



ulm university universität
uulm

Universität Ulm | 89069 Ulm | Germany

**Fakultät für
Ingenieurwissenschaften,
Informatik und Psychologie**
Institut für Datenbanken und
Informationssysteme

Konzeption und Implementierung eines Mobile Crowd Sensing Frameworks

Masterarbeit an der Universität Ulm

Vorgelegt von:

Tim Wiederhake
Tim.Wiederhake@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert
Dr. Vera Künzle

Betreuer:

Johannes Schobel, M.Sc.

2015

„Konzeption und Implementierung eines Mobile Crowd Sensing Frameworks“
Fassung vom 10. März 2016

© 2015 Tim Wiederhake

Dieses Werk ist unter der Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany
License lizenziert: <http://creativecommons.org/licenses/by-nc-sa/3.0/de>

Satz: PDF- \LaTeX 2 ϵ

Kurzfassung

Ein Großteil der Bevölkerung trägt täglich Geräte mit Internetverbindung und einer Vielzahl an Sensoren wie Mikrophon, Kamera, Beschleunigungsmesser und GPS-Empfänger bei sich: Smartphones.

Werden diese Geräte zu Sensornetzwerken verknüpft, können qualitativ und quantitativ hochwertige Daten gesammelt werden, deren Analyse vielfältig genutzt werden kann: Steuerung des Verkehrsflusses, Ausbalancierung von Stromnetzen, Erkennen von medizinischen Notfällen, Verbrechensbekämpfung und Marktanalyse sind nur einige von vielen Beispielen.

Dem Nutzen solcher Daten steht das Eindringen in die Privatsphäre und die Gefahr des Missbrauchs dieser Daten gegenüber, weswegen die Erhebung von persönlichen Daten durch das Bundesdatenschutzgesetz eingeschränkt ist.

In dieser Arbeit wird ein Framework zur massenhaften Erhebung von Daten durch aus Smartphones gebildeten Sensornetzen (Mobile Crowd Sensing), das mit den rechtlichen Vorgaben konform geht, konzipiert und implementiert.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	2
1.2	Aufbau der Arbeit	3
2	Grundlagen	5
2.1	Domänenwissen	5
2.2	Datenschutz	7
2.3	Verwandte Arbeiten	8
3	Anforderungen an ein Mobile Crowd Sensing Framework	11
3.1	Funktionale Anforderungen	11
3.2	Nichtfunktionale Anforderungen	13
4	Systemarchitektur	15
4.1	Komponenten	15
4.2	Datenbank	17
4.3	Datenformat der Fragebögen	19
4.4	REST-Schnittstelle	24
5	Implementierung der Server-Software	31
5.1	Laravel	31
5.2	Verwendung	32
6	Implementierung der Client-Software	39
6.1	Aufbau einer Android-Anwendung	39
6.2	Hauptmenü	40

Inhaltsverzeichnis

6.3	Einstellungen	42
6.4	Bearbeiten der Fragebögen	43
7	Fazit und Ausblick	47
7.1	Fazit	47
7.2	Ausblick	50
	Literatur	53
	A Beispielfragebogen	59
	Abbildungsverzeichnis	75

Kapitel 1

Einleitung

Smartphones sind ein Teil des täglichen Lebens geworden. In Europa war 2012 jedes zweite Mobiltelefon ein solches Gerät [26]. Diese Smartphones können Programme ausführen, ermöglichen mobilen Zugang zum Internet und besitzen unterschiedlichste Zusatzkomponenten. GPS-Empfänger, Kompass, Beschleunigungsmesser, Mikrofon, hochauflösende Kameras, Helligkeits- und Temperaturmesser gehören zur Palette der Sensoren, die auf modernen Smartphones fest verbaut und von Anwendungen ausgelesen und deren Daten verarbeitet werden können [2].

Die Erfassung, Nutzung, Verarbeitung und Auswertung dieser Daten ist Gegenstand der momentanen Forschung und Entwicklung. Einsatzgebiete sind beispielsweise psychologische Studien [30] und die Erfassung und Lenkung von Menschenmassen bei Großereignissen [29]. Abbildung 1.1 zeigt die Visualisierung der Positionsdaten der Teilnehmer einer Veranstaltung als Heatmap. Erkennbar sind die Dichte und Bewegung der einzelnen Teilnehmer. Mit derartigen Systemen sollen Überfüllung von Räumen und möglicherweise entstehende Paniken verhindert werden.



Abbildung 1.1: Visualisierung von mobil erfassten Positionsdaten als Heatmap an einem hypothetischen Beispiel.

1.1 Motivation

Die massenhafte Erhebung und Verarbeitung von Daten birgt das Risiko des Missbrauchs. Dem wissenschaftlichen und kommerziellen Interesse an der Nutzbarmachung dieser Sensoren steht das Recht auf Privatsphäre, informeller Selbstbestimmung und der Schutz des Persönlichkeitsrechtes, formuliert im Bundesdatenschutzgesetzes (BDSG) [40], gegenüber.

In bisherigen Systemen ist der Nutzer kaum oder gar nicht in der Lage, nachzuvollziehen, welche Daten erhoben werden und wo diese Daten wie lang gespeichert werden. Der Datenerheber sieht sich seinerseits vor der Schwierigkeit, die Nutzung dieser Daten rechtskonform abzuwickeln, da die verwendeten Systeme nicht oder nur teilweise darauf ausgelegt sind, erhobene Daten gemäß ihrer rechtlichen Bedeutung zu filtern.

Ziel dieser Arbeit ist die Konzeption und Implementierung einer Plattform zur mobilen, massenhaften Datenerhebung, unter Berücksichtigung der gesetzlichen Auflagen zur Speicherung personenbezogener Daten, sowohl auf dem Server als auch auf dem Client.

1.2 Aufbau der Arbeit

Diese Arbeit ist in 8 Kapitel unterteilt. Nach dieser Einleitung werden in Kapitel 2 einige Grundlagen für das weitere Verständnis erläutert. Dazu werden zunächst einige Begriffe aus dem Themenfeld des Mobile Crowd Sensing erläutert. Anschließend wird die Bedeutung und Relevanz des Datenschutzes diskutiert. Schließlich wird auf verwandte Arbeiten verwiesen.

In Kapitel 3 werden Anforderungen für die zu erstellende Software erarbeitet. Die aus diesen Anforderungen abgeleitete Systemarchitektur wird in Kapitel 4 vorgestellt. Insbesondere wird dabei auf den Aufbau der Datenbank, die interne Repräsentation der Fragebögen und deren Datenformat sowie die Webschnittstelle zur Kommunikation mit den Endgeräten eingegangen.

Anschließend wird in Kapitel 5 auf Details der Implementierung der Server-Software eingegangen. Neben der verwendeten Software wird die Verwendung des Webinterfaces vorgestellt. Kapitel 6 beschäftigt sich mit der Client-Software. Zunächst wird der generelle Aufbau einer Android-Anwendung dargelegt und dann die Benutzung der Client-Software erläutert.

Abschließend wird in Kapitel 7 ein Fazit dieser Arbeit gezogen und die Ergebnisse mit den anfangs gestellten Anforderungen abgeglichen. Schlussendlich wird ein Ausblick auf mögliche weitere Entwicklungen und Erweiterungen gegeben.

Kapitel 2

Grundlagen

In diesem Kapitel werden zunächst einige Grundbegriffe erklärt. Anschließend wird die Bedeutung des Datenschutzes erläutert und schließlich werden verwandte Arbeiten vorgestellt.

2.1 Domänenwissen

Dieser Abschnitt dient der Erläuterung und Bestimmung einiger Begriffe, die für das Verständnis der folgenden Kapitel benötigt werden:

Sensor Sensoren oder Messgrößen-Aufnehmer sind definiert als der Teil einer Messeinrichtung, der auf eine Messgröße unmittelbar anspricht [27]. Diese technischen Bauteile messen bestimmte physikalische oder chemische Eigenschaften, etwa Temperatur, Druck, Beschleunigung oder elektrische Spannung.

Crowd Sensing Unter Crowd Sensing versteht man das Erheben von Daten durch eine Menschenmenge [13]. Entweder werden durch die einzelnen Personen mitgeführte oder stationäre Sensoren ausgelesen oder aber die Personen fungieren selbst als Sensoren.

Dadurch können auch nichttechnische Messgrößen wie Stimmung, subjektive Eindrücke und Wahrnehmungen erhoben werden.

Smartphone Als Smartphone bezeichnet man ein Mobiltelefon, das nahezu die Funktionalität eines Computers besitzt und Programme installieren und ausführen kann [42]. Häufig besitzen Smartphones eine Reihe von internen Sensoren wie etwa Mikrofon, Kamera, Beschleunigungsmesser und GPS-Empfänger. Zumeist ist eine USB-Schnittstelle vorhanden, mit der das Smartphone mit einem Computer oder Zusatzgeräten wie externen Sensoren oder USB-Festplatten verbunden werden kann.

Mobile Crowd Sensing Werden beim Crowd Sensing Smartphones als Sensoren eingesetzt, so spricht man von Mobile Crowd Sensing [13]. Mobile Crowd Sensing unterscheidet sich von Crowd Sensing mit anderen Sensortypen insbesondere dadurch, dass durch die eingesetzten Smartphones die gesammelten Daten direkt vorverarbeitet und gefiltert, hochgeladen und Ergebnisse empfangen werden können. Dies ermöglicht ein unmittelbares Feedback für die Benutzer.

JSON Die JavaScript Object Notation [8], kurz JSON, ist ein menschenlesbares, schlankes Datenformat. JSON ist deutlich weniger komplex als beispielsweise XML und produziert weniger Overhead. Softwarebibliotheken zum Lesen und Schreiben von JSON-Daten sind für jede gängige Programmiersprache vorhanden, insbesondere PHP und Java. Die Programmiersprache JavaScript stellt hierbei eine Besonderheit dar, da JSON-Daten valide JavaScript-Programme darstellen und daher ohne zusätzlichen Aufwand direkt gelesen und verarbeitet werden können.

DataURI DataURI ist ein URI-Schema (`data://`), das das direkte Einfügen von binären Daten erlaubt, so als wären sie extern eingebunden [24]. Dies ist insbesondere dann nützlich, wenn die Menge an Daten gering und der Aufwand, die externen Daten zu laden, hoch ist. Dies ist beispielsweise bei kleinen Icons in einem HTML-Dokument der Fall, wo der

Mehraufwand eines weiteren HTTP-Requests in keinem Verhältnis zur Menge der Daten steht, einigen zehn bis hundert Bytes.

2.2 Datenschutz

Dieser Abschnitt erläutert rechtliche Grundlagen und Bedeutung des Datenschutzes im Bezug auf die Datenerhebung beim Mobile Crowd Sensing.

In Deutschland gilt seit 1977 das Bundesdatenschutzgesetz (BDSG), dessen Zweck es ist, „den Einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird“ [40, §1 (1)]. Die aktuelle Fassung dieses Gesetzes setzt die Richtlinie 95/46/EG des Europäischen Parlaments und des Rates vom 24. Oktober 1995 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr um.

Dem Interesse an der Erhebung von personenbezogenen Daten, etwa für akademische Forschung, kommerzielle Nutzung durch Erstellung von Kundenprofilen oder Verbrechensbekämpfung durch Sicherheitsbehörden, steht das Recht auf informelle Selbstbestimmung und der Datenschutz als Grundrecht gegenüber.

§ 4 Absatz 1 des BDSG [40] besagt, dass die Erhebung, Verarbeitung und Nutzung personenbezogener Daten prinzipiell verboten ist, außer sie ist explizit durch dieses Gesetz erlaubt oder durch den Betroffenen ausdrücklich gestattet worden. Die explizit erlaubten Fälle (§§ 12–14 BDSG) betreffen öffentliche Stellen, also Behörden. Für private oder akademische Zwecke ist daher nahezu immer die Zustimmung des Betroffenen, meistens in Form einer Registrierung der Benutzer, notwendig. Nichtöffentliche Stellen müssen die folgenden drei Hauptprinzipien des Datenschutzes befolgen:

Sparsamkeit Es müssen so wenig personenbezogene Daten wie möglich erhoben werden. Wenn möglich sollen diese Daten anonymisiert oder pseudonymisiert werden (§ 3a BDSG).

Erforderlichkeit Sobald die Daten nicht mehr für die Erfüllung der vorgesehenen Aufgabe erforderlich sind, sind sie zu löschen bzw. zu sperren (§ 35 BDSG).

Zweckbindung Erhobene Daten dürfen ausschließlich für den vorgesehenen Zweck verwendet werden (§§ 31 und 39 BDSG).

Besonders vor dem Hintergrund der klinischen Forschung mit Mobile Crowd Sensing ist es offensichtlich, dass persönliche Daten im Sinne des BDSG erhoben werden, und daher diese Grundprinzipien befolgt werden müssen.

Weitere relevante Gesetze sind das Telemediengesetz (TMG) [41] und das Gesetz über Rahmenbedingungen für elektronische Signaturen (SigG) [39]. Letzteres bleibt innerhalb dieser Arbeit unberührt, da keine elektronischen Signaturen verwendet werden.

Im Telemediengesetz werden unter anderem Vorschriften zur Impressumspflicht und zur Haftung im Falle von gesetzeswidrigen Inhalten gemacht. Da der konkrete Aufbau einer Befragung nicht Teil dieser Arbeit ist, findet auch dieses Gesetz hier keine Anwendung.

2.3 Verwandte Arbeiten

Dieser Abschnitt sortiert die vorliegende Arbeit in den wissenschaftlichen Kanon einiger Arbeiten zum Thema Mobile Crowd Sensing ein.

Der Artikel „Mobile crowdsensing: current state and future challenges“ [13] prägte zuerst den Begriff „Mobile Crowd Sensing“. Ausgehend vom Begriff „Internet of Things“ wird gezeigt,

dass Sensornetzwerke, bestehend aus Smartphones, sich qualitativ zu anderen Crowd Sensing Applikationen durch die hohe Anzahl, Verfügbarkeit und der stetig steigenden Rechenkapazität von Smartphones sowie der Zunahme an verfügbaren Sensoren unterscheidet.

Ein großer Vorteil von Mobile Crowd Sensing liegt in der weiten Verbreitung von Smartphones. Es erübrigt sich, eine große Menge an Messgeräten und Sensoren an Teilnehmer zu verteilen, da die Teilnehmer die benötigten Geräte bereits besitzen. Dadurch entfallen Kosten für Anschaffung, Logistik und Wartung.

Mehrere Arbeiten beschäftigen sich mit dem Nutzen der gewonnenen Daten, etwa Smart Cities [16], also vernetzte, reagierende Wohnumgebungen die beispielsweise proaktiv vor Staus warnen, Stromnetze ausbalancieren oder medizinische Notfälle erkennen können. Auch in der medizinischen Forschung wird Mobile Crowd Sensing eingesetzt, etwa bei der Tinnitus-Forschung [18, 30, 31].

Mit der Qualität der erhobenen Daten beschäftigen sich ebenfalls mehrere Arbeiten. So wird die Verknüpfung von maschinell erhobenen Daten mit von Menschen erhobenen Daten in sozialen Netzwerken beschrieben [15], sowie die Abdeckung möglichst großer räumlicher Bereiche durch optimale Übertragung und Weiterleitung der Daten [23]. Ein weiterer Forschungsansatz ist die Einführung formaler Schemata in der Erhebung von Daten durch Mobile Crowd Sensing [43].

Bereits existierende Mobile Crowd Sensing Applikationen sind beispielsweise McSense [36], Medusa [32] und Apisense [17]. Hierbei werden unterschiedliche Schwerpunkte gesetzt, etwa Benutzermotivation, Robustheit der Daten gegen Verfälschung und Einbeziehung des Benutzers als Auslöser für Messungen bei McSense [35].

Medusa wird seit Juli 2014 nicht weiterentwickelt [25]. Der aktuelle Stand der Entwicklung von Apisense ist schlecht zu beurteilen, da die Apisense-Webseite seit März 2014 [6] nur aus der Ankündigung „Coming soon“ besteht [5] und auch der Apisense-Twitteraccount seit Januar 2013 keine Aktivität zeigt [4].

Kapitel 3

Anforderungen an ein Mobile Crowd Sensing Framework

In diesem Kapitel werden die funktionalen und nichtfunktionalen Anforderungen an das zu erstellende Framework definiert.

3.1 Funktionale Anforderungen

Dieser Abschnitt definiert funktionale Anforderungen an das zu erstellende Framework.

AF1 (Aufbau als Fragebogen) Die Befragungen sollen in Form von Fragebögen stattfinden. Daher muss die Server-Software in der Lage sein, derartige Fragebögen zu erstellen und an die Client-Software auszuliefern.

AF2 (Mobile Anwendung) Die Client-Software muss als Anwendung für Smartphones konzipiert sein und die Fragebögen anzeigen, vom Benutzer ausfüllen lassen sowie die Ergebnisse zurück an die Server-Software schicken können. Insbesondere müssen die Fragebögen mit den begrenzten Ressourcen eines Smartphones verarbeitbar sein.

AF3 (Datenexport) Die Software muss über eine definierte Schnittstelle verfügen, die das Exportieren der gesammelten Daten aus dem System erlaubt. Um potentiell große und/oder lang laufende Befragungen zu unterstützen, muss diese Abfrage automatisch erfolgen können, also ohne Eingreifen eines menschlichen Akteurs.

AF4 (Vollständigkeit) Da die Software unter anderem in klinischen Studien eingesetzt werden können soll, muss sie gängige Befragungsmöglichkeiten unterstützen:

- Offene Fragen
- Geschlossene Fragen mit Einfach- und Mehrfachauswahl
- Numerische und verbale Ratingskalen

AF5 (Mehrsprachigkeit) Ein weiterer Aspekt der Universalität ist die Mehrsprachigkeit. So muss die Server-Software es ermöglichen, Fragebögen ganz oder teilweise zu übersetzen. Die Client-Software muss die Fragebögen entsprechend der eingestellten Präferenzen des Benutzers anzeigen.

AF6 (Wiederaufnehmbarkeit) Die Client-Software muss in der Lage sein, Unterbrechungen in der Bearbeitung der Fragebögen zuzulassen. Dies ist insbesondere deshalb wichtig, da Smartphones hauptsächlich unterwegs genutzt werden und keine unterbrechungsfreie Konzentration des Benutzers auf die Befragung vorausgesetzt werden kann.

AF7 (Rechteverwaltung) Mit der Software müssen den Benutzern unterschiedliche Rollen zugewiesen werden können, um eine grundlegende Rechteverwaltung zu ermöglichen. Nur dazu autorisierte Benutzer dürfen Befragungen erstellen und verändern können bzw. die Ergebnisse dieser Befragungen einsehen.

3.2 Nichtfunktionale Anforderungen

Dieser Abschnitt definiert die nichtfunktionalen Anforderungen an das zu erstellende Framework.

NFA1 (Datenschutz) Die in Deutschland geltenden Bestimmungen zum Datenschutz müssen durch Einhaltung der in Kapitel 2.2 diskutierten Hauptprinzipien des Datenschutzes erfüllt werden.

NFA2 (Universalität) Die in Kapitel 2.3 vorgestellten Mobile Crowd Sensing Applikationen sind insofern als Insellösungen zu betrachten, als dass sie jeweils nur spezielle, eng umrissene Probleme lösen. Ziel der zu erstellenden Software ist es, möglichst alle denkbaren Anwendungsfälle abzudecken.

NFA3 (Erweiterbarkeit der Sensortypen) Noch vor einigen Jahren war die Vielzahl, Genauigkeit und Verfügbarkeit von sensorbeladenen Smartphones undenkbar. Mit Blick in die Zukunft muss es die Software daher erlauben, neue Sensoren und Sensortypen unproblematisch einzubinden.

NFA4 (Bedienbarkeit) Die Bedienung sowohl der Client-Software auf dem Smartphone, als auch die Software zur Erstellung, Wartung und Auswertung der Befragung im Webbrowser sollte intuitiv und übersichtlich sein.

NFA5 (Wartbarkeit) Da die Software als Framework konzipiert wird, ist es besonders wichtig, den Quellcode gemäß der üblichen Best Practices des Software-Engineering zu strukturieren, dokumentieren und, wo möglich, ausschließlich Standardkomponenten zu verwenden. Dadurch soll erreicht werden, dass die Software in der Zukunft von Dritten einfach gewartet werden kann.

NFA6 (Erweiterbarkeit der Fragetypen) Die in Anforderung AF3 genannten Fragetypen stellen keine vollständige Liste aller existierenden Fragetypen dar. Sollten daher für einen Anwendungsfall zusätzliche Fragetypen benötigt werden, muss das Framework mit wenig Aufwand um weitere Fragetypen erweitert werden können.

Kapitel 4

Systemarchitektur

Aus den in Kapitel 3 dargelegten Anforderungen an ein Mobile Crowd Sensing Framework wird in diesem Kapitel die Architektur des Systems erarbeitet.

4.1 Komponenten

Im Rahmen dieser Arbeit wurde die Software „Monitor“ (Mobile Crowd Sensing Evaluation Framework) entwickelt. Dieses Framework besteht aus einer Client- und einer Server-Software.

Eine Applikation für das Betriebssystem Android stellt gemäß AF2 (Mobile Anwendung) die Client-Software dar. Diese wird von den Teilnehmern der Befragung verwendet, um Daten zu erheben. Sie ist in der Lage, die erhobenen Daten zu speichern und an die Server-Software zur Verarbeitung zu schicken. Die Client-Software ist außerdem in der Lage, nach dem Abschicken der Daten Rückmeldung durch die Server-Software, wie etwa Auswertungen zu empfangen und anzuzeigen. Ebenfalls ist es möglich, die Befragungen zu aktualisieren, falls diese auf dem Server verändert wurden, etwa bei nachträglichen Korrekturen.

Die Server-Software besteht aus einer Datenbank, Anwendungslogik und Web-Interface für die Erstellung, Wartung und Übersetzung der Befragungen. Über das Web-Interface können

die Benutzer des Systems außerdem die erhobenen Daten anzeigen und exportieren und die Verwaltung der Benutzer vornehmen.

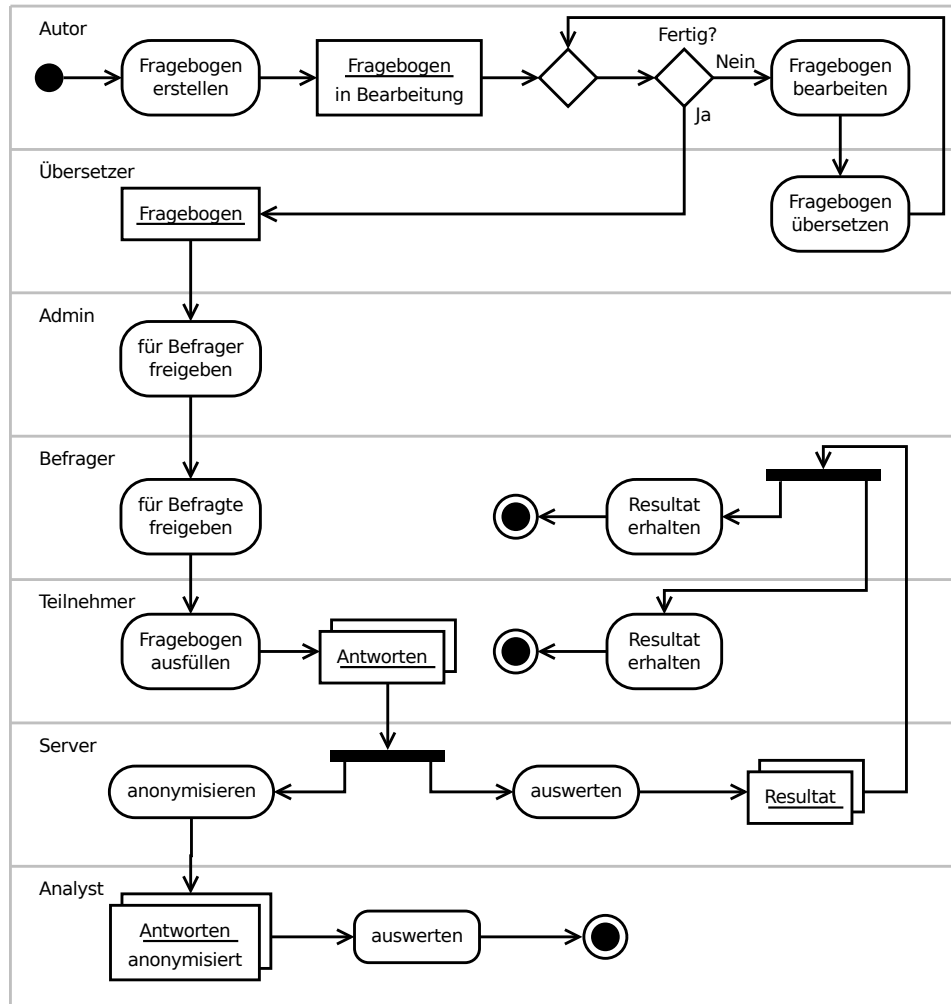


Abbildung 4.1: Der Ablauf einer Befragung als Aktivitätsdiagramm.

Abbildung 4.1 stellt den Ablauf einer Befragung mit Monitor als Aktivitätsdiagramm dar: Zunächst erstellt ein Autor mit der Server-Software in seinem Browser einen Fragebogen, siehe AF1 (Aufbau als Fragebogen). Dieser Fragebogen kann anschließend von einem Übersetzer übersetzt und von einem Autor bearbeitet werden. Wenn der Fragebogen fertig ist und die Befragung durchgeführt werden soll, wird der Fragebogen von einem Administrator für

die Befrager freigeschaltet. Diesen Befragern wiederum sind die Teilnehmer zugeordnet, welche nun den Fragebogen auf ihr Smartphone herunterladen können.

Ausgefüllte Fragebögen werden schließlich zum Server hochgeladen, ausgewertet und das Ergebnis Teilnehmer und Befrager mitgeteilt. Die ausgefüllten Fragebögen werden zusätzlich in anonymisierter Form (siehe Kapitel 5.2) gespeichert und Analysten zur Auswertung zur Verfügung gestellt.

4.2 Datenbank

Dieser Abschnitt beschreibt den Aufbau der verwendeten Datenbank.

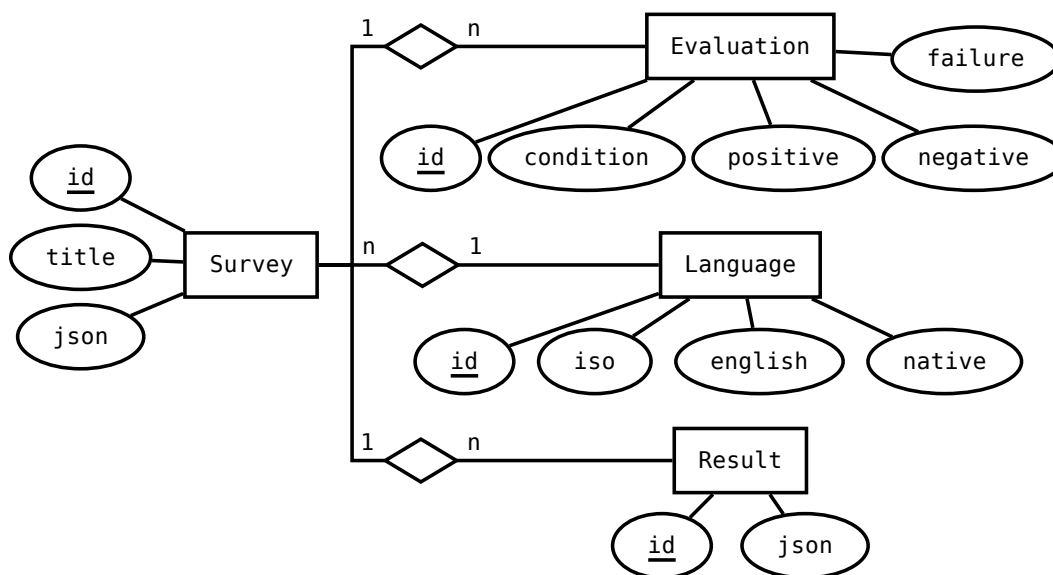


Abbildung 4.2: Ausschnitt des ER-Diagramms der Datenbank mit Fokus auf den Beziehungen zwischen Fragebogen, Auswertung und Resultate.

Abbildungen 4.2 und 4.3 stellen jeweils einen Ausschnitt der Relationen der Datenbank grafisch als ER-Diagramm dar. Die zentrale Relation der Datenbank ist der „Survey“ (Fragebogen), dem eine eindeutige Identifikationsnummer, ein Titel, die Fragebogenelemente

im JSON-Format, sowie eine Basissprache („Language“) zugeordnet ist. Die Basissprache eines Fragebogens ist diejenige Sprache, in der der Fragebogen ursprünglich erstellt wurde. Sie dient zudem als Ausweichlösung, falls der Fragebogen nur teilweise übersetzt wurde.

„Evaluation“ (Auswertung) sind Regeln, mittels derer der Server bereits ausgefüllte Fragebögen automatisiert auswertet. Die Antworten des Teilnehmers werden in die „Condition“ (Bedingung) der Auswertungen eingesetzt. Abhängig davon, ob die Bedingung nun erfüllt ist oder nicht oder ein Fehler bei der Auswertung auftrat, wird der entsprechende Textbaustein „positive“, „negative“ oder „failure“ ausgegeben.

Die ausgefüllten, anonymisierten Fragebögen werden in der Relation „Result“ zur späteren Auswertung durch Analysten gespeichert.

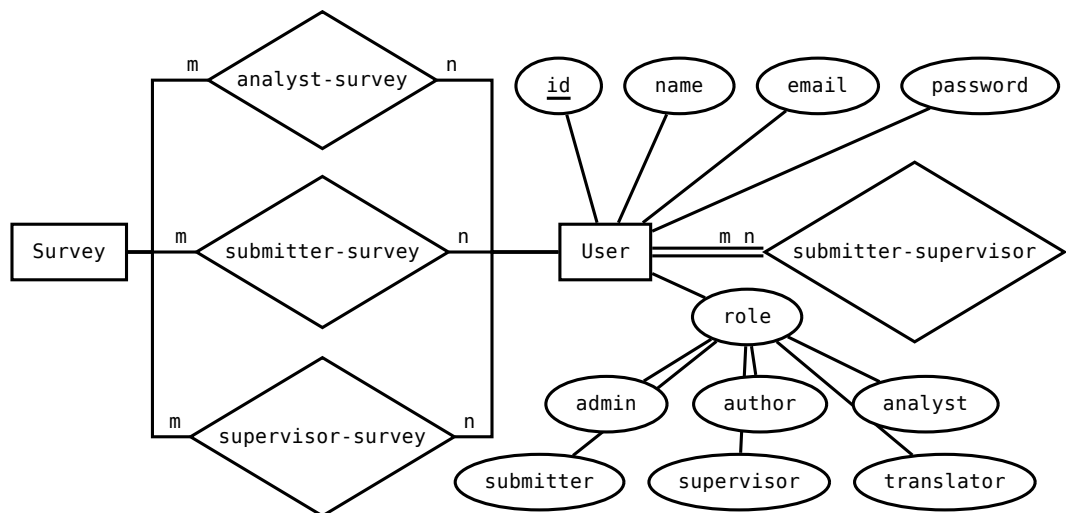


Abbildung 4.3: Ausschnitt des ER-Diagramms der Datenbank mit Fokus auf den Beziehungen zwischen den verschiedenen Benutzerrollen.

Die verwendeten Begriffe „Administrator“, „Autor“ und „Befrager“ sind Rollen, die ein Monitor-Benutzer innehaben kann. Wie in Abbildung 4.3 ersichtlich, existieren die Rollen „Admin“ (Administrator), „Author“ (Autor), „Analyst“, „Submitter“ (Teilnehmer), „Supervisor“ (Befrager) und „Translator“ (Übersetzer). Es ist möglich, dass ein Benutzer mehrere Rollen gleichzeitig inne hat.

Die dargestellten m-zu-n-Beziehungen repräsentieren die Zuordnung von Teilnehmer zu Befrager („submitter-supervisor“), Befrager zu Fragebogen („supervisor-survey“) und Teilnehmer zu Fragebogen („submitter-survey“) dar. Die Beziehung „analyst-survey“ definiert, welcher Analyst die anonymisierten Ergebnisse welcher Fragebögen einsehen darf.

4.3 Datenformat der Fragebögen

In diesem Abschnitt wird das Datenformat der Fragebögen erläutert.

Wie in Abschnitt 4.1 dargestellt, werden die eigentlichen Inhalte der Fragebögen im JSON-Format in der Datenbank gespeichert. Ein Fragebogen besteht aus einer Liste von Fragebogen-Elementen, die von jeweils einem JSON-Objekt repräsentiert werden. JSON-Objekte bestehen aus Key-Value-Paaren. Ein Key-Value-Paar mit dem Key „type“ ist stets vorhanden und definiert den Typ des jeweiligen Elements. Dies ist notwendig, da JSON nicht über die Möglichkeit verfügt, eigene Datentypen zu definieren.

Im Folgenden werden die unterschiedlichen Typen an Fragebogen-Elementen aufgezählt und die jeweiligen relevanten Key-Value-Paare dokumentiert. Diese Liste enthält die in AF4 (Vollständigkeit) genannten Fragetypen und erfüllt daher diese Anforderung. Anschließend wird die JSON-Repräsentation eines Beispielfragebogens analysiert.

Überschrift

Key	Value
type	„header“
text	Übersetzungen der Überschrift

Textblock

Key	Value
type	„text“
text	Übersetzungen des Textblocks

Trenner

Key	Value
type	„divider“

Seitenumbruch

Key	Value
type	„pagebreak“

Bild

Key	Value
type	„image“
image	Bilddaten als DataURI

Offene Frage

Key	Value
type	„textinput“
varname	Variablenname
inputtype	0 für einzeiligen Text, 1 für mehrzeiligen Text, 2 für Passworteingaben, 3 für Datumseingaben, 4 für Uhrzeiten, 5 für Zahlen
text	Übersetzungen für den angezeigten Hilfstext
optional	0 für mandatorische Eingaben, 1 für optionale Eingaben

Geschlossene Frage

Key	Value						
type	„question“						
varname	Variablenname						
min	Mindestzahl an zu gebenden Antworten						
max	Höchstzahl an zu gebenden Antworten						
options	<p>Liste an Antwortmöglichkeiten:</p> <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>value</td> <td>Zahlwert der Antwort</td> </tr> <tr> <td>text</td> <td>Übersetzungen des Antworttextes</td> </tr> </tbody> </table> <p>Lautet der Antworttext einer Antwortmöglichkeit „INPUT“, so wird ein Eingabefeld statt des Antworttextes angezeigt und der Befragte kann eine eigene Antwort eingeben.</p>	Key	Value	value	Zahlwert der Antwort	text	Übersetzungen des Antworttextes
Key	Value						
value	Zahlwert der Antwort						
text	Übersetzungen des Antworttextes						

Ratingskalen

Key	Value						
type	„rating“						
varname	Variablenname						
left	Übersetzungen für den Text des linken Pols						
right	Übersetzungen für den Text des rechten Pols						
min	Minimaler Zahlwert						
max	Maximaler Zahlwert						
value	Initialer Zahlwert						
step	Schrittgröße zwischen min und max						
display	1 wenn das Rating seinen Wert anzeigen soll, 0 wenn das Rating seinen Wert nicht anzeigen soll						
mapping	Liste an Abbildungen von Zahlwerten auf angezeigten Text: <table border="1" data-bbox="619 945 1401 1084"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>value</td> <td>Zahlwert der Antwort</td> </tr> <tr> <td>text</td> <td>Übersetzungen des Antworttextes</td> </tr> </tbody> </table>	Key	Value	value	Zahlwert der Antwort	text	Übersetzungen des Antworttextes
Key	Value						
value	Zahlwert der Antwort						
text	Übersetzungen des Antworttextes						

Dateiauswahl

Key	Value
type	„fileupload“
varname	Variablenname
size	Maximale Dateigröße in Byte
optional	0 für mandatorische Eingaben, 1 für optionale Eingaben

Sensordaten

Key	Value
type	„sensor“
varname	Variablenname
options	Sensoroptionen
optional	0 für mandatorische Eingaben, 1 für optionale Eingaben

```

1  [{
2    "type" : "header",
3    "text" : { "2" : "Fragen zur Kundenzufriedenheit" }
4  }, {
5    "type" : "text",
6    "text" : { "2" : "Würden Sie Produkt XYZ weiterempfehlen?" }
7  }, {
8    "type" : "question",
9    "varname" : "question",
10   "min" : 1,
11   "max" : 1,
12   "options" : [{
13     "value" : 0,
14     "text" : { "2" : "Ja" }
15   }, {
16     "value" : 1,
17     "text" : { "2" : "Vielleicht" }
18   }, {
19     "value" : 2,
20     "text" : { "2" : "Nein" }
21   }
22  ]
}]

```

Abbildung 4.4: JSON-Repräsentation der Elemente eines Beispielfragebogens.

Abbildung 4.4 zeigt ein Beispiel für die Elemente eines kleinen Fragebogens mit einer Überschrift, einem Text und einer Frage mit drei Antwortmöglichkeiten. Der Fragebogen ist nicht übersetzt, es existieren nur die Texte für die Sprache Nummer 2, der Basissprache des Fragebogens. Die Frage muss mit genau einer Antwort beantwortet werden (`min` und `max` sind beide „1“). Ein ausführlicheres Beispiel, das alle Elementtypen verwendet, befindet sich in Anhang A.

Dieses Datenformat erlaubt die Einführung neuer Fragetypen durch Erweiterung der möglichen Werte des „type“ Feldes der Elemente, erfüllt also Anforderung NFA6 (Erweiterbarkeit der Fragetypen).

4.4 REST-Schnittstelle

Dieser Abschnitt beschreibt die REST-Schnittstelle [11], über die Client- und Server-Software miteinander kommunizieren. Zudem können Analysten über diese Schnittstelle die für sie freigegebenen und anonymisierten Daten aus dem System exportieren, wodurch AF3 (Datenexport) erfüllt wird.

Die Rechtslage bestimmt, dass die Übertragung von persönlichen Daten verschlüsselt erfolgen muss (siehe Kapitel 2.2), so dass kein Dritter Zugriff auf diese Daten erhält. Um diese Anforderung zu erfüllen, erfolgt die Kommunikation zwischen Server und Client ausschließlich über HTTPS [33].

Das Datenformat der Schnittstelle ist ebenfalls JSON. Für jede Abfrage müssen die E-Mail-Adresse und das Passwort eines Benutzers als HTTP-GET-Parameter „email“ und „password“ angegeben werden.

Herunterladen der Liste der verfügbaren Fragebögen:

```
GET api/surveys
```

Rückgabewert ist ein assoziatives Array von Objekten, die die Kopfdaten der für den Teilnehmer freigegebenen Fragebögen darstellen. Die Schlüssel entsprechen jeweils der Identifikationsnummer des Fragebogens. Authentifizieren E-Mail und Passwort keinen Teilnehmer im System oder tritt ein anderer Fehler auf, so wird der HTTP-Fehlercode entsprechend gesetzt. Datenformat der Objekte:

Key	Value
title	Titel des Fragebogens
updated	Zeitstempel der letzten Änderung

Beispielhafte Antwort der Server-Software:

```
1 {
2   "1" : {
3     "title" : "Minifragebogen aus Objekten",
4     "updated" : 1439673672
5   },
6   "2" : {
7     "title" : "\u00dcbersetzungen",
8     "updated" : 1439673672
9   },
10  "3" : {
11    "title" : "Demofragebogen",
12    "updated" : 1439673672
13  }
14 }
```

Herunterladen eines Fragebogens:

```
GET api/survey/{survey_id}
```

Rückgabewert ist der Fragebogen mit der gegebenen Identifikationsnummer „survey_id“. Authentifizieren E-Mail und Passwort keinen Teilnehmer im System, ist der Fragebogen nicht für den Teilnehmer freigegeben oder tritt ein anderer Fehler auf, so wird der HTTP-Fehlercode entsprechend gesetzt. Datenformat des Objekts:

Key	Value
updated_at	Zeitstempel der aktuellen Fassung
language	Identifikationsnummer der Basissprache des Fragebogens
json	Fragebogen-Elemente im JSON-Datenformat gemäß Kapitel 4.2

Für ein Beispiel der Antwort der Server-Software siehe Abbildung 4.4 in Kapitel 4.3.

Absenden der Daten eines ausgefüllten Fragebogens:

```
POST api/result/{survey_id}
```

Die gegebenen Antworten müssen, im JSON-Datenformat, als HTTP-POST-Parameter „result“ übergeben werden.

Rückgabewert ist die Auswertung der Antworten durch den Server, mittels der in Kapitel 4.1 erwähnten Regeln. Üblicherweise ist dies ein HTML-Dokument. Zusätzlich wird die Auswertung per E-Mail an die zugeordneten Befrager des Teilnehmer gesandt.

Authentifizieren E-Mail und Passwort keinen Teilnehmer im System, ist der Fragebogen nicht für den Teilnehmer freigegeben oder tritt ein anderer Fehler auf, so wird der HTTP-Fehlercode entsprechend gesetzt.

Herunterladen der anonymisierten Antworten für einen Fragebogen:

```
GET api/result/{survey_id}
```

Rückgabewert ist ein Array von Objekten, die die Antworten auf den gegebenen Fragebogen darstellen. Authentifizieren E-Mail und Passwort keinen Analyst im System, ist der Fragebogen nicht für den Analyst freigegeben oder tritt ein anderer Fehler auf, so wird der HTTP-Fehlercode entsprechend gesetzt.

Key	Value
id	Identifikationsnummer der Antwort
survey_id	Identifikationsnummer des Fragebogens
created_at	Datum der Erstellung der Antwort
updated_at	Datum der letzten Änderung der Antwort
json	Gegebene Antworten im JSON-Datenformat

Beispielhafte Antwort der Server-Software:

```
1  [
2      {
3          "id" : 2
4          "survey_id" : 3,
5          "updated_at" : "2015-10-19 13:17:32",
6          "created_at" : "2015-10-19 13:17:32",
7          "json" : "...",
8      },
9      {
10         "id" : 7
11         "survey_id" : 3,
12         "updated_at" : "2015-10-19 13:33:29",
13         "created_at" : "2015-10-19 13:33:29",
14         "json" : "...",
15     },
16 ]
```

Abfrage der vorhandenen Sprachen:

GET api/languages

Rückgabewert ist ein Array von Objekten, die die vorhandenen Sprachen darstellen. Authentifizieren E-Mail und Passwort keinen Benutzer im System oder tritt ein anderer Fehler auf, so wird der HTTP-Fehlercode entsprechend gesetzt. Datenformat der Objekte:

Key	Value
id	Identifikationsnummer
iso	ISO-Ländercode [28]
english	Name der Sprache auf englisch
native	Name der Sprache in der jeweiligen Sprachen
created_at	Datum der Erstellung des Fragebogens
updated_at	Datum der letzten Änderung des Fragebogens
image	Bilddaten eines Icons oder einer Flagge, die die Sprache repräsentiert als DataURI

Beispielhafte Antwort der Server-Software:

```
1 {
2   "1" : {
3     "id" : 1,
4     "created_at" : "2015-08-15 21:21:12",
5     "updated_at" : "2015-08-15 21:21:12",
6     "iso" : "en",
7     "english" : "english",
8     "native" : "english",
9     "image" : "data:image/png;base64,..."
10  },
11  "2" : {
12    "id" : 2,
13    "created_at" : "2015-08-15 21:21:12",
14    "updated_at" : "2015-08-15 21:21:12",
```

```
15     "iso" : "de",
16     "english" : "german",
17     "native" : "deutsch",
18     "image":"data:image/png;base64,..."
19 },
20 "3" : {
21     "id" : 3,
22     "created_at" : "2015-08-15 21:21:12",
23     "updated_at" : "2015-08-15 21:21:12",
24     "iso" : "fr",
25     "english" : "french",
26     "native" : "fran\u00e7ais",
27     "image":"data:image/png;base64,..."
28 }
29 }
```


Kapitel 5

Implementierung der Server-Software

In diesem Kapitel wird die für die Programmierung der Server-Software verwendete Softwarebibliothek Laravel und die Verwendung der Server-Software beschrieben.

5.1 Laravel

In diesem Abschnitt wird die verwendete Softwarebibliothek Laravel vorgestellt.

Die Server-Software wurde in PHP (Version 5.5.9) unter Verwendung des Frameworks „Laravel“ [22] (Version 5.1) geschrieben. PHP ist eine an C und Perl angelehnte Skriptsprache, die bei den meisten Webhostern vorinstalliert ist (Verfügbarkeit: 81.5% [38]). Dies erfüllt, im Gegensatz zu Lösungen mit etwa Apache Tomcat (Verfügbarkeit: 3% [38]), Anforderung NFA5 (Wartbarkeit).

Laravel ist eine Softwarebibliothek für PHP, das dem Model-View-Controller-Muster [12] folgt und die Programmierung mit PHP vereinfacht bzw. die Fähigkeiten von PHP erweitert, etwa durch Abstraktion der Datenbankbindung.

Laravel zeichnet sich durch hervorragende Dokumentation und Lehrmaterialien, etwa den „Laracasts“ [21] genannten Lehrvideos, Unit-Tests und der freizügigen Lizenzierung aus.

Die Erzeugung eines neuen Laravel-Projektes geschieht durch Composer [9], einem Paketmanager für PHP:

```
composer create-project laravel/laravel your-project-name
```

Das Model-View-Controller-Muster [12] besagt, dass die Trennung von Daten (Model), Präsentation (View) und die Programmsteuerung (Controller) flexible und skalierbare Anwendungen ermöglichen.

Das Datenmodell wird in Laravel von „Eloquent“ bereitgestellt. Diese Komponente ermöglicht den objektorientierten Zugriff auf Relationen der Datenbank und stellt häufige Operationen, wie das Suchen bestimmter Elemente, Einfüge- oder Löschoptionen unabhängig von der zugrunde liegenden Datenbank, etwa MySQL, SQLite oder PostgreSQL, bereit.

Mit „Blade“ steht in Laravel eine umfangreiche Template-Engine bereit, die den Aufwand der Präsentation dieser Daten stark reduziert: Über eine spezielle Syntax (doppelte geschweifte Klammern) können PHP-Code und Blade-Kontrollstrukturen zusammen in (HTML-) Dokumenten eingebettet werden.

Die Programmsteuerung wird in Laravel durch Controller gehandhabt. Diese werden durch sogenannte Routen (Routes) einer oder mehreren URIs zugeordnet. Insbesondere die Erstellung von REST-Schnittstellen [11] zur Verwaltung von Ressourcen ist so sehr komfortabel möglich.

5.2 Verwendung

In diesem Abschnitt wird die Verwendung der Server-Software beschrieben.

Die hauptsächliche Aufgabe der Administratoren der Server-Software ist die Verwaltung der Systemsprachen und der Benutzer. Abbildung 5.1 zeigt die Administrationsoberfläche für einen Beispielbenutzer. Angezeigt wird der Name, die E-Mail-Adresse des Benutzers sowie die ihm zugeordneten Rollen. Da der Benutzer die Rolle „Befrager“ (Supervisor) inne hat,

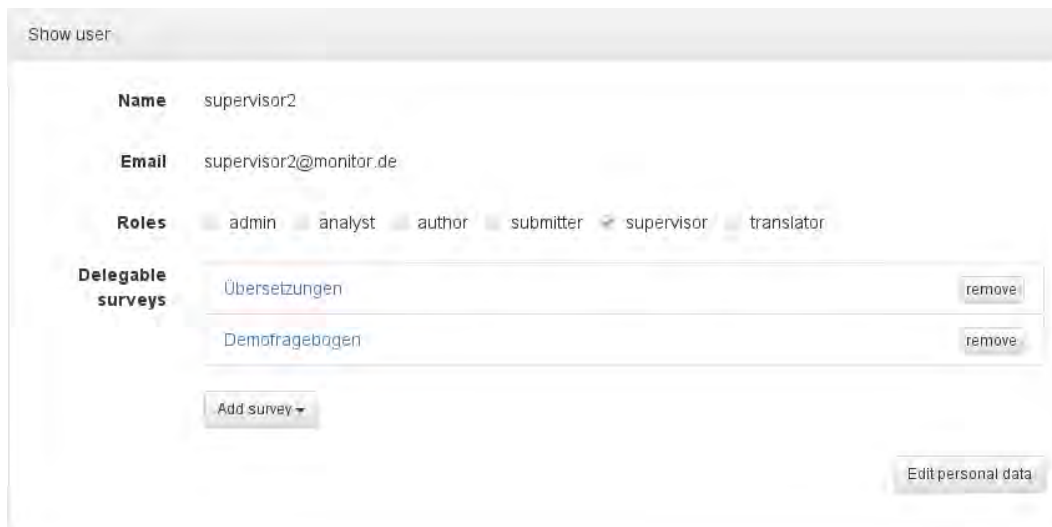


Abbildung 5.1: Benutzerverwaltung mit Monitor.

wird zusätzlich eine Liste an Fragebögen angezeigt, zu deren Verteilung dieser Benutzer berechtigt ist. Schaltflächen für die Bearbeitung dieser Liste sind vorhanden.

Wie aus Abbildung 4.1 in Kapitel 4 ersichtlich, ist der erste Schritt in einer Befragung die Erstellung bzw. Bearbeitung eines Fragebogens durch einen Autor. Abbildung 5.2 zeigt die Oberfläche zur Erstellung und Bearbeitung von Fragebögen. Unter den Kopfdaten, wie dem Titel und der Basissprache des Fragebogens, werden die einzelnen Elemente des Fragebogens angezeigt. Auf der rechten Seite befindet sich ein Katalog von Fragebogen-Elementen, die per Drag-and-Drop in den Fragebogen gezogen werden können. Ebenfalls per Drag-and-Drop lässt sich die Reihenfolge der Elemente verändern.

Zu sehen ist ebenfalls für jedes Element im Fragebogen eine Schaltfläche zum Entfernen des jeweiligen Elements und für das Eingabeelement „geschlossene Frage“ (Closed question) eine Schaltfläche, die die Sichtbarkeit der zu gebenden Antwort auf diese Frage bestimmt (offenes oder geschlossenes Auge-Symbol). Hier kann zwischen „öffentlich“ (public) und „privat“ (private) gewählt werden. Öffentliche Antworten stehen Analysten zur Verfügung, private Antworten (bei denen dieses boolesche Flag gesetzt ist) werden aus den Resultaten ausgefiltert, siehe Anforderung NFA1 (Datenschutz).

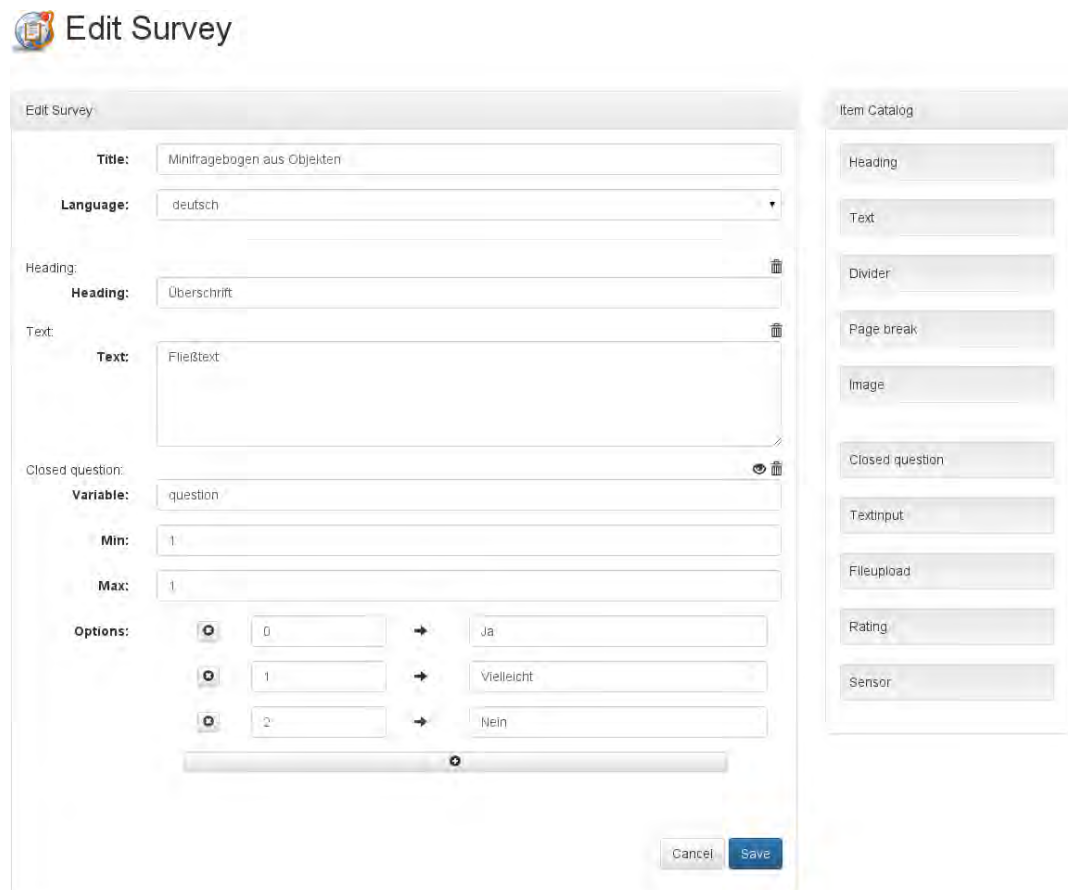
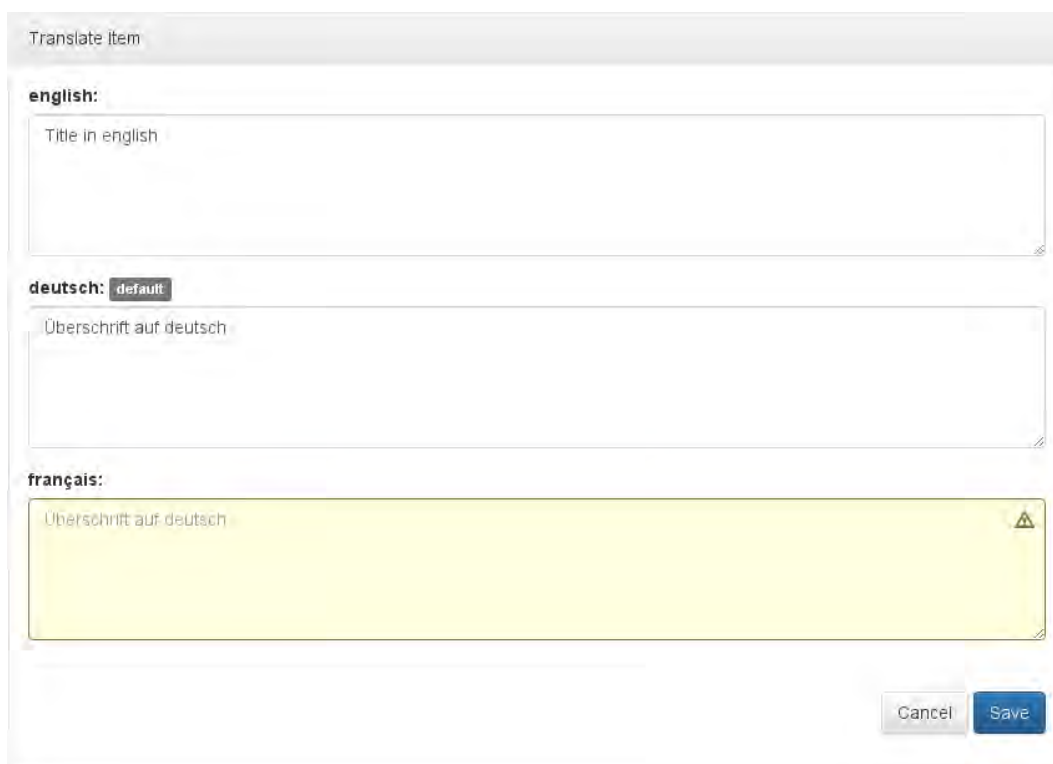


Abbildung 5.2: Oberfläche zur Erstellung und Bearbeitung von Fragebögen.

Übersetzer können im Gegensatz zu Autoren die Struktur des Fragebogens nicht ändern, sondern nur die Texte der existierenden Elemente editieren. Es ist Autoren möglich, den Fragebogen zu editieren, wenn die Übersetzung bereits begonnen wurde, wobei allerdings die Übersetzungen editierter Elemente ganz oder teilweise verloren gehen. Die Übersetzungsoberfläche für Fragebögen ist in Abbildung 5.3 dargestellt. Zu sehen ist, dass im System die Sprachen Englisch, Deutsch und Französisch eingestellt sind und der gerade übersetzte Text zu einem Fragebogen gehört, der in deutscher Sprache erstellt wurde.



The screenshot shows a 'Translate Item' dialog box with three text input fields. The first field is labeled 'english:' and contains the text 'Title in english'. The second field is labeled 'deutsch: default' and contains the text 'Überschrift auf deutsch'. The third field is labeled 'français:' and contains the text 'Überschrift auf deutsch', with a yellow background and a warning icon in the top right corner. At the bottom right, there are 'Cancel' and 'Save' buttons.

Abbildung 5.3: Übersetzung eines Fragebogenelements.

Da für Französisch noch keine Übersetzung existiert, ist dieses Feld farblich besonders hervorgehoben und der Originaltext als Hilfestellung für den Übersetzer eingeblendet. Dies erfüllt Anforderung AF5 (Mehrsprachigkeit).

Edit Evaluation Rule

Condition:

`$liter_per_day < 2`

Positive:

`<h1>Sie sollten mehr trinken.</h1>`
`<p>Es wird empfohlen, täglich mindestens zwei Liter Wasser zu trinken.</p>`

Negative:

`<h1>Gut so!</h1>`
`<p>Mit $liter_per_day Litern Wasser liegen Sie über den empfohlenen zwei Litern Wasser am Tag.</p>`

Failure:

`<p class="warning">Auswertung fehlgeschlagen: $exception</p>`

Delete Cancel Save

Abbildung 5.4: Bearbeitung einer Regel zur automatischen Auswertung der Fragebögen.

Der letzte Schritt vor der Fertigstellung eines Fragebogens ist die Programmierung der Auswertung der Antworten durch einen Autor. Abbildung 5.4 zeigt eine solche Auswertung. In der Bedingung (Condition), den beiden Fällen für erfüllte und nicht erfüllte Bedingungen (Positive und Negative) sowie dem Fehlerfall (Failure) können die Antworten als Variablen verwendet werden und mit HTML zur Anzeige formatiert werden.

Bei der automatischen Auswertung wird aus der Liste der Antworten wird eine Ersetzungsliste generiert, die jeweils dem `varname` genannten Feld eines Elementes den Wert der Antwort zuordnet. Da geschlossene Fragen mehr als eine Antwort haben können, wird dieser Fall besonders behandelt.

Für jede Antwortmöglichkeit der geschlossenen Frage werden zwei Ersetzungen generiert, jeweils nach dem Muster `varname_nummer` und `varname_nummer_text`, wobei „varname“ der verwendete Variablenname und „nummer“ der Zahlwert der Antwort. Für den Beispielfragebogen aus Abbildung 4.4 in Kapitel 4.3 sähe die Ersetzungsliste bei Auswahl der Antwort „Ja“ folgendermaßen aus:

Variable	Ersetzung
question_0	true
question_0_text	"Ja"
question_1	false
question_1_text	"Vielleicht"
question_2	false
question_2_text	"Nein"

`varname_nummer` enthält jeweils einen booleschen Wert, ob die entsprechende Antwort ausgewählt wurde oder nicht. `varname_nummer_text` beinhaltet jeweils den Antworttext. Dieser ist bei Ergänzungsmöglichkeiten (Antworttext `INPUT`, siehe Kapitel 4.3) mit dem Text der gegebenen Ergänzung gefüllt.

Kapitel 6

Implementierung der Client-Software

In diesem Kapitel wird der Aufbau einer Android-Anwendung und die Verwendung der erstellten Applikation beschrieben.

6.1 Aufbau einer Android-Anwendung

Jede Android-Anwendung besteht aus einer oder mehreren Aktivitäten (Activity). Eine Aktivität entspricht einer Bildschirmmaske und soll gemäß der Android Design Guidelines genau einen wohldefinierten Zweck erfüllen [1].

Die Oberfläche einer Aktivität wird durch ihre Layout-Datei im XML-Format definiert. Diese Dateien können von Hand erstellt werden, es gibt aber auch Programme, wie Android Studio [3], mit denen Layout-Dateien grafisch im Baukastenprinzip zusammengestellt werden können.

Neben der Entkopplung von Präsentation und Programmsteuerung ist ein weiterer Vorteil, dass in Android mehrere Layout-Dateien für die selbe Aktivität definiert werden können. Diese können für verschiedene Bildschirmgrößen angepasst werden. Android wählt dann zur Laufzeit das zur Bildschirmgröße und -orientierung am besten passende Layout für die Präsentation der Daten. Auf diese Weise wird erreicht, dass Android-Anwendungen den verfügbaren Bildschirmplatz stets optimal nutzen können.

Texte in Layouts, etwa für Beschriftungen, werden nicht direkt eingefügt, sondern über den Umweg eines symbolischen Markers. Erst zur Laufzeit des Programmes werden aus einer weiteren XML-Datei die zu den Markern gehörenden Texte geladen. Ähnlich wie bei den Layout-Dateien können mehrere Sprachdateien mit Texten für unterschiedliche Sprachen definiert werden. Dies vereinfacht die Übersetzung von Android-Programmen erheblich.

Das Wechseln, Aufrufen und die Interaktion zwischen Aktivitäten findet durch Intents (Absichtserklärungen) statt. Android unterscheidet zwischen impliziten und expliziten Intents. Explizite Intents interagieren mit einer ausdrücklich genannten Aktivität, die durch ihre Java-Klasse adressiert wird. Implizite Intents bestehen aus einem Aktionsverb aus einer vorgegebenen Liste, einem MIME-Datentyp und beliebigen Daten. Abhängig vom Aktionsverb kann die Angabe des Datentyps optional sein. Android wählt selbstständig aus der Liste der im System installierten Aktivitäten diejenigen, die angeben, die Kombination aus Aktionsverb und Datentyp / Daten handhaben zu können, zeigt dem Benutzer gegebenenfalls eine Auswahlliste der möglichen Aktivitäten an und startet letztlich diese [1].

Ein implizites Intent mit dem Aktionsverb „view“ (ansehen), dem Datentyp „text/html“ und einer URL als Daten startet beispielsweise einen Webbrowser, unabhängig davon, welchen Browser der Benutzer installiert hat. Dies führt zu robusteren Programmen, da Abhängigkeiten zu konkreten Softwarepaketen vermieden und durch Abhängigkeiten zu Funktionalitäten ersetzt werden.

6.2 Hauptmenü

Das Hauptmenü und gleichzeitig die Startaktivität der Applikation ist in Abbildung 6.1 (links) dargestellt. Zwei Fragebögen, „Demofragebogen“ und „Übersetzungen“, stehen zur Auswahl. Von einem Fragebogen liegt eine ausgefüllte Instanz vor, die entweder weiterbearbeitet oder abgeschickt werden kann. Anforderung AF6 (Wiederaufnehmbarkeit) wurde erfüllt. Die verwendeten Symbole sind aus dem „Tango“-Iconset [37], einer Sammlung an gemeinfreien Icons, entnommen und wurden für diese Anwendung angepasst.

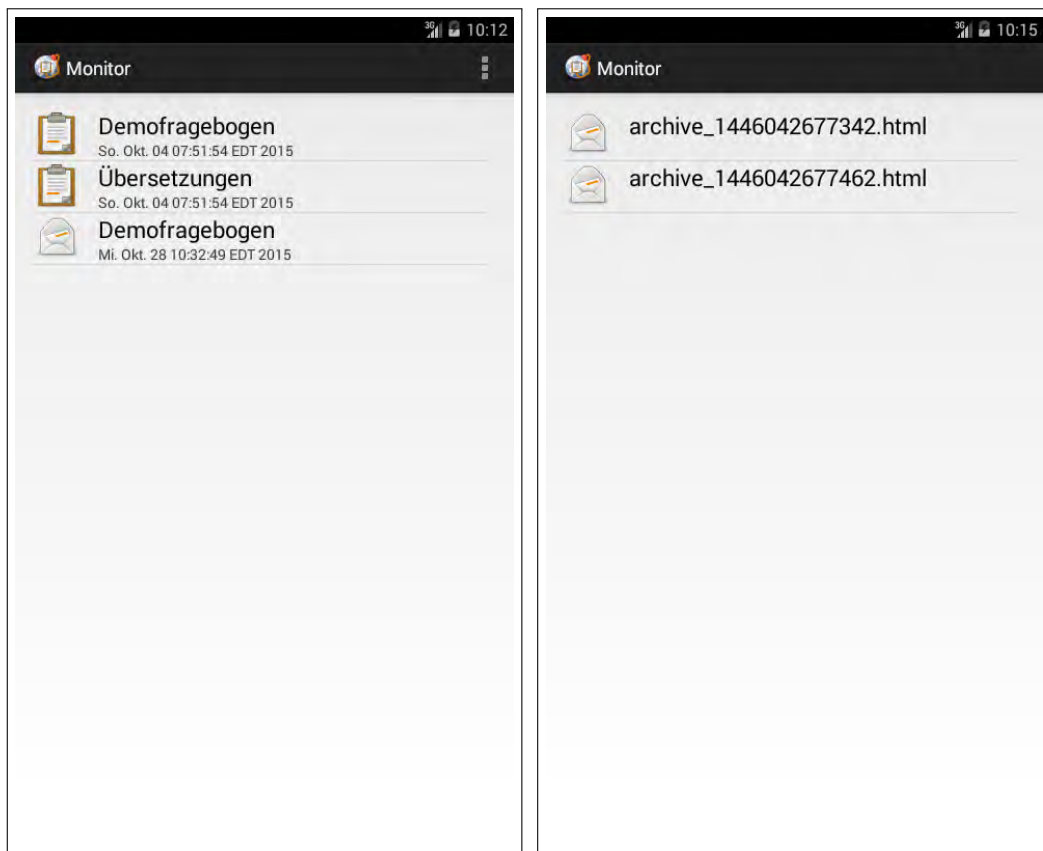


Abbildung 6.1: Hauptmenü (links) und Archiv (rechts) der Android-Anwendung.

Über das Menü (Button oben rechts) können Einstellungen vorgenommen, die Liste an Fragebögen aktualisiert und auf das Archiv der Resultate zugegriffen werden. Das Archiv ist in Abbildung 6.1 (rechts) zu sehen. Durch Antippen der Einträge können die vom Server zurückgeschickten Auswertungen der Befragungen erneut eingesehen werden. Hierbei handelt es sich um statische HTML-Dateien, die vom Browser des Benutzers angezeigt werden können.

6.3 Einstellungen

Mehrere Einstellungen können in der Client-Software vorgenommen werden. Die Angabe von E-Mail-Adresse und Passwort authentifizieren den Benutzer gegenüber der Server-Software und sind für die Benutzung des Programms unabdingbar. Wie in Anforderung AF7 (Rechteverwaltung) festgelegt, dürfen nur registrierte und authentifizierte Benutzer auf die Befragungen zugreifen.

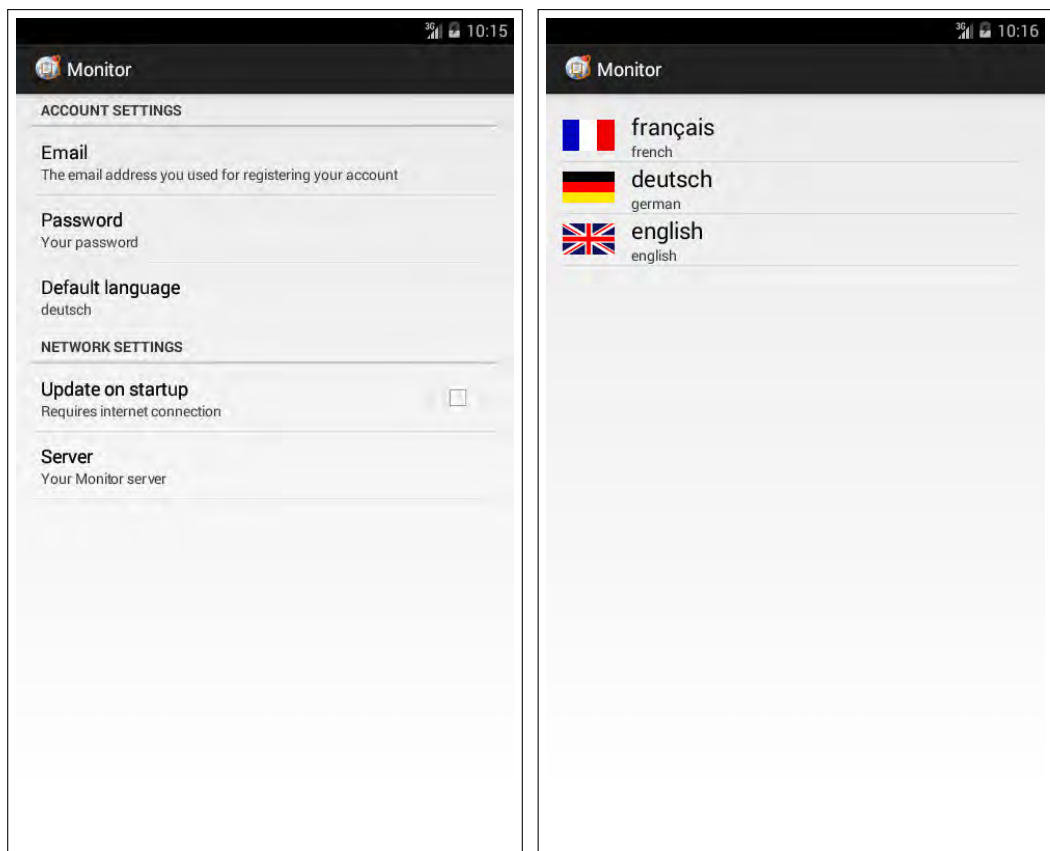


Abbildung 6.2: Einstellungsmöglichkeiten (links) und Auswahl der bevorzugten Sprache (rechts) der Android-Anwendung.

Die in Abbildung 6.2 dargestellte Liste an Einstellungen wird aus einer XML-Datei erzeugt. Sie erlaubt die automatische Zuordnung von Menüpunkten zu typisierten Einträgen in einer

anwendungslokalen Datenbank, es ist also keine weitere manuell zu erstellende Anwendungslogik notwendig. Eine Ausnahme bildet die ebenfalls in Abbildung 6.2 (rechts) dargestellte Auswahlliste an zur Verfügung stehenden Sprachen. Diese Liste wird durch die Server-Software definiert und musste durch eine weitere Aktivität programmiert werden.

Weitere Einstellungen sind die Möglichkeit, die Liste an verfügbaren Fragebögen automatisch bei Programmstart aktualisieren zu lassen und die Angabe des zu verwendenden Monitor-Servers. Auf diesem Framework aufbauende Systeme könnten diese Einstellungsmöglichkeit ausblenden und die Adresse des Servers fest in die Client-Software einprogrammieren, um eine falsche Server-Adresse als mögliche Fehlerquelle auszuschließen.

6.4 Bearbeiten der Fragebögen

Abbildung 6.3 zeigt das Ausfüllen eines Fragebogens. Auf der rechten Seite werden verschiedene Anwendungsmöglichkeiten der Ratingskala demonstriert: Als gewöhnliche Schiebelleiste, mit der ein Wert zwischen zwei Extrema ausgewählt werden kann, wobei der aktuell gewählte Wert optional nicht angezeigt wird. Die beiden Extrema, auch Pole genannt, können zur besseren Verständlichkeit beschriftet werden. Ebenfalls wird demonstriert, dass der angezeigte Wert auf Texte abgebildet werden kann. Dies ermöglicht etwa das klassische Frageschema, bei dem vom Benutzer der Grad der Zustimmung zu gegebenen Aussagen abgefragt wird. Statt den Zahlen 1 bis 5 werden dem Benutzer sinnvolle Texte angezeigt, etwa „Stimme größtenteils zu“ statt „4“.

Ebenfalls in Abbildung 6.3 sind mehrere Anwendungsmöglichkeiten der geschlossenen Frage demonstriert. Zu erkennen ist, dass Fragen, bei denen genau eine Antwort erwartet wird (single choice), als „Radiobutton“ dargestellt werden. Im Gegensatz dazu werden Fragen, bei denen mehrere oder auch keine Antwortmöglichkeit gegeben werden können (multiple choice), als „Checkbox“ dargestellt. Dies entspricht der Erwartung des Benutzers, da gängige Konventionen eingehalten werden.

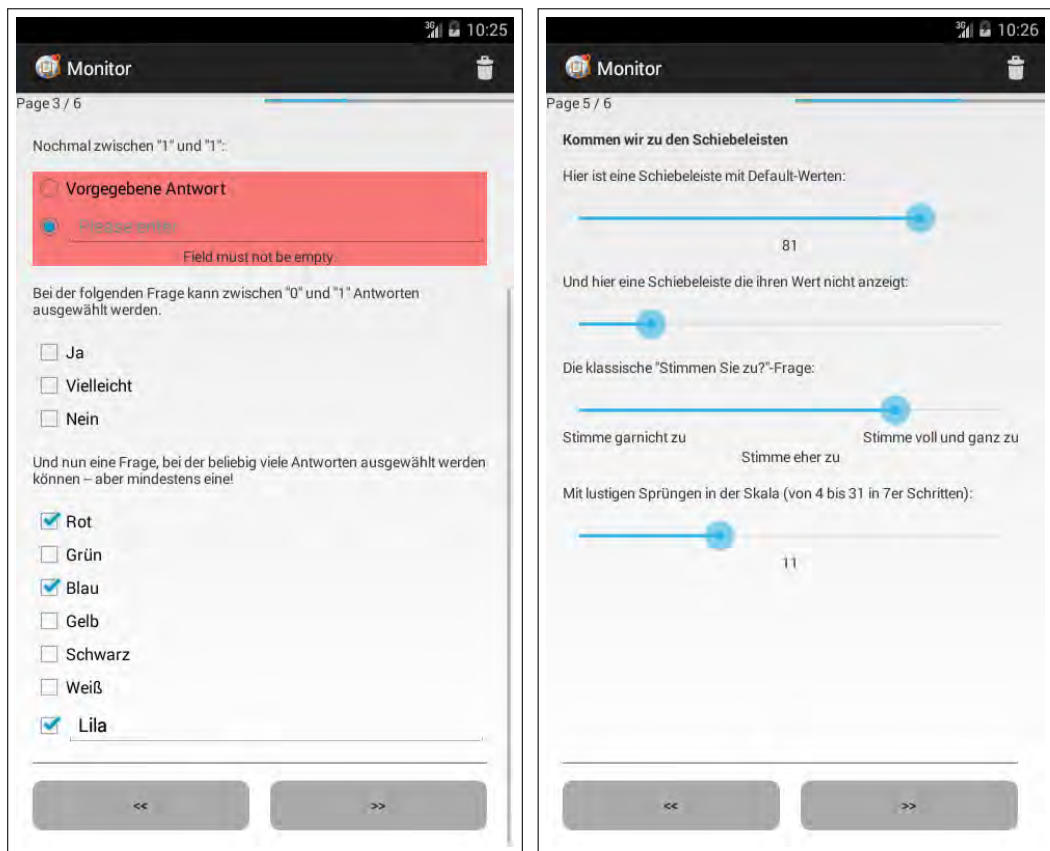


Abbildung 6.3: Ausfüllen eines Fragebogens.

Zwei der dargestellten Fragen beinhalten vom Benutzer ergänzbare Optionen. Dies wird genutzt, um die Validierung der Daten durch die Client-Software zu demonstrieren. Fehlerhafte Eingabefelder, etwa nicht ausgefüllte aber mandatorische Felder oder, wie im Beispiel, eine ausgewählte aber nicht ausgefüllte Ergänzungsoption, werden rot hinterlegt und mit einer Fehlermeldung annotiert. Der Fragebogen kann erst dann weiter ausgefüllt werden, wenn die beanstandeten Elemente korrigiert werden. Alle in Kapitel 4.3 genannten Elemente, die ein `optional`-Feld besitzen, werden validiert, insbesondere also offene und geschlossene Fragen, Dateiauswahl und Sensordaten.

Der Fortschrittsbalken in der rechten oberen Ecke und die Angabe über die aktuelle Seite und Gesamtseitenzahl geben dem Benutzer Aufschluss über den Fortschritt dieser Befra-

gung. Ist der Benutzer auf der letzten Seite angekommen, so schickt ein weiteres Antippen der „nächste Seite“-Schaltfläche die gegebenen Antworten an die Server-Software ab. Sobald die Resultate empfangen sind, werden diese angezeigt und zusätzlich als Archivdatei (siehe Kapitel 6.2) gespeichert. Die Bearbeitung des Fragebogens ist damit abgeschlossen.

Kapitel 7

Fazit und Ausblick

In diesem Kapitel wird ein Fazit aus den Ergebnissen der Arbeit gezogen und die Erfüllung der Anforderungen aus Kapitel 3 überprüft. Anschließend wird ein Ausblick auf mögliche aufbauende Arbeiten gegeben.

7.1 Fazit

In diesem Abschnitt wird die Erfüllung der Anforderungen aus Kapitel 3 überprüft.

AF1 (Aufbau als Fragebogen) Die Anforderung wurde voll erfüllt. Die Befragungen finden in der Form von Fragebögen statt, die Server-Software ist in der Lage, diese Fragebögen zu erstellen, auszuwerten und zu verwalten.

AF2 (Mobile Anwendung) Die Anforderung wurde voll erfüllt. Die Client-Software ist in der Lage, die Fragebögen herunter zu laden, anzuzeigen, vom Benutzer ausfüllen zu lassen und die Antworten auf den Server zu übertragen. Automatisch erzeugte Auswertungen für die vom Benutzer eingegebenen Daten werden präsentiert und zur Archivierung auf dem mobilen Endgerät gespeichert.

AF3 (Datenexport) Die Anforderung wurde voll erfüllt. Mit der in Kapitel 4.4 vorgestellten API ist es registrierten und authentifizierten Analysten möglich, die Ergebnisse der Befragung zu exportieren. Da durch eine wohldefinierte REST-Schnittstelle JSON-formatierte Daten exportiert werden, ist eine Anbindung an weitere Werkzeuge auch automatisiert bequem möglich.

AF4 (Vollständigkeit) Die Anforderung wurde voll erfüllt. Alle genannten Befragungsmöglichkeiten (offene Frage, geschlossene Frage, Ratingskalen) werden vollständig unterstützt.

AF5 (Mehrsprachigkeit) Die Anforderung wurde voll erfüllt. Mit dem in Kapitel 5.2 vorgestellten System der Übersetzung ist sowohl eine vollständige als auch eine teilweise Übersetzung der Fragebögen möglich. Nicht vorhandene Übersetzung wird durch ein Zurückfallen auf die Basissprache des Fragebogens gehandhabt.

AF6 (Wiederaufnehmbarkeit) Die Anforderung wurde voll erfüllt. Durch die Möglichkeit, den momentanen Bearbeitungsstand eines Fragebogens zu speichern, ist es möglich, die Bearbeitung zu unterbrechen und zu einem späteren Zeitpunkt erneut aufzunehmen.

AF7 (Rechteverwaltung) Die Anforderung wurde voll erfüllt. Die in Kapitel 4.1 und Kapitel 4.2 genannten Benutzerrollen ermöglichen eine Trennung unterschiedlicher Benutzerprofile, die den Schutz der Daten vor nicht autorisiertem Zugriff sicherstellen.

NFA1 (Datenschutz) Die Anforderung wurde im Rahmen der technischen Möglichkeiten erfüllt. Kapitel 5.2 beschreibt, wie Elemente der Fragebögen als „privat“ markiert werden können, woraufhin diese Daten nicht in der Datenbank gespeichert werden und daher Analysten nicht zur Einsicht stehen. Die Unterscheidung zwischen „persönlichen“ Daten und „nichtpersönlichen“ Daten ist allerdings nicht automatisierbar, daher ist die explizite Markierung als solche durch den Autor des Fragebogens erforderlich.

NFA2 (Universalität) Die Anforderung wurde voll erfüllt. Von beispielsweise „Medizinische Schwangerschaftsbegleitung“ bis „Mobile Zählerstandsablesung“ sind Anwendungsfälle vom medizinischen bis zum technischen Bereich möglich.

NFA3 (Erweiterbarkeit der Sensortypen) Die Anforderung wurde teilweise erfüllt. Mit dem Fragebogenelement „Sensordaten“ steht ein vielseitiges Mittel bereit, mit dem Sensoren des Smartphones konfiguriert und ausgelesen werden können. Die konkrete Implementierung des Sensor-Interfaces muss allerdings von der Anwendung, die dieses Framework nutzt, selber durchgeführt werden. Die Verwendung eines Sensor-Frameworks ähnlich [19] würde diesen Prozess stark vereinfachen.

NFA4 (Bedienbarkeit) Eine abschließende Beurteilung der Bedienbarkeit der erstellten Software ist ohne eine ausführliche Benutzerstudie nicht möglich. Allerdings wurde die Oberfläche der Server- und Client-Software möglichst simpel gehalten. Bei der Entwicklung der Client-Software wurden die Android-Guidelines [1] eingehalten. Bedienelemente der Web-Oberfläche der Server-Software wurden möglichst mit aussagekräftigen Symbolen und eindeutiger Beschriftung versehen.

NFA5 (Wartbarkeit) Die Anforderung wurde voll erfüllt. Die Software baut auf Standardkomponenten wie PHP und Android auf, deren Verbreitung und Bekanntheitsgrad sicherstellen in Zukunft von Dritten gewartet und weiterentwickelt werden zu können. Der Quellcode ist strukturiert, Änderungen können durch ein Versionskontrollsystem verfolgt werden und der Code ist unter einer freizügigen Lizenz (GNU General Public License [14]) lizenziert.

NFA6 (Erweiterbarkeit der Fragetypen) Die Anforderung wurde voll erfüllt. In Kapitel 4.3 wird beschrieben, wie die erstellte Software um weitere Fragetypen erweitert werden kann. Die Client-Software ist so geschrieben, dass sie im Fall von Inkompatibilität, also unbekanntem Fragetypen, diese ignoriert und so zumindest Teile des Fragebogens verwendet werden können.

Insgesamt ist festzustellen, dass die Ziele dieser Arbeit erreicht wurden. Die erstellte Software genügt den gestellten Ansprüchen und ist in der Lage, Mobile Crowd Sensing gemäß der Anforderungen durch das Datenschutzgesetz (siehe Kapitel 2.2) zu betreiben.

7.2 Ausblick

In diesem Abschnitt wird ein Ausblick auf mögliche aufbauende Arbeiten gegeben, die die entwickelte Software erweitern könnten.

Zum jetzigen Stand der Entwicklung sind die Fragebögen vollständig statisch. Dynamische Integration von Fragen, wie das Einblenden von Folgefragen bzw. das Ausblenden überflüssiger Fragen bei bestimmten Antworten auf vorhergehende Fragen, ist nicht möglich. Ein typisches Anwendungsszenario wäre etwa in einem medizinischen Fragebogen, dass Fragen zu einer möglichen Schwangerschaft ausgeblendet werden, wenn der Benutzer als Geschlecht „männlich“ ausgewählt hat. Eine aufbauende Arbeit könnte die Software um die Möglichkeit erweitern, Abhängigkeiten zwischen den Fragen zu definieren, um den Fragebogen interaktiver zu gestalten. Erste Konzepte sind in [34] zusammengefasst.

Von der Möglichkeit, interaktive Elemente in die Fragebögen einzubauen ausgehend, ist ein weiterer möglicher Ansatzpunkt zur Erweiterung die „Gamification“ genannte Anwendung spieltypischer Elemente [10], wie Erfahrungspunkte, Auszeichnungen (Achievements) oder Ranglisten. Das Ziel dieser Maßnahmen ist, die Motivation der Teilnehmer einer Befragung ähnlich wie bei McSense (siehe Kapitel 2.3), zu steigern und so die Anzahl der Datensätze sowie deren Qualität zu erhöhen. Dazu könnte die Implementierung weiterer Fragebogenelemente, wie etwa eingebundene Videos, nützlich sein.

Für komplexere Anwendungen könnte es notwendig werden, die Rechteverwaltung durch ein feiner einstellbares ACL (Access Control Lists [7]) System zu ersetzen. Zwar können derzeit Befrager, Teilnehmer und Analysten jeweils individuell für Fragebögen freigeschaltet werden bzw. ihnen diese Berechtigung wieder entzogen werden, allerdings existiert keine Möglichkeit, Änderungen der Berechtigungen rekursiv auf abhängige Benutzerrollen abzu-

bilden. Für Laravel existiert bereits eine ACL-Implementierung [20], die für die Benutzung in der erstellten Software angepasst werden könnte.

Eine weitere wünschenswerte aufbauende Arbeit könnte sich mit der Möglichkeit beschäftigen, die Wiederholbarkeit der Fragebögen genauer zu definieren. Im Moment können Benutzer Fragebögen, für die sie freigeschaltet sind, beliebig häufig ausfüllen. Mit zusätzlichen Kopfdaten der Fragebögen könnten Einschränkungen bezüglich der Zeit oder des Ortes gemacht werden, zu denen die Fragebögen bearbeitet werden können bzw. sollen. Mögliche Anwendungsfälle wären in der erwähnten Begleitung Schwangerer Fragebögen zu definieren, die nur in bestimmten Schwangerschaftswochen relevant sind. Fragebögen, die nur aktiviert werden wenn sich technisches Personal in der Nähe bestimmter Gebäude befindet, sind ebenfalls denkbar.

Literatur

- [1] *Android - Activity and Task Design Guidelines*. URL: <https://developer.android.com/guide/components/activities.html> (besucht am 28.09.2015).
- [2] *Android - Sensors Overview*. URL: https://developer.android.com/guide/topics/sensors/sensors_overview.html (besucht am 28.09.2015).
- [3] *Android Studio*. URL: <https://developer.android.com/sdk/index.html> (besucht am 28.09.2015).
- [4] *Apisense auf Twitter*. URL: <https://twitter.com/APISENSE> (besucht am 28.09.2015).
- [5] *Apisense Website*. URL: <http://www.apisense.com/> (besucht am 28.09.2015).
- [6] *Apisense Website, Stand 23. März 2014 im Internetarchiv archive.org*. URL: <https://web.archive.org/web/20140323172629/http://www.apisense.com/> (besucht am 28.09.2015).
- [7] John Barkley. „Comparing simple role based access control models and access control lists“. In: *In Proceedings of the second ACM workshop on Role-based access control*. ACM Press, 1997, S. 127–132.
- [8] Tim Bray, Hrsg. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 7159. Request for Comments. 2014.
- [9] *Composer - Dependency Manager for PHP*. URL: <https://getcomposer.org/> (besucht am 28.09.2015).

- [10] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara und Dan Dixon. „Gamification. Using Game-design Elements in Non-gaming Contexts“. In: *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '11. ACM. 2011, S. 2425–2428. ISBN: 978-1-4503-0268-5.
- [11] Roy Thomas Fielding. „Architectural styles and the design of network-based software architectures“. Diss. University of California, Irvine, 2000.
- [12] Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides. *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994.
- [13] Raghu K Ganti, Fan Ye und Hui Lei. „Mobile crowdsensing: current state and future challenges“. In: *Communications Magazine, IEEE* 49.11 (2011), S. 32–39.
- [14] *GNU General Public License Version 3*. URL: <http://www.gnu.org/licenses/gpl-3.0.html> (besucht am 28.09.2015).
- [15] Bin Guo, Zhiwen Yu, Xingshe Zhou und Daqing Zhang. „From participatory sensing to mobile crowd sensing“. In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*. IEEE. 2014, S. 593–598.
- [16] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y Yen, Runhe Huang und Xingshe Zhou. „Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm“. In: *ACM Computing Surveys (CSUR)* 48.1 (2015), S. 7.
- [17] Nicolas Haderer, Romain Rouvoy, Christophe Ribeiro und Lionel Seinturier. „Apsense: Crowd-sensing made easy“. In: *ERCIM News* 93 (2013), S. 28–29.
- [18] Jochen Herrmann. „Konzeption und technische Realisierung eines mobilen Frameworks zur Unterstützung tinnitusgeschädigter Patienten“. Diplomarbeit, Ulm University. 2014.
- [19] Artur Jabs. „Konzeption eines Event-basierten Sensor-Frameworks zur Datenerhebung auf mobilen Endgeräten“. Diplomarbeit, Ulm University. 2015.
- [20] *Kodeine Laravel-ACL*. URL: <https://github.com/kodeine/laravel-acl/> (besucht am 28.09.2015).
- [21] *Laracasts - The Best Laravel and PHP Screencasts*. URL: <https://laracasts.com> (besucht am 28.09.2015).

-
- [22] *Laravel - The PHP Framework For Web Artisans*. URL: <http://laravel.com> (besucht am 28.09.2015).
- [23] Huadong Ma, Dong Zhao und Peiyan Yuan. „Opportunities in mobile crowd sensing“. In: *Communications Magazine, IEEE* 52.8 (2014), S. 29–35.
- [24] L. Masinter, Hrsg. *The „data“ URL scheme*. RFC 2397. Request for Comments. 1998.
- [25] *Medusa Crowd Sensing Framework - Commit History*. URL: <https://github.com/USC-NSL/Medusa/commits/master> (besucht am 28.09.2015).
- [26] A Mohamud. „EU5 smartphone penetration reaches 55 percent in October 2012“. In: *comScore Press Release, Dec* (2012). URL: <https://www.comscore.com/Insights/Press-Releases/2012/12/EU5-Smartphone-Penetration-Reaches-55-Percent-in-October-2012> (besucht am 28.09.2015).
- [27] DIN Deutsches Institut für Normung. *Grundlagen der Meßtechnik - Teil 1: Grundbegriffe*. Techn. Ber. DIN 1319-1. Deutsches Institut für Normung e.V., 1. Jan. 1995.
- [28] A. Phillips und M. Davis, Hrsg. *Tags for Identifying Languages*. RFC 4646. Request for Comments. 2006.
- [29] Werner Pluta. *Smartphone soll Massenpanik verhindern*. 4. Sep. 2012. URL: <http://www.golem.de/news/crowd-management-smartphone-soll-massenpanik-verhindern-1209-94331.html> (besucht am 28.09.2015).
- [30] Rüdiger Pryss, Manfred Reichert, Jochen Herrmann, Berthold Langguth und Winfried Schlee. „Mobile Crowd Sensing in Clinical and Psychological Trials – A Case Study“. In: *28th IEEE Int'l Symposium on Computer-Based Medical Systems*. IEEE Computer Society Press, 2015, S. 23–24.
- [31] Rüdiger Pryss, Manfred Reichert, Berthold Langguth und Winfried Schlee. „Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research“. In: *IEEE 4th International Conference on Mobile Services (MS 2015)*. IEEE Computer Society Press, 2015, S. 352–359.

- [32] Moo-Ryong Ra, Bin Liu, Tom F La Porta und Ramesh Govindan. „Medusa: A programming framework for crowd-sensing applications“. In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM. 2012, S. 337–350.
- [33] E. Rescorla, Hrsg. *HTTP Over TLS*. RFC 2818. Request for Comments. 2000.
- [34] Johannes Schobel, Marc Schickler, Rüdiger Pryss, Fabian Maier und Manfred Reichert. „Towards Process-Driven Mobile Data Collection Applications: Requirements, Challenges, Lessons Learned“. In: *10th Int’l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps*. 2014, S. 371–382. (Besucht am 27. 10. 2015).
- [35] Manoop Talasila, Reza Curtmola und Cristian Borcea. „Crowdsensing in the Wild with Aliens and Micro-payments“. In: *IEEE Pervasive Computing Magazine* (2014).
- [36] Manoop Talasila, Reza Curtmola und Cristian Borcea. „Mobile Crowd Sensing“. In: John R. Vacca. *Handbook of Sensor Networking: Advanced Technologies and Applications*. CRC Press, 2015.
- [37] *Tango Desktop Project*. URL: <http://tango.freedesktop.org/> (besucht am 28. 09. 2015).
- [38] *Usage Statistics and Market Share of Server-side Programming Languages for Websites*. URL: http://w3techs.com/technologies/overview/programming_language/all (besucht am 28. 09. 2015).
- [39] Bundesministerium der Justiz und für Verbraucherschutz, Hrsg. *Gesetz über Rahmenbedingungen für elektronische Signaturen (SigG)*.
- [40] Bundesministerium der Justiz und für Verbraucherschutz, Hrsg. *Bundesdatenschutzgesetz (BDSG)*. Fassung vom 25.02.2015.
- [41] Bundesministerium der Justiz und für Verbraucherschutz, Hrsg. *Telemediengesetz (TMG)*.
- [42] Springer Gabler Verlag, Hrsg. *Gabler Wirtschaftslexikon, Stichwort: Smartphone*. URL: <http://wirtschaftslexikon.gabler.de/Archiv/569824/smartphone-v1.html> (besucht am 09. 10. 2015).

- [43] Daqing Zhang, Leye Wang, Haoyi Xiong und Bin Guo. „4W1H in mobile crowd sensing“. In: *Communications Magazine, IEEE* 52.8 (2014), S. 42–48.

Anhang A

Beispielfragebogen

```
1  [
2  {
3    "text" : {
4      "2" : "Als erstes geht es mal um ein wenig Text"
5    },
6    "type" : "header"
7  },
8  {
9    "type" : "text",
10   "text" : {
11     "2" : "Lorem ipsum dolor sit amet, consectetur adipisicing elit,
        sed eiusmod tempor incididunt ut labore et dolore magna aliqua.
        Ut enim ad minim veniam, quis nostrud exercitation ullamco
        laboris nisi ut aliquid ex ea commodi consequat. Quis aute
        iure reprehenderit in voluptate velit esse cillum dolore eu
        fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non
        proident, sunt in culpa qui officia deserunt mollit anim id
        est laborum."
12   }
13 },
14 {
15   "type" : "text",
16   "text" : {
```

Anhang A Beispielfragebogen

```
17     "2" : "Lorem ipsum dolor sit amet, consectetur adipisicing elit,
        sed eiusmod tempor incididunt ut labore et dolore magna aliqua.
        Ut enim ad minim veniam, quis nostrud exercitation ullamco
        laboris nisi ut aliquid ex ea commodi consequat. Quis aute
        iure reprehenderit in voluptate velit esse cillum dolore eu
        fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non
        proident, sunt in culpa qui officia deserunt mollit anim id
        est laborum.\nZeilenumbruch. Lorem ipsum dolor sit amet,
        consectetur adipisicing elit, sed eiusmod tempor incididunt ut
        labore et dolore magna aliqua. Ut enim ad minim veniam, quis
        nostrud exercitation ullamco laboris nisi ut aliquid ex ea
        commodi consequat. Quis aute iure reprehenderit in voluptate
        velit esse cillum dolore eu fugiat nulla pariatur. Excepteur
        sint obcaecat cupiditat non proident, sunt in culpa qui
        officia deserunt mollit anim id est laborum."
18     }
19 },
20 {
21     "type" : "divider"
22 },
23 {
24     "type" : "text",
25     "text" : {
26         "2" : "Trenner. Lorem ipsum dolor sit amet, consectetur adipisicing
            elit, sed eiusmod tempor incididunt ut labore et dolore magna
            aliqua. Ut enim ad minim veniam, quis nostrud exercitation
            ullamco laboris nisi ut aliquid ex ea commodi consequat. Quis
            aute iure reprehenderit in voluptate velit esse cillum dolore
            eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat
            non proident, sunt in culpa qui officia deserunt mollit anim
            id est laborum."
27     }
28 },
29 {
30     "type" : "pagebreak"
```

```
31 },
32 {
33     "type" : "header",
34     "text" : {
35         "2" : "Nun ein paar Bilder"
36     }
37 },
38 {
39     "type" : "text",
40     "text" : {
41         "2" : "Als erstes eine kleine deutsche Flagge:"
42     }
43 },
44 {
45     "type" : "image",
46     "image" : "data:image/png;base64,[...]"
47 },
48 {
49     "type" : "text",
50     "text" : {
51         "2" : "Dann der Union Jack"
52     }
53 },
54 {
55     "image" : "data:image/png;base64,[...]",
56     "type" : "image"
57 },
58 {
59     "text" : {
60         "2" : "Und zum Schluss die Trikolore:"
61     },
62     "type" : "text"
63 },
64 {
65     "type" : "image",
```

Anhang A Beispielfragebogen

```
66     "image" : "data:image/png;base64,[...]"
67   },
68   {
69     "type" : "pagebreak"
70   },
71   {
72     "text" : {
73       "2" : "Kommen wir zu den geschlossenen Fragen"
74     },
75     "type" : "header"
76   },
77   {
78     "text" : {
79       "2" : "Bei dieser Frage kann zwischen \"1\" und \"1\" Antworten
80           ausgewählt werden, sie ist also eine klassische \"Single Choice
81           \"-Frage -- und sollte deshalb auch mit Radiobuttons statt der
82           üblichen \"Checkboxen\" dargestellt sein."
83     },
84     "type" : "text"
85   },
86   {
87     "max" : 1,
88     "type" : "question",
89     "min" : 1,
90     "varname" : "jaodernein",
91     "options" : [
92       {
93         "text" : {
94           "2" : "Ja"
95         },
96         "value" : 0
97       },
98       {
99         "value" : 1,
100        "text" : {
```

```
98         "2" : "Nein"
99     }
100 }
101 ]
102 },
103 {
104     "text" : {
105         "2" : "Nochmal zwischen \"1\" und \"1\":"
106     },
107     "type" : "text"
108 },
109 {
110     "varname" : "jaodernein2",
111     "options" : [
112         {
113             "value" : 0,
114             "text" : {
115                 "2" : "Ja"
116             }
117         },
118         {
119             "text" : {
120                 "2" : "INPUT"
121             },
122             "value" : 1
123         }
124     ],
125     "min" : 1,
126     "type" : "question",
127     "max" : 1
128 },
129 {
130     "type" : "text",
131     "text" : {
```

Anhang A Beispielfragebogen

```
132     "2" : "Bei der folgenden Frage kann zwischen \"0\" und \"1\"
133         Antworten ausgewählt werden."
134   },
135   {
136     "max" : 1,
137     "varname" : "jaodernein3",
138     "options" : [
139       {
140         "text" : {
141           "2" : "Ja"
142         },
143         "value" : 0
144       },
145       {
146         "value" : 1,
147         "text" : {
148           "2" : "Vielleicht"
149         }
150       },
151       {
152         "value" : 2,
153         "text" : {
154           "2" : "Nein"
155         }
156       }
157     ],
158     "min" : 0,
159     "type" : "question"
160   },
161   {
162     "type" : "text",
163     "text" : {
164       "2" : "Und nun eine Frage, bei der beliebig viele Antworten
           ausgewählt werden können -- aber mindestens eine!"
```

```
165     }
166 },
167 {
168     "max" : 0,
169     "type" : "question",
170     "varname" : "farben",
171     "options" : [
172         {
173             "text" : {
174                 "2" : "Rot"
175             },
176             "value" : 0
177         },
178         {
179             "text" : {
180                 "2" : "Grün"
181             },
182             "value" : 1
183         },
184         {
185             "value" : 2,
186             "text" : {
187                 "2" : "Blau"
188             }
189         },
190         {
191             "value" : 3,
192             "text" : {
193                 "2" : "Gelb"
194             }
195         },
196         {
197             "value" : 4,
198             "text" : {
199                 "2" : "Schwarz"
```

Anhang A Beispielfragebogen

```
200     }
201   },
202   {
203     "text" : {
204       "2" : "Weiß"
205     },
206     "value" : 5
207   },
208   {
209     "text" : {
210       "2" : "INPUT"
211     },
212     "value" : 6
213   }
214 ],
215 "min" : 1
216 },
217 {
218   "type" : "pagebreak"
219 },
220 {
221   "type" : "header",
222   "text" : {
223     "2" : "Texteingabefelder"
224   }
225 },
226 {
227   "text" : {
228     "2" : "Ein klassisches Texteingabefeld. Ist optional, darf also
229         leer bleiben."
230   },
231   "type" : "text"
232 },
233   "inputtype" : 0,
```

```
234     "type" : "textinput",
235     "varname" : "text_empty",
236     "text" : {
237         "2" : "Darf leer bleiben"
238     },
239     "optional" : 1
240 },
241 {
242     "type" : "text",
243     "text" : {
244         "2" : "Noch ein Texteingabefeld. Dieses hier muss aber ausgefüllt
                werden."
245     }
246 },
247 {
248     "optional" : 0,
249     "inputtype" : 0,
250     "type" : "textinput",
251     "text" : {
252         "2" : "Muss gefüllt werden"
253     },
254     "varname" : "text_notempty"
255 },
256 {
257     "type" : "text",
258     "text" : {
259         "2" : "Für viel Text gibt es mehrzeilige Texteingabefelder:"
260     }
261 },
262 {
263     "type" : "textinput",
264     "inputtype" : 1,
265     "text" : {
266         "2" : "Hier kann ganz viel Text eingegeben werden"
267     },
```

Anhang A Beispielfragebogen

```
268     "varname" : "text_multiline",
269     "optional" : 1
270 },
271 {
272     "type" : "text",
273     "text" : {
274         "2" : "Und für geheimen Text Passwordeingabefelder:"
275     }
276 },
277 {
278     "optional" : 1,
279     "type" : "textinput",
280     "inputtype" : 2,
281     "text" : {
282         "2" : "Kann keiner lesen"
283     },
284     "varname" : "text_password"
285 },
286 {
287     "type" : "text",
288     "text" : {
289         "2" : "Ein Textfeld für Datumseingaben..."
290     }
291 },
292 {
293     "text" : {
294         "2" : "Wann ist dein Geburtstag?"
295     },
296     "varname" : "text_date",
297     "inputtype" : 3,
298     "type" : "textinput",
299     "optional" : 1
300 },
301 {
302     "type" : "text",
```

```
303     "text" : {
304         "2" : "... und eines für Uhrzeiten:"
305     }
306 },
307 {
308     "optional" : 1,
309     "type" : "textinput",
310     "inputtype" : 4,
311     "varname" : "text_time",
312     "text" : {
313         "2" : "Wie spät ist es?"
314     }
315 },
316 {
317     "text" : {
318         "2" : "Und zum Schluss noch ein Feld, in das man nur Nummern
           eingeben kann:"
319     },
320     "type" : "text"
321 },
322 {
323     "optional" : 1,
324     "type" : "textinput",
325     "inputtype" : 5,
326     "varname" : "text_number",
327     "text" : {
328         "2" : "Dokumentenummer"
329     }
330 },
331 {
332     "type" : "pagebreak"
333 },
334 {
335     "text" : {
336         "2" : "Kommen wir zu den Schiebeleisten"
```

Anhang A Beispielfragebogen

```
337     },
338     "type" : "header"
339 },
340 {
341     "text" : {
342         "2" : "Hier ist eine Schiebeleiste mit Default-Werten:"
343     },
344     "type" : "text"
345 },
346 {
347     "left" : {
348         "2" : ""
349     },
350     "step" : 1,
351     "max" : 100,
352     "display" : 1,
353     "varname" : "rating_default",
354     "min" : 1,
355     "mapping" : [],
356     "right" : {
357         "2" : ""
358     },
359     "value" : 50,
360     "type" : "rating"
361 },
362 {
363     "type" : "text",
364     "text" : {
365         "2" : "Und hier eine Schiebeleiste die ihren Wert nicht anzeigt:"
366     }
367 },
368 {
369     "value" : 50,
370     "type" : "rating",
371     "right" : {
```

```
372     "2" : ""
373 },
374 "mapping" : [],
375 "varname" : "rating_hidden",
376 "display" : 0,
377 "min" : 1,
378 "step" : 1,
379 "max" : 100,
380 "left" : {
381     "2" : ""
382 }
383 },
384 {
385     "type" : "text",
386     "text" : {
387         "2" : "Die klassische \"Stimmen Sie zu?\"-Frage:"
388     }
389 },
390 {
391     "varname" : "rating_agree",
392     "display" : 1,
393     "min" : 0,
394     "left" : {
395         "2" : "Stimme gar nicht zu"
396     },
397     "step" : 1,
398     "max" : 4,
399     "value" : 2,
400     "type" : "rating",
401     "right" : {
402         "2" : "Stimme voll und ganz zu"
403     },
404     "mapping" : [
405         {
406             "value" : 0,
```

Anhang A Beispielfragebogen

```
407         "text" : {
408             "2" : "Stimme gar nicht zu"
409         }
410     },
411     {
412         "text" : {
413             "2" : "Stimme eher nicht zu"
414         },
415         "value" : 1
416     },
417     {
418         "text" : {
419             "2" : "Habe keine Meinung"
420         },
421         "value" : 2
422     },
423     {
424         "text" : {
425             "2" : "Stimme eher zu"
426         },
427         "value" : 3
428     },
429     {
430         "text" : {
431             "2" : "Stimme voll und ganz zu"
432         },
433         "value" : 4
434     }
435 ]
436 },
437 {
438     "text" : {
439         "2" : "Mit Sprüngen in der Skala (von 4 bis 31 in 7er Schritten):
440         "
```

```
441     "type" : "text"
442 },
443 {
444     "left" : {
445         "2" : ""
446     },
447     "step" : 7,
448     "max" : 31,
449     "display" : 1,
450     "varname" : "rating_scale",
451     "min" : 4,
452     "mapping" : [],
453     "right" : {
454         "2" : ""
455     },
456     "value" : 4,
457     "type" : "rating"
458 },
459 {
460     "type" : "pagebreak"
461 },
462 {
463     "type" : "header",
464     "text" : {
465         "2" : "Dateiupload und Sensoren"
466     }
467 },
468 {
469     "text" : {
470         "2" : "Hier ein optionaler Dateiupload:"
471     },
472     "type" : "text"
473 },
474 {
475     "optional" : 1,
```

Anhang A Beispielfragebogen

```
476     "size" : 31337,
477     "varname" : "fileupload_optional",
478     "type" : "fileupload"
479 },
480 {
481     "text" : {
482         "2" : "Und ein Dateiupload als Pflichtfeld:"
483     },
484     "type" : "text"
485 },
486 {
487     "optional" : 0,
488     "size" : 31337,
489     "type" : "fileupload",
490     "varname" : "fileupload_mandatory"
491 },
492 {
493     "type" : "text",
494     "text" : {
495         "2" : "Hier noch der Sensor:"
496     }
497 },
498 {
499     "optional" : 1,
500     "type" : "sensor",
501     "varname" : "sensor",
502     "options" : "Sensoroptionen"
503 }
504 ]
```


Abbildungsverzeichnis

1.1	Visualisierung von mobil erfassten Positionsdaten als Heatmap. Kartenmaterial © OpenStreetMap-Mitwirkende https://www.openstreetmap.org/copyright	2
4.1	Der Ablauf einer Befragung als Aktivitätsdiagramm.	16
4.2	Ausschnitt des ER-Diagramms der Datenbank mit Fokus auf den Beziehungen zwischen Fragebogen, Auswertung und Resultate.	17
4.3	Ausschnitt des ER-Diagramms der Datenbank mit Fokus auf den Beziehungen zwischen den verschiedenen Benutzerrollen.	18
4.4	JSON-Repräsentation der Elemente eines Beispielfragebogens.	23
5.1	Benutzerverwaltung mit Monitor.	33
5.2	Oberfläche zur Erstellung und Bearbeitung von Fragebögen.	34
5.3	Übersetzung eines Fragebogenelements.	35
5.4	Bearbeitung einer Regel zur automatischen Auswertung der Fragebögen.	36
6.1	Hauptmenü und Archiv der Android-Andwendung	41
6.2	Einstellungsmöglichkeiten der Android-Anwendung	42
6.3	Ausfüllen eines Fragebogens.	44

Name: Tim Wiederhake

Matrikelnummer: 664433

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Tim Wiederhake