

Evaluierung von Tools zum Auffinden von Undefined Behavior

T3000

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Christoph Böhringer

21.06.2021

Matrikelnummer, Kurs:	3275565, TINF18-IN
Ausbildungsfirma:	Mitutoyo CTL Germany GmbH
Betreuer:	Dipl.-Inform. (FH) Thomas Weller

Erklärung

Ich versichere hiermit, dass ich meine Thesis mit dem Thema „Entwicklung eines softwaregestützten kontextabhängigen Kommunikationsassistenten “ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

Name

Abstract

In einem Softwareprodukt trat bei der Umstellung von 32 Bit auf 64 Bit ein Fehler auf, dessen Ursache sich vermutlich auf Undefined Behavior von C++ zurückführen lässt. Diese Arbeit soll den Nachweis erbringen oder widerlegen, dass Undefined Behavior die Ursache für den Fehler war. Optional kann eine mögliche Erklärung gesucht werden, warum der betroffene Code in der 32 Bit Version keinen Fehler verursacht hat.

Das betroffene Projekt wurde vor der Umstellung mehrere Jahre lang nicht verändert. Es muss daher davon ausgegangen werden, dass sich ähnliche Fehler damals auch an anderen Stellen eingeschlichen haben, jedoch noch nicht bemerkt wurden. Die Arbeit soll untersuchen, ob weitere Fehler vom Typ Undefined Behavior in diesem Projekt vorliegen.

Um Fehler dieser Art auch in anderen Projekten auszuschließen, sollen Tools gesucht und evaluiert werden, mit denen die Fehler automatisiert gefunden und berichtet werden können. Falls kein passendes Tool existiert, soll aufgezeigt werden, mit welchem manuellen Vorgehen solche Stellen erkannt werden können.

Die Tools sollen möglichst mit den bestehenden Entwicklungsumgebungen verwendet werden können und idealerweise alle Arten von Undefined Behavior erkennen.

Hinweis: Da in der Arbeit vermutlich Source Code der Mitutoyo CTL Germany GmbH offengelegt wird, ist die Arbeit unter Verschluss zu halten.

Inhaltsverzeichnis

Abstract	II
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VII
Abkürzungen	VIII
Glossar	IX
1 Einleitung	1
1.1 Problemstellung	2
1.2 Aufgabenstellung	2
2 Small Talk	4
2.1 Was ist Small Talk?	4
2.2 Small Talk führen	5
3 Stand der Technik	7
3.1 Systeme in Betrieb	7
3.2 Datenbankmodelle	9

3.3	Einordnung in den Stand der Technik	12
4	Requirements Engineering	13
4.1	Stakeholder	13
4.2	Vision	14
4.3	Anforderungen	15
4.3.1	Funktionale Anforderungen	16
4.3.2	Nichtfunktionale Anforderungen	17
4.3.3	Priorität der Anforderungen	17
4.3.4	Aufwandsschätzung der Anforderungen	17
4.3.5	User Stories	18
5	Lösungsfindung	20
5.1	Umgebung	20
5.2	Fragenfindung und -bewertung	22
5.2.1	Fragenfindung	22
5.2.2	Fragenbewertung	22
5.3	Allgemeine Hilfestellungen	24
5.4	Datenspeicherung	25
6	Lösungsbewertung	29
6.1	Umgebung	29
6.2	Fragenbewertung	31
6.3	Datenspeicherung	31

7 Umsetzung	33
7.1 Datenbank	33
7.2 Nutzerprofile	34
7.2.1 Tags	36
7.2.2 Gesprächspartner/-innen	38
7.3 Fragen/Hinweise	39
7.3.1 Umgebungsvariablen	39
7.4 Gespräch	40
7.5 Datenschutz	42
7.5.1 Schutz vor Datendiebstahl	43
8 Praxistests	44
9 Zusammenfassung	46
10 Ausblick	48
11 Anhang	50
11.1 Personas	50
11.1.1 Daniel Dacher	51
11.1.2 Susanne Schmid	53
11.1.3 Torsten Tacheles	55

Abbildungsverzeichnis

7.1	Minimierte Struktur der Datenbank	33
7.2	Anmeldung mit E-Mail-Adresse	35
7.3	Aufforderung zur Eingabe des Passworts	35
7.4	Auswahl bevorzugter und gemiedener Tags	37
7.5	Warnung, wenn weniger relevante Fragen hinzugefügt werden	41
7.6	Frage während einer Konversation	42
11.1	Generiertes Porträt von Daniel Dacher [41]	51
11.2	Generiertes Porträt von Susanne Schmid [42]	53
11.3	Generiertes Porträt von Torsten Tacheles [43]	55

Tabellenverzeichnis

4.1	Stakeholder	14
4.2	Funktionale Anforderungen	16
4.3	Nichtfunktionale Anforderungen	17
4.4	Priorität der Anforderungen	17
4.5	Aufwandsschätzung der Anforderungen	17
4.6	User Stories von Daniel Dacher, Susanne Schmid und Torsten Tacheles	19

Abkürzungen

API	Application Programming Interface
APK	Android Package
DBMS	Datenbankmanagementsystem
DSGVO	Datenschutz-Grundverordnung
IP	Internet Protocol
JSON	JavaScript Object Notation
OQL	Object Query Language
OS	Operating System
RAM	Random Access Memory
SDK	Software Development Kit
SQL	Structured Query Language
UI	User Interface
UID	User Identification
URL	Uniform Resource Locator
XML	Extensible Markup Language

Glossar

Abfragesprache	Datenbanksprache zur Informationssuche; je nach Datenbankmodell verschieden
Cloud	Über das Internet verfügbare IT-Infrastruktur, meist zu Speicherzwecken
Container	Hier: Teil der Liste, welches ein einzelnes Objekt/Element enthält
Debugger	Werkzeug zum Finden von Fehlern im Code
HashMap	Speicher für Key-Value-Paare
I/O-Stream	Eingangs-/Ausgangsstrom, wandelt Eingabe/Ausgabe in eine für das/den Programm/Menschen verständliche Sprache um
ListActivity	Android Aktivität, welche eine Liste von Objekten darstellt
Listener	Auf eine bestimmte Aktion registrierte Klasse, welche eine Funktion enthält. Die Funktion wird aufgerufen, sobald die auf den Listener registrierte Aktion ausgeführt wird
NoSQL	Datenbankmodell, welches keinen relationalen (Structured Query Language (SQL)) Ansatz verfolgt

Pop-up	Element der grafischen Oberfläche, welches eingesetzt wird um Zusatzinhalte anzuzeigen (engl. „to pop up“, „plötzlich auftauchen“)
RSS-Reader	Anwendung, welche Daten aus einem RSS-Feed liest
RSS-Feed	Abrufbare Quelle von häufig aktualisierten Informationen, wie Blogs und Nachrichten. Meist eine Zusammenfassung des originalen Inhalts
Smartwatch	Elektronische Armbanduhr, welche mit einem Smartphone verbunden werden kann und wie dieses bedient wird.
User Story	Aus Nutzersicht beschriebene Anforderungen

Kapitel 1

Einleitung

Im alltäglichen Leben kommt man in verschiedenen Situationen mit gar nicht oder nur wenig bekannten Menschen in Kontakt: an der Supermarktkasse, beim Bäcker, oder man trifft eine bekannte Person in der Stadt. In solchen Situationen kann es zu einer kurzen Unterhaltung kommen: dem Small Talk. Weitere mögliche Situationen sind: Gespräche unter Schülern/Studenten, Konversationen an einem Messestand und Diskussionen nach einer Vorlesung.

Auch in der Firma führen Mitarbeiter beispielsweise an der Kaffeemaschine und in der Frühstücks- und Mittagspause kleine Unterhaltungen. Hier kennen sich die Mitarbeiter untereinander und wissen in den meisten Fällen einige Interessen des Gesprächspartners. Schwieriger wird es, wenn man das Gegenüber nicht kennt, wie zum Beispiel auf einer Messe oder einer Fortbildung.

Die meisten Teilnehmer einer solchen Fortbildung kennen nur ein bis zwei weitere Teilnehmer. Um nicht durchgehend alleine zu sein muss also Kontakt zu fremden Personen aufgebaut werden. Dabei sollte auf die Interessen des Gegenübers und eventuelle Tabu- und Konfliktthemen geachtet werden.

1.1 Problemstellung

Konkret ist, für kommunikationsschwache Menschen, in solchen Situationen der Einstieg in das Gespräch schwierig. Aus einem geringen Informationspool über das Gegenüber gilt es einen passenden Einstieg in ein Gespräch zu finden. Standardfragen, wie beispielsweise eine Bemerkung zum Wetter, können in Betracht gezogen werden, haben aber, aus gesellschaftlicher Sicht, einen Stellenwert als unkreativen Einstieg und führen nicht zwingend zu einem angeregten Gespräch.

In einem laufenden Gespräch, kann es kommunikationsschwachen Menschen aufgrund fehlender Übung an weiterführenden Fragen, Anregungen und Themen mangeln. Die Konversation gerät ins Stocken und kann abrupt abbrechen.

Das Reden mit fremden Menschen ist für Menschen mit einer Kommunikationsschwäche oft eine Stresssituation. Stress kann dazu führen, dass das Gehirn blockiert und nicht mehr leistungsfähig ist [1, S. 79 f.]. Es kommt zu einem Blackout. Gesprächsthemen und Fragen an das Gegenüber verschwinden aus dem Gedächtnis und die Konversation wird einseitig weitergeführt oder abgebrochen.

Menschliche Hilfestellungen, beispielsweise in Form eines Mentors, sind nur vor einem Gespräch verfügbar. Eine fortlaufende Unterstützung während der Konversation ist durch eine Begleitperson realisierbar, dadurch kann allerdings auch ein Dialog zwischen der Begleitperson und dem Gesprächspartner entstehen, wobei die Hilfe benötigende Person außen vor gelassen wird.

1.2 Aufgabenstellung

In dieser Praxisarbeit soll eine Softwarelösung entwickelt werden, welche dem Benutzer Fragen bereit stellt, um eine Konversation zu beginnen.

Die Fragen sollen aufgrund von Umgebungsvariablen ausgewählt werden, damit ein passender Einstieg in ein Gespräch ermöglicht wird. Zudem können Anwender Profile von sich selbst und ihren Gegenübern erstellen. Die Einstellungen der Profile sollen auch in die Auswahl der Fragen miteinbezogen werden.

Die Anwendung soll leicht zu bedienen sein und über ein übersichtliches User Interface (UI) verfügen. Dies soll eine Nutzung während einer Konversation ermöglichen, ohne dass der Redefluss stark unterbrochen wird.

Die Software soll ihrem Anwender im Idealfall genug theoretische Hilfestellungen bieten, damit mögliche Anwender über die Zeit eigenständig Gespräche beginnen und diese auch weiterführen können.

Kapitel 2

Small Talk

2.1 Was ist Small Talk?

Als Small Talk (engl. *small* „unbedeutend, klein“ und *to talk* „sich unterhalten“) wird ein spontanes, zufällig entstandenes, lockeres Gespräch bezeichnet. Die häufigsten Themen beziehen sich dabei meist auf das Privatleben der Involvierten, oder auf das Geschehen um das Gespräch. Der Ton des Gesprächs ist informell.[2]

Im Alltag kann es in verschiedenen Situationen zu einem Small Talk kommen:

- Auf dem Weg zur Arbeit
- In der Mittagspause
- Beim Einkaufen
- Beim Spaziergehen
- An der Supermarktkasse
- In öffentlichen Verkehrsmitteln
- Auf einer Party
- ...

Small Talk kann in verschiedenen Situationen, beispielsweise in einem Bewerbungsgespräch, als „Eisbrecher“ verwendet werden. Das Führen von Small Talk hilft dabei Interesse am Gegenüber zu zeigen. In einem beruflichen Umfeld können somit Beziehungen zu Kollegen/-innen geknüpft werden.

Die Auswahl der Gesprächsthemen orientiert sich dabei daran, wie gut man das Gegenüber bereits kennt. Handelt es sich um eine/-n Freund/-in, so kann der Small Talk auch übersprungen werden und das Gespräch kann mit einem bevorzugten Thema begonnen werden. [2]

2.2 Small Talk führen

Typische Fragen im Small Talk, wie

- Wie findest du das Wetter?
- Wie geht es dir/deiner Familie?

können als Gesprächseinstieg dienen, sind aber nicht weiterführend und können somit die Konversation zum Stehen bringen.

Offene Fragen, also Fragen welche nicht mit „Ja“ oder „Nein“ beantwortet werden können, erfordern eine detailliertere Antwort vom Gegenüber und fördern somit den Gesprächsfluss. Beispiele für Anfänge einer offenen Frage sind „Warum“ oder „Was“.

Gemeinsamkeiten, wie der Ort des Gespräches, stellen eine gute Grundlage für Themen dar. Durch diese wird eine vertraute Atmosphäre geschaffen. [3, S. 104]

Während des Small Talks bietet es sich an, öfters das Thema zu wechseln. Small Talk ist ein unbeschwertes Gespräch und wird durch Themenwechsel abwechslungsreich, leicht und locker. [3, S. 108]

Neben der allgemeinen Gesprächsführung spielen, für eine gelungene Konversation, auch Mimik, Gestik und Haltung eine Rolle.

Bei der Mimik ist hierbei auf Augenkontakt zu achten, schüchterne Menschen können dabei auch auf den Mund schauen. Des Weiteren haben ein Lächeln und das Vermeiden von Grimassen einen positiven Einfluss auf das Gespräch.

Das Verwenden der Arme und Hände zur Gestikulation in einem Gespräch machen dieses lebendiger. Eine belebte Körpersprache zeigt Offenheit und Extraversion. Versteckte Handflächen und verschränkte Arme deuten hingegen auf Verslossenheit hin. [3, S. 119]

Eine starke und selbstbewusste Haltung zieht Aufmerksamkeit auf den/die Redner/-in. Dies kann auch das Selbstbewusstsein des/der Redners/-in steigern, was schüchternen Menschen bei der Gesprächsführung helfen kann. Zu einer selbstbewussten Körperhaltung gehören: [4]

- Fester Stand: beide Beine fest und ungefähr schulterbreit auf dem Boden
- Aufrechte Schultern
- Gerader Rücken
- Nach vorne gerichteter Kopf, nicht nach unten schauen

Wenn das Gegenüber eine zurückhaltende Person ist, sollte eine zu selbstbewusste Haltung vermieden werden, da sich das Gegenüber sonst unter Druck gesetzt fühlen kann und in der Konversation somit in die Defensive gerät.

Kapitel 3

Stand der Technik

3.1 Systeme in Betrieb

Zur Verbesserung der eigenen Kommunikations- und Small-Talk-Fähigkeiten, existieren bereits einige theoretische (sowohl digital als auch gedruckt), menschliche und anderweitige Hilfestellungen:

- **Menschliche Hilfestellungen**

Menschen mit Kommunikationsschwäche können sich einen Coach zurate ziehen, welcher ihnen die benötigten Kenntnisse vermittelt und eventuell in Praxiseinsätzen vertieft.

Ähnlich zu einem Coach können auch Seminare besucht werden, welche durch eine höhere Teilnehmerzahl mehr Praxisnähe bieten.

Als weitere menschliche Unterstützung bietet sich das Hinzunehmen einer Begleitperson, wie beispielsweise einem/r Freund/in an. Sie kann beim Einstieg in ein Gespräch zur Hilfe genommen werden oder das gesamte Gespräch über eine moderierende Rolle einnehmen.

- **Theoretische Hilfestellungen**

Es können auch Internetportale oder Bücher als Quellen für Tipps und Ratschläge verwendet werden. So bietet beispielsweise „Kommunikation Lernen“ [5] verschiedene Blogbeiträge mit Tipps, Methoden und Checklisten. Die Beiträge dienen als

Ratgeber und können Anregungen für zukünftige Gesprächssituationen liefern. Ähnliche Hilfestellungen kann man auch aus Büchern beziehen. So haben sich Psychologen, wie zum Beispiel Frank Naumann in seinem Buch „Die Kunst Des Smalltalk“ [6], mit der Thematik des lockeren Gesprächs befasst und Hilfestellungen bereit gestellt.

- **Weitere Hilfestellungen**

„Im Verlauf der Konversation“ [Abstract] deutet darauf hin, dass der Assistent mobil sein soll. Daraus ergeben sich drei Gerätearten, auf welchen die Software laufen kann: Laptops, mobile Endgeräte und Smartwatches. Im mobilen Sektor bestehen bereits Möglichkeiten, um die eigenen Small Talk-Fähigkeiten zu verbessern. Die folgenden Apps implementieren Lösungsmöglichkeiten zu einzelnen Aspekten der in Kapitel 1.1 beschriebenen Problematik.

- „Topicks - Gesprächsthemen und mehr!“ [7] stellt Nutzern Gesprächsstoff für verschiedene Themen bereit. Fragen und Anregungen können zufällig oder über verschiedene Kategorien (Technologie, Kunst, Beruf, Sport, Musik, Filme und Serien, ...) ausgewählt werden. Zu bereits bestehenden Kategorien können auch weitere Themen hinzugefügt werden, diese sind aber nur lokal verfügbar.
- Mit der App „Beyond Small Talk“ [8] werden Anwendern Fragen vorgeschlagen. Die Fragen sollen ein tiefergehendes Gespräch aufbauen, welches nicht auf der oberflächlichen Ebene des Small Talks bleibt. Die App wurde von einem Psychologieprofessor entwickelt und soll Beziehungen zu Familienmitgliedern, Freunden und anderen Bekannten stärken.
- „Trending - All in one app“ [9] bietet einen Überblick der aktuellen Themen im Internet. Mit Hilfe von Google Trends [10] werden die am stärksten steigenden Suchanfragen mit Bildern und Artikeln bereitgestellt. Des Weiteren werden die neuesten „top posts“ von Reddit, aktuelle Tweets und Hashtags, virale Youtube Videos und die meistgelesenen Wikipedia-Artikel angezeigt. Das durch die App bereitgestellte Informationsangebot bietet eine Grundlage für Gesprächsthemen und Gesprächseinstiege.
- Eine weitere App, zur Bereitstellung von Informationen, mit welchen man

Small Talk betreiben kann ist „small talk“ [11]. Die App präsentiert täglich sechs Themengebiete: Sprüche zum Nachdenken, lustige Fakten, Witze, Wort des Tages, historische Ereignisse des aktuellen Tages und ein Tipp. Des weiteren stehen auch die Beiträge von vorherigen Tagen zur Verfügung. Mit der Verwendung dieser App können Themen und „fun facts“ zur Verwendung im Small Talk angehäuft werden.

Des weiteren stellen Apps wie „Nwsty“ [12] aktuelle Schlagzeilen bereit und „Clio“ [13] zeigt historische Stätten in der Umgebung und wie man den Weg zu diesen findet. Diese Apps bieten aber weniger Hilfestellungen für den Small Talk als die oben vorgestellten Anwendungen.

Es besteht auch der spezielle Fall des Datings, für welchen mehrere Lösungsmöglichkeiten, wie Tinder, Lovoo, Parship, Badoo und ähnliche Anwendungen, vorhanden sind. Dieser Aspekt wird gezieht nicht behandelt, da die Partnersuche nicht Teil der Ziele der zu entwickelnden Software ist.

3.2 Datenbankmodelle

Da das Erstellen und Speichern von benutzerdefinierten Profilen und Fragen eine Anforderung an diese Arbeit ist, bietet es sich an ein System zur Speicherung und Verwaltung der Daten zu verwenden. Dies kann durch das Schreiben der Daten in eine Textdatei, das Erstellen und Serialisieren von Objekten, Speicherung der Daten in einem Array oder durch die Verwendung eines Datenbankmanagementsystem (DBMS) umgesetzt werden.

- Daten können hartkodiert in Arrays oder Listen gespeichert werden. Das bedeutet, dass die Daten im Quelltext als Konstanten vorhanden sind.
- Beim Schreiben in Textdateien werden die Daten in Strings umgewandelt und beispielsweise durch Kommata, Tabulatorzeichen und Absätze getrennt. Das Trennen der Daten ermöglicht, dass diese auch wieder ausgelesen und zugeordnet werden können.

- Serialisierung wandelt Objekte in Bytefolgen um, welche in eine Datei geschrieben werden. Um das ursprüngliche Objekt zu erhalten werden die Bytes solange deserialisiert, bis daraus ein Objekt entsteht.

Datenbanksysteme dienen der elektronischen Datenverwaltung. Sie bestehen aus den zu verwaltenden Daten und der Verwaltungssoftware, dem DBMS. Das DBMS ist zuständig für Lese- und Schreibzugriffe auf die Datenbank und verwaltet die interne Datenspeicherung.

Für die interne Strukturierung der Daten gibt es verschiedene Datenbankmodelle. Eine erste Unterteilung der Datenbankmodelle erfolgt bei der Aufteilung in SQL und NoSQL. Die Bezeichnung SQL trifft dabei nur auf ein Datenbankmodell zu, während NoSQL alle weiteren Modelle einschließt. Einige der relevanteren Datenbankmodelle [14] werden hier aufgelistet:

- **Relationale Datenbank**

Eine relationale Datenbank speichert Daten zeilenweise in Tabellen. Relevant für einzelne Zeilen ist dabei, dass jede Spalte nur ein Element enthält (Atomarität). Elemente können auch auf weitere Tabellen verweisen. Grundprinzipien einer relationalen Datenbank sind die Konsistenz und die Redundanzfreiheit der Daten. Um dies zu erfüllen müssen Daten eindeutig identifizierbar und nur einmal in der Datenbank enthalten sein. Das Abfragen der Daten erfolgt über die Abfragesprache SQL.

Relationale Datenbanken sind das am weitesten verbreitete Datenbankmodell. Sie besitzen eine hohe Verarbeitungsgeschwindigkeit und können fast beliebige Dateiformate speichern. Relationale Datenbanken eignen sich besonders für die Speicherung fester Strukturen und die Verknüpfung von Daten.

- **Objektorientierte Datenbank**

Bei objektorientierten Datenbanken werden Daten und deren Funktionen in Objekten gespeichert. Diese Objekte übernehmen dann die interne Datenverwaltung. Das Auslesen der Daten erfolgt über die objektinternen Funktionen oder die Objekt-Abfragesprache. Das Abfragen der Daten erfolgt ähnlich wie bei relationalen Datenbanken, weshalb der Name der Abfragesprache auch ähnlich ist: Object

Query Language (OQL). Durch den vermehrten Einsatz objektorientierter Programmiersprachen (Java, C#, .NET) steigt die Relevanz objektorientierter Datenbanken.

- **Graphdatenbank**

Graphdatenbanken benutzen, wie der Name sagt, Graphen um vernetzte Daten darzustellen. Die Daten werden in Knoten gespeichert. Knoten werden durch Kanten in Beziehung zueinander gesetzt, wobei eine Kante genau zwei Knoten miteinander verbindet. Ein Knoten kann beliebig viele Kanten besitzen. Knoten sind durch einen Bezeichner eindeutig identifizierbar und können eine beliebige Menge an Eigenschaften enthalten. Kanten lassen sich zudem gewichten, um beispielsweise die Relevanz der Beziehung, der verbundenen Knoten, darzustellen. Graphdatenbanken werden aufgrund ihrer Struktur oft für soziale Netzwerke (Facebook, Instagram, ...), wo mehrere Personenprofile über verschiedene Beziehungen miteinander verbunden sind, eingesetzt.

- **Dokumentenorientierte Datenbank**

Eine dokumentenorientierte Datenbank besitzt Dokumente als Grundlage zur Speicherung der Daten. Ein einzelnes Dokument ist dabei vergleichbar mit einer Zeile in einer relationalen Datenbank. Dokumente besitzen einen eindeutigen Identifikator und benötigen kein intern vorgegebenes Schema, nach welchem die Daten strukturiert sein müssen. Somit bestimmen die gespeicherten Daten das Schema zur Datenspeicherung. Es können auch verschiedene Dokumenttypen abgespeichert werden, was dokumentenorientierte Datenbanken zu einer häufigen Wahl beim Speichern von unstrukturierten und wechselhaften Daten und Dateien macht.

Es existieren noch weitere Datenbankmodelle (Netzwerkdatenbankmodell, hierarchisches Datenbankmodell, Key-value, ...). Diese werden im Rahmen dieser Arbeit aber nicht weiter ausgeführt.

3.3 Einordnung in den Stand der Technik

Aus dem Stand der Technik geht hervor, dass die in 3.1 vorgestellten Softwarelösungen, die in 1.1 beschriebenen Probleme nur in Teilen lösen. Die Apps bieten dabei hauptsächlich Informationen zu aktuellen Themen und generelle Informationen, welche in ein Gespräch eingebaut werden können. Nur die App „Beyond Small talk“ [8] stellt einem Nutzer Fragen zu Verfügung. Jedoch zielen diese Fragen auf eine Vertiefung des Gesprächs und bieten keine Hilfestellung um eine Konversation zu beginnen.

Menschliche Hilfestellungen sind kostenintensiv (Coaches und Seminare) und können zudem eine Abhängigkeit von der Begleitperson aufbauen. Wenn die Begleitperson eine Führungsrolle übernimmt, kann dies dazu führen, dass die Hilfe benötigende Person nicht zum Nachdenken angeregt wird und nur das Verhalten der Begleitperson kopiert, was den Lerneffekt schmälert und die Eigenständigkeit nicht fördert.

Hilfsmittel wie Bücher und Internetportale bieten eine gute theoretische Grundlage, lassen aber praktische Aspekte außer Acht.

Mit dieser Arbeit soll eine Softwarelösung entwickelt werden, welche sowohl theoretische als auch praktische Hilfestellungen bietet und die Funktionen der bereits existierenden Lösungen in einem Tool zusammenfasst. Die zu entwickelnde Hilfestellung soll zudem Gesprächseinstiege und individuelle Fragen und Profile bereitstellen.

Kapitel 4

Requirements Engineering

Das Requirements Engineering bezieht sich in diesem Projekt auf das Erfassen von Stakeholdern und das Ermitteln von Anforderungen.

Wegen fehlenden oder falschen Anforderungen können viele Projekte nicht die gewünschten Ziele erreichen [15, S. 4-9]. Aus diesem Grund ist es sinnvoll zum Beginn eines Projektes Stakeholder zu erfassen, eine Vision aufzustellen und Anforderungen an das Endprodukt zu ermitteln.

4.1 Stakeholder

Die Stakeholder-Analyse identifiziert wichtige Anspruchsträger des Projektes. Die nachfolgende Tabelle wurde nach [15, S. 58-60] ausgefüllt. Die Spalte „Verantwortung“ wurde mit der Spalte „Berechtigte Interessen“ ersetzt. Die Spalte „Aufgabenbeschreibung“ wurde ausgelassen, da sie in „Berechtigte Interessen“ indirekt enthalten ist und somit keine weiteren Informationen für dieses Projekt bereitstellt.

Name	Rolle	Berechtigte Interessen	Konfliktpotential	Verfügbarkeit
TW	Trainer (Studenten) Trainer (Mitarbeiter)	Projekt entspricht dem Kenntnisstand des Studenten Projekt soll innerhalb der zeitlichen Vorgaben der DHBW zu einem sichtbaren Ergebnis kommen	Kann als Trainer (Vorgesetzter) Entscheidungen treffen. Gegebenenfalls auch gegen das Projekt	Direkt (Stakeholder arbeitet in der gleichen Firma/Abteilung)
KS	Manager Verantwortlich für die Ausbildung der Studenten	Student hat Zugang zu den nötigen Mitteln für das Projekt Projekt wird zeitgerecht fertiggestellt	Zeitvorstellung/-einteilung des Projekts Besitzt als Vorgesetzter und Manager von TW die größte Entscheidungskraft	Direkt
DB	Auftraggeber	Projekt enthält Anforderungen der ursprünglichen Idee	Anforderungen der ursprünglichen Idee können neuen Anforderungen gegenüber stehen	Direkt
DB, KS, TW	Gutachter	Dokumentation muss nach Vorgaben der DHBW erstellt sein Faire und nachvollziehbare Bewertung Dokumentation muss nach Vorgaben von Mitutoyo CTL erstellt sein	-	Direkt
Prüfungsausschuss	Zweitgutachter	Dokumentation entspricht den Anforderungen der DHBW	Konfliktpotential mit Erstgutachtern	-
Mitarbeiter	Endkunden	Die Softwarelösung soll benutzbar sein Die Softwarelösung erfüllt ihren Zweck	-	Direkt
RE	Datenschutzbeauftragter	Die Softwarelösung entspricht der DSGVO	Datenverwaltung	Direkt
PK	Geschäftsführer	Projekt wird fristgerecht abgeschlossen	Ressourcenverteilung (Zeit, Verfügbarkeit der Trainer, ...)	Direkt
CB	Entwickler	Erfolg des Projekts Lauffähige Anwendung	-	-

Tabelle 4.1: Stakeholder

4.2 Vision

Vor der Festlegung von Anforderungen ist es notwendig die Vision und Ziele zu definieren. Dies ermöglicht einen Abgleich der Anforderungen mit der Vision und den Zielen. Daraus kann entschieden werden, ob die Anforderung zielführend ist. [16, S. 456 ff.]

Vision:

Die Softwarelösung bringt Menschen mit Kommunikationsschwäche dazu, den ersten Schritt in ein Gespräch zu wagen. Dazu schlägt sie Sätze vor, die in der Situation des Benutzers als geeignet erscheinen, ein längeres Gespräch mit einem bestimmten Ziel zu verfolgen.

Für die Softwarelösung wird im weiteren Verlauf der Name „Strike Up“ (englisch: „to strike sth. up“ = „an etwas anknüpfen“) verwendet.

Strike Up unterscheidet sich insbesondere dadurch von anderen Small-Talk-Ratschlägen, dass nicht nur plumpe Floskeln, sondern maßgeschneiderte persönliche Formulierungen angeboten werden.

Zudem ist Strike Up erweiterbar und kann um neue Formulierungen und neue Situationen ergänzt werden.

Ziele:

- Es werden Sätze und Fragen zur Eröffnung eines Dialogs bereitgestellt.
- Sätze und Fragen werden auf den Anwender angepasst.
- Sätze und Fragen werden auf das Gegenüber angepasst.
- Sätze und Fragen werden auf das Ziel und die Umgebung der Unterhaltung angepasst.
- Strike Up lässt sich leicht bedienen.

4.3 Anforderungen

Durch die Befragung mehrerer Stakeholder wurden Anforderungen an das Projekt ermittelt.

Anforderungen erhalten die Anforderungsattribute Identifikator, Beschreibung, Priorität und Aufwand [16, S. 479 f.]. Da die in diesem Projekt entwickelte Softwarelösung keinen direkten Schaden an Menschen verursachen kann, wird das Anforderungsattribut „Kritikalität“ außer Acht gelassen.

4.3.1 Funktionale Anforderungen

ID	Beschreibung	Stakeholder
F-1	Es werden allgemeine Hinweise für eine bessere Gesprächsführung bereitgestellt.	KS
F-2	Strike Up besitzt eine Feedbackfunktion, mit welcher Hinweise als gut oder schlecht bewertet werden können.	KS, DB, TW
F-3	Als schlecht empfundene Hinweise werden kein zweites Mal vorgeschlagen.	KS
F-4	Es werden Statistiken für die Nützlichkeit der Hinweise bereitgestellt.	KS
F-5	Es werden mehrere Fragen/Hinweise gleichzeitig bereitgestellt.	DB, KS, TW
F-6	Es lassen sich Personenprofile erstellen (von sich selbst und vom Gegenüber)	DB
F-7	Personenprofile können personalisierte Notizen enthalten, welche in einem Gespräch abrufbar sind.	KS
F-8	Es werden aktuelle Themen angeboten, über welche der Benutzer Gesprächsstoff beziehen kann.	KS
F-9	Es kann ausgewählt werden von welchen Quellen die Themen stammen.	KS
F-10	Nach einem Gespräch wird dem Benutzer angeboten Feedback zu senden. Dies geschieht über einen Button.	TW, KS
F-11	Benutzer können ihr Profil und damit verbundene Daten löschen.	RE
F-12	Fragen/Hinweise werden an den Benutzer und das Gegenüber angepasst.	-
F-13	Fragen/Hinweise werden an die Umgebung des Gesprächs angepasst.	-
F-14	Im Gesprächsverlauf werden weiterführende Fragen/Hinweise bereitgestellt.	-

Tabelle 4.2: Funktionale Anforderungen

4.3.2 Nichtfunktionale Anforderungen

ID	Beschreibung	Stakeholder
NF-1	Strike Up ist portabel.	TW, KS, DB
NF-2	Strike Up ist unauffällig und kann leicht während einem Gespräch verwendet werden.	KS, DB, TW
NF-3	Die Benutzeroberfläche der Softwarelösung ist übersichtlich.	DB, CB
NF-4	Personenbezogene Daten werden nicht an Dritte weitergegeben.	RE

Tabelle 4.3: Nichtfunktionale Anforderungen

4.3.3 Priorität der Anforderungen

Priorität	IDs
Niedrig (optional)	F-4, F-9
Mittel	F-2, F-5, F-7, F-8, F-10, F-11
Hoch	NF-1, NF-2, F-1, F-3, F-6, NF-3, NF-4, F-12, F-13, F-14

Tabelle 4.4: Priorität der Anforderungen

4.3.4 Aufwandsschätzung der Anforderungen

Aufwand	IDs
Niedrig	NF-1, F-5, NF-4
Mittel	NF-2, F-1, F-3, F-7, F-8, F-9, F-10, F-11, F-14
Hoch	F-2, F-4, F-6, NF-3, F-12, F-13

Tabelle 4.5: Aufwandsschätzung der Anforderungen

4.3.5 User Stories

Chris Rupp verweist beim Requirementsengineering auf verschiedene Spezifikationslevel. Diese beschreiben die genaue Spezifikation einer Anforderung.

Eine Anforderung des Spezifikationslevels Null entspricht dabei einer Idee, einer vagen Formulierung oder einem allgemeinen Ziel. Eine Anforderung auf dem vierten Spezifikationslevel entspricht einer Technical User Story oder einem Backlog Item. [17, S. 38 ff.]

Für dieses Projekt ist eine genaue Spezifikation der Anforderung auf das Level drei und vier nicht sinnvoll. Jedoch befinden sich die bisherigen Anforderungen auf dem nullten und ersten Spezifikationslevel. Um Anforderungen des zweiten Spezifikationslevels zu generieren werden nun, mit den in 11.1 erstellten Personas, User Stories geschaffen.

Beschreibung	Anforderung
Als Daniel Dacher möchte ich, dass meine an Strike Up gesendeten Daten nicht an Dritte weitergegeben werden.	NF-4
Als Daniel Dacher möchte ich, dass ich mit einem Klick mein Konto und alle damit verbundenen Daten jederzeit löschen kann.	F-11
Als Susanne Schmid möchte ich, dass ich Strike Up leicht während der Fahrt mit Bus und Bahn verwenden kann.	NF-1, NF-2
Als Susanne Schmid möchte ich, dass ich in einem Gespräch schnell auf die Funktionen von Strike Up zugreifen kann.	NF-2
Als Susanne Schmid möchte ich, dass ich keine schlechten Fragen/Hinweise bekomme, da meine Gesprächszeit oft beschränkt ist.	F-3, F-12
Als Susanne Schmid möchte ich eine Möglichkeit haben, mit welcher ich mich im Voraus auf ein Gespräch vorbereiten kann.	F-1
Als Torsten Tacheles, möchte ich durch Strike Up neue Kommunikationstechniken lernen.	F-1
Als Torsten Tacheles möchte ich durch Strike Up täglich etwas Neues lernen, was ich meinen Klienten beibringen kann.	F-1, F-8
Als Torsten Tacheles möchte ich, dass Strike Up auch meinen Klienten weiterhilft.	-

Tabelle 4.6: User Stories von Daniel Dacher, Susanne Schmid und Torsten Tacheles

Kapitel 5

Lösungsfindung

5.1 Umgebung

Da Strike Up portabel und während einem Gespräch verwendbar sein soll (vgl. 4.3 NF-1, NF-2), schränkt dies die Verfügbarkeit der möglichen Plattformen ein.

Eine Softwarelösung für einen Computer/Laptop ist in diesem Fall nicht sinnvoll, da eine unauffällige Verwendung eines Laptops (oder eines Computers) während einem Gespräch nur möglich ist, wenn die betroffene Person bereits an einem Laptop sitzt. Dies ist in öffentlichen Plätzen wie Bus, Bahn, Park, Geschäftsessen oder einer Messe meist nicht der Fall.

Als portable Umgebung bieten sich daher Smartphones und Smartwatches an.

In Deutschland besitzen 81% der Bevölkerung ab 14 Jahren ein Smartphone, in jungen Altersgruppen steigt der Prozentsatz auf über 95% [18]. Smartphones können innerhalb weniger Sekunden aus der Tasche geholt und gestartet werden, dies ermöglicht einen schnellen Gesprächseinstieg mit Strike Up.

Für Smartphones bestehen zwei dominante Betriebssysteme: Android und iOS. Mit einem Marktanteil von 78,2% ist Android in Deutschland Marktführer für Smartphone-betriebssysteme, gefolgt von iOS mit 21,3%. Andere Betriebssysteme, wie Windows und Blackberry, besitzen einen Marktanteil von unter 0,5%. [19].

Für Android entwickelte Apps basieren auf Java oder Kotlin, während iOS-Apps in Swift oder Objective-C entwickelt werden. Des weiteren gibt es Tools und Software Deve-

lopment Kits(SDK) mit welchen Apps entwickelt werden können, welche mit wenigen Einschränkungen auf beiden Betriebssystemen lauffähig sind:

- **React Native:** React Native ist ein JavaScript Framework, welches die UI in native (Android oder iOS spezifische) Elemente umwandelt. Die Logik bleibt dabei unverändert. Das Framework wird von Facebook, Instagram und Uber benutzt.
- **Xamarin:** Xamarin ermöglicht es Entwicklern eine gemeinsame Logik für Android und iOS zu schreiben. Die jeweilige UI wird allerdings in einer nativen Programmiersprache entwickelt.
- **Flutter:** Flutter ist ein SDK, welches von Google erstellt, und im Jahr 2018 erstmals in der Version 1.0 veröffentlicht wurde. Das SDK verwendet die, ebenso von Google entwickelte, Programmiersprache Dart. Flutter ermöglicht es UI-Komponenten zu entwickeln, welche auf beiden Betriebssystemen konsistent sind.

In Deutschland besaßen 2019 circa ein Drittel der Bevölkerung (36%) [20] eine Smartwatch. Das Gerät ermöglicht Nutzern/Nutzerinnen das Lesen und Verfassen von Nachrichten (auch über Spracheingabe), die Überwachung von sportlichen Aktivitäten (Pulsmessung, zurückgelegte Distanz, Schrittmesser, ...) und über eine App auch das Abrufen von Karten und Planen von Routen. Für eine optimale Nutzung sollte die Smartwatch dabei mit dem Smartphone gekoppelt sein. Auf dem Smartphone ist eine App installiert, welche die App auf der Smartwatch unterstützt. Damit eine Kopplung möglich ist, müssen die Betriebssysteme der beiden Geräte kompatibel sein.

Wie bei Smartphones ist auch der Smartwatchmarkt unter mehreren Marken wie Apple, Samsung, Huawei, Garmin [21] und fitbit [22] aufgeteilt. Jede Marke benutzt dabei ihr eigenes Betriebssystem. Zum Beispiel sind Watch Operating System (OS) und Android Wear reduzierte Versionen der originalen Betriebssysteme (iOS und Android).

Den Großteil des Marktanteils, im ersten Quartal 2020, besitzt Apple mit 36,3%. Darauf folgen Huawei (14,9%), Samsung (12,4%), Garmin (7,3%) und fitbit (6,2%) [23].

5.2 Fragenfindung und -bewertung

5.2.1 Fragenfindung

Strike Up soll auf den/die Nutzer/-in spezifizierte Fragen bereitstellen, dies kann durch individuelle Personenprofile erzieht werden (vgl. 4.2 F-12, F-6). Die Profile enthalten dabei Daten wie Name, Geschlecht, und Alter. Daraus lassen sich bereits einige Gesprächsthemen generieren. So würde eine Person unter 25 Jahren eher über Videospiele und Influencer oder bekannte Youtuber reden, als ein Person in einem Alter von über 75 Jahren.

Für ein personalisiertes Gespräch sollten aber noch weitere Faktoren miteinbezogen werden, da die oben genannten Merkmale nur eine oberflächliche Beschreibung der Person ermöglichen.

Des Weiteren spielen Umgebungsvariablen eine Rolle bei der Fragenauswahl (vgl. 4.2 F-13). Beispiele für Umgebungsvariablen sind Ort, Jahreszeit und die Beziehung zum Gegenüber. Beispielsweise beinhaltet eine Konversation auf einer geschäftlichen Messe andere Themen, als eine Konversation am Strand im Sommer.

Umgebungsvariablen haben einen direkten Einfluss auf die Auswahl der Fragen/Hinweise und werden entweder von Strike Up generiert (Umgebungsvariablen welche mit Uhrzeit, Datum oder Einstellungen in den Nutzerprofilen zusammenhängen) oder vom/von dem/der Nutzer/-in vor einem Gespräch ausgewählt (geschäftlich oder privat, kennt der/die Nutzer/-in das Gegenüber, ...).

5.2.2 Fragenbewertung

Um Fragen und Hinweise während und bereits vor dem Gespräch an das Gegenüber anzupassen, könnten Fragen und Hinweise durch den/die Benutzer/-in im Voraus und im Gesprächsverlauf bewertet werden.

Die Bewertung wird mittels einer Punktzahl von null bis 100 realisiert, wobei 100 die optimale Punktzahl darstellt. Der/die Nutzer/-in bewertet dabei auch Fragen/Hinweise für das Gegenüber, soweit dies möglich ist und die Bewertung nicht durch das Gegen-

über selbst stattfindet. In einem Gespräch wird bei Fragen/Hinweisen die Punktzahl des/der Nutzers/-in mit der Punktzahl des Gegenübers addiert, wodurch eine maximale Punktzahl von 200 erreicht werden kann. Des Weiteren können Umgebungsvariablen, abhängig von ihrer Stimmigkeit, Punkte zu Fragen/Hinweisen addieren oder subtrahieren.

Fragen und Hinweise werden anschließend nach ihrer Punktzahl geordnet und für die Konversation bereit gestellt. Fragen/Hinweise mit einer Gesamtbewertung von unter 100 Punkten werden nicht angezeigt. Sollten jedoch weniger als zehn Fragen/Hinweise in einem Gespräch verfügbar sein, so werden auch Punktzahlen kleiner 100 eingebunden und der/die Nutzer/-in wird über einen Warnhinweis informiert, dass möglicherweise Fragen angeboten werden, welche ihm/ihr oder dem Gegenüber nicht gefallen.

Eine weitere Möglichkeit zur Fragenbewertung ist das einführen von „Tags“. Ein Tag dient dabei als Schlagwort um Fragen/Hinweise zu kategorisieren. So hätte die Frage „Was ist ihr Lieblingstier?“ die Tags „Tier“ und „kennenlernen“. Der/die Nutzer/-in kann in seinen/ihren Profileinstellungen aus einer Liste aller verfügbaren Tags auswählen, ob er/sie das Tag positiv (soll im Gespräch vorkommen), neutral (egal) oder negativ (soll in einem Gespräch vermieden werden) bewertet. Nach demselben Prinzip werden die Tags der Gesprächspartnerprofile bearbeitet. Standardmäßig werden alle Tags als neutral bewertet. Da es vorkommen kann, dass wenig Informationen über das Gegenüber bekannt sind, kann dessen Profil und die damit verbundenen Tags auch während einer Konversation bearbeitet werden.

Fragen und Hinweise besitzen auch Tags, welche für diese als passend oder unpassend bewertet werden. Kommt ein Tag nicht in der Liste der passenden oder unpassenden Tags vor, so wird es als neutral bewertet. Umgebungsvariablen verfügen in gleicher Weise über passende und unpassende Tags.

Fragen/Hinweise werden, nach Ermittlung der Umgebungsvariablen, durch ihre eigenen Tags, die Tags der Umgebungsvariablen, die Tags des/der Nutzers/-in und die Tags des Gegenübers bewertet. Hierzu wird die Liste aller unpassenden Tags einer/eines Frage/Hinweises mit den Listen unpassender Tags von Nutzer/-in, Umgebungsvariablen und Gegenüber verglichen. Immer, wenn ein Tag aus der Liste der/des Frage/Hinweises mit einem Tag aus einer anderen Liste übereinstimmt, wird auf die/den aktuelle Frage/Hinweis ein Minuspunkt addiert. Dasselbe geschieht mit den Listen der passenden

Tags, hierbei wird bei einem übereinstimmenden Tag jedoch ein Pluspunkt addiert. Anschließend werden die Fragen/Hinweise in absteigender Reihenfolge nach der erreichten Punktzahl sortiert. Fragen/Hinweise mit Null oder weniger Punkten werden aus dem Pool für das Gespräch entfernt. Sind weniger als zehn Fragen/Hinweise im Pool vorhanden, so wird dieser zuerst mit „neutralen“ (Bewertung entspricht Null) und anschließend mit „negativen“ (Bewertung kleiner Null) Fragen aufgefüllt und der/die Nutzer/-in erhält einen Warnhinweis, welcher auf möglicherweise als unpassend empfundene Fragen/Hinweise hinweist.

5.3 Allgemeine Hilfestellungen

Aus F-1 und F-8 (4.2) geht hervor, dass Strike Up auch allgemeine Hinweise und Gesprächsthemen bereitstellen soll. Der /die Nutzer/-in kann sich mit Strike Up über aktuelle Themen informieren und somit dieses Wissen in Gespräche einbringen.

Allgemeine Hinweise können Tips für Haltung, Gestik, Mimik und Gesprächstechniken geben. Dadurch kann das Selbstbewusstsein und die Eloquenz des/der Nutzers/-in verbessert werden.

Aktuelle Themen können mit Hilfe eines RSS-Readers abgerufen werden. Dieser ist für den/die Nutzer/-in über einen Button auf der Startseite von Strike Up erreichbar. In den Einstellungen des RSS-Readers kann angegeben werden, von welchen Quellen die Themen bezogen werden sollen. Damit ist auch F-9 erfüllt. Die Auswahlmöglichkeiten sind hierbei vordefiniert, da von dem/der durchschnittlichen Nutzer/-in nicht erwartet wird über RSS-Feeds informiert zu sein.

Die Themen werden mit Titel und einer Zusammenfassung angezeigt. Über einen Button wird der RSS-Reader aufgefordert den RSS-Feed erneut abzurufen und die Daten zu aktualisieren. Wird auf den Titel oder Zusammenfassung einer/-s Meldung/Themas gedrückt, so wird der/die Nutzer/-in zum vollständigen Artikel weitergeleitet.

Auf allgemeine Hinweise zur Gesprächsführung wurde bereits in 2 eingegangen. Das Realisieren dieser Hinweise innerhalb von Strike Up kann über tägliche Tipps umgesetzt werden. Beim ersten täglichen Öffnen der Anwendung wird dem/der Nutzer/-in ein

Pop-up angezeigt, welches einen allgemeinen Hinweis zur Gesprächsführung enthält. Innerhalb dieses Pop-ups können frühere und zukünftige Hinweise aufgerufen werden.

5.4 Datenspeicherung

Die von Strike Up bereitgestellten und verwalteten Daten benötigen einen Speicherort, welcher für Nutzer/-innen zugänglich ist und von Zugriffen durch Unbekannte geschützt ist. Das Speichern der Daten kann in einer Cloud oder in dem geräteinternen Speicher des Smartphones durchgeführt werden.

Wichtige Kriterien für eine Auswahl sind hierbei:

- Speichervolumen
- Strukturierung der Daten
- Lese- und Schreibgeschwindigkeit
- Sicherheit vor Fremdzugriffen
- Verwaltung und Sicherung der Daten

Zur Datenspeicherung im geräteinternen Speicher gibt es Android SharedPreferences [24]. Daten werden hierbei als Key-Value-Paare in XML-Dateien gespeichert. Innerhalb der SharedPreferences einer App können Dateien erstellt, gelöscht und bearbeitet werden. Das Erstellen von neuen Ordnern und die damit verbundene Strukturierung der Daten ist jedoch nicht möglich.

Mit zusätzlichen Bibliotheken wie Gson [25] können Java Objekte in JavaScript Object Notation (JSON) Objekte umgewandelt werden und somit als Key-Value-Paar in den Shared Preferences gespeichert werden.

SharedPreferences eignen sich jedoch nur für Datensätze unter 100KB, da die Daten im Random Access Memory (RAM) gespeichert werden. Der RAM kann durch eine zu große Datenmenge überlastet werden, wodurch die App nicht mehr wie gewollt funktioniert. Für eine größere Speicherkapazität kann der app-spezifische Speicher [26] verwendet werden. Der Unterschied zu den SharedPreferences besteht darin, dass Dateien nicht im

RAM, sondern im geräteinternen Speicher gespeichert werden. Zum Lesen, Schreiben und Bearbeiten der Dateien wird deshalb ein I/O-Stream benutzt. Eine Strukturierung der Daten durch Verzeichnisse ist auch hier nicht möglich, es wird aber die Verwendung von Datenbanken wie SQLite [27] ermöglicht. SQLite ist eine SQL-Datenbank und ermöglicht somit das Strukturieren der Daten in Tabellen, sowie das Verwenden aller SQL-typischen Abfragebefehle.

Werden die Fragen von Strike Up im geräteinternen Speicher gespeichert, so müssen die Daten im Source Code enthalten sein und sind anschließend nur durch Updates veränderbar. Änderungen der Daten durch den/die Nutzer/-in finden nur lokal statt und ein Austausch unter Nutzern/-innen (z.B. zur Bewertung von Fragen/Hinweisen) ist nicht möglich.

Im geräteinternen Speicher gespeicherte Daten, können durch Nutzer/-innen eingesehen werden, da die Daten im Source Code enthalten sind, welcher aus der Android Package (APK) extrahiert werden kann. Für andere installierte Apps sind die Daten jedoch nicht sichtbar.

Bei einer Deinstallation der App werden alle durch den/die Nutzer/-in erzeugten Daten gelöscht. Dies bedeutet, dass ein/eine Nutzer/-in von Strike Up, nach einer Deinstallation und Neuinstallation der App, ein neues Konto erstellen muss. Zudem werden alle durch den/die Nutzer/-in erstellten Gesprächspartner/-innen gelöscht.

Datenspeicherung in einer Cloud ermöglicht die Aktualisierung der durch Strike Up bereitgestellten Fragen/Hinweisen, ohne dass die Anwendung selbst aktualisiert wird. Fragen/Hinweise, Nutzerprofile und Profile von Gesprächspartnern/-innen werden in der Cloud gespeichert und zur Laufzeit abgerufen.

Der für die Cloud benötigte Server kann durch einen privaten Server realisiert werden. Die Realisierung durch das Aufsetzen eines privaten physikalischen Servers kann jedoch aus mehreren Gründen ausgeschlossen werden:

- Die Internet Protocol (IP)-Adresse kann sich verändern
- Eine Authentifizierung muss erstellt werden, damit Nutzer/-innen von Strike Up auf die Daten zugreifen können
- Die Authentifizierung muss Zugriffe durch Fremde verhindern, um die Sicherheit

persönlicher Daten zu wahren

- Die Hardware muss bereitgestellt und gewartet werden
- Strom- und Internetausfälle schalten den gesamten Server aus

Aus diesen Gründen bietet sich das Nutzen von Cloudservern anderer Anbieter wie Google, Amazon und Microsoft an. Diese (und andere) Unternehmen bieten auch fertige Cloud-Datenbanken an.

Google bietet mit Firebase [28] eine Plattform mit verschiedenen Tools, wie einer Echtzeitdatenbank, Authentifizierung, Cloud-Speicher, Crash Reports und Weiteren an. Für Strike Up sind hierbei besonders die Echtzeitdatenbank und die Authentifizierung relevant. Die Echtzeitdatenbank speichert Daten in Form von JSON-Objekten und ist somit eine NoSQL-Datenbank. Gespeicherte Daten können in der Datenbank durch Verschachtelung strukturiert werden. Die Authentifizierung ermöglicht das Anlegen von Nutzern, welche auf die Datenbank zugreifen können. Durch Regeln kann festgelegt werden, ob Nutzer/-innen, oder Fremde, in bestimmten Bereichen der Datenbank Lese- und/oder Schreibrechte haben. Somit kann festgelegt werden, dass nur authentifizierte Nutzer/-innen auf die gespeicherten Daten zugreifen können. Des Weiteren können die Zugriffsrechte der Nutzer/-innen auf ihre eigenen Daten beschränkt werden.

Firebase besitzt ein Free-Tier, bei welchem ein Gesamtspeicher von 1GB und eine monatliche Download-Größe von 10GB zur Verfügung gestellt wird. Bei einer höheren Nutzung können die Werte kostenpflichtig vergrößert werden.

Amazon DynamoDB [29] ist eine von Amazon gehostete NoSQL-Datenbank, welche Daten als Key-Value-Paare in Dokumenten speichert. Wie bei Firebase ist auch hier ein Nutzermanagement möglich. Dies geschieht über das Definieren von Zugriffsrechten für Nutzer/-innen und das Aufteilen von Nutzergruppen in Rollen (Nutzer/-in, Moderator, ...).

Im Free-Tier sind 25GB Speicher und 25 Write Capacity Units (nach Angaben von Amazon entspricht dies der Verarbeitung von ca. 200 Millionen Anfragen pro Monat) enthalten[30].

Microsoft stellt mit Azure Cosmos DB [31] eine auf App-Entwicklung spezialisierte NoSQL-Datenbank bereit. Cosmos DB unterstützt fünf verschiedene Datenbankmodelle:

MonogDB-API (dokumentorientiert), Apache-Cassandra-API (Key-Value), Gremlin-API (Graphdatenbank) und Microsofts Document SQL und Table API. Die Datenbankstrukturen und Application Programming Interfaces (APIs) werden dabei auf die interne Struktur der Cosmos DB abgebildet [32]. Zugriffsrechte auf gespeicherte Daten können bis auf die Ebene eines Key-Value-Paares individuell eingestellt werden. Nutzer/-innen erhalten dabei Ressource-Tokens, welche Zugriff auf bestimmte Teile der Datenbank gestatten.

Der Free-Tarif für ein Azure Cosmos DB-Konto enthält 5GB Speicher, sowie 400RU/s (Anforderungseinheiten pro Sekunde).

Kapitel 6

Lösungsbewertung

6.1 Umgebung

Smartwatches sind internetfähig und können ohne die Unterstützung eines Smartphones auskommen. Da Smartwatches wie eine Armbanduhr am Handgelenk des/der Nutzers/-in sind, kann Strike Up schnell und unauffällig verwendet werden. Besonders während einem Gespräch ist ein kurzer Blick auf die Uhr weniger auffällig als ein Blick auf das Smartphone.

Die Schwächen einer Smartwatch liegen allerdings in der geringen Display-Größe. Das Erstellen und Bearbeiten von Gesprächspartnern/-innen wird dadurch erschwert. Zudem schränkt die Displaygröße die Übersichtlichkeit der App ein. Das Verwenden einer Smartwatch zusätzlich zu einer App auf dem Smartphone wäre eine gute Lösung; jedoch müssen so zwei Apps entwickelt werden.

Aus diesem Grund wird in dieser Arbeit Strike Up in Form einer App für Smartphones umgesetzt. Die Unterstützung durch eine Smartwatch ist eine mögliche Verbesserung für die Zukunft.

Damit Apps im App Store von Apple veröffentlicht werden dürfen, benötigt der/die Entwickler/-in eine Mitgliedschaft im Apple Developer Program. Diese kostet pro Jahr 99 US-Dollar [33]. Des Weiteren kann eine für iOS entwickelte Anwendung nur in einer Apple-Umgebung (Macbook, ...) kompiliert werden.

Aus diesen Gründen, und wegen dem Marktanteil von 21,3% in Deutschland, ist das

Entwickeln von Strike Up für eine reine iOS-Umgebung nicht sinnvoll.

Um Apps im Google Play Store zu veröffentlichen, wird eine einmalige Registrierungsgebühr von 25 US-Dollar benötigt [34]. Eine jährliche Gebühr ist nicht vorhanden.

React Native ermöglicht zwar das Entwickeln einer Anwendung für iOS und Android, jedoch werden nicht alle plattformspezifischen APIs unterstützt. Nicht unterstützte APIs müssen in der nativen Programmiersprache erstellt werden. Es muss somit trotzdem für iOS und Android separat entwickelt werden. [35]

Die nicht vollständige Unterstützung nativer APIs ist ein Schwachpunkt aller Frameworks, welche das gleichzeitige Entwickeln für iOS und Android ermöglichen.

Xamarin bietet durch die Unterstützung von .Net und Microsoft Visual Studio die beste Entwicklungsumgebung. Jedoch mindert die geringe Popularität die verfügbaren Hilfestellungen durch andere Entwickler/-innen [36].

Laut einer Umfrage von Stack Overflow ist Flutter das beliebteste der drei vorgestellten Frameworks [37]. Es ist jedoch auch das jüngste (Version 1.0 im Jahr 2018) und kann somit auf den geringsten Anteil an verfügbaren Hilfestellungen zurückgreifen. Auch die Unterstützung durch Bibliotheken von Dritten ist noch nicht so gut wie bei den anderen Frameworks. [36]

Ein weiterer negativer Aspekt der Cross-Platform-Entwicklung sind Systemupdates von iOS oder Android. Ein Systemupdate kann neue Funktionen hinzufügen und alte verändern oder entfernen. Während bei nativen Plattformen darauf geachtet wird, dass alle Änderungen rückwärtskompatibel sind, müssen sich Cross-Platform-Frameworks erst an die Änderungen anpassen. Dadurch kann es zu fehlerhaftem Verhalten der App kommen. Bei der Entwicklung einer App, welche sich in Zukunft durch Verbesserungen und weitere Funktionen noch ändern kann, ist somit das Entwickeln auf einer nativen Plattform sinnvoller.

Strike Up ist darauf ausgelegt den/die Nutzer/-in in und vor Gesprächen zu unterstützen. Um diese Aufgabe optimal zu erfüllen, ist die App auf Feedback der Nutzer/-innen angewiesen. Daraus resultiert, dass Strike Up nach der Veröffentlichung nicht vollständig ist und somit weiterhin aktualisiert wird.

Die Wahl des Betriebssystems fällt somit auf Android.

6.2 Fragenbewertung

Bei einer punktebasierten Fragenbewertung wird von dem/der Nutzer/-in verlangt, möglichst viele Fragen im Voraus zu bewerten. Dies ist bei einer hohen Fragenanzahl nicht sinnvoll. Des Weiteren wird für ein individuelles Gespräch auch das Gegenüber in die Auswahl der Fragen miteinbezogen. Somit muss der/die Nutzer/-in präemptiv entscheiden, ob dem Gegenüber einzelne Fragen gefallen. Da Strike Up bei der Auswahl passender Fragen helfen soll (vgl. 4.2: F-12, F-13), ist die Auswahl passender Fragen durch den/die Nutzer/-in nicht zielführend.

Auch bei einer Fragenbewertung durch Tags wird von dem/der Nutzer/-in erwartet vor einem Gespräch auszuwählen was ihm/ihr und dem Gegenüber gefällt. Dies geschieht jedoch auf einer oberflächlicheren Basis. Es ist beispielsweise leichter zu sagen, ob das Gegenüber lieber klassische Musik oder Heavy Metal hört, als zu sagen wie gut dem Gegenüber die Frage „Und wie war ihr Wochenende so?“ gefällt.

Falls einige Tags nicht den Erwartungen entsprochen haben, können Tags auch während dem Gesprächsverlauf angepasst werden.

Die Auswahl der Fragen wird somit durch das Verwenden von Tags umgesetzt.

6.3 Datenspeicherung

Strike Up wird in Zukunft durch neue Fragen und Hinweise erweitert. Eine laufende Integration neuer Fragen und Hinweise ist mit dem Verwenden des geräteinternen Speichers nicht möglich. Um dem internen Speicher neue Fragen hinzuzufügen, müsste Strike Up im Google Play Store aktualisiert werden. Aus diesem Grund wird eine Cloud-Datenbank verwendet.

Firebase ermöglicht Datenverwaltung über eine zentrale Konsole, auf welche nur der/die Besitzer/-in der Datenbank Zugriff hat. Die Konsole bietet weitere Funktionen, wie Crash-Reports, Datennutzung und ein Dashboard mit Nutzerinformationen.

Da Daten als JSON-Objekte gespeichert werden, können von Strike Up generierte Objekte direkt in der Datenbank gespeichert und daraus wieder als Objekte gelesen werden.

Firebase besitzt mit 1GB Speicher den geringsten frei verfügbaren Speicherplatz der vorgestellten Datenbanken.

Da Amazon DynamoDB Daten als Key-Value-Paare speichert, werden von Strike Up generierte Objekte entweder durch Code in JSON-Objekte umgewandelt und anschließend gespeichert, oder in Form von einzelnen Attributen als Key-Value-Paar gespeichert. Beim Schreiben und Lesen von Daten sind somit Zwischenschritte nötig.

Mit 25GB frei verfügbaren Speicherplatz bietet Amazon DynamoDB den größten Speicherplatz der vorgestellten Datenbanken.

Microsoft Azure CosmosDB bietet dem/der Nutzer/-in die Wahl des Datenbankmodells. Die Wahl fällt für Strike Up auf eine objektorientierte Datenbank.

Um Daten zu lesen und zu schreiben, wird die URL und das Passwort der Datenbank benötigt. Da diese Informationen im Klartext von Strike Up hinterlegt werden, kann jede/-r Nutzer/-in auf diese zugreifen und somit freien Zugriff auf die Datenbank erhalten.

Microsoft Azure CosmosDB wird somit als mögliche Option ausgeschlossen.

Die Wahl der Cloud-Datenbank fällt auf Firebase, da die Konsole eine übersichtliche Datenverwaltung und ein integriertes Nutzermanagement ermöglicht. Des Weiteren ermöglicht das vorhandene Datenbankmodell das Speichern von Objekten.

Kapitel 7

Umsetzung

7.1 Datenbank

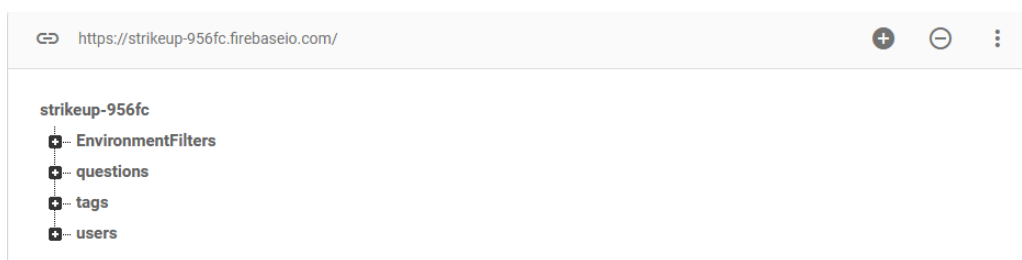


Abbildung 7.1: Minimierte Struktur der Datenbank

Daten werden in der Datenbank in Form vier verschiedener Kategorien gespeichert: „EnvironmentFilters“ (Umgebungsvariablen), „questions“ (Fragen), „tags“ (Tags) und „users“ (Nutzer/-innen) (vgl. 7.1).

Die Verwaltung der Datenbank erfolgt durch den Besitzer des Firebase-Projektes. Dies geschieht über eine Konsole. Um auf diese Konsole zuzugreifen, wird das entsprechende Google-Konto benötigt.

Über die Konsole können in der Datenbank gespeicherte Daten beliebig geändert, gelöscht und hinzugefügt werden. Nutzerkonten können deaktiviert oder gelöscht werden.

Bei einer Deaktivierung eines Nutzerkontos kann der/die entsprechende Nutzer/-in sich nicht mehr bei Strike Up anmelden. Des Weiteren kann eine Zurücksetzung des

Passworts angefordert werden.

Über die Konsole kann der aktuell genutzte Speicherplatz, sowie die in diesem und im vorherigen Monat heruntergeladene Datenmenge eingesehen werden.

Zu den Nutzern/-innen werden (anonymisiert) das Land, in welchem sie Strike Up verwenden, das verwendete Betriebssystem, sowie die in der App verbrachte Zeit angezeigt.

Außerdem können Zugriffsregeln für individuelle Abschnitte der Datenbank definiert werden.

Lese- und Schreibaufrufe werden im Code asynchron ausgeführt. Das bedeutet der Code „läuft“ weiter, während der Lese-/Schreibaufruf im Hintergrund ausgeführt wird. Dies verhindert das Finden von Fehlern mit einem Debugger, da Variablen, welche Daten aus der Datenbank enthalten, zur Laufzeit immer Null sind. Eine Fehlersuche im Zusammenhang mit Leseaufrufen der Datenbank gestaltet sich somit schwierig.

Wenn Daten aus der Datenbank in darauffolgendem Code benötigt werden, so muss auf die Fertigstellung des Lesevorgangs gewartet werden, da ansonsten mit Null-Objekten gearbeitet wird. Null-Objekte würden zu Fehlern in der Anwendung führen.

Ein Leseaufruf besteht aus einem Listener, welcher auf einen Knotenpunkt der Datenbank registriert wird. Beim Setzen eines Listeners (und wahlweise bei Veränderung der Daten unter dem registrierten Knotenpunkt) wird dessen Funktion einmal ausgeführt. Um Null-Objekte zu verhindern, werden, auf einen Leseaufruf folgende, Funktionen in dem zugehörigen Listener aufgerufen. Dies umgeht das asynchrone Ausführen der Leseaufrufe. Bei mehreren aufeinanderfolgenden Leseaufrufen, verzögert sich jedoch die Ausführung des Codes, da bei jedem Aufruf auf dessen Fertigstellung gewartet wird. Dies kann in Extremfällen, und bei langsamem Internet, die Ausführung der App verlangsamen.

7.2 Nutzerprofile

Damit Nutzer/-innen Zugriff auf die von Strike Up bereitgestellten Funktionen haben, müssen sie sich zuerst registrieren. Eine Registrierung verhindert, in beschränktem Maße, dass die von Strike Up genutzte Datenbank mit ungenutzten Nutzerprofilen

gefüllt wird. Dies spart Speicherplatz, welcher im kostenlosen Vertrag beschränkt ist. Die Anmeldung erfolgt über eine E-Mail-Adresse und ein Passwort (siehe 7.2 und

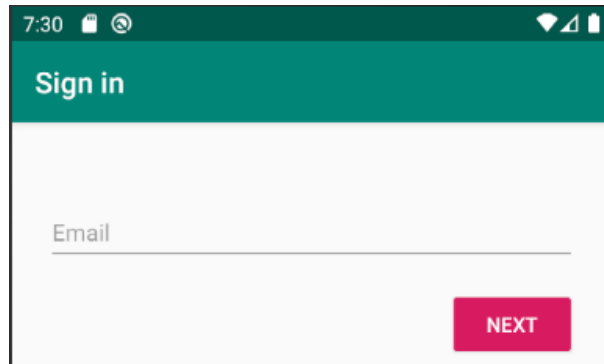
A screenshot of a mobile application's 'Sign in' screen. The screen has a teal header with the text 'Sign in'. Below the header is a light gray background. There is a text input field labeled 'Email' with a thin gray underline. To the right of the input field is a pink button with the text 'NEXT' in white capital letters. The top status bar shows the time '7:30' and various icons.

Abbildung 7.2: Anmeldung mit E-Mail-Adresse

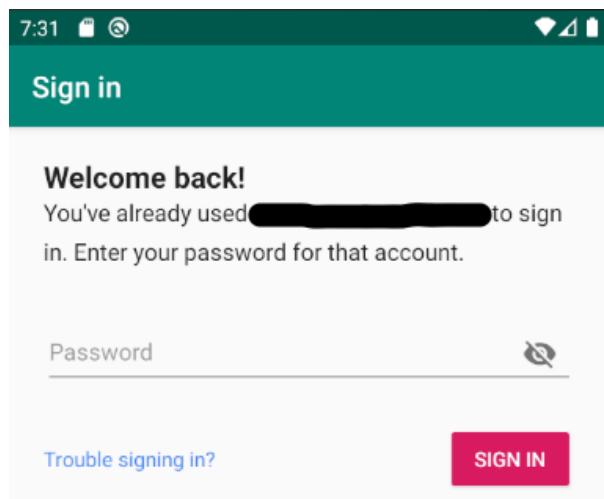
A screenshot of a mobile application's 'Sign in' screen. The screen has a teal header with the text 'Sign in'. Below the header is a light gray background. There is a bold black text 'Welcome back!' followed by a line of text: 'You've already used [redacted] to sign in. Enter your password for that account.' Below this is a text input field labeled 'Password' with a thin gray underline and a small eye icon to its right. At the bottom left is a blue link 'Trouble signing in?'. At the bottom right is a pink button with the text 'SIGN IN' in white capital letters. The top status bar shows the time '7:31' and various icons.

Abbildung 7.3: Aufforderung zur Eingabe des Passworts

7.3). Ist die E-Mail-Adresse bereits auf ein Konto registriert, so wird der/die Nutzer/-in aufgefordert das dazugehörige Passwort einzugeben (7.3). Wenn mit der E-Mail-Adresse ein neues Konto angelegt wird, so muss der/die Nutzer/-in ein neues Passwort für diesen Account festlegen.

In der Datenbank werden Nutzer/-innen unter ihrer User Identification (UID) gespeichert. Die UID ist eine einzigartige, von Firebase automatisch generierte Kennung. Anhand dieser einzigartigen Kennung wird jede/-r Nutzer/-in von Strike Up identifiziert. Wird eine Nutzeraccount gelöscht, so wird auch dessen UID entfernt.

Abgesehen von der UID besitzt ein Nutzerprofil folgende Attribute:

- Name: Der Name des/der Nutzers/-in
- Email: Die E-Mail-Adresse des/der Nutzers/-in
- Alter: Das Alter des/der Nutzers/-in
- PreferredTags: Eine Liste aller Tags, welche der/die Nutzer/-in bevorzugt
- IgnoredTags: Eine Liste aller Tags, welche der/die Nutzer/-in vermeiden will
- ConversationalPartners: Eine Liste aller gespeicherten Gesprächspartner/-innen (siehe 7.2.2) des/der Nutzers/-in

7.2.1 Tags

Tags werden in der Datenbank unter dem Knoten „tags“ gespeichert. Ein Tag besitzt keine Attribute. Tags sind für jeden authentifizierten Nutzer einsehbar.

Nutzer/-innen wählen aus der Liste aller verfügbaren Tags aus, ob ihnen ein Tag gefällt oder nicht. Das Auswählen der bevorzugten und gemiedenen Tags geschieht dabei über zwei separate Buttons.

Tags, welche bereits in der Liste bevorzugter Tags sind, werden beim Hinzufügen neuer bevorzugter Tags nicht mehr zur Auswahl angeboten. Das Gleiche gilt für gemiedene Tags. Beim Hinzufügen neuer bevorzugter Tags werden jedoch auch Tags angezeigt, welche in der Liste gemiedener Tags vorhanden sind. Dies gilt auch umgekehrt.

Damit Tags nur in einer Liste vorkommen, und um die Auswahl der Tags zu vereinfachen, wird eine neue Funktion entwickelt, welche über mehrere Checkboxes das Aufteilen der Tags in die entsprechenden Listen ermöglicht (7.4). Rot steht dabei für gemiedene Tags und grün für bevorzugte Tags. Gelb markierte Tags sind in keiner der beiden Listen vorhanden und werden somit neutral bewertet.

Ein Problem tut sich dabei bei der Implementation einer ListActivity in Android auf. Da mehr Objekte in der anzuzeigenden Liste sind, als auf den Bildschirm passen, verwendet Android bereits für angezeigte Elemente verwendete Container, um noch nicht angezeigte Elemente zu laden [38]. Das bedeutet: Wenn ein, mit einem Objekt gefüllter, Container (durch scrollen) vom Bildschirm verschwindet, wird dieser Container benutzt um das neu



Abbildung 7.4: Auswahl bevorzugter und gemiedener Tags

erschienene Element anzuzeigen. Der Container behält dabei die Position des Hakens der Checkbox. Beim Scrollen, wird somit bei neu erschienen Tags der Haken an der Position der neu ausgeblendeten Tags gesetzt. Dies lässt sich durch das Verwenden einer Hashmap, welche die Positionen der Haken speichert, umgehen. Dabei ist der Name eines Tags der „Key“ und die Position des Hakens das „Value“.

Aus Zeitgründen ist diese Funktion noch nicht fehlerfrei lauffähig. Deshalb geschieht die Auswahl bevorzugter und gemiedener Tags weiterhin über zwei separate Buttons.

Während der Entwicklung wurden Tags um eine ID erweitert; Tags werden somit unter ihrer ID gespeichert und besitzen das Attribut „Name“.

Das Verwenden einer ID ermöglicht, dass Strike Up auf weitere Sprachen erweitert werden kann. So können Tags beispielsweise um die Attribute „Name-en“ oder „Name-fr“ erweitert werden.

7.2.2 Gesprächspartner/-innen

Gesprächspartner/-innen werden als eine, zu einem/-r Nutzer/-in gehörende, Liste gespeichert. Jede/-r Nutzer/-in kann beliebig viele Gesprächspartner/-innen hinzufügen. Ein/-e Gesprächspartner/-in besitzt die Attribute:

- Name: Name des/der Gesprächspartners/-in
- Gender: Geschlecht des/der Gesprächspartners/-in
- Alter: Alter des/der Gesprächspartners/-in
- IgnoredTags: Eine Liste aller Tags, welche der/die Gesprächspartner/-in vermeiden will
- PreferredTags: Eine Liste aller Tags, welche der/die Gesprächspartner/-in bevorzugt
- Notizen: Von dem/der Nutzer/-in verfasste Notizen zu dem/der Gesprächspartner/-in

Das Bearbeiten und Auswählen von Tags geschieht dabei gleich wie bei einem/-r Nutzer/-in.

Gesprächspartner/-innen werden um das Attribut ID erweitert, da sie bisher unter ihrem Namen in der Datenbank gespeichert wurden. Das Speichern unter dem Namen erzeugte Probleme, wenn der Name geändert wurde, da so zwei Objekte desselben Partnerprofils in der Datenbank hinterlegt waren. Gebraucht wird aber nur ein Objekt. Es entsanden also redundante Daten.

Das Einführen einer unveränderlichen und einzigartigen ID ermöglicht eine problemfreie Änderung des Namens.

Die ID wird von Firebase generiert. Zum Erstellen dieser einzigartigen ID benutzt Firebase den gewünschten Speicherort innerhalb der Datenbank und das aktuelle Datum.

Nachdem das Profil eines/-r Gesprächspartners/-in erstellt ist, kann dieses bearbeitet und wieder gelöscht werden.

Strike Up besitzt individuelle Profile für den/die Nutzer/-in und für Gesprächspartner/-innen. Somit ist F-6 erfüllt (vgl. 4.2).

7.3 Fragen/Hinweise

Fragen und Hinweise werden in der Datenbank unter „questions“ gespeichert. Eine einzelne Frage ist dabei unter ihrer ID zu finden.

Fragen besitzen folgende Attribute:

- ID: Eine eindeutige ID, unter welcher die Frage gespeichert ist
- tags: Eine Liste aller Tags, welche zu dieser Frage passen
- text: Der Text der Frage

Fragen und Hinweise können nicht von Nutzern/-innen bearbeitet werden. Das Verändern, Hinzufügen und Entfernen von Fragen obliegt somit dem/der Verwalter/-in der Datenbank.

Ursprünglich war geplant, dass Nutzer/-innen sowohl Fragen als auch Tags zur Datenbank hinzufügen können, da die Datenbank besonders in den Anfängen von Strike Up noch nicht vollständig ist. Diese Idee wurde jedoch aus Angst vor Missbrauch wieder gestrichen.

7.3.1 Umgebungsvariablen

Eine Umgebungsvariable wird mit folgenden Attributen in der Datenbank gespeichert:

- ID: Eine eindeutige ID, unter welcher die Umgebungsvariable gespeichert ist
- Name: Der Name der Umgebungsvariable
- IgnoredTags: Liste aller Tags, auf welche die Umgebungsvariable einen negativen Einfluss hat
- Preferred Tags: Liste aller Tags, auf welche die Umgebungsvariable einen positiven Einfluss hat

Umgebungsvariablen werden vor einem Gespräch entweder automatisch generiert oder durch den/die Nutzer/-in gesetzt.

Die Umgebungsvariable Alter (jung/alt) wird automatisch generiert und der/die Nutzer/-in kann die Umgebungsvariable „Privat“ oder „Geschäftlich“ auswählen. Alternativ kann die Auswahl einer Umgebungsvariable auch übersprungen werden.

Das automatische Generieren und manuelle Auswählen von Umgebungsvariablen ist noch erweiterbar. Die bisherige Implementation stellt einen konzeptionellen Beweis der Erfüllbarkeit von F-13 dar.

Im Laufe der Arbeit kam auf, dass die Eingabe eines statischen Alters bei Nutzerprofilen und Gesprächspartnern/-innen nicht sinnvoll ist, da sich dieses jährlich ändert. Fragen und Hinweise beziehen sich auf das Alter und Strike Up setzt automatisch Umgebungsvariablen zu diesem (z.B. Jung oder Alt). Aus diesem Grund wird Alter als Profil-Attribut entfernt. Stattdessen wird das Geburtsdatum verwendet. Aus diesem wird, bei einem Datenbankabruf des Profils, das Alter ermittelt. Somit aktualisiert sich das Alter eines Profils am Geburtstag.

7.4 Gespräch

Vor einem Gespräch wird aus der Liste aller verfügbaren Gesprächspartner/-innen ein/-e Gesprächspartner/-in ausgewählt oder alternativ ein/-e neue/-r Gesprächspartner/-in erstellt. Anschließend kann der/die Gesprächspartner/-in noch einmal bearbeitet werden. Bevor die Konversation beginnt wählt der/die Nutzer/-in noch manuell Umgebungsvariablen aus.

Die Listen bevorzugter Tags des/der Nutzers/-in, des Gegenübers und der Umgebungsvariablen werden addiert. Doppelte (dreifache, vierfache, ...) Tags sind dabei erlaubt. Das Gleiche passiert mit den entsprechenden Listen gemiedener Tags.

Ist die Listen fertig, so wird diese mit den entsprechenden Listen jeder verfügbaren Frage verglichen. Hierfür werden alle Tags der bevorzugten/gemiedenen Listen einzeln miteinander verglichen.

Kommt ein Tag sowohl in der Liste der bevorzugten (gemiedenen) Tags einer Frage, als auch in der kombinierten Liste bevorzugter Tags von Nutzers/-in, Gegenüber und Umgebungsvariablen vor, so wird bei der Frage ein Punkt addiert (subtrahiert). Fragen

starten dabei mit einer Punktzahl von null.



Abbildung 7.5: Warnung, wenn weniger relevante Fragen hinzugefügt werden

Für die Konversation werden nur Fragen mit einer Punktzahl über null ausgewählt. Sollten somit jedoch weniger als zehn Fragen verfügbar sein, so werden auch Fragen mit einer Punktzahl von null hinzugefügt. Sind immer noch weniger als zehn Fragen vorhanden, so werden auch negativ bewertete Fragen hinzugefügt. Der/die Nutzer/-in wird durch ein Pop-up auf diesen Umstand aufmerksam gemacht (vgl. 7.5).

Während einem Gespräch sieht die UI wie in 7.6 aus. Über die Buttons „vorherige“ und „nächste“ kann zwischen den für das Gespräch verfügbaren Fragen gewechselt werden (F-5). Der Text der Fragen wird zentral im Bildschirm angezeigt, während andere Funktionen, wie Buttons, am Rand vorhanden sind (NF-3). Das farbige Quadrat unter „Du“ zeigt die Bewertung der Frage aus Sicht des Nutzers an. Das Quadrat unter „Partner“ zeigt die Bewertung aus Sicht des Gegenübers und das große Quadrat zeigt die Gesamtbewertung. Eine grüne Farbe steht dabei für eine positive, gelb für eine neutrale und rot für eine negative Bewertung.

Über „Tabela Tingel“ wird das Profil des Gegenübers aufgerufen. Somit sind Notizen des Gegenübers während einem Gespräch verfügbar (F-7). Das Profil des/der Gesprächspartners/-in kann somit auch erst während einer Konversation mit Informationen befüllt werden.

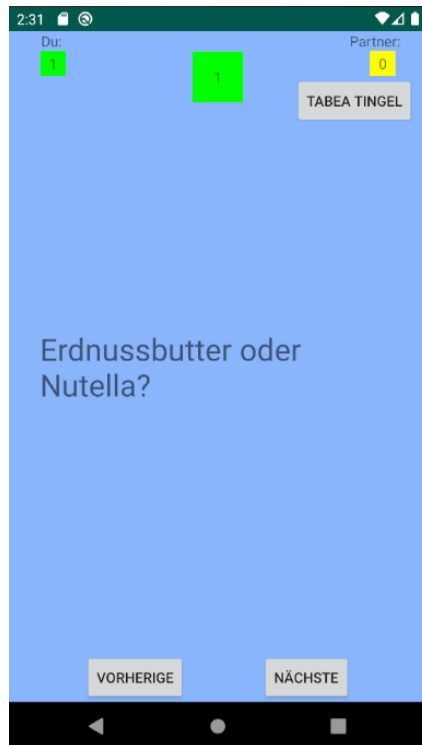


Abbildung 7.6: Frage während einer Konversation

7.5 Datenschutz

Strike Up soll keine personenbezogenen Daten an Dritte weitergeben (vgl. 4.3, NF-4). Als „Dritte“ ist hierbei Google in der Rolle des Verwalter der Daten, zu verstehen.

Google ist hierbei ein „processor of [...] Customer Personal Data under the European Data Protection Legislation“ [39]. Dies bedeutet, dass Google ein Verwalter der Daten ist, diese aber nicht einsehen oder versenden darf.

Da die zu Strike Up gehörende Datenbank von der EU aus bearbeitet wird, gilt die Datenschutz-Grundverordnung (DSGVO). Somit muss Google Daten an Behörden weiterleiten, falls dies gefordert wird.

Des Weiteren ist der Ersteller von Strike Up für das Einhalten der DSGVO verantwortlich. Um dies zu garantieren wird beim ersten Öffnen der Anwendung ein Dialogfenster geöffnet, welches den/die Nutzer/-in auffordert zu bestätigen, dass er/sie über 18 Jahre alt ist. Außerdem willigt er/sie ein, dass alle von Strike Up verarbeiteten Daten gespeichert werden.

Wird eine dieser zwei Bedingungen nicht erfüllt, so schließt sich Strike Up und der Ablauf wird beim nächsten Öffnen wiederholt.

7.5.1 Schutz vor Datendiebstahl

Damit Strike Up auf die Cloud-Datenbank zugreifen kann, wird eine Uniform Resource Locator (URL) zu dieser benötigt. Die URL ist dabei im Code der Anwendung enthalten. Diese URL ermöglicht Angreifern jedoch keinen Zugriff auf die Datenbank, da zudem noch das verbundene Google-Konto benötigt wird.

Eine weitere Angriffsmöglichkeit ist der im Quellcode hinterlegte API-Key, welcher Strike Up bei der Datenbank registriert. Über den API-Key können Angreifer auf alle in der Datenbank gespeicherten Daten zugreifen.

Dies wird durch das Aufstellen von Regeln verhindert. Regeln beziehen sich auf definierte Bereiche der Datenbank. Die für Strike Up definierten Regeln erlauben jedem/-r authentifizierten Nutzer/-in das Lesen der Bereiche „EnvironmentFilters“, „questions“ und „tags“ (vgl. 7.1).

Da der Bereich „users“ nutzerspezifische Daten enthält, ist nicht jedem/-r Nutzer/-in erlaubt alle darunter liegenden Knoten zu lesen. Jede/-r Nutzer/-in kann dabei nur auf „unter“ seiner/ihrer UID gespeicherte Daten zugreifen. Der/die Nutzer/-in hat für diesen Bereich Lese- und Schreibrechte.

Kapitel 8

Praxistests

Strike Up soll Menschen beim Eröffnen und Führen von Gesprächen helfen (4.2). Um die Wirksamkeit dieser Hilfestellungen zu testen, wurden mit der entwickelten Applikation Praxistests durchgeführt.

Gesprächspartner/-innen waren dabei zufällige Passanten auf der Straße und in Drogeriemärkten.

Die folgenden Erkenntnisse beruhen auf empirischen Erfahrungen.

Während den Tests erwies sich besonders das Vorbereiten auf das Gespräch als hilfreich. Präparierte Gesprächsthemen halfen dabei, eine Konversation interessant zu gestalten und am Laufen zu halten.

Da die Versorgung mit Gesprächsthemen durch Strike Up aus zeitlichen Gründen in dieser Arbeit nicht zureichend entwickelt wurde, wurden externe Quellen (News-Webseiten, Nachrichten im Fernsehen) hinzugezogen. Mit diesen Quellen wurde das von Strike Up gewünschte Verhalten nach F-9 und F-10 simuliert.

Auf sich selbst und das Gegenüber angepasste Fragen erwiesen sich als vergleichsweise durchschnittlich nützlich. Fragen entstanden eher spontan und aus den bereits vorbereiteten Themen heraus.

Das Vermeiden unpassender Fragen hingegen sorgte dafür, dass für sich selbst und das Gegenüber unangenehme Themen nicht angesprochen wurden. Dies verhinderte einen abrupten Abbruch der Konversation.

Eine unerwartete Entdeckung war, dass das Gegenüber, wenn es die Zuhilfenahme der App bemerkte, meist positiv mit einem Lachen reagierte. Als Grund für das Lachen wurde genannt, dass das Verwenden der App den/die Benutzer/-in sympathisch darstelle. Der/die Nutzer/-in würde eine Schwäche einsehen und versuche diese zu verbessern. Es kamen aber auch Fälle vor, bei welchen sich das Gegenüber ablenend gegenüber der App äußerte und das Gespräch somit beendete. Das Smartphone würde dabei dem/der Nutzer/-in die Konversationsführung wegnehmen und somit zu einer Verdummung führen.

Es wurde aber auch festgestellt, dass Strike Up um eine Schnellgespräch-Funktion erweitert werden sollte.

Will der/die Nutzer/-in spontan ein Gespräch beginnen, so muss zuerst ein/-e Gesprächspartner/-in ausgewählt werden und anschließend Umgebungsvariablen gesetzt werden. Dieser Prozess kann, besonders beim Erstellen eines/-r neuen Gesprächspartners/-in Zeit in Anspruch nehmen.

Um diesen Prozess zu umgehen, kann im Hauptbildschirm ein Button hinzugefügt werden, welcher sofort ein Gespräch startet. Die UI des Gespräches entspricht dabei der UI einer normalen Konversation. Das Profil des/der Gesprächspartners/-in wird während der Konversation entdeckt/erstellt.

Kapitel 9

Zusammenfassung

Mit dieser Arbeit sollte ein kontextabhhniger Kommunikationsassistent entwickelt werden (4.2). Dies wurde in Form einer Applikation namens Strike Up umgesetzt.

Strike Up ist eine fr Android entwickelte App, welche dem/der Nutzer/-in ermglicht ein eigenes Profil, sowie Profile von Gesprchspartnern/-innen, zu erstellen. Diese Profile enthalten bevorzugte und gemiedene Tags, anhand welcher whrend einer Konversation Fragen ausgewhlt und vorgeschlagen werden. Des Weiteren haben automatisch generierte und durch den/die Nutzer/-in ausgewhlte Umgebungsvariablen einen Einfluss auf die Auswahl der Fragen.

Fragen beziehen sich somit auf bevorzugte und weniger bevorzugte Themen des/der Nutzers/-in und des Gegenbers, sowie auf die Umgebung der Konversation.

Um Strike Up nutzen zu knnen, mssen sich Nutzer/-innen mit einer E-Mail-Adresse und einem Passwort anmelden. Anschlieend wird eine Altersbesttigung, sowie eine Einwilligung zur Speicherung der Daten gefordert.

Jegliche von Strike Up zur Verfgung gestellte, sowie durch den/die Nutzer/-in erstellte, Daten werden in der, von Google gehosteten, Cloud-Datenbank Firebase gespeichert. Nutzer/-innen knnen dabei alle Teile der Datenbank lesen, welche keine Informationen anderer Nutzer/-innen enthalten. Schreibzugriffe werden Nutzern/-innen nur bei selbst erstellten Daten (eigenes Profil, Gesprchspartner/-innen) gestattet.

Als besonders interessant haben sich die im Zuge dieser Arbeit durchgefhrten Praxis-

tests erwiesen.

Die zufällig ausgewählten Gesprächspartner/-innen reagierten meist positiv auf das Verwenden einer App als Hilfestellung und fanden dies sogar sympathisch.

Während einem Gespräch hat sich dabei die Vorbereitung auf das Gespräch als am hilfreichsten erwiesen. Weniger hilfreich war die Auswahl bevorzugter Fragen. Hierbei hat sich das Ausschließen unpassender Fragen als nützlicher erwiesen.

Kapitel 10

Ausblick

Bevor Strike Up im Google Play Store, oder firmenintern, veröffentlicht werden kann, müssen noch einige Features verbessert werden: Die Benutzeroberfläche sollte ansprechender werden und das editieren von Tags könnte, für eine bessere Übersicht, in einem (statt zwei verschiedenen) Bildschirm stattfinden.

Fragen und Hinweise können noch nicht durch den/die Nutzer/-in bewertet werden und ein Feedback am Ende eines Gespräches ist auch nicht vorhanden (vgl. 4.2, F-2, F-4, F-10). Des Weiteren können Fragen noch nicht aus der Auswahl ausgeschlossen werden (F-3). Auch die Versorgung mit aktuellen Themen, beispielsweise durch einen RSS-Feed, wurde nicht umgesetzt (F-8, F-9). Es gibt auch keine einfache Funktion, mit welcher ein/-e Nutzer/-in sein/ihr Konto und damit verbundene Daten löschen kann (F-11). Dies ist möglich, indem der Verwalter der Datenbank kontaktiert wird, jedoch ist diese Funktion nicht in Strike Up integriert.

In der aktuellen Form kann Strike Up noch in viele Richtungen verbessert und erweitert werden: Die Gesprächsunterstützung könnte von einer Unterstützung bei One-on-One-Gesprächen auf eine Unterstützung während Gruppengesprächen und Diskussionen erweitert werden.

Das Bewerten von Fragen kann durch ein Fragen-Rating, ähnlich wie bei der Fragenbewertung vor einem Gespräch, ermöglicht werden.

Nutzer/-innen könnten, durch drücken eines Buttons, die aktuell vorgeschlagene Frage in eine Liste ignorierte Fragen setzen. Vor einem Gespräch würden, für die Konversation

vorgeschlagene, Fragen mit der Liste der ignorierten Fragen verglichen werden. Wenn eine Fragen in beiden Listen vorkommt, so würde diese nicht während dem Gespräch vorgeschlagen werden.

Strike Up könnte während einer Konversation Fragen als Push-Benachrichtigung anzeigen. Somit müsste das Smartphone zum Einsehen einer Frage nicht mehr entsperrt werden.

Die App kann um eine Vorlese-Funktion erweitert werden. Fragen werden dem/der Nutzer/-in über Kopfhörer vorgelesen. Mit den Kopfhörfunktionen (überspringen, zurück) könnte zwischen Fragen gewechselt werden.

Des Weiteren könnte der aktuelle Standort benutzt werden um auf Attraktionen in der Nähe hinzuweisen oder um Wetterinformationen abzurufen. Die Wetterinformationen könnten dabei über OpenWeather [40] abgerufen werden.

Strike Up kann über eine zusätzliche App für Smartwatches erweitert werden. Während einem Gespräch könnten somit Fragen und Hinweise auf der Smartwatch angezeigt werden und das Smartphone bliebe somit in der Tasche.

Kapitel 11

Anhang

11.1 Personas

Eine Persona ist eine fiktionale Person. Das erstellen einer Persona kann helfen weitere Anforderungen an ein Projekt zu identifizieren, indem Personas mit unterschiedlichen Ausgangssituationen und Interessen kreiert werden.

11.1.1 Daniel Dacher



Abbildung 11.1: Generiertes Porträt von Daniel Dacher [41]

Name:

Daniel Dacher

Anfangsbuchstabe D wie Datenschutzbeauftragter.

Relevanz:

Ein Datenschutzbeauftragter wurde in der ersten Version der Stakeholderanalyse nicht berücksichtigt.

Wohnort:

Stuttgart

Firma:

Wir-schützen-Daten GmbH

Die Firma untersucht neue Softwarelösungen auf deren Verträglichkeit mit der DSGVO. Dies geschieht entweder Stichprobenweise oder auf Kundenanfrage.

Arbeitsort:

Stuttgart

Alter:

37 Jahre

Familienstand:

Verlobt

Kenntnisse unnd Fähigkeiten:

Ausbildung zum Datenschutzbeauftragten

DSGVO Datenschutzbeauftragter Zertifizierung

Aus der Persona „Daniel Dacher“ geht hervor, dass auf den Umgang mit personenbezogenen Daten geachtet werden muss.

Dies ist insbesondere relevant, da die Anforderung A-8 das Erstellen von persönlichen Profilen fordert.

11.1.2 Susanne Schmid



Abbildung 11.2: Generiertes Porträt von Susanne Schmid [42]

Name:

Susanne Schmid

Standardbenutzerin von Strike Up mit dem Anfangsbuchstaben S.

Relevanz:

Stellt den voraussichtlichen Durchschnittsbenutzer von Strike Up dar.

Wohnort:

Ehrenfeld, Köln

Ehrenfeld ist ein Stadtteil von Köln mit ungefähr 37.500 Einwohnern.

Firma:

Lebensmittelladen

Lebensmittelladen ist eine kleinere Version von ähnlichen Firmen, wie Aldi, Netto, Lidl und weiteren.

Arbeitsort:

Chorweiler, Köln

Der Weg zur Arbeit mit öffentlichen Verkehrsmitteln dauert ungefähr 30 Minuten.

Jobtitel/Rollen:

Susanne ist Assistant Managerin von Lebensmittelladen. Sie kümmert sich um den Vertrieb, geht gelegentlich aber auch an die Kasse.

Arbeitszeit:

40h-Woche typisch von 7:00-16:00

Alter:

35 Jahre

Familienstand:

Verheiratet, zwei Kinder

Kenntnisse unnd Fähigkeiten:

Ausbildung bei Lebensmittelladen

Persönliche Eigenschaften:

leicht reizbar/ungeduldig, hauptsächlich machen sich diese Eigenschaften wegen der frühen Arbeitszeit bemerkbar. Dies könnte zu allgemein schlechteren Reaktionen auf Vorschläge von Strike Up und das mehrfache Klicken von Buttons führen. offen für Neues, Susanne ist interessiert daran neue Menschen zu treffen und neue Dinge auszuprobieren

schüchtern/zurückhaltend, ihre schüchterheit hält Susanne oft davon ab ein Gespräch zu beginnen

11.1.3 Torsten Tacheles



Abbildung 11.3: Generiertes Porträt von Torsten Tacheles [43]

Name:

Torsten Tacheles

Anfangsbuchstabe T wie Trainer.

Relevanz:

Kommunikationstrainer der Strike Up benutzt um seinen Kenntnisstand zu erweitern und neue Ideen anzureichern.

Kann Strike Up an Kunden weiterempfehlen.

Strike Up war ursprünglich nur zur Gesprächsführung gedacht.

Wohnort:

Berlin

Firma:

Kommunikation Lernen 1on1

Torsten hat die Firma gemeinsam mit einem Freund gegründet. Es gibt keine weiteren Mitarbeiter.

Die Firma ist darauf spezialisiert Menschen Kommunikationstechniken und allge-

meine Rhetorik beizubringen.

Arbeitsort:

Berlin

Jobtitel/Rollen:

Torsten ist der Chef der Firma. Er kümmert sich neben Trainingskursen auch um alle weiteren Angelegenheiten der Firma.

Arbeitszeit:

45h-Woche typisch von 7:00-17:00

Alter:

57 Jahre

Familienstand:

Ledig

Kenntnisse unnd Fähigkeiten:

Studium der Psychologie

Weiterbildung zum Kommunikationstrainer

Online-Seminare um die Kenntnisse zum Leiten einer Firma zu erlangen.

Persönliche Eigenschaften:

gesprächsfreudig, Torstens Job ist es Menschen Kommunikationstechniken beizubringen, es ist für ihn gewissermaßen Pflicht gerne zu reden.

zynisch, Torstens Zynismus hat ihn manchmal bereits Kunden gekostet

Literatur

- [1] Tobias Roesse. *Lernen OHNE Stress. Wie Sie Ihrem Kind wirklich helfen können*. epubli, 2014.
- [2] *De.wikipedia.org, Alltagsgespräch*. URL: <https://de.wikipedia.org/wiki/Alltagsgespr%C3%A4ch> (besucht am 07.09.2020).
- [3] Dominik Birgelen. *Alles, was Sie über das Verkaufen wissen müssen: Ich und der Kunde*. Springer Gabler, 2013.
- [4] *Rhetorik-lernen.de, In 11 Mindsets zu einem starkem Selbstbewusstsein auf der Bühne*. URL: <https://rhetorik-lernen.net/selbstbewusstsein/> (besucht am 07.09.2020).
- [5] *Kommunikation-lernen.de, Blog*. URL: <https://kommunikation-lernen.de/blog/> (besucht am 20.02.2020).
- [6] Frank Naumann. *Die Kunst Des Smalltalk. Leicht ins Gespräch kommen, locker Kontakte knüpfen*. 2012.
- [7] *Play.google.com, Topicks - Gesprächsthemen und mehr!* URL: <https://play.google.com/store/apps/details?id=de.waghann.talking.release&hl=de> (besucht am 20.02.2020).
- [8] *Play.google.com, Beyond Small Talk*. URL: <https://play.google.com/store/apps/details?id=com.beyondsmalltalk.beyondsmalltalk1&hl=de> (besucht am 20.02.2020).
- [9] *Play.google.co, Trending - All in one app*. URL: <https://play.google.com/store/apps/details?id=com.anandroid.trending&hl=de> (besucht am 21.02.2020).
- [10] *Trends.google.com, Homepage*. URL: <https://trends.google.de/trends/?geo=DE> (besucht am 09.09.2020).

- [11] *Play.google.com, small talk*. URL: <https://play.google.com/store/apps/details?id=com.getsmalltalk&hl=de> (besucht am 21.02.2020).
- [12] *Play.google.com, Nwsty - Headlines & Daily Breaking News Summaries*. URL: <https://play.google.com/store/apps/details?id=com.instancea.nwsty&hl=en> (besucht am 26.02.2020).
- [13] *Play.google.com, Clio - Discover Nearby History and Walking Tours*. URL: <https://play.google.com/store/apps/details?id=com.theclio.clio&hl=en> (besucht am 26.02.2020).
- [14] *Db-engines.com, Ranking*. URL: <https://db-engines.com/de/ranking> (besucht am 19.02.2020).
- [15] Christof Ebert. *Systematisches Requirements Engineering. Anforderungen ermitteln, dokumentieren analysieren und verwalten*. Heidelberg: dpunkt.verlag, 2019.
- [16] Helmut Balzert. *Lehrbuch der Softwaretechnik. Basiskonzepte und Requirements Engineering*. Heidelberg: Spektrum Akad. Verl., 2009.
- [17] Chris Rupp. *Requirements-Engineering und -Management. Aus der Praxis von klassisch bis agil*. München: Carl Hanser Verlag, 2014.
- [18] *Marktforschung.de, EITO: Smartphonennutzung*. URL: <https://www.marktforschung.de/aktuelles/marktforschung/acht-von-zehn-menschen-in-deutschland-nutzen-ein-smartphone/> (besucht am 12.08.2020).
- [19] *Kantarworldpanel.com, Smartphone OS Market Share*. URL: <https://www.kantarworldpanel.com/global/smartphone-os-market-share/> (besucht am 19.06.2020).
- [20] *Statista.de, Anteil der Smartwatch-Nutzer in Deutschland in den Jahren 2015 bis 2019*. URL: <https://de.statista.com/statistik/daten/studie/1047586/umfrage/anteil-der-smartwatch-nutzer-in-deutschland/#statisticContainer> (besucht am 12.08.2020).
- [21] *Garmin.com, Homepage*. URL: <https://www.garmin.com/de-DE/> (besucht am 02.09.2020).
- [22] *Fitbit.com, Homepage*. URL: <https://www.fitbit.com/de/home> (besucht am 02.09.2020).

- [23] *Canalys.com, Worldwide smartwatch shipments Q1 2020*. URL: <https://www.canalys.com/newsroom/canalys-worldwide-smartwatch-shipments-q1-2020> (besucht am 02.09.2020).
- [24] *Developer.android.com, Save key-value data*. URL: <https://developer.android.com/training/data-storage/shared-preferences> (besucht am 03.09.2020).
- [25] *Github.com, Gson Java library*. URL: <https://github.com/google/gson> (besucht am 03.09.2020).
- [26] *Developer.android.com, Access app-specific files*. URL: <https://developer.android.com/training/data-storage/app-specific> (besucht am 03.09.2020).
- [27] *Sqlite.org, What is SQLite*. URL: <https://www.sqlite.org/index.html> (besucht am 03.09.2020).
- [28] *Firebase.google.com, Homepage*. URL: <https://firebase.google.com/> (besucht am 04.09.2020).
- [29] *Aws.amazon.com, DynamoDB*. URL: <https://aws.amazon.com/dynamodb/> (besucht am 04.09.2020).
- [30] *Aws.amazon.com, Free Tier*. URL: <https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc> (besucht am 04.09.2020).
- [31] *Azure.microsoft.com, Azure Cosmos DB*. URL: <https://azure.microsoft.com/de-de/services/cosmos-db/> (besucht am 04.09.2020).
- [32] *Heise.de, Eine Einführung in die Azure Cosmos DB*. URL: <https://m.heise.de/developer/artikel/Eine-Einfuehrung-in-die-Azure-Cosmos-DB-4446403.html?seite=all> (besucht am 04.09.2020).
- [33] *Developer.apple.com, Abschluss und Aktivierung*. URL: <https://developer.apple.com/de/support/purchase-activation/> (besucht am 04.09.2020).
- [34] *Play.google.com, Google Play Console*. URL: <https://play.google.com/apps/publish/signup/> (besucht am 06.09.2020).

- [35] *Medium.com, React Native vs iOS/Android Native: what is the best choice for mobile development today?* URL: <https://medium.com/@smartym.pro/react-native-vs-ios-android-native-what-is-the-best-choice-for-mobile-development-today-67cde42fc8bc> (besucht am 05.09.2020).
- [36] *Blog.logrocket.com, Flutter vs. React Native vs. Xamarin.* URL: <https://blog.logrocket.com/flutter-vs-react-native-vs-xamarin/> (besucht am 05.09.2020).
- [37] *Insights.stackoverflow.com, Most loved dreaded and wanted.* URL: <https://insights.stackoverflow.com/survey/2019#most-loved-dreaded-and-wanted> (besucht am 05.09.2020).
- [38] *Developer.android.com, ListView.* URL: <https://developer.android.com/reference/android/widget/ListView> (besucht am 08.09.2020).
- [39] *Firebase.google.com, Data processing terms.* URL: <https://firebase.google.com/terms/data-processing-terms> (besucht am 08.09.2020).
- [40] *Openweathermap.org, Homepage.* URL: <https://openweathermap.org/> (besucht am 09.09.2020).
- [41] *Thispersondoesnotexist.com, Daniel Dacher.* URL: <https://thispersondoesnotexist.com/> (besucht am 17.06.2020).
- [42] *Thispersondoesnotexist.com, Susanne Schmid.* URL: <https://thispersondoesnotexist.com/> (besucht am 17.06.2020).
- [43] *Thispersondoesnotexist.com, Torsten Tacheles.* URL: <https://thispersondoesnotexist.com/> (besucht am 18.06.2020).