschreitung umkoordinieren [47:10]

- Sehr selten

5. Trotz gebuchten Raumes auf anderen ausweichen [47:51]

- 1-2 mal passiert in 5 Monaten
- Grund: Kundentermin im Raum → Hatte Vorrang

[Ergänzende Fragen]

1. Termine mit externen Leuten [49:02]

- Eher nicht so oft

2. Besprechungsräume nicht in akzeptablen Zustand [49:20]

- 50/50 (Whiteboards teilweise nicht sauber, Glas stehen geblieben, Stühle im R6 öfters nicht richtig rein geschoben)

3. Benachrichtigung zum Überprüfen des Raumzustands [50:38]

- Nicht gut

[Anmerkung: I2 schaltet Benachrichtigungen zu Terminen generell aus]

4. Zusätzliche Ausstattung in Meeting [51:43]

- Post-It's und Stifte nahezu immer



A Anhang

5. Beantwortung von Fragen nach zusätzlicher Ausstattung durch System hilfreich [52:33]

- Nein, brauche nur Post-It's und Stifte
- Manchmal extra Whiteboard

6. Bevorzugte Räume [53:11]

- R1 und R2 (Fenster, Ausblick)
- R4 weil man nicht sitzt, sondern steht

7. Unbrauchbare Räume [54:21]

- R3, zu großer Tisch für kleinen Raum, dunkel
- R6 nicht praktikabel, zu viele Tische, Schlauchform, Wände nicht von überall gut als Arbeitsfläche einsehbar

8. Aspekte, die „Buchen und Nutzen von Besprechungsräumen“ schwierig gestalten [55:28]

- Auflösen von kleinem Termin → das großer Termin Platz hat

9. Buchen und Nutzen von Besprechungsräumen besser gestalten [56:40]

- Konflikte beim Suchen von (längerem) Termin
- Szenario: Zeitraum vorschlagen mit gewissen Personen → automatischer Terminvorschlag, wenn alle (+Raum) verfügbar sind
- Mittagessen, Kekse, Nüsse bei externen Leuten (einheitliche Regelung und Ansprechstation)
- Kontext des Termingeschäfts erschließen (Schulung, Kundentermine, etc.)

10. Offene Punkte oder Anmerkungen [59:40]

- Bezug auf Fragebogen: Frage, wie lange Termin üblicherweise dauert
 - Zeit variiert sehr stark
 - Von 15 min bis 2 Tage alles möglich



A Anhang

A.4.3 Interviewter I3

[Vorgehen]

1. Arten von Besprechungen [00:34]

- Vorstellungsgespräche
- Mitarbeitergespräche
- Sämtliche Meetings die man so hat (z.B. Produktpräsentation)
 - Alles einmalige Ereignisse, keine wiederkehrenden

2. Anpassung des Meetings

[Anmerkung: Frage wurde I3 nicht gestellt, da er keine wiederkehrenden Besprechungen bucht]

3. Arbeitsschritte vor, während und nach dem Meeting [01:19]

- Vorher: Welchen Raum?
- Für Anforderungen passend
 - Größe (nicht R6 für Vorstellungsgespräch)
 - Diskretion (R3/R4 für Mitarbeitergespräch)

4. Raum zu bestimmtem Zeitpunkt in der Zukunft buchen [03:18]

- Raum zu dieser Zeit frei? Welcher Raum? Passend für Bedürfnisse
- Mindestens eine Woche vorher buchen

5. Raum jetzt sofort buchen [04:42]

- [1] Erst auf E-Paper Joan schauen
- [2] Raum auch physisch frei
- [3] Wenn Raum frei → Drücken „jetzt den Raum buchen“ direkt am E-Paper Display

6. Wichtige Informationen zur Buchung [06:02]



A Anhang

- Anzahl der Plätze
- Diskretion

[Anmerkung: I3 bucht in der Regel nur Vorstellungsgespräche oder Mitarbeitergespräche mit wenigen Personen]

[Raumbuchung]

1. Nutzung des Web Interfaces von Google [07:10]

- Ja

2. Wozu Web Interface von Google [07:37]

- Raumbuchungen
- Personen zu Termin einladen

3. Stärken des Web Interfaces von Google [09:06]

- Sehr übersichtlich
- Gleichzeitig freie Zeiten von anderen Personen/Räumen
- Dadurch Möglichkeit, Puffer einzuplanen (Überzogene Termine, Wegezeit)

[Anmerkung: I3 würde am liebsten über den Apple Kalender buchen und hat dies auch ausprobiert, funktioniert aber scheinbar nicht oder nicht gut]

4. Schwächen des Web Interfaces von Google [11:18]

- Keine, alle notwendigen Informationen da

[Anmerkung: I3 arbeitet erst seit einigen Monaten mit dem Google Kalender]

5. Verbesserungen bzw. vermisste Features beim Web Interface von Google [11:53]

- Nein

[Anmerkung: I3 arbeitet erst seit einigen Monaten mit dem Google Kalender]

6. Verwendetes Device beim Web Interface von Google [12:08]

- MacBook → Browser



A Anhang

- Smartphone noch nie ausprobiert

7. Nutzung des Slackbots Joan [12:44]

- Nein

[Anmerkung: I3 kennt den Slackbot Joan nicht, nur das E-Paper Display]

8. Wozu Slackbot Joan

[Anmerkung: Die Frage wurde I3 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

9. Stärken des Slackbots Joan

[Anmerkung: Die Frage wurde I3 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

10. Schwächen des Slackbots Joan

[Anmerkung: Die Frage wurde I3 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

11. Verbesserungen bzw. vermisste Features beim Slackbot Joan

[Anmerkung: Die Frage wurde I3 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

12. Verwendetes Device beim Slackbot Joan

[Anmerkung: Die Frage wurde I3 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

13. Bestätigung der Raumbuchung sinnvoll [14:31]

- Nein

14. Bestätigung der Raumbuchung in Ordnung, wenn intuitiv [14:49]

- Ja, falls Termine ausfallen (um Raum wieder frei zu geben, keine Leerbuchungen)
- Zeitraum etwa 15min nach Terminstart
- Bestätigung nicht am Rechner, sondern über E-Paper oder Smartphone



A Anhang

15. Ist manuelle Bestätigung aus dem alten Office bekannt [17:00]

- Nein

16. Hat manuelle Bestätigung aus dem alten Office funktioniert

[Anmerkung: Die Frage wurde I3 nicht gestellt, da die manuelle Bestätigung der Raumbuchung aus dem alten Office nicht bekannt war]

[E-Paper Display]

1. Nutzung des E-Paper Displays, um Raum sofort zu suchen [19:03]

- Ja → für spontane Meetings
- Buchen dann direkt am E-Paper Display

2. Nutzung des E-Paper Displays, um Raum zu späteren Zeitpunkt zu buchen [20:08]

- Nein, wenn genug Zeit ist dann direkt am Rechner über das Google Web Interface

3. Übereinstimmung Belegungsstatus des Raumes mit E-Paper Display [21:18]

- Eindruck, dass es meistens stimmt

4. Derzeitige Informationen auf E-Paper Display [22:20]

- Uhrzeit
- Status des Raums (frei/belegt)

5. Sinnvolle Informationen auf E-Paper Display [22:40]

- Wunsch: „Privatisierung“ bei Mitarbeitergespräch
- Möglichkeit für Personal, Titel auf „privat“ ändern
- Generelle Möglichkeit, auch im Kalender der jeweiligen Person

[Zeitmanagement]

1. Einhaltung Terminlänge [25:27]

- So gut wie nie zu lange



A Anhang

- Eher zu kurz (reichlich Zeit eingeplant)

[Anmerkung: I3 bucht seine Termine in der Regel mit etwas Pufferzeit]

2. Reaktion bei Terminüberschreitung [26:40]

- Gespräch in anderen Raum verlegen
- Neuer Termin zu anderer Zeit
- Anderen Ort im Office suchen

3. Information über Raumbelegung nach eigenem Termin [27:33]

- E-Paper Display (I3 schaut aber nicht jedes mal vorher auf das Display)

[Anmerkung: Beim alten Arbeitgeber von I3 gab es einen 30-Minuten Puffer zwischen allen Terminen, für Getränke und Catering]

4. Termin aufgrund von Zeitüberschreitung umkoordinieren [28:49]

- Noch nie

5. Trotz gebuchten Raumes auf anderen ausweichen [29:21]

- Noch nie

[Ergänzende Fragen]

1. Termine mit externen Leuten [29:43]

- 75% externe Leute (durch Vorstellungsgespräche)

2. Besprechungsräume nicht in akzeptablen Zustand [30:42]

- Sehr selten

[Anmerkung: I3 geht meist 15min vor dem Termin in den Raum und schaut, ob dieser in Ordnung ist]

3. Benachrichtigung zum Überprüfen des Raumzustands [32:11]

- Nicht notwendig, schaut sowieso vorher nach

4. Zusätzliche Ausstattung in Meeting [33:38]



A Anhang

- Ausstattung nahezu überall gleich und bestens
5. Beantwortung von Fragen nach zusätzlicher Ausstattung durch System hilfreich [35:30]
- Generell ja
 - Möglicherweise aktiver Hinweis nach der Buchung (Moderatorenkoffer, Post-It's, Flipchart, HDMI-Anschluss, Getränke)
6. Bevorzugte Räume [36:38]
- R1 und R2 (Heller durch Fenster, Ausblick)
 - Nachteil im R1/R2: Man kann nicht an der Stirnseite sitzen (bevorzugt bei Vorstellungsgesprächen)
 - R3 und R4 (Diskretion, lieber für Vorstellungsgespräche und Mitarbeitergespräche, hohe Stühle im R4 kein Problem)
7. Unbrauchbare Räume [38:55]
- Kein zu großer Raum (R6) für Vorstellungsgespräch
8. Aspekte, die „Buchen und Nutzen von Besprechungsräumen“ schwierig gestalten [39:52]
- Nein, bisher noch keine Probleme
9. Buchen und Nutzen von Besprechungsräumen besser gestalten [41:59]
- Kalender funktioniert gut
 - Eigene Tools zu umständlich
 - Wunsch nach „Privatisierung“ von bestimmten Terminen
10. Offene Punkte oder Anmerkungen [43:14]
- E-Paper funktioniert gut (Status des Raums, schnelles reingehen und buchen direkt am Display)



A.4.4 Interviewter I4

[Vorgehen]

1. Arten von Besprechungen [00:32]

- Projektbezogene Meetings (Reviews, Retrospektive, Plannings, Scrum)
- Jour-fixe
- Bewerbungsgespräche, Mitarbeitergespräche
- Kundengespräche
- Workshops

→ Sowohl Single aus auch Repeated Meetings

2. Anpassung des Meetings [01:45]

- So gut wie gar nicht

3. Arbeitsschritte vor, während und nach dem Meeting [02:13]

- Wie viele Personen?
- Welcher Raum? (Passend für Bedürfnisse bzw. Art des Meetings)
 - R4 (Kreativraum)
 - R3 (Videokonferenz)
 - R1/R2 als „normale“ Räume
 - R6 bei mehr Personen und längerer Zeit (Luft besser)
- Vorbereitungen für Meeting direkt davor
- Inhalte der Meetings meist „on the fly“ entwickelt

4. Raum zu bestimmtem Zeitpunkt in der Zukunft buchen [04:25]

- Web Interface Google Kalender
 - Meist Zeit vom Kunden vorgegeben (feste Zeit)
 - Verfügbarkeit von anderen Teilnehmern



A Anhang

- Welcher Raum ist verfügbar? Passend für Meeting
- Raum buchen (Buchungen innerhalb eines Monats voraus, aus [Zeitmanagement/Frage 3] aufgenommen)

[Anmerkung: Für spontane Meetings mit wenigen Personen nutzt I4 oft die Arbeitsboxen]

5. Raum jetzt sofort buchen [06:26]

- [1] Physikalisches schauen, ob Raum frei ist
- [2] Wenn ja → E-Paper schauen, ob frei
- [3] Direkt buchen über E-Paper Display

6. Wichtige Informationen zur Buchung [07:12]

- Raumgröße
- Anzahl der Personen
- Ausstattung im Allgemeinen (Wunsch nach Beamer)

[Raumbuchung]

1. Nutzung des Web Interfaces von Google [08:36]

- Ja

2. Wozu Web Interface von Google [08:44]

- Raumbuchungen
- Terminplanung
- Abgleich mit den Kalendern von Kollegen
- Verwendung nur zum Planen von Terminen, nicht wenn sofort ein Raum benötigt wird

3. Stärken des Web Interfaces von Google [10:30]

- Überall verfügbar (ob Office oder privater Rechner)

4. Schwächen des Web Interfaces von Google [12:45]



A Anhang

- Teams/Gruppen definieren und dazu einen Termin finden (funktioniert dann aber nur mit internen Mitarbeitern)

5. Verbesserungen bzw. vermisste Features beim Web Interface von Google [15:00]

- Outlook: Auswahl von Teilnehmern → dann Vorschlag eines Raumes, der zu Anzahl Personen und Zeitintervall passt
- Platzoptimierung (nicht Umschalten zwischen GUESTS/ROOMS)
- Ein Kalender für alles (Zep Integration in Google Kalender)

6. Verwendetes Device beim Web Interface von Google [17:30]

- Planen und Räume buchen: MacBook → Browser
- Übersicht unterwegs: Google Kalender App am Smartphone

7. Nutzung des Slackbots Joan [18:30]

- Nein

8. Wozu Slackbot Joan

[Anmerkung: Die Frage wurde I4 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

9. Stärken des Slackbots Joan

[Anmerkung: Die Frage wurde I4 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

10. Schwächen des Slackbots Joan

[Anmerkung: Die Frage wurde I4 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

11. Verbesserungen bzw. vermisste Features beim Slackbot Joan

[Anmerkung: Die Frage wurde I4 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

12. Verwendetes Device beim Slackbot Joan



A Anhang

[Anmerkung: Die Frage wurde I4 nicht gestellt, da der Slackbot Joan noch nie genutzt wurde]

13. Bestätigung der Raumbuchung sinnvoll [18:52]

- Ja

[Anmerkungen]

- Art der Bestätigung
 - * An Tür: In Ordnung
 - * Mit Device: Würde nerven
- I4 hält eine Bestätigung der Raumbuchung generell für sinnvoll, um Leerbuchungen zu vermeiden
- Idee: Serientermine, die 3 Mal nicht stattgefunden haben, herausnehmen bzw. nachfragen, alternativ: Serientermine nur über bestimmten Zeitraum (z.B. quartalsweise)
- Problem: Wird ein Raum aufgrund einer nicht erfolgten Bestätigung frei gegeben, kann er auch nur „spontan“ genutzt werden und nicht von vornherein in die Planung von anderen Personen mit einfließen

14. Bestätigung der Raumbuchung in Ordnung, wenn intuitiv

[Anmerkung: Diese Frage wurde I4 nicht gestellt, da er Frage 13 mit „Ja“ beantwortet hat]

15. Ist manuelle Bestätigung aus dem alten Office bekannt [21:58]

- Nein

16. Hat manuelle Bestätigung aus dem alten Office funktioniert

[Anmerkung: Die Frage wurde I4 nicht gestellt, da die manuelle Bestätigung der Raumbuchung aus dem alten Office nicht bekannt war]

[E-Paper Display]

1. Nutzung des E-Paper Displays, um Raum sofort zu suchen [22:35]



A Anhang

- [1] Physikalisches schauen ob Raum frei
 - [2] Wenn frei → E-Paper Display frei?
 - a) Wenn frei → E-Paper drücken und buchen
 - b) Wenn belegt → Trotzdem rein gehen
2. Nutzung des E-Paper Displays, um Raum zu späteren Zeitpunkt zu buchen [24:03]
- Nein, wenn alles belegt → Alternative (Arbeitsbox, Couch)
3. Übereinstimmung Belegungsstatus des Raumes mit E-Paper Display [25:10]
- In letzter Zeit: weniger, immer wieder Räume frei obwohl laut Display gebucht
(Vermutete Ursache: Urlaubszeit)
[Anmerkung: I4 ist erst seit einigen Monaten in der Firma und fehlt möglicherweise die Referenz]
4. Derzeitige Informationen auf E-Paper Display [26:35]
- Status des Raums (frei/belegt)
 - Von wann bis wann gebucht
 - Nächster Termin
5. Sinnvolle Informationen auf E-Paper Display [27:08]
- (Wenn belegt) → Aktuell freie Räume (Vgl. grüne Lampe im Parkhaus)
[Zeitmanagement]
1. Einhaltung Terminlänge [28:27]
- Selbst: Relativ oft (Aufgrund von Durchtaktung der Termine)
 - Andere: Auch meist, da jeder wieder Anschlusstermine hat
2. Reaktion bei Terminüberschreitung [30:15]
- Kurzer Austausch mit Folgetermin
 - Fragen, ob für Alternative für Teilnehmer akzeptabel



A Anhang

- Raum verlassen

3. Information über Raumbelegung nach eigenem Termin [31:05]

- Auf E-Paper Display vor dem Raum schauen (nicht immer vor dem Termin, teilweise auch währenddessen)

[Anmerkung: Information in welchem Raum man sich befindet wäre hilfreich]

4. Termin aufgrund von Zeitüberschreitung umkoordinieren [34:25]

- So gut wie gar nicht

5. Trotz gebuchten Raumes auf anderen ausweichen [34:43]

- • Sehr selten

[Ergänzende Fragen]

1. Termine mit externen Leuten [35:15]

- Relativ häufig (>50 %) durch Vertriebstätigkeit oder Projekte mit Externen

2. Besprechungsräume nicht in akzeptablen Zustand [35:52]

- Whiteboards oft nicht sauber

[Anmerkung: Ein Teil der Antworten wurde in Punkt 4 aufgenommen]

3. Benachrichtigung zum Überprüfen des Raumzustands [38:35]

- Nein, keine Zeit dafür (Extern unterwegs, Taktung von Terminen)

4. Zusätzliche Ausstattung in Meeting [39:04]

- Beamer im R6 wäre gut
- USB-Ports zum Aufladen von Handy, Tablet, etc.
- Adapterkoffer (HDMI- und DisplayPort-Kabel)
- Whiteboard Stifte und Wischmöglichkeit
- Privacy/Diskretion: Bei Terminen mit Kunden, Gespräche Führungskreis, Mitarbeitergespräch



A Anhang

- Kleiner Tisch für Gläser, Getränke, Block und Stifte

[Anmerkung: Ein Teil der Antworten wurde aus Punkt 2 aufgenommen]

5. Beantwortung von Fragen nach zusätzlicher Ausstattung durch System hilfreich [41:36]

- Initial schon

[Anmerkung: Ein Teil der Antworten wurde in Punkt 7 aufgenommen]

6. Bevorzugte Räume [47:10]

- R1 und R2 (hell, angenehme Größe)

[Anmerkung: Ein Teil der Antworten wurde in Punkt 7 aufgenommen]

7. Unbrauchbare Räume [48:46]

- Vernünftiger Konferenzraum: R3 ist „semi-gut“, Mikro-Kabel zu kurz, 2 Kameras (Sprechender, Alle Personen), 2 Monitore (eigener Screen, Gesprächspartner), Ton über Lautsprecher, technische Bedingungen im Allgemeinen verbessern

- R3: Zu laut durch Klimaanlage, dunkel

- R4: Zu laut durch Klimaanlage und Kicker, für Workshops ok, für Kunden eher nicht wegen Hochstühlen

- R6 ist groß aber dunkel, da mitten im Office

[Anmerkung: Ein Teil der Antworten wurde aus den Punkten 5 und 6 aufgenommen]

8. Aspekte, die „Buchen und Nutzen von Besprechungsräumen“ schwierig gestalten [49:47]

- Nein

9. Buchen und Nutzen von Besprechungsräumen besser gestalten [50:07]

- Automatischer Vorschlag eines Raums, wenn 3-4 Teilnehmer zu einem Termin hinzugefügt werden



A Anhang

- „Dashboard“ zur Raumbelegung, Idee: Smartphone App oder Widget, das verfügbare Räume anzeigt

10. Offene Punkte oder Anmerkungen [54:24]

- R6 nach Bedarf in zwei Räume unterteilbar machen (Tür dazwischen)
- Alternativen in Form von externen Räumen (Räume woanders buchen)
- Möglichkeit, Tische flexibler umzubauen je nach Art des Meetings
- Fernseher sind zu klein (vor allem im R6, wenn jemand ganz hinten sitzt) → Wunsch nach Beamer



A Anhang

A.5 Personae



Abbildung A.1: Persona Heiko Neumann aus dem Bereich *Development*



A Anhang



Abbildung A.2: Persona Sandra Böhm aus dem Bereich *Coaching*



A Anhang





A Anhang



ERIC SCHWEIZER

„Nur entspannt kann ich locker mit dem Kunden sprechen.“

ZIELE
Ich brauche eine gute Übersicht, um mich gut auf die zeitliche Verfügbarkeit meiner Kunden einzustellen

ARBEITSORT
Office ● ○ ○ ○ ○
Kunde ○ ○ ○ ○ ○

SOFTSKILLS
Teamfähigkeit ● ● ○ ○ ○
Konfliktfähigkeit ● ● ● ○ ○
Organisation ● ● ● ● ●
Kommunikation ● ● ● ● ●

PERSONA
ECKDATEN
45 Jahre
Wohnt in Baiersdorf, 30 km zum Arbeitsplatz
Verheiratet, zwei Kinder (11, 14 Jahre)

HOBBIES, LEIDENSCHAFTEN
Haus & Garten, Zeit mit der Familie verbringen, Fitnessstudio, Sauna

BERUFSEFAHRUNG
Ausbildung Fachinformatiker Systemintegration
12 Jahre Berufserfahrung in der Entwicklung
6 Jahre Berufserfahrung in der Projektleitung
8 Jahre Berufserfahrung im Vertrieb

BUCHUNGSVERHALTEN
Aktiv: Kundetermine, Workshops, Jour fixe
Passiv: Mitarbeitergespräche

EMOTIONEN
+ Technische Spielereien, Erlebnisbäder, Candle-Light-Dinner
- Elternabende, Staus, Autopannen

Eng getaktete Termine sind für Eric an der Tagesordnung. Dabei pendelt er meist zwischen unterschiedlichen Orten und muss auch seine Wegezeit von und zu externen Terminen beachten. Er nutzt sein Handy, um trotz seines vollen Terminkalenders den Überblick nicht zu verlieren. Sein Smartphone ist sozusagen sein Office-in-the-pocket.





Abbildung A.4: Persona Eric Schweizer aus dem Bereich *Vertrieb*

A.6 UI-Elemente aus dem Prototyping Tool

Ich bin dein persönlicher Raumbuchungs-Assistent. Ich helfe dir gerne dabei, einen Besprechungsraum für deine Termine zu finden und zu buchen.

Abbildung A.5: UI-Element *TextMessage* aus dem Prototyping Tool

Bitte wähle die Art deines Termins aus:

Vorstellungsgespräch	<input type="radio"/>
Mitarbeitergespräch	<input type="radio"/>
Sonstiges	<input type="radio"/>

Abbildung A.6: UI-Element *TypeOfMeeting* aus dem Prototyping Tool

Wieviele Personen nehmen am Meeting teil?

7 TeilnehmerInnen

Abbildung A.7: UI-Element *SliderParticipants* aus dem Prototyping Tool

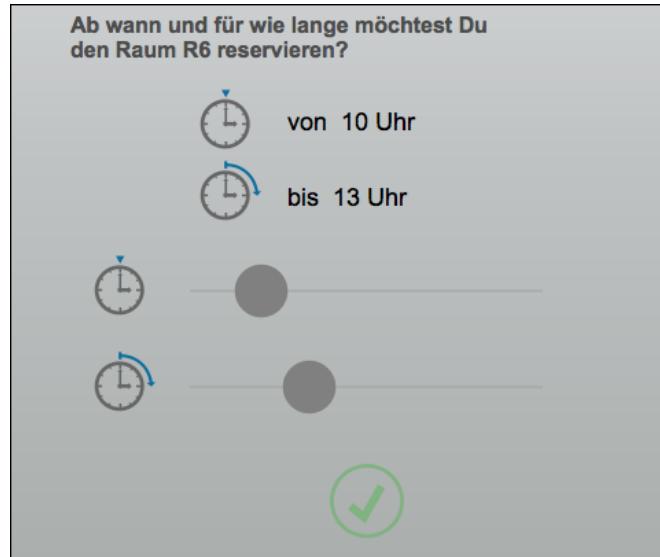


Abbildung A.8: UI-Element *SliderStartTime* aus dem Prototyping Tool

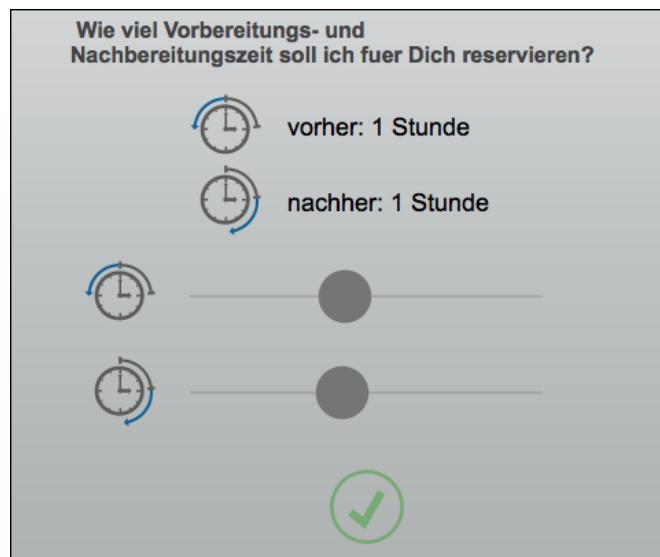


Abbildung A.9: UI-Element *SliderTimeBuffer* aus dem Prototyping Tool

A Anhang

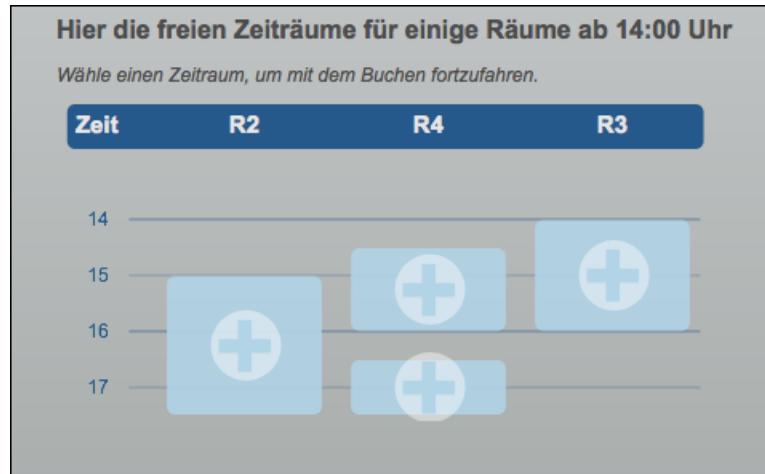


Abbildung A.10: UI-Element *CalendarView* aus dem Prototyping Tool



Abbildung A.11: UI-Element *CollectionViewDates* aus dem Prototyping Tool

A Anhang

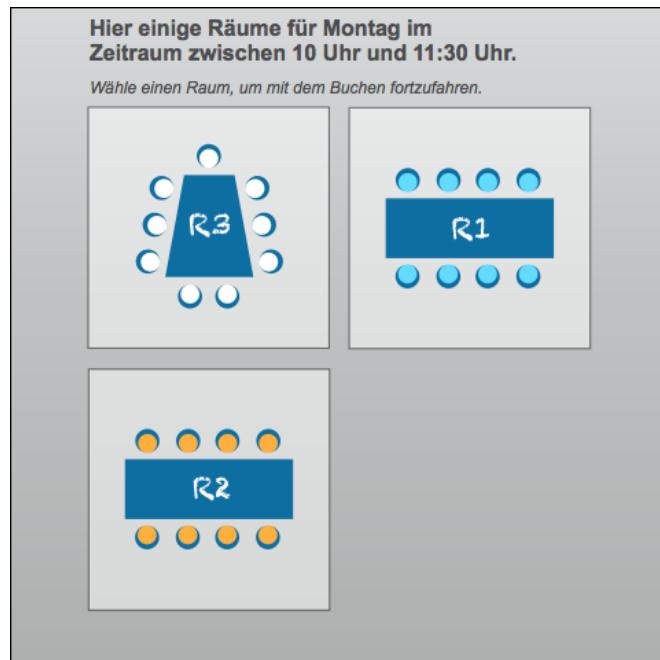


Abbildung A.12: UI-Element *CollectionViewRooms* aus dem Prototyping Tool

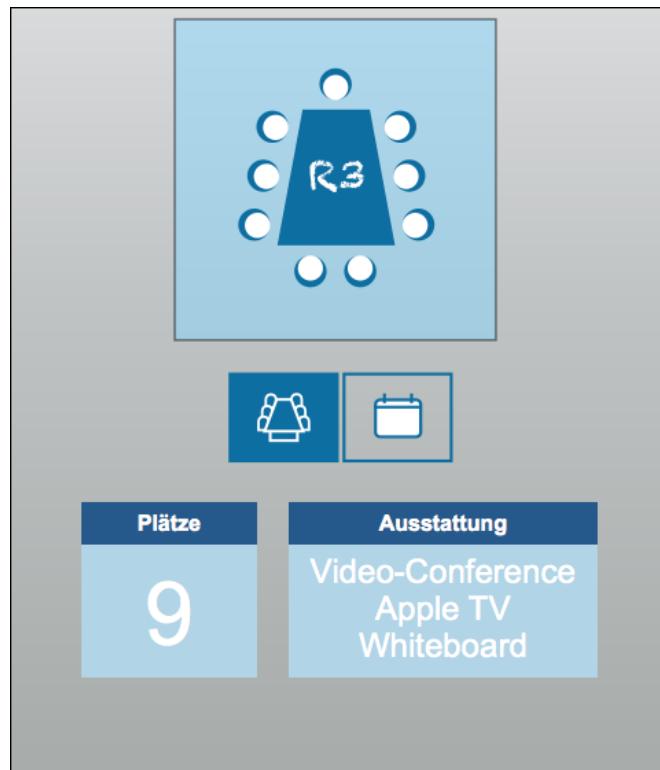


Abbildung A.13: UI-Element *DetailView* aus dem Prototyping Tool

A Anhang

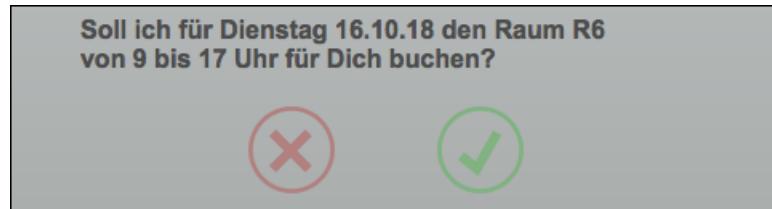


Abbildung A.14: UI-Element *ConfirmBooking* aus dem Prototyping Tool



Abbildung A.15: UI-Element *Certificate* aus dem Prototyping Tool



Abbildung A.16: UI-Element *RoomMap* aus dem Prototyping Tool



A Anhang

A.7 Einverständniserklärung zum User Test

Die folgende Einverständniserklärung wird den Teilnehmern vor der Durchführung des User Tests zur Unterschrift vorgelegt.

Usability-Test zur Erfassung von Erfahrungen und Erwartungen in Hinblick auf die Nutzung eines Chatbots um einen Raum zu buchen

Im Rahmen einer Masterarbeit an der Technischen Hochschule Nürnberg in Zusammenarbeit mit der adorsys GmbH & Co. KG beschäftigen wir uns mit dem Design, der Bedienung und der Funktionalität eines Chatbots, mit dessen Hilfe Mitarbeiter und Mitarbeiterinnen einen Besprechungsraum reservieren können. Hierbei geht es um:

- die Benutzerführung für einen entsprechenden Chatbot
- die Informationen, welche Nutzer von einem solchen Chatbot erwarten
- den Kommunikationsstil, den Nutzer erwarten

Hierfür führen wir einen Usability-Test mit einem von uns entwickelten Prototyping-Tool durch. Das Ziel ist es, mehr über Erwartungen von potentiellen Anwendern und die von ihnen verwendeten Formulierungen bei Nutzung eines Conversational Interfaces zu erfahren, und diese Erkenntnisse in die Entwicklung des Chatbots einfließen lassen zu können.

Alle Fragen und Aufgaben in diesem Usability-Test beziehen sich auf die Einstufung und Verbesserung des Chatbots und nicht darauf, die Fähigkeiten und Fertigkeiten der Testperson zu testen. Alle Angaben werden vertraulich behandelt und ausschließlich anonymisiert verarbeitet und archiviert. Weiterhin werden diese Angaben nur im Rahmen der Entwicklung dieses Chatbots und in hierauf bezogene Präsentationen verwendet.

- Unternehmen: adorsys GmbH & Co. KG
- Projektleitung: Steffen Blümm, Technical Lead iOS / CUI
- Method: Think-Aloud
- Dauer:
- Datum:



A Anhang

Einverständniserklärung

Ich erkläre mich dazu bereit, im Rahmen des oben beschriebenen Projekts an einen Usability-Test teilzunehmen. Ich wurde über die Ziele des Projekts informiert. Ich kann den Test jederzeit abbrechen, weitere Tests ablehnen und meine Einwilligung in eine Aufzeichnung und Niederschrift des Tests jederzeit zurückziehen, ohne dass mir dadurch irgendwelche Nachteile entstehen.

Ich bin damit einverstanden, dass der Test mit einem Aufnahmegerät aufgezeichnet und sodann von den MitarbeiterInnen der adorsys ausgewertet wird. Für die Auswertung des Usability-Tests werden alle Angaben zu meiner Person aus dem Text entfernt und/oder anonymisiert. Mir wurde außerdem versichert, dass der Test in Veröffentlichungen nur in Ausschnitten zitiert wird, um sicherzustellen, dass ich auch durch die Reihenfolge von im Test erwähnten Ereignissen nicht für Dritte erkennbar sein werde.

Datum, Unterschrift der Testperson

A.8 Feedback zum Usability Testessen

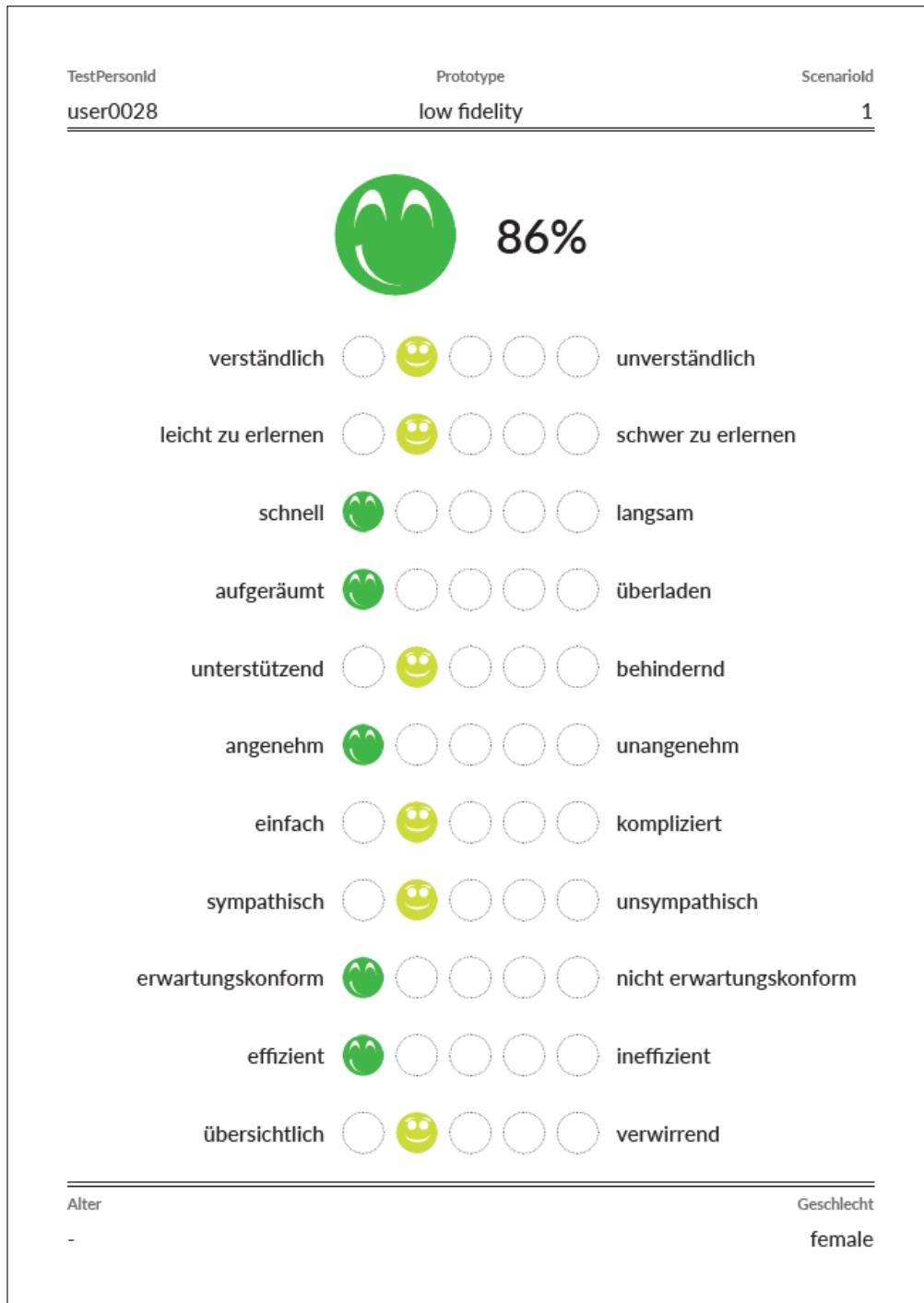


Abbildung A.17: Ergebnisse zum Fragebogen von user0028



A Anhang

TestPersonId	Prototype	Scenarioid
user0028	low fidelity	1
Anmerkungen		
"Wenn das bei uns so einfach funktionieren würde, wäre das schon toll."		
"waren jetzt nicht soooo viele Schritte."		
-> Daten komprimiert in Stenno-Stil		
Positiv: - UI-Elemente einleuchtend und verständlich - Confirmation als gute empfunden -> Absicherung		
Negativ: - Lageplan - Farbliche Markierung des Raumes geht unter		
Anregung: - Ausstattung des Raumes abfragen		
<hr/> <hr/>		
Alter		Geschlecht
- (38)		female

Abbildung A.18: Notizen zum Fragebogen von user0028

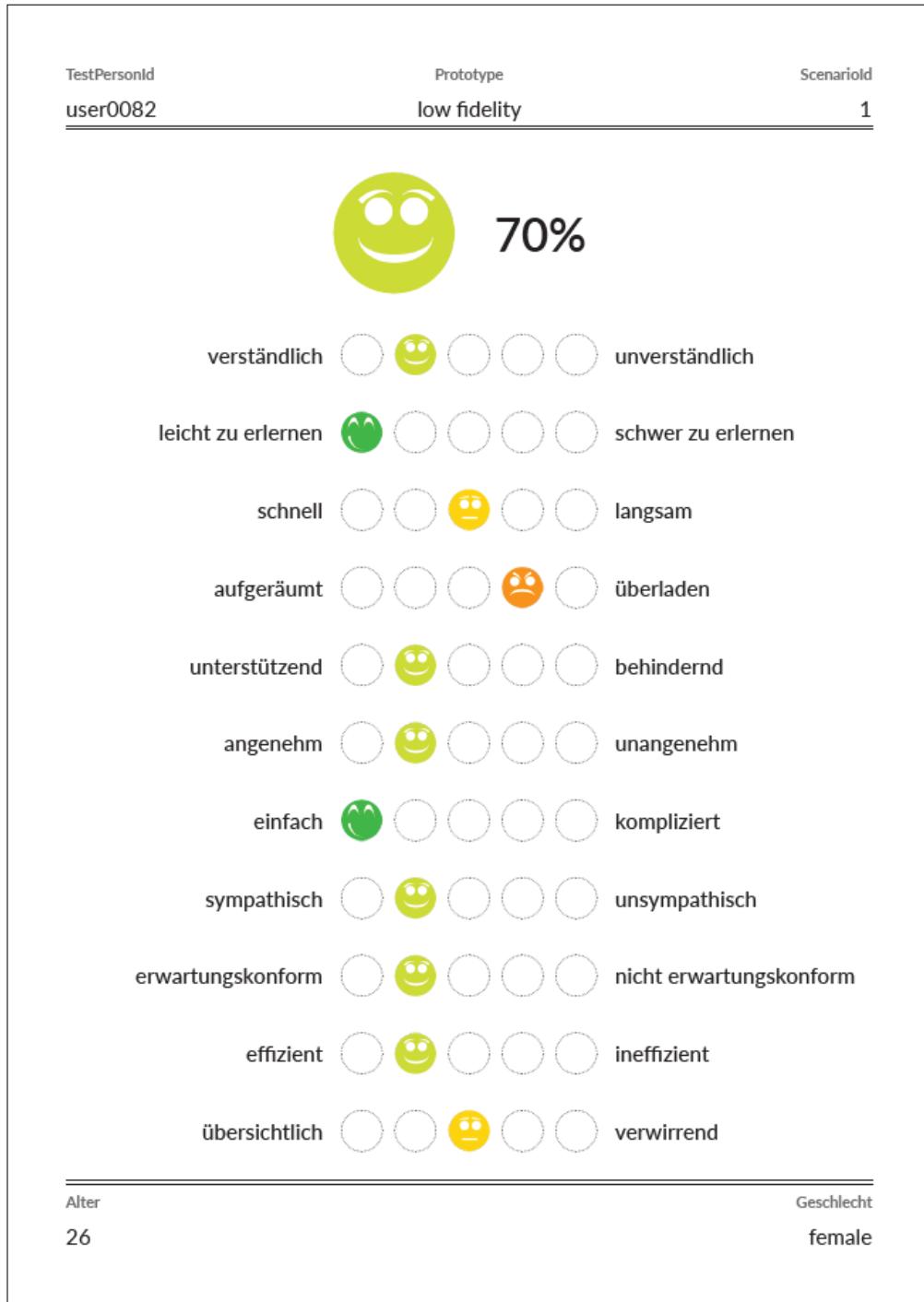


Abbildung A.19: Ergebnisse zum Fragebogen von user0082



A Anhang

TestPersonId user0082	Prototype low fidelity	Scenarioid 1
Anmerkungen		
schnell langsam		
Testperson merkte an, dass sich diese Bewertung eher auf die Antwortgeschwindigkeit des Chatbots / WOz bezieht, als auf die Geschwindigkeit des Prozesses. Sie äußerte sich in die Richtung, dass der Prozeß durch das Conversational Interface eher beschleunigt wird.		
Positiv: - Raumicons - Form der Sitzanordnung + Anzahl Personen erkennbar - "Chatbot is thinking" - "Entschuldigung, das muss ich noch lernen." -> macht Chatbot sympathisch		
Negativ: - Lageplan: Markierung nicht stark genug gegenüber Küche & Werkzeugraum - Duration-Slider: Icon redundant - da zweimal vorhanden		
Anregung - Eintrag in Kalender bzw. Vorschlag im Kalendar zum bestätigen - Ausstattung - Auto-Element-Nachrichten sollten sich stärker von <Nachrichten um User> unterscheiden		
Alter 26	Geschlecht female	

Abbildung A.20: Notizen zum Fragebogen von user0082

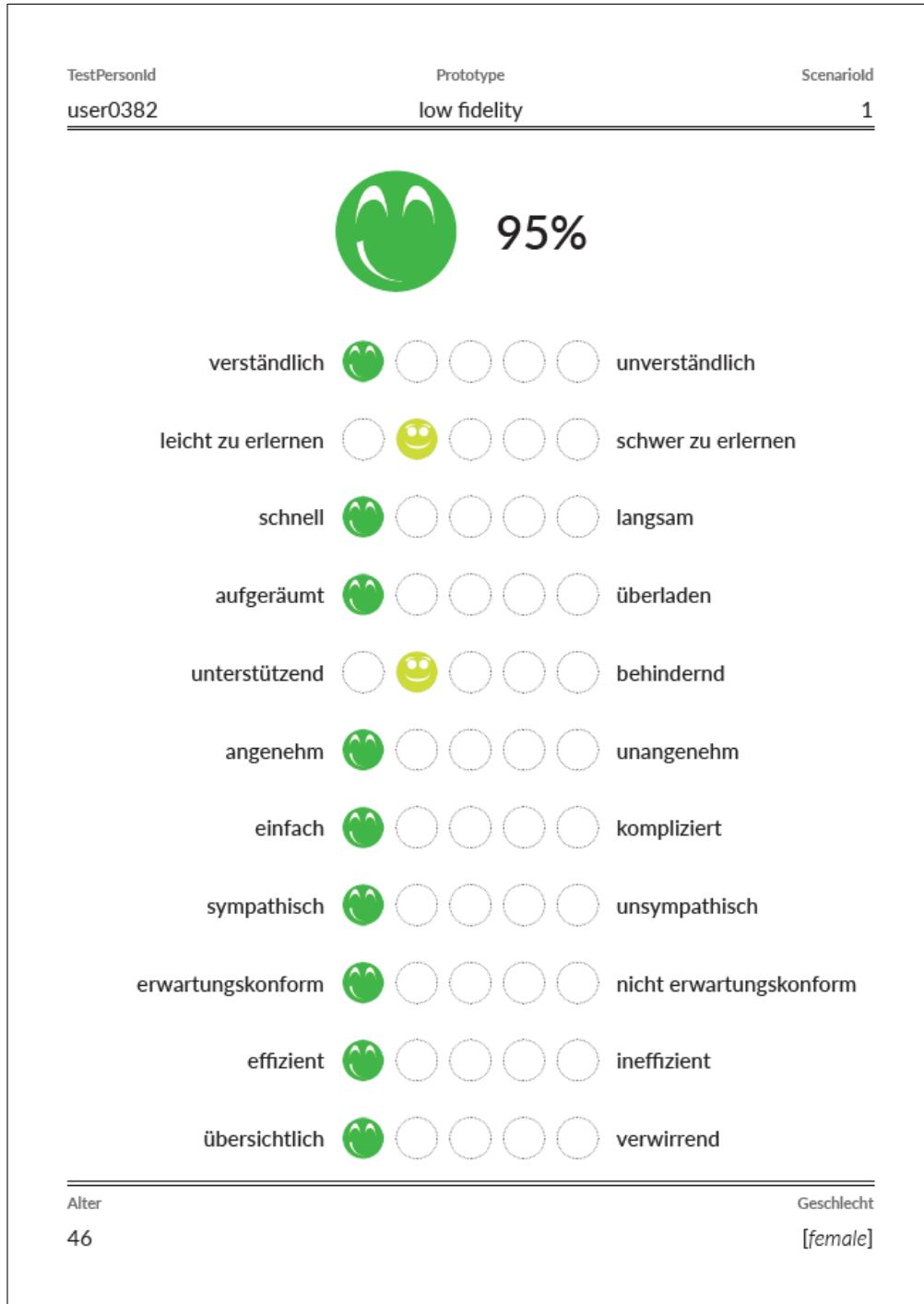


Abbildung A.21: Ergebnisse zum Fragebogen von user0382



A Anhang

TestPersonId	Prototype	Scenarioid
user0382	low fidelity	1
Anmerkungen		
"Das hätte ich auch gerne, ich kenne das komplizierter."		
Positiv: - Chatbot is thinking - Lageplan		
Anregung: - Gibt es WLAN - Getränke, Catering anfordern/planen		
Alter	Geschlecht	
46	[female]	

Abbildung A.22: Notizen zum Fragebogen von user0382

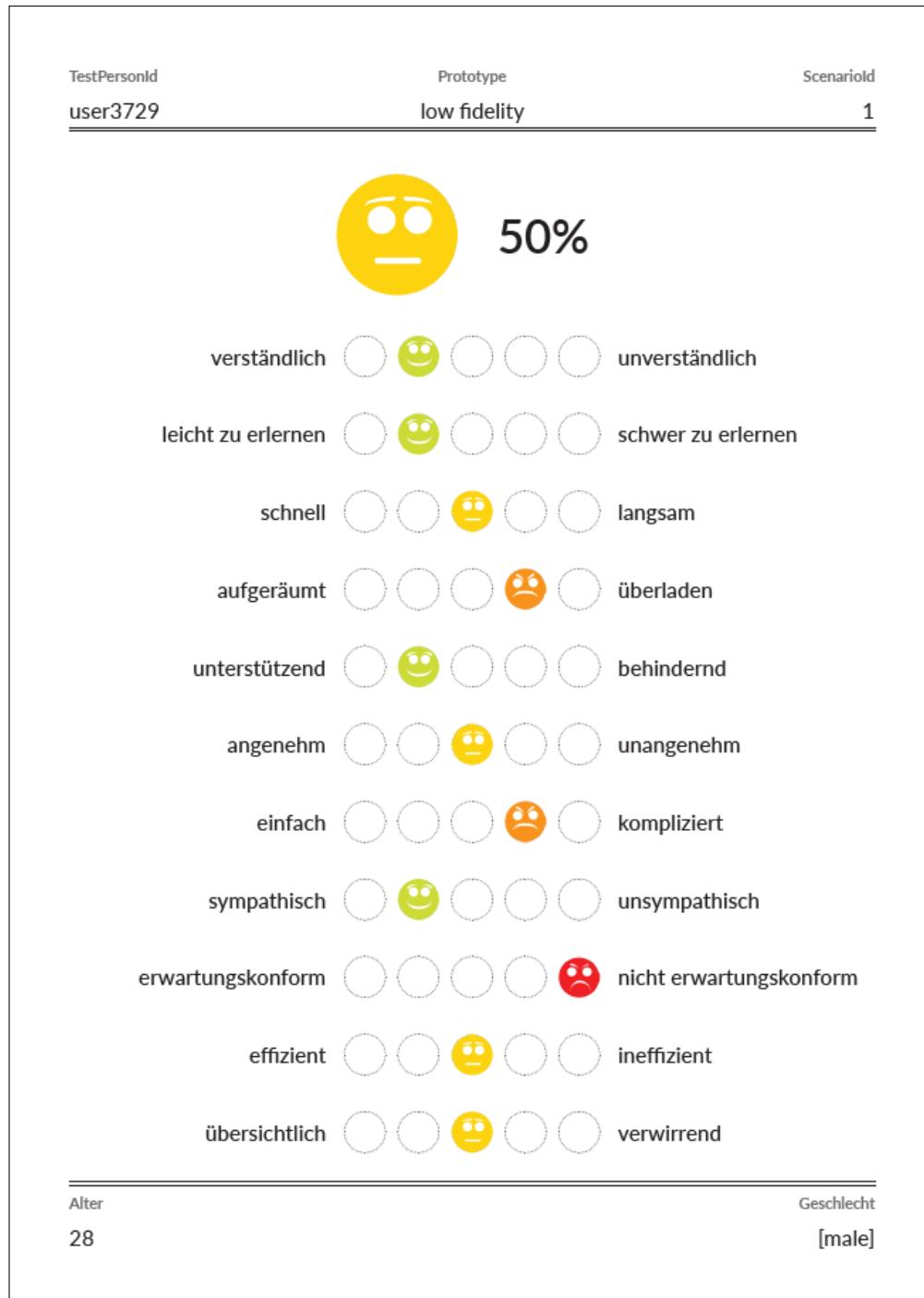


Abbildung A.23: Ergebnisse zum Fragebogen von user3729



A Anhang

TestPersonId	Prototype	Scenarioid
user3729	low fidelity	1
Anmerkungen		
"ich finde es interessant, dass er sagt dass er denkt"		
Positiv: - Kalender leicht verstanden - Interaktion ist schnell erlernbar		
Negativ: - Erwartet: Text als Rückgabe - erwartet Textnachrichten vom Chatbot - Auto-Nachrichten irritierend - Auto-Element zu schnell - Farben		
Alter	Geschlecht	
28	[male]	

Abbildung A.24: Notizen zum Fragebogen von user3729

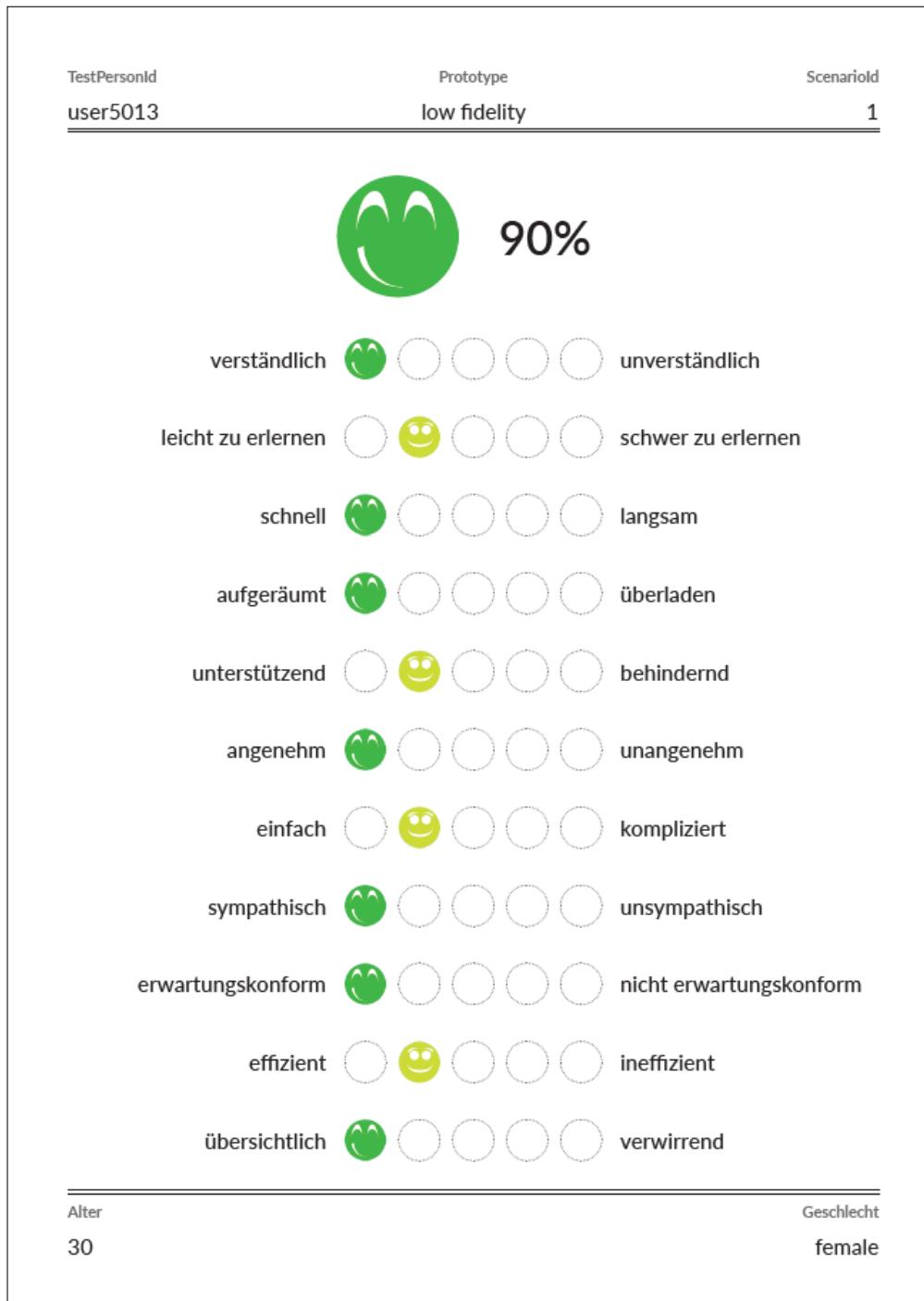


Abbildung A.25: Ergebnisse zum Fragebogen von user5013



A Anhang

TestPersonId user5013	Prototype low fidelity	Scenarioid 1
<p>Anmerkungen</p> <p>"Coole Sache!"</p> <p>-> Daten komprimiert in Steno-Stil</p> <p>Positiv:</p> <ul style="list-style-type: none">- UI-Elemente einfach & schnell- Eingabe des Datums über relatives Datum (e.g. morgen vormittag) <p>Negativ:</p> <ul style="list-style-type: none">- Was bedeuten die Farben? (Raumicons) <p>Anregung:</p> <ul style="list-style-type: none">- Ausstattung des Raumes abfragen (Projektor, Dockingstation, Whiteboard)- Mouse-Over-PopUps mit Infos zum Raum bei Lageplan <p>Klick-dummy:</p> <ul style="list-style-type: none">- Raumausstattung positiv		
<hr/>		
Alter 30	Geschlecht female	

Abbildung A.26: Notizen zum Fragebogen von user5013

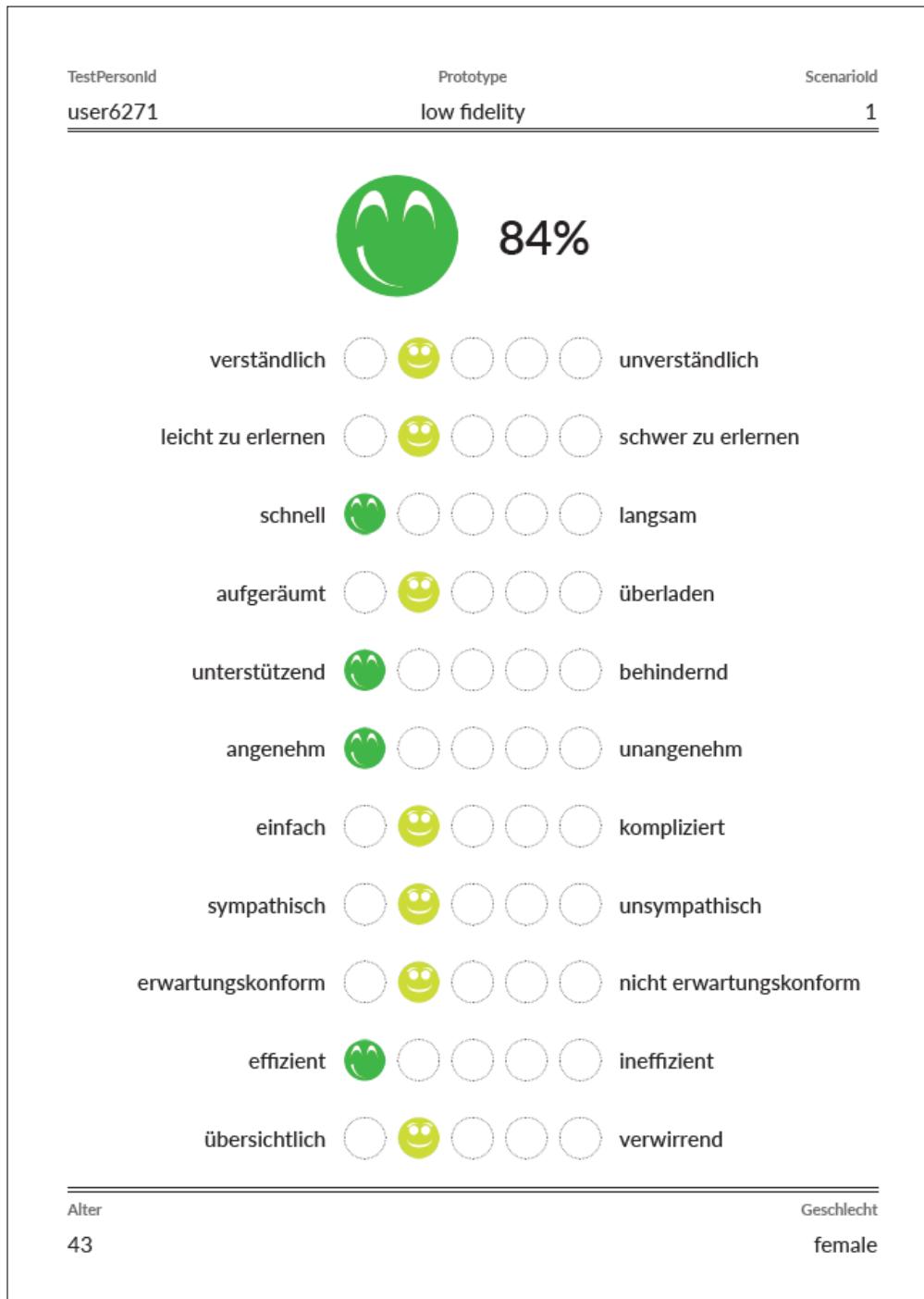


Abbildung A.27: Ergebnisse zum Fragebogen von user6271



A Anhang

TestPersonId	Prototype	Scenarioid
user6271	low fidelity	1
<hr/>		
Anmerkungen		
<p>“Das ist Klasse! Finde ich gut.” (Lageplan) “Cool, das finde ich gut.” (Chatbot is thinking)</p>		
<p>Positiv:</p> <ul style="list-style-type: none">- Lagepan: cool / klasse- CollectionView: direkt und unmissverständlich bedienbar.		
<p>Negativ:</p> <ul style="list-style-type: none">- keine Skala am Slider- Icon zeigt Viertelstunde und man kann nur halbstündig auswählen		
<p>Anregung:</p> <ul style="list-style-type: none">- Buchungsbestätigung per EMail / Eintrag im Kalender- kein Slider - Zeit (auch) mit inkrement/dekrement einstellbar- Info: Wifi-Gastzugang / Moderatorenkoffer / Metaplanwand / Flipchart- Bot-Character- Anzahl Personen als Info		
<hr/>		
Alter	Geschlecht	
43	female	

Abbildung A.28: Notizen zum Fragebogen von user6271

A.9 Feedback zum User Test

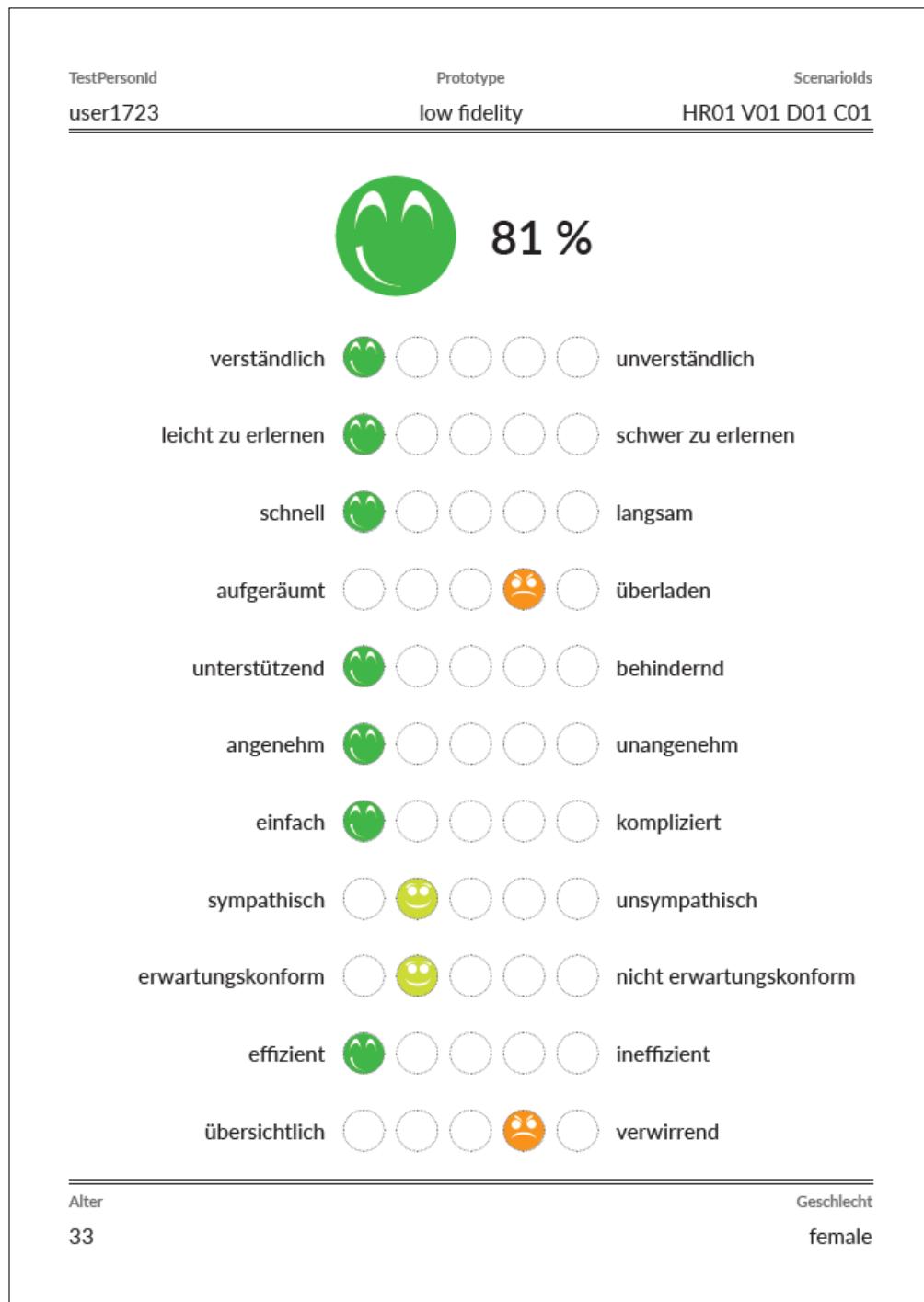


Abbildung A.29: Ergebnisse zum Fragebogen von user1723



A Anhang

TestPersonId	Prototype	Scenarioids
user1723	low fidelity	HR01 V01 D01 C01
Anmerkungen		
CalendarView: "ist relativ simpel"		
Buchungsbestätigung: "Ja, prima. Das ist ein schönes Ergebnis." ... [später] "Die ist ja immer gut."		
"Das fand ich richtig gut" -> Chatbot versteht natürlichen Ausdruck "für eine Stunde"		
"Das hier noch ne Bestätigung kommt, finde ich sehr gut." (Confirm-Message -> "ich bin ein Mensch der Bestätigungen mag.")		
Dates-CollectionView: "Finde ich sehr einfach gehalten. In dem Sinne auch gut. Ist ja nur ein Klick."		
Vor-/Nachbereitungszeit (Einstellung über 2 Slider) nach Erläuterung: "Das finde ich ganz gut"		
"Mir taugt das ganz gut." (Zusammenfassend)		
Wunsch / Verbesserungsanmerkung		
Wunsch: Möglichkeit den letzten Schritt zu verwerfen.		
Wertungsanmerkungen		
verständlich-unverständlich-Wertung		
"Er hat verstanden was ich wollte, sofern er es konnte."		
leicht zu erlernen-schwierig zu erlernen		
"Wenn man's einmal gemacht hat [...] eher angefangen zu variieren was man so als Daten eingeben kann."		
sympathisch - unsympathisch		
"Spricht mich mit meinem Namen an."		
"Möchte mir helfen."		
"Für ein Tool das mit mir interagiert finde ich's schon sympathisch"		
"Nett fand ich das 'Entschuldigung, das muss ich noch lernen.'"		
Alter	Geschlecht	
33	female	

Abbildung A.30: Notizen zum Fragebogen von user1723

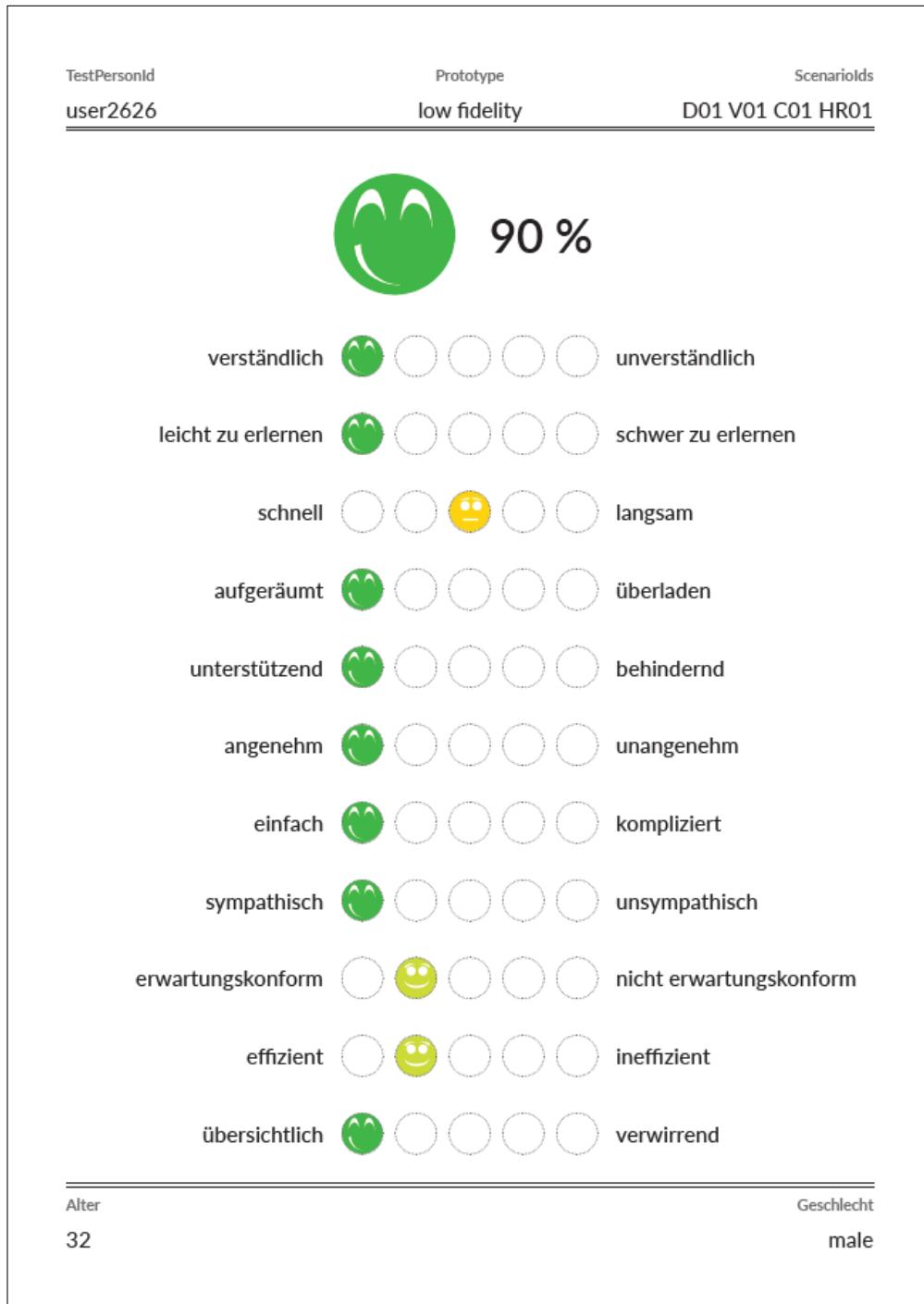


Abbildung A.31: Ergebnisse zum Fragebogen von user2626



A Anhang

TestPersonId	Prototype	Scenarios
user2626	low fidelity	D01 V01 C01 HR01

Anmerkungen

Buchungsbestätigung: "Cool"
"Dass nochmal so 'ne Bestätigung kommt, ist eigentlich ziemlich cool!"

Datumsbuttons (Szenarium C01): "Gut"

Positiv: Anzeige der Räume mit Farbe, Anordnung der Stühle und Tische

"Ich denke, dass er mich versteht."

Positiv: Alternativen werden aufgezeigt

Der konsistente Workflow – dann irgendwann Titel, Bestätigen, Zertifikat – wurde positiv hervorgehoben.

Wunsch / Verbesserungsanmerkung

Wünsche / Verbesserungswünsche
Koordiniert Termine mit mehreren Teilnehmern, vielleicht mit Prozentangabe 80% der angeforderten Teilnehmer haben an dem Termin Zeit / oder Einstellungen: finde einen Termin, an dem mindestens 80% der Teilnehmer Zeit haben

Wertungsanmerkungen

schnell-langsam-Wertung

Proband hob hervor, dass er unterbrochen wurde.

erwartungskonform - nicht erwartungskonform
- eigentlich passt es soweit

effizient - ineffizient
- Leute mit einladen, würde es effizienter machen

Alter Geschlecht
32 male

Abbildung A.32: Notizen zum Fragebogen von user2626

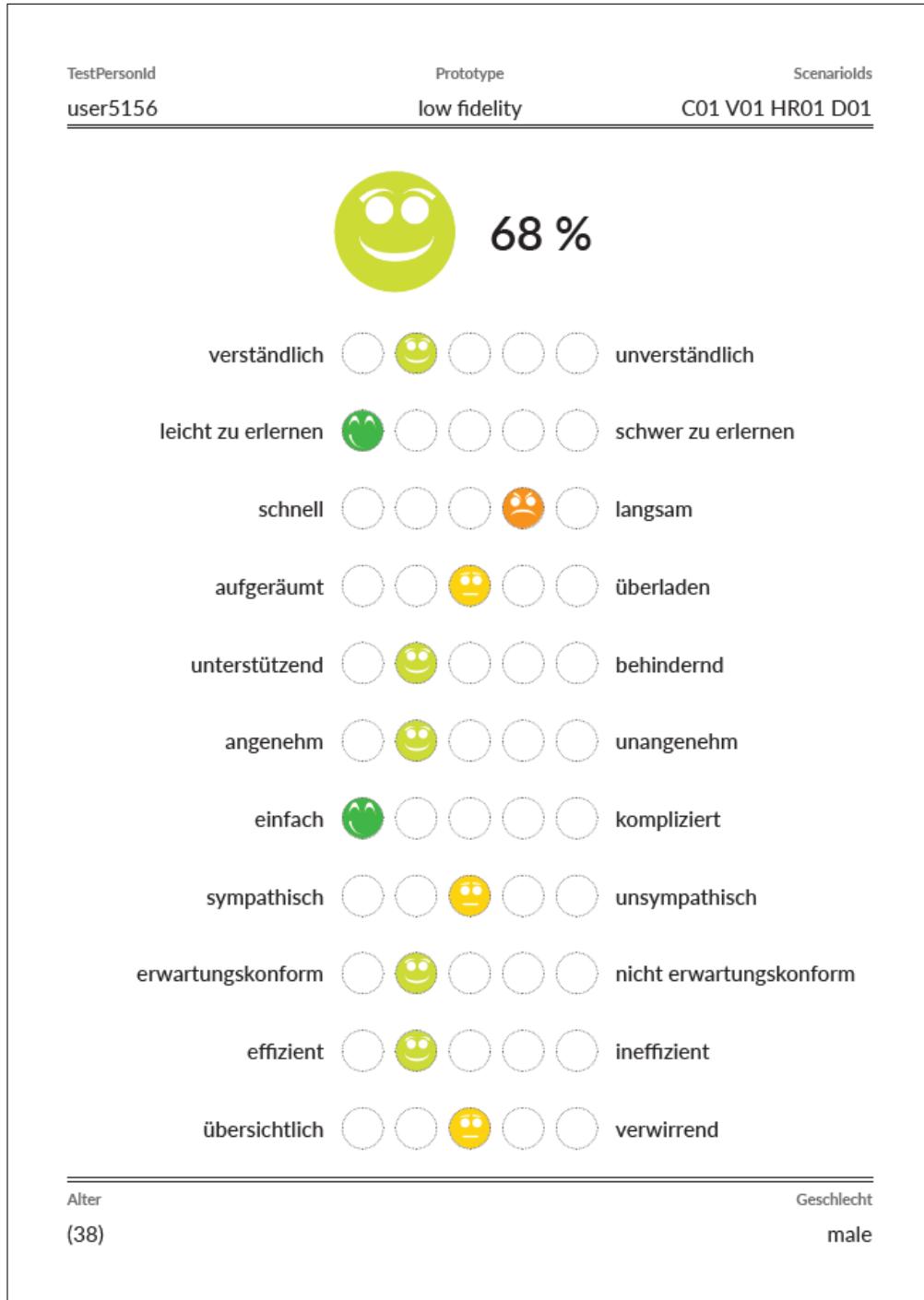


Abbildung A.33: Ergebnisse zum Fragebogen von user5156



A Anhang

TestPersonId	Prototype	Scenarioids
user5156	low fidelity	C01 V01 HR01 D01
Anmerkungen		
"Des ist jetzt wieder was tolles." (Buchungsbestätigung)		
Raum-Abbildungen als Buttons sind positiv aufgefallen (CollectionView) + Stuhlfarben einleuchtend		
"Ja, cool, verrückt." "Das ist schön." (Buchungsabschluss / Buchungsbestätigung)		
Verwendung von UI-Element für kleinere Eingaben (Startzeit / Dauer / etc.) als zu umständlich empfunden.		
Bei Einstellung der Vor- und Nachbereitungszeit: Proband wünscht sicher eher zuerst Nachfrage nach Vor-/Nachbereitungszeit, bevor Regler präsentiert werden.		
Proband wünscht sich mehr "Smalltalk" mit dem Chatbot (u.a. Name, Charakter, Reaktion auf Bedanken)		
Wertungsanmerkungen		
<i>verständlich-unverständlich-Wertung</i> Proband merkte an, dass Regler anfänglich nicht sofort verstanden werden.		
<i>schnell-langsam-Wertung</i> Proband hob hervor, dass teilweise zu viele Zwischenfragen (-> Fehler im Testsystem, Konzept sah weniger Zwischenfragen vor)		
<i>aufgeräumt-überladen-Wertung</i> Proband findet UI überladen, allerdings Prototypenstatus		
<i>unterstützend-behindernd-Wertung</i> Proband erwarten, dass Personen eingeladen werden können		
<hr/>		
Alter	(38)	Geschlecht
		male

Abbildung A.34: Notizen zum Fragebogen von user5156

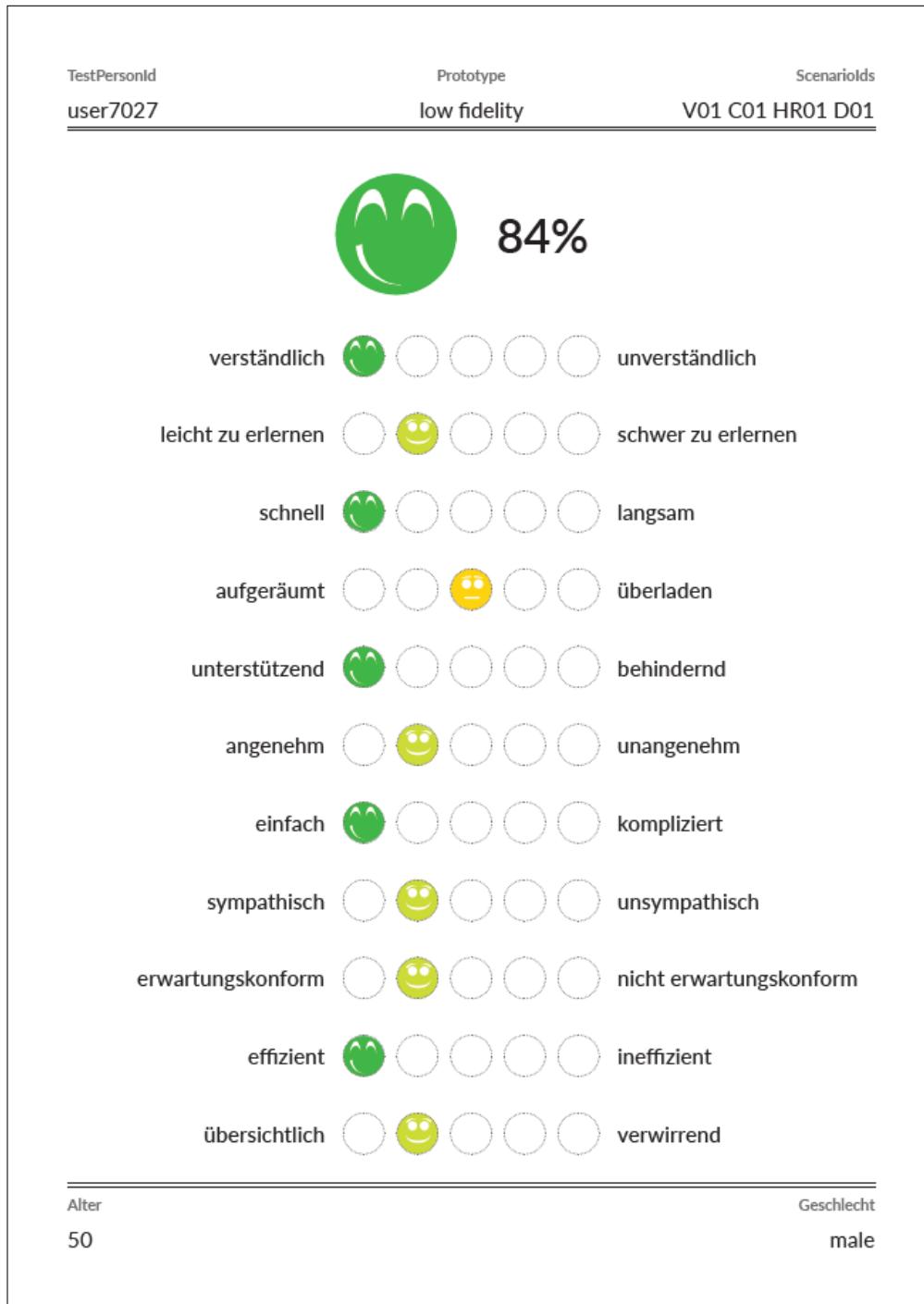


Abbildung A.35: Ergebnisse zum Fragebogen von user7027



A Anhang

TestPersonId	Prototype	Scenarios
user7027	low fidelity	V01 C01 HR01 D01
Anmerkungen		
"Bin erst einmal angetan, dass ich hier etwas anklicken kann." (CalendarView)		
"Ging des eigentlich sehr schnell, das hätte ich eigentlich schlechter erwartet." (Workflow)		
"Das hat mich überrascht, wie gut das geht." (Effizienz)		
Proband bemängelt, dass Chatbot bei Korrekturen nicht explizit bestätigt, dass er es verstanden hat.		
"Provide just enough info!" - Formulierung bei durch das System getroffene Vorauswahl. ("Hier die Räume ... anstelle von "Hier einige Räume ...")		
Proband wünscht eine ausformulierte Antwort für die auto-generierten Antworten.		
Proband empfand die Verwendung von UI-Elementen teilweise zu umfangreich; lenkte von der Möglichkeit der Texteingabe ab, obwohl diese teilweise als flüssiger empfunden worden wäre.		
Proband hätte sich mehr Persönlichkeit (Name, Avatar) vermutet.		
Wertungsanmerkungen		
<i>leicht zu erlernen-schwer zu erlernen-Wertung</i> UI-Elemente explorativ erfassbar		
<i>schnell-langsam-Wertung</i> erstaunlich schnell		
Alter	Geschlecht	
50	male	

Abbildung A.36: Notizen zum Fragebogen von user7027

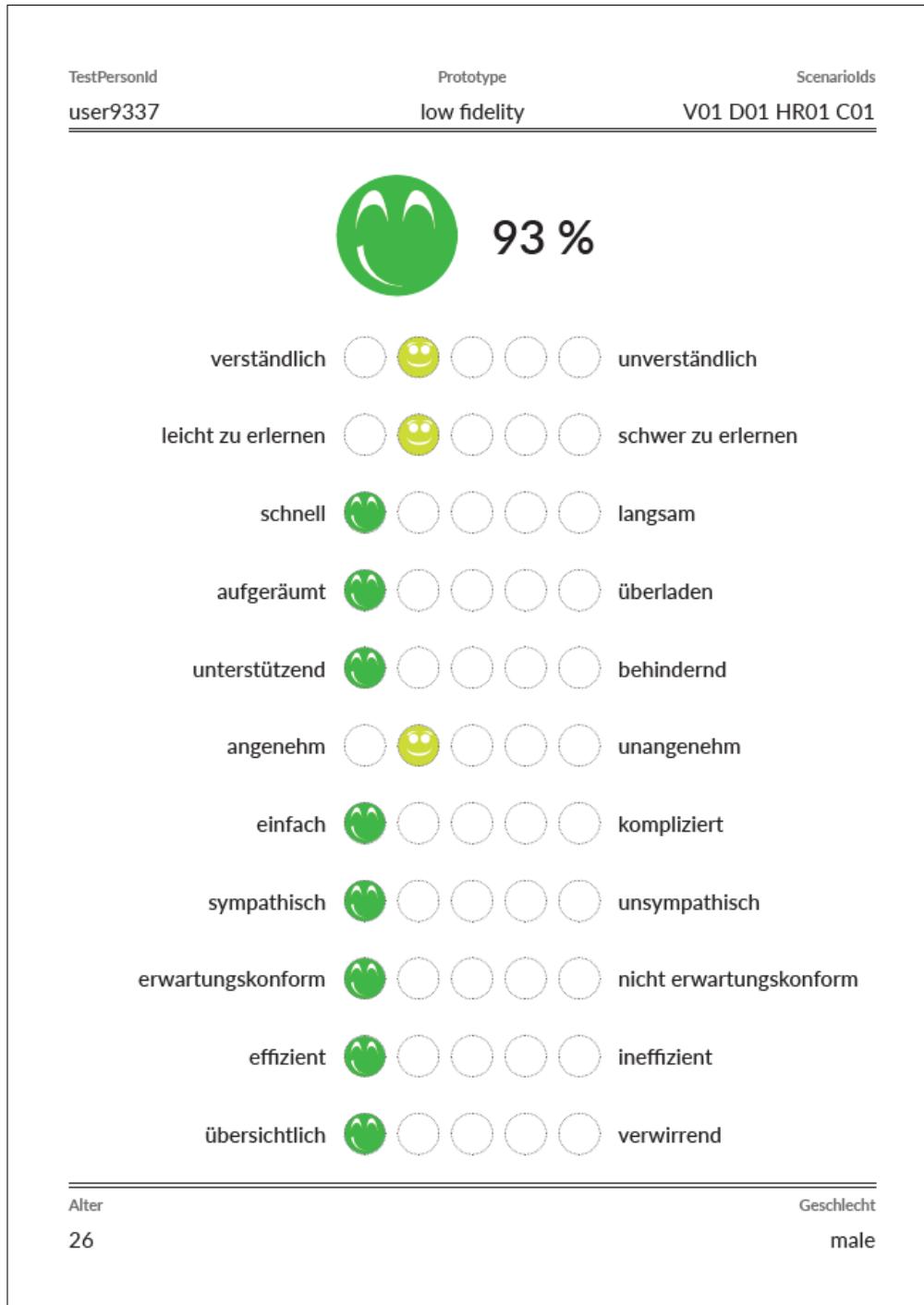


Abbildung A.37: Ergebnisse zum Fragebogen von user9337



A Anhang

TestPersonId	Prototype	Scenarioids
user9337	low fidelity	V01 D01 HR01 C01
Anmerkungen		
"Also das finde ich ziemlich cool." (CalendarView)		
"Ja, alles klar. Cool!" (Bestätigung)		
"Ich finde es zum Beispiel auch ganz cool, dass das Bild auch schon die Form von dem Tisch hat." (DetailView + Bestätigung)		
"Ja, cool" -> Autogenerierter Titel für die Buchung ("Meeting für <Name des Probanden>")		
HR-Termin-Typ-Auswahl: "Finde ich cool." Buchungsbestätigung: "Jup, schaut doch gut aus!" (Positive Äußerung kann sich auch auf 'schnell' Anchluss des Szenarios beziehen.)		
"Ah ja, cool. Das schaut jetzt besser aus." (nachdem Proband bessere Formulierung für seine Anfrage gefunden hatte)		
Wunsch / Verbesserungsanmerkung Findet Ablehnen-Button (rotes X) bei optionalem Titel zu hart, an dieser Stelle lieber ein "Weiter". (-> Erzeugt Unsicherheit, on dadurch der gesamte Buchungsprozess verworfen wird.) Allerdings für Abbrechen-Button in Bestätigungsdialog als passend empfunden.		
Wertungsanmerkungen		
<i>verständlich-unverständlich-Wertung</i> "Ingesamt fand ich die Antworten eigentlich ganz cool. Wurde teilweise sogar positiv überrascht mit den Graphiken, usw."		
<i>leicht zu erlernen - schwer zu erlernen</i> "Man bekommt ja den Dialog zurück, wenn was fehlt."		
<i>angenehm - unangenehm</i> "Ist immer erst einmal komisch, aber insgesamt eher angenehm."		
<i>erwartungskonform - nicht erwartungskonform</i> "Teilweise wurde ich sogar übertrffen."		
<hr/> Alter		
26		
<hr/> Geschlecht		
male		

Abbildung A.38: Notizen zum Fragebogen von user9337

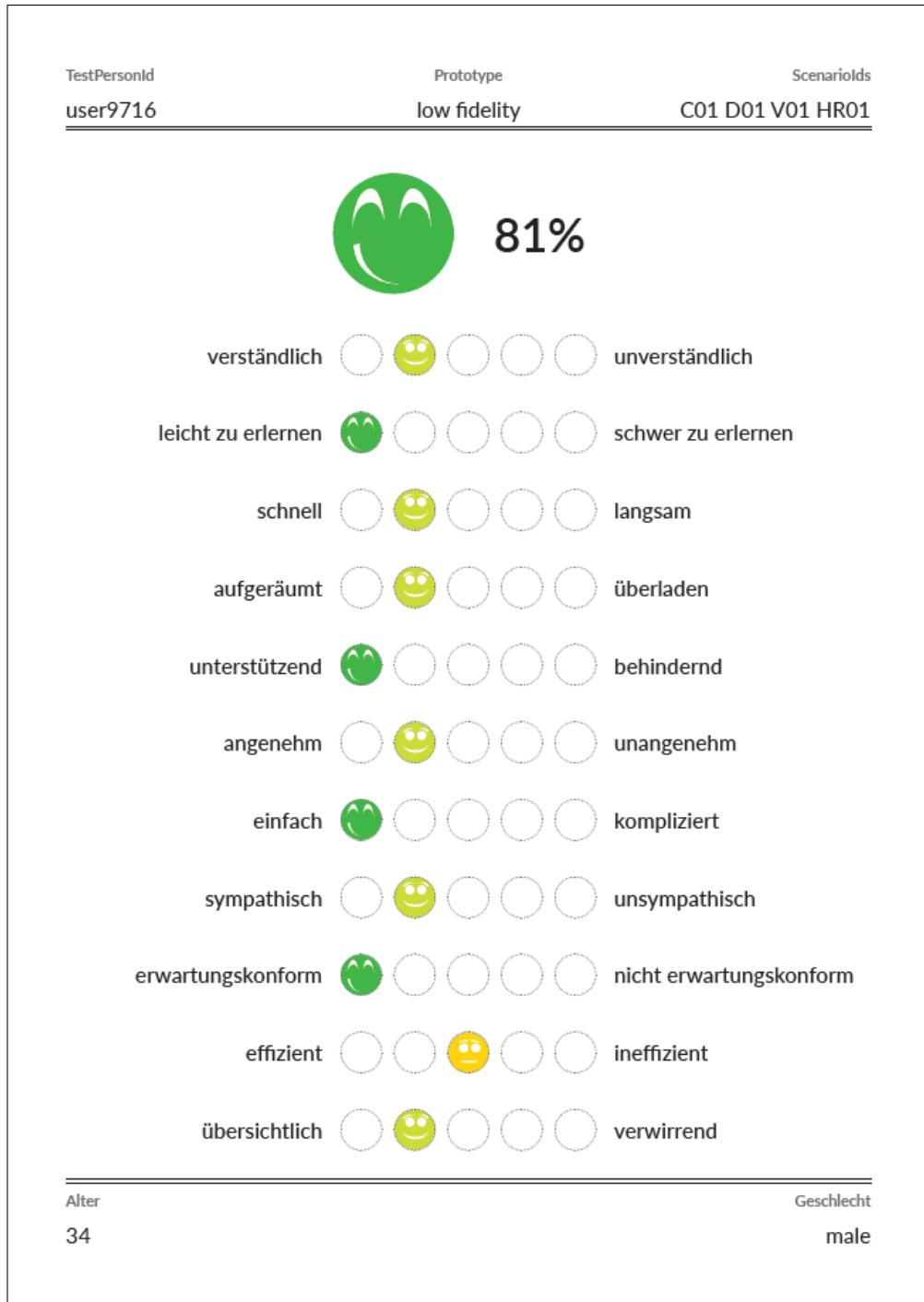


Abbildung A.39: Ergebnisse zum Fragebogen von user9716



A Anhang

TestPersonId	Prototype	Scenarioids
user9716	low fidelity	C01 D01 V01 HR01
Anmerkungen		
"Allerdings ist das natürlich auch ne schöne Lösung." (Vor- / Nachbereitungszeiten ausgewiesen)		
"Wenn man alles einmal benutzt hat, ist es einfacher." (Bedienung -> UI-Elemente)		
"Passt! Des ging doch flott. Des war gut."		
"Ok, cool, ja das ging doch auch schnell."		
"Prinzipiell ist es super" (zum Design, Kritik am Visual Design, aber Bestätigung des Konzepts)		
"Oh, ich werde sogar angesprochen."		
"Chatbot is thinking" is Proband zu menschlich (schlägt alternativ "Chatbot is typing" vor)		
Keine Reaktion auf Begrüßung bzw. Smalltalk.		
Proband würde sich eine Einführung wünschen (Verstehen der UIElemente, dass diese interaktiv sind)		
Feedback bei Korrektur erwünscht.		
Weitergehende Personalisierung für Sympathie noch wünschenswert (Avatar, Charakter)		
Wertungsanmerkungen		
<i>effizient-ineffizient-Wertung</i> Da keine Einladung von Personen und Abgleich mit deren Kalendern integriert ist, fällt diese Bewertung schwer.		
Alter	Geschlecht	
34	male	

Abbildung A.40: Notizen zum Fragebogen von user9716



A.10 CD Verzeichnisstruktur

- └ **Latex-Files/** ⇒ *editierbare L^AT_EX Dateien*
 - └ **bilder/** ⇒ *Alle verwendeten Bilder*
 - └ **hauptkapitel/** ⇒ *Fünf Hauptkapitel*
 - └ **literatur/** ⇒ *Bibliotheksdatei und Zitierstil*
 - └ **nebenkapitel/** ⇒ *Deckblatt, Abstract, Anhang, ...*
 - └ config.tex
 - └ main.tex
- |
- └ **Literatur/**
 - └ **Android/**
 - └ **CUI/**
 - └ **Server-Technologien/**
 - └ **Usability/**
- |
- └ **Quelltexte/** ⇒
 - └ **AndroidApp/** ⇒ *Android Applikation*
 - └ **Dialogflow/** ⇒ *Dialogflow Agent*
 - └ **PrototypingTool/** ⇒ *JSON-Files der Szenarien*
 - └ **Server/** ⇒ *Node.js Webservice*
- |
- └ **Weiteres/**
 - └ **AppVideos/** ⇒ *Beispielvideos zur Android Applikation*
 - └ **Dokumente/** ⇒ *Exposee, Masterarbeit, Praesentation*
- |