

BACHELORTHESIS

Luis Carlos Alvis Gil

Drag&Drop Survey Creator für mobile Geräte

FAKULTÄT TECHNIK UND INFORMATIK

Department Informatik

Faculty of Computer Science and Engineering

Department Computer Science

Luis Carlos Alvis Gil

Drag&Drop Survey Creator für mobile Geräte

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Informatik Technischer Systeme*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Prüferin: Prof. Birgit Wendholt
Zweitgutachter: Prof. Olaf Zukunft

Eingereicht am: 01.06.2022

Luis Carlos Alvis Gil

Thema der Arbeit

Drag&Drop Survey Creator für mobile Geräte

Stichworte

Drag & Drop, Lokalisierung, Soundwalk, Fragebogen, Mobilgeräte, Web, Template, Tonaufnahme

Kurzzusammenfassung

Der Schwerpunkt dieser Arbeit liegt auf der Entwicklung einer Anwendung, die zur Erstellung und Organisation von Fragebögen und Soundwalks verwendet werden kann. Dabei werden bestehenden Fragebogen-Editoren untersucht und ihre Funktionalitäten auf mobilen Geräten verglichen. Auf mobilen Geräten ist es schwer auf den Benutzeroberflächen die Anwendungen darzustellen, da hierfür nur eingeschränkte Möglichkeiten zur Verfügung stehen. Auf Grund dessen wird in dieser Arbeit eine Anwendung entwickelt, in der Drag & Drop, zur Umgehung dieser Einschränkung benutzt wird. Dabei werden spezifische Funktionalitäten wie: eine Geolokalisierung für Fragebögen, ein Template-Mechanismus und die Möglichkeit Tonaufnahmen-hinzuzufügen, für Soundwalks implementiert. Den Abschluss dieser Arbeit bildet die Anzeige der Ergebnisse, in der gezeigt wird, ob die umzusetzenden Anforderungen akzeptiert werden.

Luis Carlos Alvis Gil

Title of Thesis

Drag&Drop Survey Creator for mobile devices

Keywords

Drag & Drop, Localisation, Soundwalk, Questionnaire, Mobile, Web, Template, Sound Recording

Abstract

The focus of this work is on the development of an application that can be used to create and organise questionnaires and soundwalks. Existing questionnaire editors are examined and their functionalities on mobile devices are compared. On mobile devices, it is difficult to display the applications on the user interfaces, as only limited possibilities are available for this. For this reason, an application is developed in this work in which drag & drop is used to circumvent this limitation. Specific functionalities such as geolocation for questionnaires, a template mechanism and the possibility to add sound recordings for sound-walks are implemented. The final part of this work is the display of the results, which shows whether the requirements to be implemented are accepted.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	v
Abbildungsverzeichnis.....	ix
Listingverzeichnis.....	xi
Tabellenverzeichnis.....	xii
1 Einleitung.....	13
1.1 Motivation.....	13
1.2 Ziel.....	14
1.3 Ausblick.....	15
2 Stand der Technik und Wissenschaft.....	16
2.1 Vergleichbare Arbeiten und Projekten.....	16
2.1.1 Kriterien zur Auswertung der Fragebogen-Editoren.....	16
Open vs. Closed Source.....	16
Mobile Kompatibilität des Fragebogen-Editors.....	17
Template Mechanismus.....	17
Umfang der Fragetypen.....	18
Bedienung und Bedienbarkeit.....	18
Lokalisierung.....	19
Medienunterstützung.....	19
Privatsphäre.....	20
Erweiterbarkeit.....	20
2.1.2 SurveyJS.....	20
Open vs. Closed Source.....	21
Mobile Kompatibilität des Fragebogen-Editor.....	21
Template Mechanismus.....	22
Umfang der Fragetypen.....	22
Bedienung und Bedienbarkeit.....	23
Lokalisierung.....	24

Medienunterstützung	24
Privatsphäre.....	24
Erweiterbarkeit.....	24
2.1.3 Forms.app.....	24
Open vs. Closed Source	24
Mobile Kompatibilität des Fragebogen-Editors	25
Template Mechanismus	25
Umfang der Fragetypen.....	26
Bedienung und Bedienbarkeit	26
Lokalisierung.....	28
Medienunterstützung.....	28
Privatsphäre.....	28
Erweiterbarkeit.....	29
2.1.4 Google Forms.....	29
Open vs. Closed Source	29
Mobile Kompatibilität des Fragebogen-Editors	29
Template Mechanismus	30
Umfang der Fragetypen.....	31
Bedienung und Bedienbarkeit	31
Lokalisierung.....	33
Medienunterstützung.....	33
Privatsphäre.....	33
Erweiterbarkeit.....	33
2.1.5 SurveyMonkey	34
Open vs. Closed Source	34
Mobile Kompatibilität des Fragebogen-Editors	34
Template Mechanismus	35
Umfang der Fragetypen.....	36
Bedienung und Bedienbarkeit	36
Lokalisierung.....	38
Medienunterstützung.....	38

Privatsphäre.....	38
Erweiterbarkeit.....	38
2.2 Vergleich und Zusammenfassung des Stands der Technik und Wissenschaft.....	39
3 Anforderungsanalyse	41
3.1 Funktionale Anforderungen	41
3.1.1 Dashboard	41
3.1.2 Fragebogen-Editor.....	43
3.1.3 Administrator und Teilnehmer	45
3.2 Nicht-Funktionale Anforderungen	46
3.2.1 Zuverlässigkeit	46
3.2.2 Benutzbarkeit	47
3.2.3 Weiterentwickelbarkeit	48
3.3 Zusammenfassung der Anforderungsanalyse.....	49
4 Entwurf	51
4.1 Systemarchitektur.....	51
4.1.1 Model-View-Controller.....	53
4.1.2 Model-View-Controller im Drag & Drop Survey Creator	53
4.1.3 Web-Architektur.....	56
4.1.4 Web-Architektur im Drag & Drop Survey Creator.....	58
4.1.5 Zusammenfassung der Systemarchitektur.....	59
4.2 Umsetzung der Anforderungen	60
4.2.1 Drag & Drop	60
4.2.2 Template-Mechanismus	61
4.2.3 Gruppierung von Fragebögen.....	62
4.2.4 Tonaufnahmen hinzufügen.....	62
4.2.5 Lokalisierung.....	62
4.2.6 Trennung der Administrator und Teilnehmer-Bereich.....	63
4.3 Arbeitsablauf und Gestaltung der Benutzeroberfläche	63
4.3.1 Arbeitsablauf.....	63
4.3.2 Landing Page.....	65
4.3.3 Dashboard	66

4.3.4	Fragebogen-Editor.....	67
4.3.5	Vorschau	68
4.4	Persistenz.....	69
4.5	Klassenkonzept	71
4.6	Zusammenfassung des Entwurfs.....	72
5	Zusammenfassung und Ausblick.....	74
5.1	Ausblick	75
	Literaturverzeichnis.....	76

Abbildungsverzeichnis

Abbildung 1: SurveyJS Struktur. Quelle: Eigenes Werk	21
Abbildung 2: SurveyJS-Creator auf mobilen Geräten (links) und auf dem Desktop (rechts), Quelle: Screenshot aus SurveyJS https://surveyjs.io/create-survey	22
Abbildung 3: Forms.app auf mobilen Geräten (links) und auf dem Desktop (rechts), Quelle: Screenshot aus Forms.app https://forms.app	25
Abbildung 4: Dashboard Ansicht von Forms.app und die „Duplizieren“ Funktionalität, Quelle: Screenshot aus Forms.app https://forms.app	27
Abbildung 5: Google Forms auf mobilen Geräten (links) und auf dem Desktop (rechts), Quelle: Screenshot aus Google Forms https://docs.google.com/forms	30
Abbildung 6: Dashboard Ansicht von Google Forms, Quelle: Screenshot aus Google Forms https://docs.google.com/forms	32
Abbildung 7: SurveyMonkey auf mobilen Geräten (links) und auf dem Desktop (rechts), Quelle: Screenshot aus SurveyMonkey https://de.surveymonkey.com	35
Abbildung 8: Verteilungsdiagramm der Kommunikation zwischen Komponenten, Quelle: Eigenes Werk	53
Abbildung 9: Design nach dem MVC-Prinzip, Quelle: Eigenes Werk.....	55
Abbildung 10: Web-Architektur nach dem MVC-Muster nach Balzert, Quelle: (Balzert, 2011 S. 197)	56
Abbildung 11: Beispiel für eine RIA-Web-Architektur nach Balzert, Quelle: (Balzert, 2011 S. 199)	57

Abbildung 12: Design der Komponenten nach der Web-Architektur MVC-Pattern, Eigenes Werk.....	58
Abbildung 13: Implementierung der Lokalisierung, Eigenes Werk	63
Abbildung 14: Skizze zum Arbeitsablauf der Anwendung, Quelle: Eigenes Werk.....	64
Abbildung 15: Landing Page der Anwendung auf dem Desktop-Browser (links) und Mobile-Browser (rechts), Quelle: Eigenes Werk.....	66
Abbildung 16: Dashboard Page der Anwendung auf dem Desktop-Browser (links) und Mobile-Browser (rechts), Quelle: Eigenes Werk.....	67
Abbildung 17: Fragebogen-Editor Page der Anwendung auf dem Desktop-Browser (links) und Mobile-Browser (rechts), Quelle: Eigenes Werk	68
Abbildung 18: Vorschau vom Fragebogen auf dem Desktop-Browser (links) und Mobile-Browser (rechts), Quelle: Eigenes Werk.....	69
Abbildung 19: JSON-Objekte und deren Beziehungen, Quelle: Eigenes Werk	70
Abbildung 20: Klassenkonzept, Eigenes Werk.....	71

Listingverzeichnis

Listing 1: vereinfachtes Soundwalk-Model im Server, Quelle: Eigenes Werk.....	54
Listing 2: vereinfachte Dashboard View-Komponente, Quelle: Eigenes Werk.....	54
Listing 3: vereinfachte Dashboard Controller-Komponente, Quelle: Eigenes Werk.....	55

Tabellenverzeichnis

Tabelle 1: Zusammenfassung der Kriterien zur Auswertung von Fragebogen-Editoren.....	39
Tabelle 2: Zusammenfassung der Anforderungen an das Dashboard	43
Tabelle 3: Zusammenfassung der Anforderungen an den Fragebogen-Editor.....	45
Tabelle 4: Zusammenfassung der Anforderungen an die Rollen	46
Tabelle 5: Spezifizierung der Zuverlässigkeit.....	47
Tabelle 6: Spezifizierung der Benutzbarkeit	48
Tabelle 7: Spezifizierung der Weiterentwickelbarkeit	49
Tabelle 8: Erfüllte nicht-funktionale Anforderungen anhand der Architektur.....	60
Tabelle 9: Überblick der umgesetzten Anforderungen	73

1 Einleitung

In diesem Kapitel wird die aktuelle Problemstellung dargestellt und die Ziele dieser Arbeit angeführt. Zum Schluss folgt ein Ausblick hinsichtlich des Aufbaus dieser Arbeit.

1.1 Motivation

Mit dem Voranschreiten der technischen Entwicklungen werden regelmäßig vergangene Entwicklungen erneut aufgegriffen und unter dem Einfluss der weiterentwickelten Technologie wieder aufgelegt. Im Jahr 2021 besaßen 62,6 Millionen Menschen in Deutschland Smartphones, die stets weiterentwickelt wurden und neue Funktionalitäten dazugewannen (Tenzer, 2022). Solche Funktionalitäten könnten in unterschiedlichen Anwendungsbereichen eingeführt werden.

Die Klima-Krise schreitet voran, weshalb der Aufbau regenerativer Energiequellen weiter ausgebaut werden muss. Hierfür werden bestehende Technologien stets weiterentwickelt (Edenhofer, et al., 2011). Diese Technologien werden von der in Deutschland lebenden Bevölkerung nicht immer positiv angenommen. Im Fall der Windenergie ist der Aufbau von Windkraftanlagen in der Nähe von Wohngebieten, stark umstritten. Sie sollen sowohl eine visuelle Beeinträchtigung durch Schattenwurf darstellen als auch die Landschaft zerstören. Zudem sollen sie durch die verursachten Lärmemissionen auch gesundheitsschädlich sein (WHO, kein Datum).

Der Windpark Curslack, welcher von der HAW -Hamburg in Hamburg- Bergedorf betrieben wird, wird in dieser Arbeit als Projekt untersucht. Ziel dieses Projektes ist es, das Verständnis zu spezifischen Fragestellungen von Windenergieerzeugung, -anlagenbetrieb und Netzintegration sowie Speicherung zu erweitern. Um die Akzeptanz von Windkraftanlagen auf eine wissenschaftliche Grundlage zu stellen, haben Wissenschaftler der HAW-Hamburg eine Methode

entwickelt, mit welcher diese untersucht werden kann. Hierfür sollen bestimmte Teilnehmer diese Anlagen begehen und bewerten. Hierzu laufen sie größere Bereiche mit Hilfe von Wegpunkten ab und beurteilen punktuell, anhand dafür entwickelter Fragebögen beispielweise die Lautstärke der Windkraftanlagen an jedem Beobachtungspunkt bereitstehen. Diese Methode nennt man „*Soundwalk*“. Hierfür wird heutzutage Stift und Papier verwendet, um eine spätere manuelle Übertragung auf dem Computer durchzuführen.

Es gibt bereits Anwendungen, die Fragebögen online erzeugen. Diese stellen jedoch nur allgemeine Funktionalitäten zur Verfügung. Die meisten sind für Desktops optimiert und die, die mobil verfügbar sind, werden von einer „*Paywall*“ (Bezahlschranke) eingeschränkt. Außerdem sind die Funktionalitäten dieser Anwendungen für das Fachgebiet der Soundwalks nicht ausreichend oder stehen in den kostenlosen Versionen der jeweiligen Software nicht zur Verfügung. Einige wichtige Aspekte bei den Soundwalks liegen in der Möglichkeit „*Templates*“ (Vorlagen) anzulegen und zu bearbeiten sowie die Gruppierung mehrere Templates in einer großen Befragung zusammenzufassen. Hierzu ist die Ortsbestimmung (Lokalisierung) der einzelnen Beobachtungspunkte und das Erstellen eines digitalen Fragebogens per Drag&Drop-Mechanismus die Voraussetzung. Diese Funktionalitäten sind bei vielen Anbietern entweder nicht möglich oder nur in den kostenpflichtigen Varianten verfügbar.

Gewünscht ist eine digitale und mobile Anwendung für die Ersteller der Soundwalks. Die Fragebögen sollen flexibel zusammenstellbar und bei der Vor-Ort-Begehung auf dem Smartphone oder Tablet erstellbar und bearbeitbar sein.

1.2 Ziel

Ziel dieser Bachelorarbeit ist es eine mobile und digitale Anwendung zu entwickeln, die eine einfache und nutzerfreundliche Erstellung von Fragebögen mittels Drag & Drop auf mobilen Endgeräten ermöglicht, um schnell und effektiv neue Fragebögen unterwegs erstellen zu können. Dabei soll es möglich sein, mittels Lokalisierung einen bestimmten Fragebogen zuzuordnen. Die Fragebögen sollen verschiedene Medientypen, wie Tonaufnahmen, Bilder und Videos, unterstützen. Es sollen die üblichen Frage-Typen, wie Einfach- und Mehrfachauswahl, Matrixfrage, Skala, Rankingfrage, Freier-Text und weitere vorhanden sein. Diese sollen per Drag & Drop einem Fragebogen hinzugefügt werden können. Zudem sollen Templates von

Fragebögen zur weiteren Nutzung gespeichert werden können. Diese Fragebögen sollen auch per Drag & Drop in einer größeren Befragung gruppiert und als Liste mittels Lokalisierung aufgerufen werden können.

Als Hardware werden alltägliche mobile Endgeräte wie Smartphones und Tablets benutzt. Da die Software auf unterschiedlichen Plattformen betrieben werden soll, soll sie möglichst alle unterstützen. Durch die vielfältige Kompatibilität eines Webbrowsers eignet sich dieser, als für alle Plattformen kompatibles Medium. Deswegen wird das Ziel eine Webanwendung sein, die für mobile Geräte optimiert ist und freie Software für wissenschaftliche Anwendungen nutzt.

Diese Arbeit befasst sich nicht mit dem Auswerten und Speichern der Fragebogenergebnisse. Der Fokus liegt auf der Entwicklung eines Tools, mit dem Fragebögen für Soundwalks und die Soundwalks selbst erstellt werden können. Es soll eine Möglichkeit geben die Fragebogenvorlagen zu speichern.

1.3 Ausblick

In Kapitel 2 werden anhand vergleichbarer Arbeiten die unterschiedlichen und bereits vorhandenen Fragebogen-Editoren vorgestellt. Anschließend wird die Auswahl der für diese Bachelorarbeit relevanten Funktionen auf Grundlage einer Bewertungstabelle getroffen.

Die zu entwickelnden Komponenten werden in Kapitel 3 als funktionale und nicht-funktionale Anforderungen formuliert.

Kapitel 4 beschreibt den Systementwurf, die detaillierte Umsetzung der einzelnen Komponenten, sowie die Architektur des Systems, unter Berücksichtigung der Ergebnisse aus Kapitel 2 und 3.

In Kapitel 5 werden die Ergebnisse dieser Arbeit zusammengefasst sowie auf mögliche Erweiterungen der entwickelten Anwendung eingegangen.

2 Stand der Technik und Wissenschaft

Dieses Kapitel befasst sich mit dem aktuellen Stand der Fragebogen-Editoren und wie sie sich unterscheiden. Im ersten Teilkapitel 2.1 werden die Kriterien zur Auswertung dieser Editoren und die sich auf dem Markt befindenden Produkte vorgestellt. Im Teilkapitel 2.2 wird die Funktionsweise der verschiedenen Produkte erklärt, um Ideen für ein eigenes Konzept zu erschaffen. Ebenso wird darauf geschaut, warum die bestehenden Produkte nicht für Soundwalk-Fragebögen geeignet sind.

2.1 Vergleichbare Arbeiten und Projekten

Ein Fragebogen-Editor ermöglicht das Erstellen von digitalen Fragebögen mittels einer Sammlung von Fragebogenkomponenten. In dieser Sammlung befinden sich die Elemente, die dem Fragebogen hinzugefügt werden können. Diese Elemente können entweder verschiedene Fragetypen oder Anordnungselemente sein. Dazu können die Fragebögen auf einem Server gespeichert oder exportiert werden, um weiter benutzt werden zu können. Einige Editoren unterstützen auch die Personalisierung der kompletten Benutzeroberfläche mit verschiedenen Themen und Farbschemas, sowie das Auswerten von Fragebogenergebnissen.

2.1.1 Kriterien zur Auswertung der Fragebogen-Editoren

Es gibt verschiedene Editoren, welche die Erstellung von Fragebögen online unterstützen. Im Folgenden werden einige dieser vorgestellt. Damit ein sinnvoller Vergleich erfolgen kann, wird eine Vergleichsbasis definiert. Hierzu werden zuerst die relevanten Kriterien für diese Arbeit aufgelistet und beschrieben:

Open vs. Closed Source

Wenn die Anbieter ihr fertiges Produkt der Öffentlichkeit zugänglich machen, müssen sie entscheiden, ob sie den Code als „*Open Source*“ oder als „*Closed Source*“ zur Verfügung stellen wollen. Bei Open Source ist der Code für jeden öffentlich zugänglich und die Programmierer können den Code lesen oder ändern. Dabei ist es auch erlaubt, diese Änderungen weiter zu

verbreiten. Bei Closed Source (auch als proprietär bezeichnet), wird der Öffentlichkeit kein Zugang zum Code gewährt. Der Code kann durch unterschiedliche Mittel angesehen, aber nicht erweitert werden.

Mobile Kompatibilität des Fragebogen-Editors

Der Fragebogen-Editor soll auf mobilen Geräten richtig dargestellt und einfach bedient werden können. Bei einem Wechsel der Ansicht auf eine mobile Variante sollen keine Layout-Fehler auftreten. Dazu soll die allgemeine Funktionalität des Fragebogen-Editors durch einen Wechsel der Ansicht nicht eingeschränkt werden. Des Weiteren sollen kleine Displays in der Größe von Smartphones im Fokus liegen, weil die Soundwalk-Fragebogen hauptsächlich unterwegs erstellt werden sollen. Wenn der Fragebogen-Editor mehrere Plattformen unterstützen soll, sollen seine Anzeige und Funktionalitäten konsistent untereinander bleiben.

Template Mechanismus

Bei einem Template handelt es sich um eine Vorlage oder Schablone für Fragebögen, welche die wichtigsten Layouts- und Design-Elemente enthält und mit grafischen und textuellen Inhalten gefüllt werden kann. Durch diesen Mechanismus können Templates von Fragebögen angelegt und bearbeitet sowie gespeichert und exportiert werden. Dabei soll es möglich sein, einen Header oder einen Footer einem Fragebogen hinzuzufügen. Ein Header oder Footer bezeichnet einen Bereich im Fragebogen entweder an der Kopf- oder Fußzeile, in dem es keine Fragen gibt. Stattdessen werden dort Randinformationen, wie Titel, Beschreibung und Bilder, hinterlegt. Bei einem Soundwalk sehen die Fragebögen oft ähnlich aus. Deswegen sind ein passender Titel, Beschreibung und Bild oder ein Logo für die Erkennung des Fragebogens hilfreich. Dazu können in diese Bereiche zusätzliche Informationen, wie Geokoordinaten, abgelegt werden.

Ein Soundwalk besteht aus beliebig vielen Beobachtungspunkten. Alle Beobachtungspunkte haben einen entsprechenden Fragebogen. Die Fragen bei den unterschiedlichen Beobachtungspunkten sind oft ähnlich, der Rahmen bzw. der Ort ändert sich jedoch. Deswegen werden vo-

rinstanzierte und bearbeitbare Templates für Soundwalks vorausgesetzt. Jeder generierte Fragebogen soll als ein Template benutzt werden können. Diese Templates sollen zur Wiederverwendung dem Nutzer zur Verfügung stehen.

Umfang der Fragetypen

Ein Fragebogen-Editor besitzt unterschiedliche Fragetypen als Design-Elemente des Fragebogens. Für Soundwalks werden nicht unbedingt alle mögliche Fragetypen benötigt. Es sollen u. a. die folgenden Fragetypen unterstützt werden:

- Einfachauswahl: Hier kann aus mehreren Alternativen genau eine ausgewählt werden.
- Mehrfachauswahl: Hier kann aus mehreren Alternativen mehrere ausgewählt werden. Es können entweder beliebig viele sein oder es gibt eine feste Vorgabe, wie viele ausgewählt werden können.
- Matrixfrage: Die Matrixfrage bedient verschiedene inhaltlich zusammenhängende Fragen, die auf einer einheitlichen Skala beantwortet werden. Diese Fragen können sowohl Einfach- als auch Mehrfachauswahl sein.
- Freier Text: Hier wird ein Textfeld angeboten. Es kann entweder beliebig lang sein oder es gibt eine feste Zeichenanzahl.
- Skala: Hier wird ein Wert aus einer vorgegebenen Skala ausgewählt.
- Rankingfrage: Bei dieser Frage werden Elemente miteinander verglichen und in einer Reihenfolge angeordnet.

Bedienung und Bedienbarkeit

Ein Fragebogen-Editor soll bestimmte Eigenschaften erfüllen, um die Akzeptanz der Nutzer zu steigern. Die Nachbearbeitung existierender Fragebögen steht im Mittelpunkt der Fragebogen-Editoren. Diese Funktionalität optimiert den gesamten Fragebogenerstellungsprozess. Dazu soll es möglich sein die Fragebogenkomponenten per Drag & Drop auf eine beliebige Position zu bewegen. Diese Funktionalität erleichtert das Aufstellen des Layouts der Fragebögen.

Um ähnliche Fragebögen untereinander unterscheiden zu können, soll ein Header und/oder ein Footer zur Verfügung stehen. Sie sind leicht zu bearbeiten und bieten einen Überblick über den

aktuellen Fragebogen. Da die Fragebogen-Editoren auf mobilen Geräten bedient werden sollen, soll ein entsprechendes Layout angeboten werden. Große Fragebögen können zum Beispiel nicht vollständig auf einem Smartphone angezeigt werden. Mit der Unterstützung der Touch-Bedienung, soll die Navigation des Fragebogen-Editors benutzerfreundlicher werden.

Durch verschiedene Erweiterung- und Integrationsmöglichkeiten soll sich ein Fragebogen-Editor an verschiedene Anwendungsdomäne anpassen können. Deswegen ist die Skalierung der Fragebogen-Editoren eine wichtige Eigenschaft. Es soll dazu auch eine Möglichkeit geben, eine Vorschau für den Fragebogenersteller zu zeigen. Hierbei soll der Ersteller sehen, wie der Fragebogen im Auge des Teilnehmers aussieht.

Lokalisierung

Unter Lokalisierung wird das Bestimmen des Aufenthaltsorts einer Person oder eines Objektes, mittels dafür geeigneter Technologien verstanden. Beispiele hierfür sind die Ortung des Globalen Navigationssatellitensystems (GPS), das Globale System für mobile Kommunikation (GSM) im Mobilfunknetz und die WLAN-basierte Ortung.

Die Fragebogen-Editoren sollen Fragebögen georeferenzieren können. Dabei soll der Fragebogen nur für die Teilnehmer, die sich in einem bestimmten Radius befinden, freigeschaltet sein. Diese Funktion soll jedoch vom Fragebogenersteller ein- und ausgeschaltet werden können.

Medienunterstützung

Das Hochladen von Medien unterschiedlicher Formate erlaubt eine breite Datenerfassung. Insbesondere spielen im Bereich der Soundwalks Ton- und Videoaufnahmen eine wichtige Rolle. Das Dateien-Hochladen in einem Fragebogen-Editor soll unterschiedliche Medienformate unterstützen. Diese Medien können entweder als einzelner Frage-Typ oder als Anhang in den anderen Fragetypen in den Fragebogen eingefügt werden. Deswegen soll es bei jedem Fragebogen möglich sein, mindestens eine Tonaufnahme zu hinterlegen.

Privatsphäre

Die Fragebögen sollen anonym erstellt und ausgefüllt werden können. Trotzdem soll eine Unterscheidung zwischen Fragebogenersteller und -Teilnehmer stattfinden. Außerdem sollen die Teilnehmer auch untereinander unterscheidbar sein. Dabei werden keine persönlichen Daten gespeichert. Eine Anmeldung oder Registrierung soll nicht erforderlich sein.

Erweiterbarkeit

Durch Erweiterungsmöglichkeiten lassen sich Fragebogen-Editoren an neue Anforderungen anpassen. Hierfür soll durch einen einfachen Entwurf und die Dezentralisierung der Software in Module mit hoher Autonomie die Erweiterbarkeit begünstigt werden. Die Integration in andere Plattformen als soziale Medien-Verlinkung, wird hierbei nicht als Erweiterung betrachtet. Stattdessen soll die Software komplett oder teilweise verändert werden können, um die gewünschte Funktionalität zu ermöglichen.

2.1.2 SurveyJS

SurveyJS ist ein von „Devsoft Baltic OÜ“ entwickeltes JavaScript Projekt. Mit SurveyJS kann ein Benutzer Fragebögen erstellen, die in andere Systeme integriert werden können. Es bietet nicht nur einen Fragebogendienst, sondern hilft auch bei der Durchführung von Fragebögen auf der eigenen Domain. Darüber hinaus kann es die Anzeige von Fragebögen an das Erscheinungsbild und das Design des eigenen Systems über CSS anpassen. (SurveyJS, 2022)

SurveyJS besteht aus der SurveyJS-Library, dem SurveyJS-Creator und dem SurveyJS-Service. Die Struktur ist auf Abbildung 1 zu sehen. Die SurveyJS-Library erlaubt es, Anwendungen SurveyJS-Fragebögen in ihren Ansichten ausführen zu können. Der SurveyJS-Creator wird benutzt, um Fragebögen zu erstellen und zu bearbeiten. Dieser kann entweder direkt in der SurveyJS Webseite oder in der eigenen Anwendung mit der entsprechenden SurveyJS-Schnittstelle benutzt werden. Der SurveyJS-Service ist für die Speicherung und Analyse der Ergebnisse sowie für den Kunden-Support verantwortlich. Er ist für diese Arbeit nicht relevant.

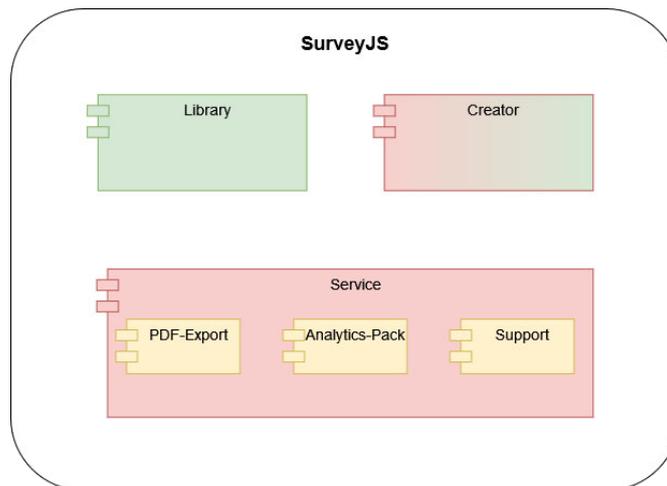


Abbildung 1: SurveyJS Struktur. Quelle: Eigenes Werk

Die grüne Farbe repräsentiert die Komponenten, die kostenfrei für Nutzer zugänglich sind. Die rote Farbe bezeichnet die kostenpflichtigen Komponenten. Der SurveyJS-Creator besitzt beide, weil er in der Webseite von SurveyJS kostenfrei benutzt werden kann und auch als komplettes Paket kostenpflichtig in der eigenen Web-Seite integriert werden kann.

Open vs. Closed Source

SurveyJS ist teilweise Open-Source und Closed-Source. Nur die SurveyJS-Bibliothek hat ihren Quellcode öffentlich. Die restlichen Teile werden als Dienste verkauft. Die SurveyJS-Bibliothek und der Fragebogen-Editor (SurveyJS-Creator) auf der Webseite von SurveyJS werden kostenlos zur Verfügung gestellt. Das Einbauen dieses Editors in der eigenen Anwendung ist jedoch kostenpflichtig.

Mobile Kompatibilität des Fragebogen-Editor

Der SurveyJS Fragebogen-Editor (Creator) funktioniert mit Drag & Drop nur im Desktop-Browser oder auf relativ großen Bildschirmen, wie bei Tablets. Bei kleineren mobilen Endgeräten wird die Drag & Drop Funktionalität der Fragetypen gestrichen. Wie auf der Abbildung 2 zu erkennen ist, ist die Funktionalität des Fragebogen-Editors unabhängig vom benutzten Gerät. Die Menüs sind jedoch etwas überfüllt. Für Endnutzer könnte dies eine hohe Erlernbarkeit bedeuten.

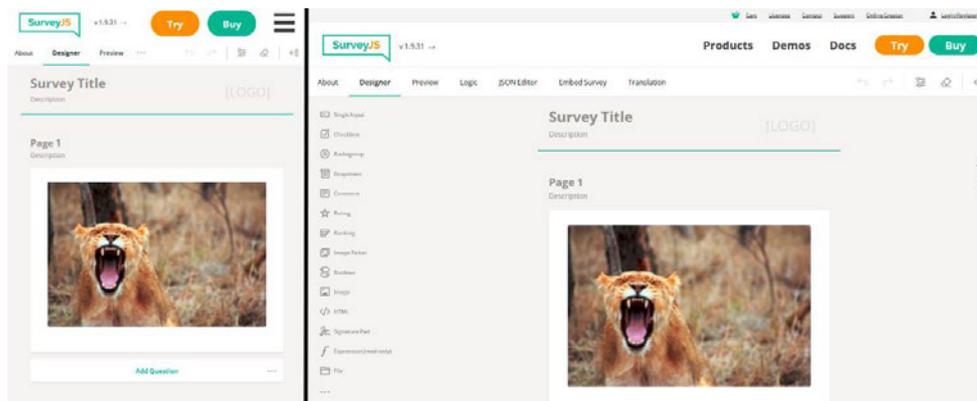


Abbildung 2: SurveyJS-Creator auf mobilen Geräten (links) und auf dem Desktop (rechts),
Quelle: Screenshot aus SurveyJS <https://surveyjs.io/create-survey>

Alle Elemente in der SurveyJS Bibliothek lassen sich einzeln über die API benutzen. Dadurch ist es möglich über die API einen eigenen Fragebogen-Editor aufzubauen. Diese Elemente können sich der eigenen Benutzeroberfläche anpassen.

Template Mechanismus

Das Anlegen und Bearbeiten von Vorlagen (Templates) wird in SurveyJS nicht unterstützt. Es können nur Fragebögen erstellt werden und diese dann direkt in einer externen Anwendung integriert werden. Es existiert jedoch eine Alternative, um die Fragebögen zu speichern: Die Fragebogen-Definitionen in SurveyJS sind JSON Objekte. Diese lassen sich aus dem JSON-Editor vom SurveyJS kopieren und in einen neuen Fragebogen einfügen oder in der eigenen Datenbank speichern. Mit dieser JSON-Definition wird der Fragebogen generiert. Ein Header mit Titel, Beschreibung und Logo werden unterstützt. Dieser Header kann nicht ausgeblendet werden und eine Sammlung von Fragebögen, als große Befragung anzulegen, ist in SurveyJS nicht möglich.

Die erstellten Fragebögen werden nicht bei SurveyJS gespeichert, da diese Funktionalität kostenpflichtig ist und nicht für kostenlose Nutzer zur Verfügung steht.

Umfang der Fragetypen

Die einzelnen Fragen lassen sich in der Größe konfigurieren. Jede Frage mit Auswahlmöglichkeit kann beliebig viele Items enthalten.

SurveyJS unterstützt 20 Fragebogenkomponenten. Einige davon sind nur Layout Komponenten, die für diese Arbeit nicht relevant sind. Mehrfachauswahl, Dropdown, Matrix, Bewertung und Freier-Text Fragen sind zur Auswahl vorhanden. Dazu gibt es spezielle Frage-Typen, wie HTML, Unterschrift und Funktionsfragen.

Eine Antwortvalidierung, die abhängig vom Frage-Typ ist, gibt es bei SurveyJS auch. So können genaue Informationen, wie E-Mail-Adresse, Datum oder Telefonnummer abgefragt und direkt überprüft werden. Diese Validierung kann komplett personalisiert werden, da SurveyJS diese mit unterschiedlichen Ausdrücken bearbeiten kann.

Bedienung und Bedienbarkeit

In der SurveyJS Webseite wird der Fragebogen-Editor durch das Ausführen des Survey-Creator geöffnet. Dabei wird eine neue Seite aufgerufen, welche einen leeren Fragebogen und die verschiedenen Frage-Typen, die zur Verfügung stehen, anzeigt.

Um eine Frage hinzuzufügen, wählt der Nutzer aus der Toolbox eine der verfügbaren Fragetypen aus und zieht ihn per Drag & Drop in die gewünschte Position im Fragebogen. Diese Funktionalität ist jedoch nur in der Desktop-Version anwendbar. Die Frage kann anschließend vom Nutzer bearbeitet werden. Die Reihenfolge der Fragen kann, nachdem sie dem Fragebogen hinzugefügt wurde, auch mit Drag & Drop geändert werden. Die Fragen lassen sich auch in einen anderen Frage-Typ umwandeln. Die Daten werden dabei behalten, wenn der neue Frage-Typ die entsprechenden Optionen dafür enthält.

Wenn der Fragebogen fertig ist oder die Arbeit gespeichert werden muss, kann der Nutzer den Fragebogen einbetten oder die aktuelle Definition in den JSON-Editor kopieren und in einer Textdatei bzw. lokal speichern. Der aktuelle Zustand wird im Browser zwischengespeichert und kann bei einem erneuten Zugriff auf die Webseite weiter genutzt werden. Jede Fragebogen-Definition lässt sich auf diese Weise als ein Template benutzen.

Eine Vorschau vom aktuellen Fragebogen bietet SurveyJS auch. Der Fragebogen-Ersteller kann den Fragebogen als Teilnehmer ausfüllen. Die Vorschau hat verschiedene Geräte-Ansichten zur Auswahl.

Lokalisierung

Die Verknüpfung von Fragebogen und Lokalisierung wird in SurveyJS nicht unterstützt. Es können daher keine Fragebögen durch einen Standort eingegrenzt werden.

Medienunterstützung

Es gibt einen speziellen Frage-Typ („File“) für alle Dateiformate (mp3, mp4, wav und viele mehr). Dieser Frage-Typ unterstützt höchstens eine Datei pro Frage. Diese Datei kann entweder per Drag & Drop oder per Durchsuchen der eigenen Dateien eingefügt werden.

Privatsphäre

Um Fragebögen zu erstellen und zu beantworten, wird keine Anmeldung benötigt. Ein Konto wird gebraucht, wenn weitere Dienste vom SurveyJS in Anspruch genommen werden.

Erweiterbarkeit

SurveyJS besitzt eine kostenfreie Bibliothek zur Erweiterung des Fragebogen-Editors. Sie kann in der eigenen Anwendung als eine externe Bibliothek eingebunden werden. Diese Bibliothek besitzt eine API, welche die gesamte Funktionalität des Fragebogen-Editors enthält und komplett personalisiert werden kann. Durch diese API können Entwickler ihren eigenen Fragebogen-Editor basierend auf dem vom SurveyJS bauen.

2.1.3 Forms.app

„*Forms.app*“ ist eine Anwendung der gleichnamigen Firma und ist eine Plattform zur Formular- und Fragebogenerstellung. Alle erstellten Fragebögen werden direkt in ihren Servern gespeichert und dort auch für die Teilnehmer zur Verfügung gestellt. Diese Anwendung lässt sich auch auf mobilen Geräten in einer App verwenden. (Forms.app, 2022)

Open vs. Closed Source

Forms.app ist eine proprietäre Anwendung und bietet vier verschiedene Preismodelle, wobei die kostenlose Version alle Funktionalitäten limitiert beinhaltet. In der kostenlosen Variante

hat der Nutzer Zugriff auf 10 Fragebögen, 150 Fragebogen-Antworten und 1000 Fragebogen-Ansichten pro Monat, 1000 Antworten Speicher und 100 MB Datenspeicher. Die Premium Versionen, die monatlich ab 12 USD (ca. 11€) kosten, bieten eine unbegrenzte Anzahl an Fragebögen und Fragebogen-Ansichten, bis zu 100.000 Fragebogen-Antworten pro Monat, sowie unbegrenzte Antworten- und 100 GB Datenspeicher. (Forms.app, 2022)

Mobile Kompatibilität des Fragebogen-Editors

Forms.app ist im Google Play Store und im AppStore für Android- und IOS-Mobilgeräte verfügbar und kann auch über einen Webbrowser aufgerufen werden.

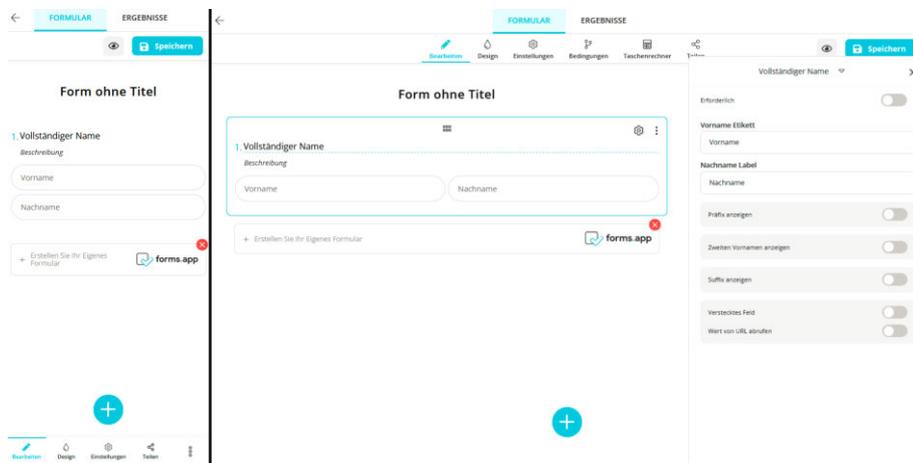


Abbildung 3: Forms.app auf mobilen Geräten (links) und auf dem Desktop (rechts),
Quelle: Screenshot aus Forms.app <https://forms.app>

Wie auf die Abbildung 3 zu erkennen ist, ist die Funktionalität des Fragebogen-Editors unabhängig von dem benutzten Gerät, wie es bei SurveyJS auch war. Durch die relative Anordnung der Elemente wird sichergestellt, dass die Bedienung konsistent auf verschiedenen Geräten bleibt. Um die Touch-Bedienung der mobilen Geräte zu verbessern, wird die Menüleiste nach unten auf die Seite gebracht. Forms.app ist hierdurch optimiert für mobile Geräte.

Template Mechanismus

Alle Fragebögen in Forms.app sind auch als Templates zu betrachten und haben einen Header, der aus einem Titel in Textform besteht. Dieser Header kann mit einer Schaltfunktion entweder

ein- oder ausgeblendet werden. Die Fragebögen können auf der Startseite dupliziert, geteilt und bearbeitet werden. Forms.app bietet auch vorgefertigte Fragebögen und Formulare für verschiedene Situationen an. Es gibt u. a. Templates für Bewerbungsformulare, Website-Kontaktformulare und Feedback-formulare. Diese Templates werden von Forms.app veröffentlicht und können von jedem Nutzer eingesetzt und verändert werden.

Die Möglichkeit mehrere Fragebögen in einer großen Befragung zusammenzufassen gibt es in Forms.app nicht. Somit besteht keine Möglichkeit ein Soundwalk, wie oben definiert, anzulegen.

Die erstellten Fragebögen werden in der Forms.app Cloud gespeichert und können über soziale Medien mit einer URL geteilt werden. Diese URL verweist auf die Teilnehmer-Ansicht des Fragebogens auf den Forms.app Servern. In dieser Ansicht können die Teilnehmer den Fragebogen ausfüllen.

Umfang der Fragetypen

Forms.app unterstützt 28 Fragebogenkomponenten. Davon ist einer ein „Trenner“, der nicht als Frage-Typ mitzählt. Trotzdem sind einige diese Fragetypen doppelt vorhanden. Zum Beispiel werden Fragetypen für vollständige Namen, Adressen oder Telefonnummern bereitgestellt, obwohl diese mit einer Textfrage beantwortet werden könnten. Mehrfachauswahl, Skala, Matrix, Bewertung und Freier Text Fragen sind auch zur Auswahl vorhanden.

Dazu bietet Forms.app eine Antwortvalidierung, die abhängig vom Frage-Typ ist. So können genaue Informationen, wie E-Mail-Adresse, Datum oder Telefonnummer abgefragt und direkt überprüft werden.

Bedienung und Bedienbarkeit

Auf der Startseite wird ein neuer Fragebogen per Klick auf den Knopf “Formular erstellen” erstellt. Dabei wird eine neue Seite aufgerufen, wo schon vorgefertigte Templates zur Verfügung stehen. Der Fragebogen-Ersteller kann eines diese Templates verwenden oder ein “leeres” Template benutzen. Hierbei kann der Nutzer auswählen, wie der Fragebogen für die Teilnehmer dargestellt werden soll. Der Nutzer kann entweder eine Listenansicht oder eine

Schrittansicht, wobei jede Frage auf einer neuen Seite angezeigt wird, auswählen. Danach wird eine Seite mit einem leeren Fragebogen präsentiert. Eine andere Möglichkeit einen Fragebogen zu erstellen, ist ein vom Benutzer erstelltes Template zu verwenden. Dazu wird der gewünschte Fragebogen ausgewählt und dupliziert. Hierbei entsteht eine modifizierbare Kopie des ausgewählten Fragebogens.

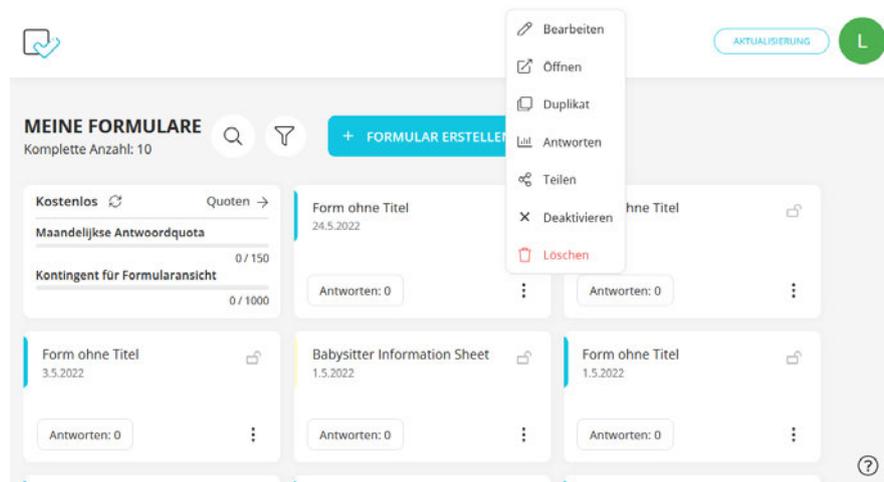


Abbildung 4: Dashboard Ansicht von Forms.app und die „Duplizieren“ Funktionalität, Quelle: Screenshot aus Forms.app <https://forms.app>

Um eine Frage hinzuzufügen, klickt der Nutzer auf einen "Floating Action Button (FAB)", wobei eine Toolbox geöffnet wird, wo sich alle Frage-Typen befinden. Die Fragetypen in der Toolbox können nicht mit Drag & Drop im Fragebogen eingefügt werden, weil diese als Drop-Down-Auswahl zur Verfügung stehen. Beim Auswählen einer der Fragetypen wird die Frage nach der ausgewählten Frage im Fragebogen eingefügt und kann anschließend vom Nutzer bearbeitet werden. Die Reihenfolge der Fragen kann, nachdem sie dem Fragebogen hinzugefügt wurde, mit Drag & Drop geändert werden. Die Fragen lassen sich auch in einen anderen Frage-Typ umwandeln, jedoch gehen dabei alle Daten in der Frage verloren.

Wenn der Fragebogen fertig ist oder die Arbeit gespeichert werden muss, kann der Nutzer den Fragebogen speichern. Der aktuelle Zustand wird gespeichert und kann jederzeit von der Startseite als Template genutzt werden. Dadurch ist jeder Fragebogen auch eine Vorlage in Forms.app. Sobald der Fragebogen gespeichert wird, wird diese auch veröffentlicht. Mittels einer URL können Teilnehmer auf den Fragebogen zugreifen und ihn beantworten.

Forms.app unterstützt auch eine Fragebogen-Vorschau. Der Nutzer kann dabei den Fragebogen als ein Teilnehmer abarbeiten. Der Vorschaumodus bietet auch die Möglichkeit die Ansicht zwischen Desktop, Tablet und Handy zu wechseln. Hiermit kann die Anzeige der gestellten Fragen in den verschiedenen Formaten überprüft werden.

Lokalisierung

In den Einstellungen kann eine Ortungs-Einstellung aktiviert werden. Hierbei wird ein Fragebogen einem Lokalisierungspunkt zugeordnet. Der Nutzer kann entweder ein Land oder eine Stadt angeben. Außerdem kann ein Punkt auf einer Karte mit einem entsprechenden Radius ausgewählt werden. Der minimale Radius beträgt einen Kilometer und der eigene Standort kann dabei nicht ausgewählt werden. Durch diese Einschränkung können nur Teilnehmer im ausgewählten Gebiet an der Befragung durch den entsprechenden Fragebogen teilnehmen.

Medienunterstützung

Medien werden in Forms.app mittels der Frage-Typen “Bild-Hochladen“ sowie “Datei-Hochladen” unterstützt. Beide Fragetypen können eins bis zehn Dateien akzeptieren. Trotzdem gibt es keine Möglichkeit ein Bild oder eine Datei an eine Frage anzuhängen.

Privatsphäre

Um Fragebögen zu erstellen und ausfüllen wird keine Anmeldung benötigt. Trotzdem ist eine Anmeldung erforderlich, um die Fragebögen zu teilen bzw. zu veröffentlichen. Der Fragebogen-Ersteller kann entscheiden, ob eine Anmeldung bei den Teilnehmern nachgefragt wird und bei den Datenschutzeinstellungen eingeben, wer Zugriff auf den Fragebogen hat. Es werden vier verschiedene Datenschutzeinstellungen angeboten:

- Privat: Nur der Fragebogen-Ersteller darf auf dem Fragebogen zugreifen.
- Eingeschränkt: Nur bestimmte Personen, die der Fragebogen-Ersteller auflistet, können auf den Fragebogen zugreifen.
- Nicht gelistet: Personen, die einen Link für den Fragebogen bekommen haben, können auf den Fragebogen zugreifen.
- Öffentlich: Für alle zugänglich.

Erweiterbarkeit

Forms.app bietet keine Unterstützung für weitere Entwicklungen an. Es gibt keine API, wo man auf Funktionen zugreifen könnte und der Code ist nicht öffentlich verfügbar. Deswegen sind Erweiterungen nicht für externe Nutzer möglich. Forms.app bietet stattdessen verschiedene Integrationsmöglichkeiten. Es können u. a. Google Sheets und Microsoft Excel Synchronisierungen stattfinden. Soziale Medien können auch in den Fragebögen integriert werden. Solche Funktionalitäten sind jedoch für diese Arbeit nicht relevant.

2.1.4 Google Forms

„*Google Forms*“ ist die Umfrage-Verwaltungssoftware von Google. Sie gehört zu den webbasierten Google-Suite-Diensten und ist wie alle ihre anderen Dienste nur im Browser verfügbar. Sie erschien 2006 und wurde seitdem von Google weiterentwickelt. (Google, 2022)

Open vs. Closed Source

Google Forms ist wie Forms.app eine proprietäre Anwendung und bietet neben der kostenlosen Version verschiedene Preis-Modelle. Die Premium Versionen sind mit Google Workspace verfügbar und bieten verbesserte Sicherheit, einen größeren Datenspeicher und erweiterten Support. Die Kernfunktionalitäten des Google Forms Fragebogen-Editors werden in den teureren Versionen nicht geändert. Somit kann der Fragebogen-Editor in der kostenlosen Version fast ohne Einschränkungen verwendet werden.

Mobile Kompatibilität des Fragebogen-Editors

Google Forms ist für alle mobilen Geräte, die einen Browser unterstützen, verfügbar. Es besitzt keine eigene App und ist nur durch einen Browser erreichbar.

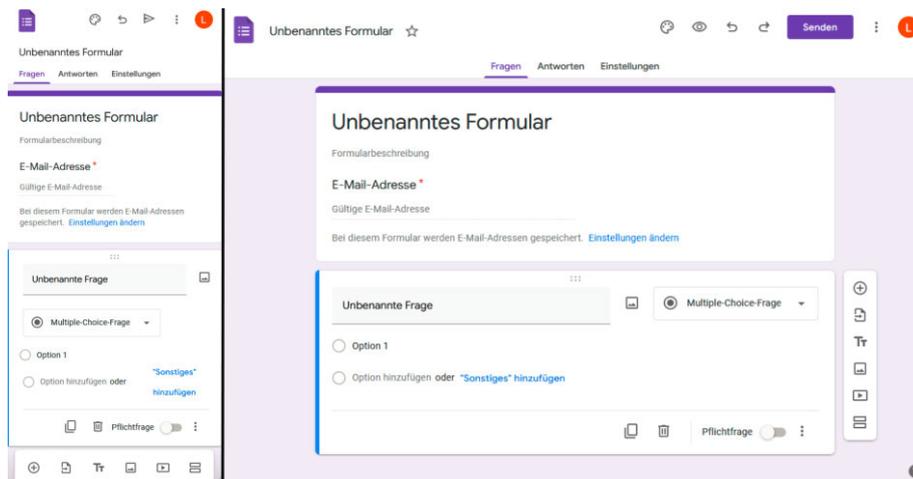


Abbildung 5: Google Forms auf mobilen Geräten (links) und auf dem Desktop (rechts),
Quelle: Screenshot aus Google Forms <https://docs.google.com/forms>

Wie auch bei Forms.app, ist auf dem Bild zu erkennen, dass die Funktionalität des Fragebogen-Editors konsistent unabhängig vom benutzten Gerät bleibt. Auf mobilen Geräten wird die Menüleiste im unteren Bereich der Ansicht geankert, um die Touch-Bedienung auf den mobilen Geräten zu verbessern. Der Google Forms Fragebogen-Editor ist hierdurch auch optimiert für mobile Geräte.

Template Mechanismus

Google Forms bietet auch vorgefertigte Fragebögen und Formulare an. Diese Templates werden von Google Forms veröffentlicht und können von jedem Nutzer eingesetzt und verändert werden. Analog zu Forms.app können die Fragebögen auf der Startseite entfernt und bearbeitet werden. Der Header des Fragebogens besteht aus dem Titel und einer Beschreibung. Es wird im Header kein Bild oder Logo unterstützt. Alle Fragebögen haben einen Header, welcher nur aus einem Titel in Textform besteht. Der Header kann in Google Forms nicht ausgeblendet werden. Jeder Fragebogen muss also einen Titel und eine Beschreibung haben.

Die Möglichkeit mehrere Fragebögen in einer großen Befragung zusammenzufassen gibt es in Google Forms auch nicht.

Die Fragebögen werden in Google Forms automatisch auf den Google Servern gespeichert. Der aktuelle Zustand von einem Fragebogen kann jederzeit von der Startseite als Template

benutzt und weiterbearbeitet werden. Dadurch ist jeder Fragebogen auch ein Template in Google Forms. Sobald der Fragebogen gespeichert wird, wird dieser auch veröffentlicht. Der Zugriff auf den Fragebogen erfolgt mit Einladung per E-Mail oder Link. Dazu kann der Fragebogen auch in einer HTML als iFrame eingebettet werden.

Umfang der Fragetypen

Google Forms unterstützt elf unterschiedliche Fragebogen-Komponenten. Es werden zwei Fragetypen zur Freitexteingabe unterstützt. Als Auswahlmöglichkeiten bietet Google Forms „Multiple-Choice-Fragen“ als Einfachauswahlfragen, „Kästchen“ für Mehrfachauswahlfragen und „Drop-down“ als eine Alternative zur Multiple-Choice-Fragen. Eine Frage zum Datei-Hochladen ist auch vorhanden. Dazu gibt es noch Skala und Raster Fragen, die einer Matrix-Frage entsprechen.

Die Fragen können per Knopfdruck dupliziert werden. Dabei wird die Kopie der Frage direkt unter der kopierten Frage angehängt. Dazu kann zwischen jeder Frage ein Bild, Video oder Titel und Beschreibung, sowie ein „Abschnitt“ hinzugefügt werden. Alle Fragen können mit einem Bild ergänzt werden. Hierbei kann das Bild aus den eigenen Dateien hochgeladen, direkt aus der Kamera importiert, per URL eingefügt oder aus der Google Bildersuche entnommen werden.

Die Antwort Validierung kann auch für jede Frage aktiviert werden. Sie ist für jeden Fragetyp anders definiert. Zum Beispiel wird für Text-Antworten eine Länge-Beschränkung oder eine Regulären-Ausdruck-Suche angeboten. Eine Einschränkung auf Zahlen, Text, E-Mail-Adresse oder URL ist auch möglich. Bei Multiple-Choice-Fragen kann auch eine bestimmte Anzahl definiert werden, die beantwortet werden muss. Alle diese Validierungen können auch mit einer benutzerdefinierten Fehlermeldung versehen werden.

Bedienung und Bedienbarkeit

Auf der Startseite, wo schon vorgefertigte Templates zur Verfügung stehen, wird ein neuer Fragebogen per Klick auf das Template „Leer“ erstellt. Dabei wird eine neue Seite aufgerufen und ein leerer Fragebogen präsentiert. Diese ist die einzige Möglichkeit einen neuen Fragebogen zu erstellen.

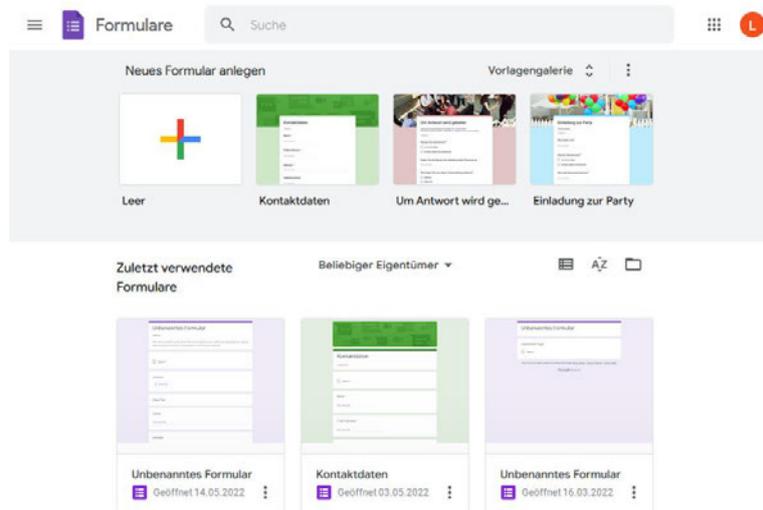


Abbildung 6: Dashboard Ansicht von Google Forms, Quelle: Screenshot aus Google Forms <https://docs.google.com/forms>

Das Hinzufügen von Fragen funktioniert wie bei Forms.app. Dabei wird standardmäßig eine Multiple-Choice -Frage, die vom Benutzer bearbeitet werden kann, erstellt. Die Fragetypen in der Toolbox können auch nicht mit Drag & Drop im Fragebogen eingefügt werden, weil diese als Drop-Down Auswahl zur Verfügung stehen. Trotzdem kann die Reihenfolge der Fragen, nachdem sie dem Fragebogen hinzugefügt wurden, mit Drag & Drop geändert werden. Die Fragen lassen sich auch in einen anderen Frage-Typ umwandeln. Die Daten werden dabei übernommen, wenn die passenden Optionen im neuen Frage-Typ zur Verfügung stehen.

Durch die “Frage-Importieren”-Funktion werden alle Fragen von einem anderen erstellten Fragebogen zur Verfügung gestellt. Der Nutzer kann dabei die gewünschten Fragen auswählen und im aktuellen Fragebogen verwenden. Dies entspricht dem Template Mechanismus von Google Forms.

Google Forms unterstützt auch eine Fragebogen-Vorschau. Hierzu wird die Ansicht gewechselt und der Nutzer kann den Fragebogen als Teilnehmer abarbeiten. Der Vorschaumodus unterstützt keine unterschiedlichen Geräte-Ansichten. Trotzdem reagiert die Seite auf Anzeige-Veränderungen und wechselt dementsprechend in den mobilen Anzeigemodus. Neben dem Fragebogen-Editor besitzt Google Forms auch ein Live-Tracking System mit Diagrammen für Fragebogenergebnisse. Sobald ein Teilnehmer den Fragebogen ausgefüllt und ihn eingereicht hat, kann der Fragebogen-Ersteller die Antworten sehen und exportieren.

Lokalisierung

Google Forms bietet keine Lokalisierung für Fragebögen. Es gibt auch kein Add-On der diese Funktionalität ergänzt.

Medienunterstützung

Wie in Forms.app werden Medien in Google Forms durch den entsprechenden Frage-Typ „Datei-Hochladen“ eingefügt. Es können bis zu zehn Bild-, Audio-, Video- und andere Dateien mit jeweils einer Maximalgröße von zehn GB hochgeladen werden. Trotzdem gibt es keine Möglichkeit als Teilnehmer ein Bild oder eine Datei an eine andere Frage anzuhängen.

Privatsphäre

Um Fragebögen zu erstellen und zu beantworten, wird eine Anmeldung mit einer Google E-Mail-Adresse benötigt. Trotzdem kann der Fragebogen-Ersteller entscheiden, die E-Mail-Adressen der Teilnehmer nicht zu erfassen. Dazu wird auch die Möglichkeit angeboten, die Anzahl der Antworten pro Teilnehmer auf maximal eine Antwort pro Person zu begrenzen.

Erweiterbarkeit

Bei Google Forms wird eine REST-API zur Erstellung und Änderung eines Fragebogens zur Verfügung gestellt. Das Abrufen von Antworten und das Lesen von Inhalten und Metadaten ist durch diese API auch möglich. Dazu kann das Einrichten und Empfangen von Push-Benachrichtigungen eingestellt werden. Die Nutzung der API-Funktionalität ist jedoch nicht kostenlos. Ein Zugang zur Google Cloud Plattform wird dafür benötigt. (Google, 2022)

Google Forms bietet auch einen Marketplace für Drittanbieter-Add-Ons. Es können zum Beispiel Zeitschaltuhren an die Fragebögen angebunden werden. Dazu ist es möglich die E-Mail-Benachrichtigungen von Google Forms mit Add-Ons zu bearbeiten. Erweiterte Symbolbibliotheken stehen auch zur Verfügung. Diese Funktionalitäten sind jedoch für diese Arbeit nicht interessant.

2.1.5 SurveyMonkey

„*SurveyMonkey*“ ist eine der größten Plattformen für Online-Umfragen auf dem Markt. Sie wurde von Ryan Finley 1999 gegründet und seitdem weiter ausgebaut. Sie bietet eine mobile App und eine webbasierte Anwendung für die Erstellung von Fragebögen. (SurveyMonkey, 2022)

Open vs. Closed Source

SurveyMonkey ist eine proprietäre Anwendung und bietet neben der kostenlosen Version verschiedene Preis-Modelle an. Die kostenlose Variante hat einige Einschränkungen. Zum Beispiel werden 10 Fragen per Fragebogen erlaubt und die maximale Anzahl an Antworten liegt bei 100. Hierdurch ist die Anzahl der Fragetypen auch begrenzt. Ein Datenexport wird in der kostenlosen Variante auch nicht angeboten. Die kostenpflichtigen Modelle bieten mehr Personalisierungsmöglichkeiten, Fragetypen und Optionen, sowie Analysemöglichkeiten, eine erweiterte Fragebogen-Logik und Kundensupport. (SurveyMonkey, 2022)

Mobile Kompatibilität des Fragebogen-Editors

SurveyMonkey ist im Google Play Store und im AppStore für Android- und IOS-Mobilgeräte verfügbar und kann auch über einen Webbrowser aufgerufen werden. Die mobile und die Browser-Version unterscheiden sich in der Benutzeroberfläche und im Umfang der Fragetypen. Die Browser Version kann auch auf mobilen Geräten verwendet werden. Die Anzeige wird dabei nicht an die Größe vom Display angepasst.

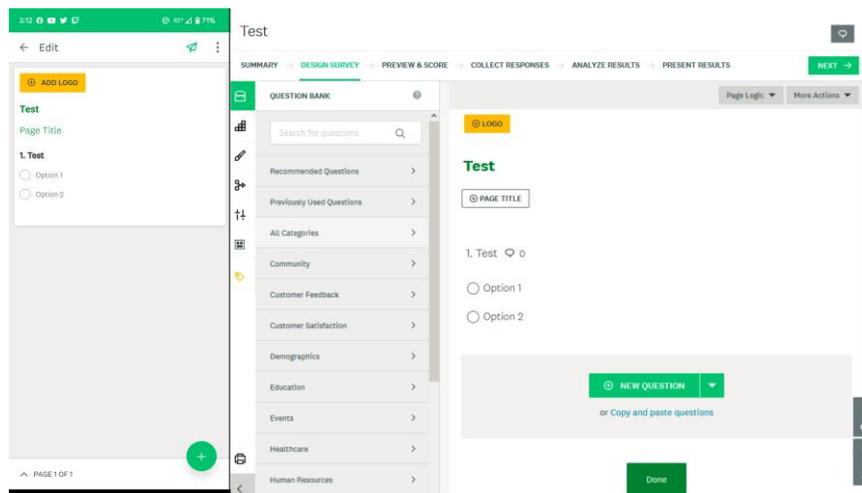


Abbildung 7: SurveyMonkey auf mobilen Geräten (links) und auf dem Desktop (rechts),
Quelle: Screenshot aus SurveyMonkey <https://de.surveymonkey.com>

SurveyMonkey erkennt, dass der Nutzer ein mobiles Gerät benutzt und wechselt bei Smartphones und Tablets zur App-Ansicht. Diese Ansicht ist für mobile Geräte optimiert und ändert den kompletten Fragebogen-Editor. Dabei wird wie bei Forms.app ein „FAB“ für die Frage-Erstellung verwendet und die Menüs, wie in Abbildung 7 zu sehen ist, vereinfacht.

Template Mechanismus

SurveyMonkey bietet die Möglichkeit ein Template aus einem Fragebogen zu bilden. Dabei können nur eigene Fragebögen benutzt werden. Es werden auch vorgefertigte Templates für verschiedene Situationen angeboten. Diese Templates sind jedoch kostenpflichtig.

Der Header des Fragebogens besteht aus dem Titel und einer Beschreibung. Dazu können ein Seitentitel und eine Seitenbeschreibung für mehrseitige Fragebögen eingegeben werden. Ein Logo oder ein Bild in dem Header hinzuzufügen, ist möglich. Hierfür wird jedoch die kostenpflichtige Version benötigt. Der Header kann in SurveyMonkey auch nicht ausgeblendet werden. Die Möglichkeit, mehrere Fragebögen in einer großen Befragung zusammenzufassen gibt es in SurveyMonkey nicht.

Die Fragebögen können von der Startseite entfernt, bearbeitet und geteilt werden. Dazu können Ergebnisse von aktiven Fragebögen in dieser Ansicht direkt analysiert werden. Die Fragebögen

werden in SurveyMonkey automatisch auf den SurveyMonkey Servern gespeichert. Der aktuelle Zustand kann jederzeit von der Startseite als Template benutzt und weiterbearbeitet werden. Dadurch kann jeder Fragebogen in SurveyMonkey auch ein Template sein. Sobald der Fragebogen gespeichert wird, wird dieser auch veröffentlicht. Der Zugriff auf den Fragebogen erfolgt durch eine Einladung über soziale Medien oder einen Link. Dazu kann der Fragebogen auch in einer HTML als iFrame oder in eine mobile App eingebettet werden.

Umfang der Fragetypen

SurveyMonkey unterstützt eine unterschiedliche Anzahl an Fragetypen in der App und im Browser. In der App werden sechs Fragetypen unterstützt. Einer dieser Fragetypen ist eine Suche von oft verwendeten Fragen. Diese Fragen sind in einer Fragendatenbank von SurveyMonkey abgelegt und können als Vorlage benutzt werden. Die klassischen Multiple-Choice-, Text- und Dropdown-Fragen können in der App genutzt werden. Dazu können ein Bild und ein Absatz dem Fragebogen hinzugefügt werden. Die Browser-Version bietet insgesamt 22 Fragetypen. Dabei werden neben den sechs Fragetypen noch erweiterte Fragetypen, wie Sternebewertungen, Dateiuploads, Rankings, Matrizen u.v.m. unterstützt. Einige dieser Fragetypen sind jedoch nicht kostenlos. Da die Dateiupload-Frage kostenpflichtig ist, ist es nur möglich für den Fragebogen-Ersteller Medien in dem Fragebogen hinzuzufügen. Die Teilnehmer können, ohne diesen Frage-Typ ihre Aufnahmen nicht hinzufügen.

Alle Fragen können mit einem Link, Bild, Video oder Tooltip ergänzt werden. Hierbei werden das Bild und das Video per URL eingefügt. Die Antwort-Validierung kann auch in jeder Textfrage benutzt werden. Sie können zum Beispiel nur Zahlen, Daten oder E-Mail-Adressen beinhalten. Dabei kann die Zeichenlänge eingestellt werden. Diese Validierungen können nicht mit einer benutzerdefinierten Fehlermeldung versehen werden. Außerdem werden die Antworten in Auswahl-Fragen nicht validiert.

Bedienung und Bedienbarkeit

Auf der Startseite wird ein neuer Fragebogen per Klick auf den "Erstellen"-Knopf erstellt. Dabei wird eine neue Seite aufgerufen, wo der Nutzer einen vorherigen Fragebogen kopieren oder

einen Fragebogen von Grund auf neu erstellen kann. Danach wird eine Seite mit entweder einem leeren oder die Kopie eines alten Fragebogens angezeigt. Bei einem leeren Fragebogen wird der Nutzer, wie bei Forms.app, nach dem Format des Fragebogens gefragt. Hierbei kann er auch zwischen einer Listen- und Schritt-Ansicht auswählen. Bei dem kopierten Fragebogen werden alle Fragen direkt in den neuen Fragebogen kopiert.

Um eine Frage hinzuzufügen, gibt es zwei Möglichkeiten. In der Desktop Version klickt der Nutzer auf den „Neue Frage“ Knopf am Ende des Fragebogens. In der App-Version klickt der Nutzer auf den „FAB“ Knopf im unteren Bereich der Ansicht. Die Frage wird in beiden Fällen am Ende eingefügt. Dabei wird eine leere Frage erstellt. Diese kann anschließend vom Benutzer bearbeitet werden. Neben dem Fragetext, wie im Google Forms, hat der Nutzer die Auswahlmöglichkeit, welchen Frage-Typ er verwendet möchte. Es gibt in diesem Fall keine Toolbox und kein Drag & Drop von Frage-Typen im Fragebogen. Trotzdem kann die Reihenfolge der Fragen, nachdem sie dem Fragebogen hinzugefügt wurden, mit Drag & Drop geändert werden, wie es bei allen anderen Anbietern auch möglich ist. Die Fragen lassen sich auch in einen anderen Frage-Typ umwandeln, wobei die Daten übernommen werden, wenn die passenden Optionen im neuen Frage-Typ zur Verfügung stehen.

Im „Kioskmodus“ von SurveyMonkey läuft die Umfrage in einer Schleife. So kann ein Gerät als Umfrage-Station verwendet werden. Dazu kann der Fragebogen-Ersteller auch die Antworten pro Person begrenzen. Die Live-Ergebnisse können auch auf der Startseite verfolgt werden. Sobald die Teilnehmer den Fragebogen ausgefüllt und eingereicht haben, kann der Fragebogenersteller die Antworten sehen und exportieren.

SurveyMonkey unterstützt auch eine Fragebogen-Vorschau. Durch Klicken auf dem „Weiter“ Knopf wird die Ansicht gewechselt und der Nutzer kann den Fragebogen ausprobieren. Der Vorschaumodus unterstützt unterschiedliche Geräte-Ansichten. Dabei kann zwischen drei Gerätetypen ausgewählt werden. Es stehen Desktop-, Tablet- und Smartphone-Ansichten zur Verfügung. Hierbei kann der Nutzer die Fragen so sehen, wie ein Teilnehmer sie auf seinem Gerät sehen würde.

Lokalisierung

SurveyMonkey bietet wie auch Google Forms keine Lokalisierung für Fragebogen. Deswegen gibt es keine Möglichkeit ein Fragebogen einem geographischen Punkt zuzuordnen.

Medienunterstützung

Es gibt ein Frage-Typ, der das Datei-Hochladen erlaubt, allerdings ist dieser Frage-Typ eine kostenpflichtige Funktionalität. Dies ist auch die einzige Möglichkeit, als Teilnehmer Medien dem Fragebogen hinzuzufügen. Als Dateitypen können Bilder und Dokumente hochgeladen werden.

Privatsphäre

Um Fragebögen zu erstellen, wird eine Anmeldung mit einer E-Mail-Adresse benötigt. Trotzdem kann der Fragebogen-Ersteller entscheiden, die E-Mail-Adressen der Teilnehmer nicht zu erfassen. Dazu wird auch die Möglichkeit angeboten, die Anzahl der Antworten pro Teilnehmer auf maximal eine Antwort pro Person zu begrenzen.

Erweiterbarkeit

Wie Google Forms hat SurveyMonkey auch eine REST-API zur Verfügung. Diese ist jedoch nur begrenzt verfügbar. Um diese API zu benutzen, muss der Nutzer zuerst eine App in seinem SurveyMonkey Profil registrieren. Nach 90 Tage wird diese entweder öffentlich oder privat bereitgestellt. Beide Optionen sind auch kostenpflichtig. Öffentliche Apps sind für alle Nutzer mit einem SurveyMonkey Konto verfügbar und müssen von SurveyMonkey überprüft werden. Private Apps können außerhalb von SurveyMonkey betrieben werden. Die REST-API unterstützt die Erstellung und Bearbeitung von Fragebögen sowie das Abrufen von deren Ergebnissen.

2.2 Vergleich und Zusammenfassung des Stands der Technik und Wissenschaft

Wie in diesem Kapitel zu sehen ist, existieren bereits verschiedene mobile Fragebogen-Editoren. Jedoch erfüllen diese die, im Teilkapitel 2.1 formulierten Kriterien nicht vollständig. Im Folgenden werden mit Hilfe einer Bewertungstabelle die Kriterien zur Auswertung von Fragebogen-Editoren zusammengefasst und verglichen.

Tabelle 1: Zusammenfassung der Kriterien zur Auswertung von Fragebogen-Editoren

<i>Kriterien</i>	<i>SurveyJS</i>	<i>Forms.app</i>	<i>Google Forms</i>	<i>SurveyMonkey</i>
Open vs. Closed Source	Open	Closed	Closed	Closed
Mobile Kompatibilität	Web	App und Web	Web	App und Web
Template Mechanismus	Keine	Limitiert	Limitiert	Limitiert
Umfang Frage-Typen	20	28	11	6 oder 22
Bedienung	D&D*	Limitiert D&D	Limitiert D&D	Limitiert D&D
Lokalisierung	Keine	Limitiert	Keine	Keine
Medienunterstützung	Ja	Ja	Ja	Limitiert
Privatsphäre	Keine	Anmeldung	Anmeldung	Anmeldung
Erweiterbarkeit	API	Keine	Limitiert	Limitiert

* Drag & Drop

Die betrachteten Fragebogen-Editoren unterstützen eine Mobilgeräte-Ansicht entweder durch eine App oder einen Webbrowser. Sie beinhalten die erforderliche Menge an Frage-Typen und die Unterstützung von Medien wird durch einen dieser Frage-Typen ermöglicht. Außerdem ist die Bedienung bei den Fragebogen-Editoren ähnlich. Dies macht die Nutzung von verschiedenen Fragebogen-Editoren einfacher, da das Prinzip der Bedienung unabhängig vom benutzten Editor erhalten bleibt.

Keiner der Fragebogen-Editoren unterstützt eine genaue Lokalisierung. Für die Soundwalks werden kleine Radien benötigt, die kein Anbieter anbieten kann. Zudem gibt es auch keinen ausgereiften Template Mechanismus in den Fragebogen-Editoren. Dieser Mechanismus muss Soundwalks, welche mehrere Fragebögen in einer großen Befragung zusammenfassen, erstellen können. Dazu muss der Nutzer sich bei drei der Anbieter anmelden, um ein Fragebogen erstellen oder veröffentlichen zu können. Dies ist bei SurveyJS jedoch nicht der Fall, da es dort keine Unterscheidung zwischen Ersteller und Teilnehmer geben kann. In diesem Sinne kann keiner der Fragebogen-Editoren die gewünschte Funktionalität für die Privatsphäre erreichen. SurveyJS unterscheidet sich auch von den anderen Anbietern, indem es der einzige Anbieter ist, der eine Open Source Bibliothek kostenlos zur Verfügung stellt. Diese Bibliothek kann benutzt werden, um einen eigenen Fragebogen-Editor zu entwickeln und die Funktionalität vom SurveyJS zu erweitern.

Auf Grundlage der Ergebnisse dieses Kapitels und den Anforderungen aus Kapitel 3, wird in Kapitel 4, unter Verwendung von der SurveyJS-Library ein Konzept für einen Drag & Drop Fragebogen-Editor entwickelt. Dazu werden die erweiterten Aspekte der Verwaltung von Fragebogen- und Soundwalk-Templates sowie die Integration von Geolokalisierung im Fokus stehen.

3 Anforderungsanalyse

Dieser Abschnitt befasst sich mit der Bereitstellung, der für den Entwurf-Prozess benötigten Daten. In Teilkapitel 3.1 wird in die funktionalen Anforderungen eingeführt, welche für die verschiedenen Funktionen des zu entwickelnden Fragebogen-Editor formuliert werden müssen. In Teilkapitel 3.2 werden die nicht-funktionalen Anforderungen bezüglich der Bedienbarkeit erläutert. Im letzten Teilkapitel 3.3 werden die Ergebnisse der Analyse diskutiert und ein Ausblick auf den Software-Entwurf dargestellt.

3.1 Funktionale Anforderungen

In den funktionalen Anforderungen wird festgelegt, was das System ausführen soll. (Balzert, 2009) Die Anwendung soll ein Soundwalk-Verwaltungs-System mit Drag & Drop Funktionalitäten implementieren. Um dies zu realisieren, soll die Anwendung in zwei Hauptbereiche unterteilt werden. Für die gesamte Verwaltung soll ein Dashboard implementiert werden, wo alle Fragebögen und Soundwalks angezeigt und bearbeitet werden können. Dabei soll jeder Fragebogen und jeder Soundwalk als Template für weitere Fragebögen bzw. Soundwalks benutzt werden können. Der zweite Hauptbereich soll den Fragebogen-Editor umfassen, wo der Nutzer ein Fragebogen per Drag & Drop zusammenstellen und sich eine Vorschau davon zeigen lassen kann. Dazu soll es für den Benutzer möglich sein, einen Lokalisierungspunkt für den Fragebogen zu setzen. Die Soundwalks und die Fragebögen sollen in einer Datenbank gespeichert werden können, um zur weiteren Nutzung zur Verfügung zu stehen.

3.1.1 Dashboard

Das Dashboard soll einen Überblick über alle erstellten Soundwalks und Fragebögen bieten. Soundwalks sind Sammlungen von Fragebögen. Fragebögen sind Sammlungen von Fragebogen-Komponenten. Diese Sammlungen sollen im Dashboard übersichtlich dargestellt werden. Dazu soll es möglich sein, einen neuen Soundwalk oder Fragebogen im Dashboard zu erstellen und weiter bearbeiten zu können. Anschließend soll ein Fragebogen per Drag & Drop einem Soundwalk hinzugefügt werden können.

Der Template-Mechanismus soll im Dashboard zur Verfügung stehen. Der Mechanismus in dieser Anwendung soll eine Kopie eines Fragebogens oder einen Soundwalk erstellen. Dies soll im Dashboard erfolgen, weil der Nutzer in diesem Bereich ein Überblick über alle Fragebögen und Soundwalks hat.

Für Soundwalks soll das „Bearbeiten“ den Wechsel der Ansicht auf die Sammlung von Fragebögen des ausgewählten Soundwalks bewirken. Auf dieser Seite soll der Nutzer Fragebögen aus dem Soundwalk entfernen oder hinzufügen können. Solche Änderungen werden aktiv in der Datenbank per Knopfdruck gespeichert. Die Anwendung soll speichern, wenn die entsprechende Aktion vom Benutzer ausgeführt wird. Das „Löschen“ des Soundwalks soll den Soundwalk und seine Fragebögen entfernen. Beim „Kopieren“ soll der gesamte Soundwalk noch einmal nachgebildet werden. Durch das Kopieren wird ein Soundwalk als Template für andere Soundwalks benutzt.

Bei den Fragebögen soll durch das „Bearbeiten“ die Ansicht im Fragebogen-Editor gewechselt werden. Die Anforderungen an den Fragebogen-Editor werden im nächsten Teilkapitel vorgestellt. Das „Löschen“ soll den Fragebogen aus der Datenbank entfernen. Das „Kopieren“ soll analog wie beim Soundwalk funktionieren und dem Template-Mechanismus entsprechen.

Tabelle 2: Zusammenfassung der Anforderungen an das Dashboard

<i>Anforderung (AF)</i>	<i>Beschreibung</i>
AF01	Die Anzeige von Fragebögen und Soundwalks soll übersichtlich dargestellt werden
AF02	Das Anlegen von Fragebögen und Soundwalks soll im Dashboard unterstützt werden.
AF03	Fragebögen und Soundwalks sollen gelöscht, kopiert (Template-Mechanismus) und bearbeitet werden können.
AF04	Die Fragebögen sollen per Drag & Drop zu den Soundwalks hinzugefügt werden können.
AF05	Soundwalks und Fragebögen sollen in einer Datenbank gespeichert werden können.

3.1.2 Fragebogen-Editor

Auf den Fragebogen-Editor soll der Nutzer, wie oben erörtert, durch das Klicken auf ein bestehendes Template aus dem Dashboard kommen. Die Erstellung von Fragebögen soll einem Baukasten ähneln. Jeder Fragebogen soll einen Header, der nicht gelöscht und vom Benutzer bearbeitet werden kann beinhalten. Mit Hilfe einer Toolbox soll der Nutzer in der Lage sein, unterschiedliche Frage-Typen per Drag & Drop in diesen Baukasten hinzuzufügen. Der Nutzer soll dazu ein Element (Frage-Typ) aus der Toolbox oder aus dem Fragebogen wählen und es in die gewünschte Position hineinziehen. Die sich bereits im Fragebogen befindenden Fragen sollen nach oben bzw. nach unten verschoben werden.

Es sollen mindestens die Frage-Typen Einfach-, Mehrfachauswahl, Ranking bzw. Bewertung, Matrix und Freier-Text, sowie die Möglichkeit Dateien wie Bilder, Videos und Tonaufnahmen

hochzuladen, zur Verfügung stehen. Insbesondere Tonaufnahmen sind für die Soundwalks relevant. Deswegen soll es möglich sein, jeden Fragebogen mit mindestens einer Tonaufnahme zu versehen.

Das Kopieren der Fragebögen verursacht das Kopieren der Fragen. Durch diesen Template-Mechanismus sollen die Fragen leicht zwischen den Fragebögen importiert werden.

Wenn der Nutzer eine Frage bearbeiten möchte, soll er auf diese klicken können. Hierdurch soll die Bearbeitung von der Frage aktiviert werden. Anschließend sollen die Änderungen gespeichert und die Daten in der Datenbank aktualisiert werden.

Um die Fragebögen vor der Veröffentlichung zu testen, soll eine Vorschau für den Fragebogen-Ersteller zur Verfügung stehen. Dabei soll der Nutzer sehen können, wie der Fragebogen auf seinem Gerät aussehen würde.

Wenn ein Fragebogen an einem bestimmten Ort durchgeführt werden soll, soll der Fragebogen-Editor die Lokalisierung unterstützen. Den Fragebögen soll ein Lokalisierungspunkt zugewiesen werden können.

Tabelle 3: Zusammenfassung der Anforderungen an den Fragebogen-Editor

<i>Anforderung (AF)</i>	<i>Beschreibung</i>
AF06	Der Fragebogen-Editor soll für das Hinzufügen und Bearbeiten von Fragen Drag & Drop-Unterstützung anbieten.
AF07	Jeder Fragebogen soll einen Header besitzen.
AF08	Fragetypen Einfach-, Mehrfachauswahl, Ranking bzw. Bewertung, Matrix, Text, mehrfacher Text, sowie Dateien (Tonaufnahmen) sollen zur Verfügung stehen.
AF09	Bearbeitete Fragen sollen gespeichert und in der Datenbank aktualisiert werden.
AF10	Eine Fragebogenvorschau soll dem Nutzer zur Verfügung stehen.
AF11	Fragebögen sollen einem Lokalisierungspunkt zugewiesen werden können.

3.1.3 Administrator und Teilnehmer

In der Anwendung sollen zwei Rollen unterstützt werden. Einerseits gibt es den Fragebogen-Ersteller bzw. den Administrator und andererseits die Fragebogen-Teilnehmer. Um den Administrator-Bereich vor Teilnehmer-Zugriffen zu schützen, soll dieser Passwort geschützt werden. Deswegen sollen zwei Ansichten implementiert werden. In der einen Ansicht sollen für den Fragebogen-Ersteller die Fragebögen und Soundwalks zur Bearbeitung zur Verfügung stehen, während in der Teilnehmer Ansicht nur die eingblendeten Soundwalks angezeigt werden sollen. Die Teilnehmer-Ansicht ist jedoch nicht Teil dieser Arbeit.

Tabelle 4: Zusammenfassung der Anforderungen an die Rollen

<i>Anforderung (AF)</i>	<i>Beschreibung</i>
AF12	Es soll eine Trennung vom Administrator- und Teilnehmer-Bereich stattfinden.

3.2 Nicht-Funktionale Anforderungen

In diesem Abschnitt werden die nicht-funktionalen Eigenschaften als weitere Anforderungen aufgestellt. Diese sollen die verschiedenen Konzepte der Fragebogen-Editor-Entwicklung unterstützen und zu einer höheren Akzeptanz beitragen. Nicht-funktionale Anforderungen beschreiben in welcher Qualität das System die Leistung erbringen soll und welche Randbedingungen eingehalten werden müssen. Diese Eigenschaften untergliedern sich in Funktionalität, Zuverlässigkeit, Weiterentwickelbarkeit, Benutzbarkeit, Effizienz, Wartbarkeit und Portabilität der Anwendung. (Balzert, 2011)

In den folgenden Tabellen werden alle wichtigen Qualitätsanforderungen und deren Unterpunkte auf ihre Wichtigkeit für die Anwendung geprüft und eingeordnet. Jedoch werden nur die, für diese Arbeit relevanten nicht-funktionale Anforderungen vorgestellt.

3.2.1 Zuverlässigkeit

Ein Softwareprodukt ist zuverlässig, wenn die „*Fähigkeit des Softwareproduktes, ein spezifiziertes Leistungsniveau zu bewahren, wenn es unter festgelegten Bedingungen benutzt, wird*“ besteht (Balzert, 2011 S. 124). Der relevanteste Unterpunkt dieser Qualitätsanforderung ist die Fehlertoleranz. Für die umzusetzende Anwendung ist eine Internetverbindung notwendig. Dabei ist davon auszugehen, dass das Mobilgerät nicht immer mit der besten Übertragungsgeschwindigkeit versorgt wird. Auf Grund dessen sollen beim Laden der Fragebögen und der darin genutzten Medien entsprechende Anpassungen vorgenommen werden. Die Auflösung der Bilder soll zum Beispiel niedrig gehalten werden. Dazu sollen die Benutzereingaben auf Korrektheit überprüft, sowie mögliche Fehlerquellen durch entsprechende Ausnahme-Regeln

abgefangen werden. Die Gewichtung der Unterpunkte dieser nicht-funktionale Anforderung für diese Arbeit zeigt die folgende Tabelle.

Tabelle 5: Spezifizierung der Zuverlässigkeit

<i>Systemqualität</i>	<i>Sehr wichtig</i>	<i>Wichtig</i>	<i>Normal</i>	<i>Irrelevant</i>
Zuverlässigkeit				
Reife			x	
Fehlertoleranz		x		
Wiederherstellbarkeit		x		

3.2.2 Benutzbarkeit

Die Benutzbarkeit wird definiert durch die „*Fähigkeit des Softwareproduktes, vom Benutzer verstanden und benutzt zu werden sowie für den Benutzer erlernbar und attraktiv zu sein, wenn es unter den festgelegten Bedingungen benutzt, wird*“ (Balzert, 2011 S. 130). Die wichtigsten Qualitätsmerkmale liegen in der Benutzbarkeit der Anwendung. Der Nutzer muss in der Lage sein, die Benutzeroberfläche sofort zu begreifen und die Elemente darin zu bedienen.

Für das Erreichen eines intuitiven Verständnisses dieser Anwendung wird auf die Erlernbarkeit großen Wert gelegt. Diese ermöglicht dem Nutzer die Anwendung einfach zu bedienen. Um diese Bedienbarkeit zu erreichen, soll Drag & Drop überall, wo es sinnvoll ist, angewendet werden. Die Oberflächengestaltung ist in diesem Fall die Attraktivität der Anwendung. Sie soll auf das Design und die Standardfarben der HAW-Hamburg Rücksicht nehmen.

Im Gegensatz zu Desktop-Umgebungen, ist bei der Entwicklung von Anwendungen für mobile Geräte der wesentlich kleinere Bildschirm zu beachten. Aus diesem Grund sollen die Benutzeroberflächen entsprechend angepasst werden, um die UI-Elemente angemessen darstellen zu können. Vor allem sollte das horizontale Scrollen vermieden werden.

Die Bedienung bei mobilen Anwendungen ist meistens auf einen Touchscreen ausgelegt, weshalb große Bedienelemente verwendet werden müssen. Diese großen Elemente sollen auch die Drag & Drop Funktionalität unterstützen. Darüber hinaus soll es dem Nutzer möglich sein, sein Ziel schnell zu erreichen. Dafür ist eine intuitive Navigation zu realisieren. Die Gewichtung der Unterpunkte dieser nicht-funktionale Anforderung für diese Arbeit zeigt die folgende Tabelle.

Tabelle 6: Spezifizierung der Benutzbarkeit

<i>Systemqualität</i>	<i>Sehr wichtig</i>	<i>Wichtig</i>	<i>Normal</i>	<i>Irrelevant</i>
Benutzbarkeit				
Verständlichkeit		x		
Erlernbarkeit	x			
Bedienbarkeit	x			
Attraktivität		x		

3.2.3 Weiterentwickelbarkeit

Die Weiterentwickelbarkeit bezeichnet „die Fähigkeit eines Softwareprodukts sich langfristig an geänderte Anforderungen und Techniken, die einen Einfluss auf die Architektur und/oder die funktionalen Erweiterungen haben, anzupassen, ohne die architektonische Integrität zu verletzen.“ (Balzert, 2011 S. 119). Die relevantesten Unterpunkte dieser Qualitätsanforderung für die zu implementierende Anwendung, werden die Änderbarkeit und die Erweiterbarkeit sein. Um diese Anforderungen gewährleisten zu können, soll die Anwendung mit Komponenten-Modellen arbeiten. Dadurch kann das Softwareprodukt Erweiterungen der Software um zusätzliche Funktionen gewährleisten. Neben den bisher genannten funktionalen Anforderungen sollen auf einfachem Weg neue Funktionen hinzugefügt werden können. Die Gewichtung der Unterpunkte dieser nicht-funktionale Anforderung für diese Arbeit zeigt die folgende Tabelle.

Tabelle 7: Spezifizierung der Weiterentwickelbarkeit

<i>Systemqualität</i>	<i>Sehr wichtig</i>	<i>Wichtig</i>	<i>Normal</i>	<i>Irrelevant</i>
Effizienz				
Analysierbarkeit		x		
Änderbarkeit	x			
Testbarkeit			x	
Portabilität			x	
Erweiterbarkeit		x		
Integrität der Architektur		x		

3.3 Zusammenfassung der Anforderungsanalyse

Die zu entwickelnde Anwendung geht über einen Fragebogen-Editor hinaus. Durch die Gruppierung von Fragebögen in Soundwalks und ein angepasster Template-Mechanismus, wo die bestehende Soundwalks und Fragebögen kopiert werden sowie die Unterstützung von Drag & Drop und Geolokalisierung ist eine Lösung notwendig, die alle diese Anforderungen erfüllt.

Das Dashboard ist der Einstiegspunkt in die Anwendung und gibt einen Überblick über die Soundwalks und Fragebögen. Im Gegensatz zu den vorgestellten Lösungen in Kapitel 2 sollen die Fragebögen in dieser Anwendung lokalisiert werden können und die Gruppierung mehrerer Fragebögen in einem Soundwalk unterstützt werden. Ebenso sollen Tonaufnahmen zu jedem Fragebogen hinzugefügt werden können. Des Weiteren soll der Administrator-Bereich durch ein Passwort gesichert werden, um unerlaubte Zugriffe auf den Bereich zu sperren.

Eine wichtige Anforderung an diese Anwendung ist ein hoher Grad der Nutzerfreundlichkeit, um die Anwendung für den Nutzer so einfach wie möglich zu gestalten. Für die Implementie-

ung bedeutet diese Vorgabe, dass die Navigation des Nutzers, durch die Anwendung möglichst reibungslos erfolgen soll. Außerdem soll der Nutzer überall, wo es sinnvoll ist, Drag & Drop nutzen können. Die zu entwickelnde Anwendung muss zudem den Aspekt des mobilen Einsatzes erfüllen. Die Bedienflächen sind klein und die Darstellung der Inhalte sollen dem Benutzen Gerät angepasst werden. Außerdem soll die Internet-Verbindung der Benutzer die Anzeige von Medien, wie Bilder beeinflussen, um diese in einer passenden Qualität darzustellen. Dazu sollen die Bedienelemente eine Touch-Kompatibilität besitzen. Letztendlich soll die Bedienbarkeit durch eine einfache Navigation unterstützt werden.

Auf Basis dieser Anforderungen wird im nächsten Kapitel ein Entwurf für die zu entwickelnde Software dargestellt.

4 Entwurf

In diesem Kapitel wird der Entwurf der Software vorgestellt. In Teilkapitel 4.1 wird in den Aufbau und die Verteilung der Komponenten der Systemarchitektur eingeführt. Das Teilkapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** beschreibt, wie die Anforderungen aus Kapitel 3 in dieser Arbeit umgesetzt wurden. Im Folgekapitel 4.3 wird die Gestaltung der verschiedenen Ansichten, sowie der allgemeine Arbeitsablauf der Software vorgestellt. Im Teilkapitel 4.4 wird die Persistenz der Software angesprochen und wie die Datenbankobjekte strukturiert sind. In Teilkapitel 4.5 wird ein Klassenkonzept vorgestellt. Anschließend wird in Teilkapitel 4.6 eine Zusammenfassung des Entwurfs formuliert, wo die Konzepte aus der zuvor definierten Anforderungsanalyse, auf ihre Akzeptanz geprüft werden.

4.1 Systemarchitektur

Die zu entwickelnde Anwendung kann nur von einem Administrator gleichzeitig bedient werden. Dies entspricht der Single-User-System-Umsetzung. Der Administrator greift auf die Anwendung und Daten über das Internet zu. Um die Möglichkeit zu haben das System in Zukunft zu erweitern, wurde für die Systemarchitektur ein verteiltes System ausgewählt. Hierdurch wird die Anwendung in Subsysteme, wo eine geringe Kopplung besteht, strukturiert. Es entsteht dabei eine Drei-Schichten-Architektur. Sie besteht aus einer Darstellungsschicht, wo der Endbenutzer mit der Anwendung interagiert, einer Anwendungsschicht, wo die Informationen verarbeitet werden, und einer Datenschicht, wo die verarbeiteten Informationen gespeichert und verwaltet werden. Der Vorteil dieser Architektur ist, dass mehrere Prozesse parallel ausgeführt werden können. Dies ermöglicht eine bessere Skalierung, da durch das Hinzufügen weiterer Systeme in die Architektur die Rechenleistung des Systems ohne Weiteres gesteigert werden kann. (Starke, et al., 2009)

Wie in den Anforderungen in Kapitel 3 geschildert, soll das System eine Plattformunabhängigkeit ermöglichen. Native Lösungen sind zwar für ihr jeweiliges System optimiert, jedoch erhöhen sie den Programmieraufwand, wodurch die Portabilität beeinträchtigt wird. Aus diesem Grund wird die Anwendung als Web-Client realisiert. Um ein Web-Client zu realisieren, gibt

es mehrere Frameworks, welche für diese Arbeit von Nutzen sein könnten. Es wird ein Framework ausgewählt, welches auf die vorherigen Anforderungen eingeht. „*React*“ ermöglicht dies. React ist ein JavaScript-Framework zur Erstellung von webbasierten Benutzeroberflächen und sogenannten „*Single-Page-Apps*“ oder SPAs. SPAs sind Web-Anwendungen, die aus verschiedenen Ansichten bestehen. Diese Ansichten befinden sich alle innerhalb einer einzelnen Seite. Dies ist durch die Status-Verwaltung von React möglich. Der Status kann als aktueller Zustand der Anwendung beschrieben werden. Dazu benutzt React Komponenten, die die Ansichten darstellen. Diese Komponenten oder auch Code-Blöcke sind wiederverwendbar und tragen dazu bei, die Gestaltung von Benutzeroberflächen wesentlich schneller zu ermöglichen. Die SurveyJS-Library ist zum Beispiel auch eine React-Komponente und lässt sich einfach in React-Anwendungen integrieren. In React gibt es auch andere Komponenten bzw. Bibliotheken, die die Drag & Drop Funktionalität anbieten. (Zammeti, 2020)

Im Web-Server wird ein Server für die Datenabstraktion benötigt. Die Datenabstraktion ermöglicht das Abstrahieren oder Verbergen von Daten in der Datenbank. In dieser Arbeit wird ein *NodeJS* Server betrieben, weil dieser auch JavaScript benutzt und eine hohe Kompatibilität mit React hat. Dies vereinfacht die Komplexität der gesamten Anwendung, indem überall die gleiche Programmiersprache benutzt wird. NodeJS ist eine plattformübergreifende Laufzeitumgebung für die Erstellung von Serverseiten und Netzwerkanwendungen. Sie verwendet eine ereignisgesteuerte, nicht blockierende I/O-Architektur, die für Echtzeitanwendungen geeignet ist. (Zammeti, 2020) Dieser Server kommuniziert mit der „*MongoDB*“ Datenbank. MongoDB ist eine Datenbank, die auf einem nicht-relationalen Dokumentmodell basiert (NoSQL). Sie benutzt das JSON-Dokumentenmodell und ermöglicht dadurch eine höhere Flexibilität und Skalierbarkeit der Datenspeicherung und Datennutzung. MongoDB lässt sich auch leicht in NodeJS integrieren. Die gesamte Kommunikation zwischen den verschiedenen Komponenten erfolgt über das Internet durch eine REST API. (Mardan, 2018)

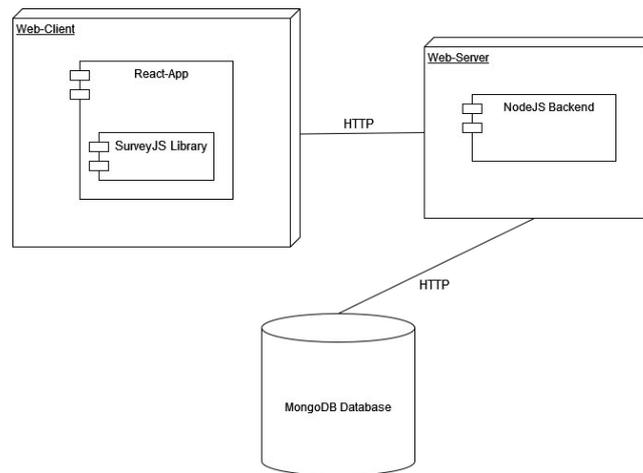


Abbildung 8: Verteilungsdiagramm der Kommunikation zwischen Komponenten, Quelle: Eigenes Werk

4.1.1 Model-View-Controller

„*Model-View-Controller*“ (MVC) ist ein Design-Pattern, welches auch als Architektur-Pattern verwendet wird. Ziel der MVC ist die Aufteilung einer Anwendung in drei Subsysteme. „*View*“ repräsentiert die Benutzeroberfläche, „*Model*“ die Anwendungslogik und „*Controller*“ die Steuerung der Anwendung. Dabei können mehrere Controller den Zustand eines Objektes bzw. ein Model ändern. Die Views zeigen den Zustand des Models auf der Benutzeroberfläche. Bei Zustandsänderungen passen sie ihre Darstellungen an. Dadurch arbeiten die View und Controller Komponenten eng zusammen und lassen sich getrennt von dem Model entwickeln und testen. (Balzert, 2011)

Das MVC-Pattern kann auch durch einen „*Observer*“ (Beobachter) erweitert werden. Dies wäre notwendig, wenn mehrere Administratoren auf die gleiche Datensätze Zugriff hätten. Erst dann musste das Model die anderen Komponenten benachrichtigen.

4.1.2 Model-View-Controller im Drag & Drop Survey Creator

Wie im Abschnitt 4.1.1 erwähnt, eignet sich das MVC-Pattern am besten für Anwendungen, bei denen keinen Änderungen an der Persistenz von externen Anwendungen vorgenommen werden. Die Architektur der Drag & Drop Survey Creator Anwendung basiert auf dieser Variante von MVC.

Entsprechend der Verteilung der MVC-Verantwortungen lassen sich die Komponenten der Drag & Drop Survey Creator Anwendung mit Hilfe von React zerlegen. Für diese Arbeit wird React die Rolle der View und des Controllers übernehmen.

Model: Das Model verwaltet alle Informationen der Anwendung, wie z. B. Fragebogenobjekte. Dies ist das einzige Subsystem, das nicht vom React übernommen wird. Das Model liegt im Server und kümmert sich um die Umwandlung von JSON-Objekten aus dem Client in das passende Format und gibt die umgewandelten Objekte an die Datenbank weiter. Das Model sieht beispielweise so aus:

Listing 1: vereinfachtes Soundwalk-Model im Server, Quelle: Eigenes Werk

```
const soundwalkSchema = Schema({
  title: String,
  description: String,
  surveys: JSON-Array,
  Status: ENUM
})
```

View: Der Controller kümmert sich um das Abrufen der Fragebögen und Soundwalks, sowie das Rendering ihre entsprechende View-Komponenten. Dadurch werden Benutzereingaben interpretiert und an das Model weitergeleitet. Die Komponenten der Controller sehen zum Beispiel wie folgt aus:

Listing 2: vereinfachte Dashboard View-Komponente, Quelle: Eigenes Werk

```
// Dashboard.js
import React from "react";

const Dashboard = comments => (
  <ul>
    {soundwalks.map(({ soundwalk, key }) =>
      <li>{soundwalk}-{key}</li>
    )}
  </ul>
)
```

Controller: Der Controller kümmert sich um das Abrufen der Fragebögen und Soundwalks, sowie das Rendering ihre entsprechende View-Komponenten. Dadurch werden Benutzereingaben interpretiert und an das Model weitergeleitet. Die Komponenten der Controller sehen zum Beispiel wie folgt aus:

Listing 3: vereinfachte Dashboard Controller-Komponente, Quelle: Eigenes Werk

```
// DashboardController.js
import React from "react";
import Dashboard from "../Dashboard";

class DashboardContainer extends React.Component {
  constructor() {
    super();
    this.state = { soundwalks: [] }
  }

  componentDidMount() {
    fetch("/my-soundwalks.json")
      .then(res => res.json())
      .then(soundwalks => this.setState({ soundwalks }));
  }

  render() {
    return <Soundwalk comments={this.state.soundwalk} />;
  }
}
```

Durch diese Verteilung der Komponenten entsteht folgendes Design:

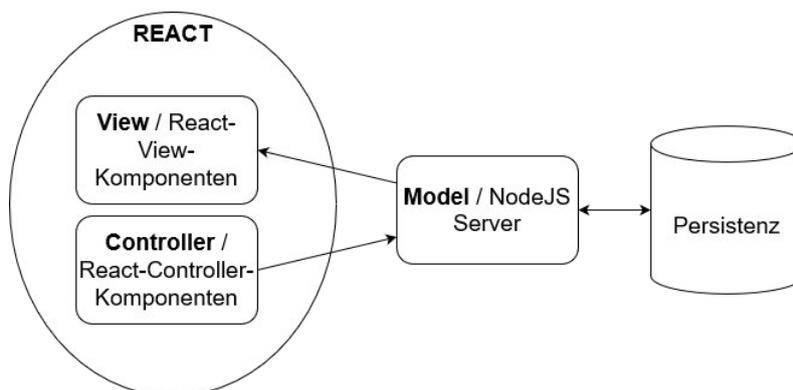


Abbildung 9: Design nach dem MVC-Prinzip, Quelle: Eigenes Werk

Die View-Komponente erhält die View-Daten aus dem NodeJS Server zur Darstellung der Fragebögen und Soundwalks. Die Controller-Komponente kommuniziert alle Interaktion des Benutzers an die Logik bzw. den NodeJS Server.

4.1.3 Web-Architektur

Das Web-Modell beschreibt eine Möglichkeit, Aufgaben und Dienstleistungen innerhalb eines verteilten Systems zu verteilen. Diese Aufgaben und Dienstleistungen werden von Systemkomponenten erledigt. Diese werden wiederum in Clients und Server unterteilt. Der Client kann jederzeit eine Aufgabe vom Server anfordern. Der Server beantwortet diese Anfrage und stellt das gewünschte Paket zur Verfügung. Diese Architektur ähnelt der Client-Server-Architektur. Dabei werden logische Schichten bzw. Subsysteme auf Clients und auf einen oder mehrere Server verteilt, wobei Web-Techniken eingesetzt werden. (Balzert, 2011)

In dieser Architektur befinden sich auf den Clients in der Regel Teile der GUI-Schicht. Diese Teile werden in einem Web-Browser ausgeführt. Der Web-Browser funktioniert dabei als Betriebssystem oder Laufzeitumgebung des Web-Clients. Serverseitig nimmt ein Web-Server die Anfragen von Web-Clients entgegen und beantwortet sie. Dabei wird das HTTP-Protokoll zur Kommunikation eingesetzt. Dieser Web-Server enthält sowohl Teile der Applikations-Schicht und der GUI-Schicht. Die Datenbank bzw. die Persistenz-Schicht kann sich auf einem anderen Server befinden. (Balzert, 2011)

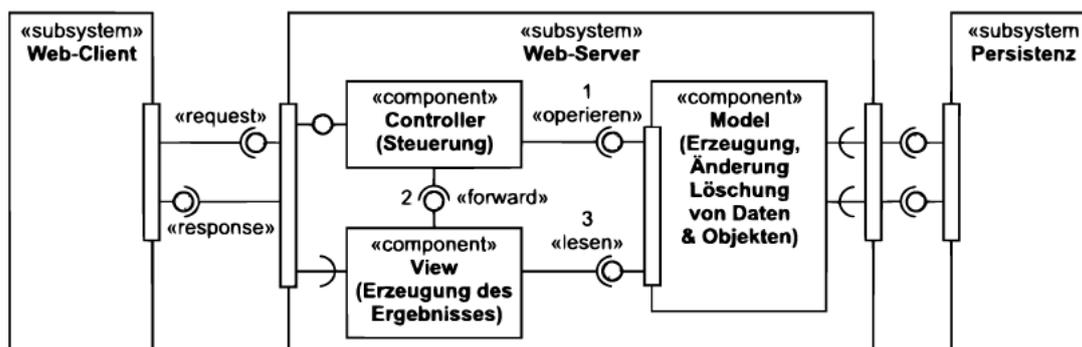


Abbildung 10: Web-Architektur nach dem MVC-Muster nach Balzert, Quelle: (Balzert, 2011 S. 197)

Jede Benutzer-Anfrage einer Web-Seite durch das HTTP-Protokoll baut eine TCP-Verbindung mit dem Web-Browser auf. Der Server bearbeitet diese Anfrage. Nach der Rücksendung der Antwort wird die TCP-Verbindung aufgelöst. Dadurch gibt es zwischen Web-Client und Web-Browser keine permanente Verbindung. Der Zugriff auf Webanwendungen ist einer beliebigen Anzahl von Benutzern möglich. Die Entwickler haben dabei keinen Einfluss auf die Laufzeitumgebung des Web-Clients. (Balzert, 2011)

In dieser Architektur erfolgten die gesamte Verarbeitung und Vorbereitung der Ansicht auf dem Server. Dadurch wird der Server zu einem Engpass. Um dieses Problem zu lösen, kann das Rendering von den Daten durch den Web-Client übernommen werden. Basierend auf dieser Vorstellung wurde die RIA-Web-Architektur entwickelt. (Balzert, 2011)

RIA-Web-Architektur

RIA steht für „Rich Internet Application“. Diese Architektur wurde entworfen, um die Web-Server-Last zu reduzieren. Dabei werden Funktionen, wie der Einsatz von HTML5, CSS und JavaScript, an den Web-Client übergeben. RIAs sind Client-Server-Anwendungen, welche über das Internet verbunden sind. Sie sind auf allen gängigen Betriebssystemen und Browser-Plattformen ohne sichtbare Unterschiede verfügbar. Dadurch ist die Bereitstellung der Anwendung leicht zu realisieren und die Portabilität wird erhöht. Zusätzlich ermöglicht eine RIA-Web-Architektur die Nutzung von Desktop-Funktionalitäten wie Drag & Drop. (Balzert, 2011)

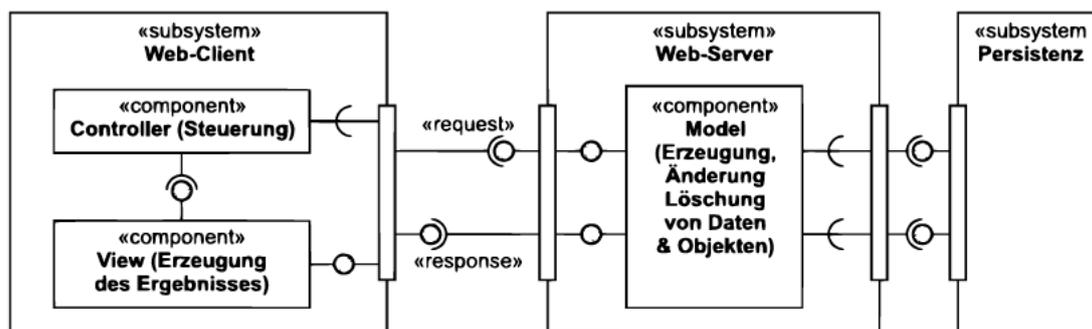


Abbildung 11: Beispiel für eine RIA-Web-Architektur nach Balzert, Quelle: (Balzert, 2011 S. 199)

4.1.4 Web-Architektur im Drag & Drop Survey Creator

Aus Benutzersicht ist der Web-Client, der vom Benutzer verwendete Webbrowser, der über eine URL auf die Anwendung zugreift.

Aus Implementierungssicht, wurde ein Großteil der Anwendung clientseitig mit React implementiert. Dadurch übernimmt der Server die Hauptaufgabe der Kommunikation und Datenabstraktion zwischen Datenbank und Nutzern.

Im Web-Client befinden sich die View und die Controller Komponenten, welche beide mit React realisiert wurden. Die SurveyJS-Library liest JSON-Objekte. Um sie zu benutzen, müssen die Daten im passenden Format an die Bibliothek übergeben werden. Dafür werden die JSON formatierten Strings in JSON geparkt. Diese werden dann an den Server weitergegeben, wo sie zur Speicherung in der Datenbank vorbereitet und weitergeleitet werden. Der Server kann wiederum die Daten aus der Datenbank holen, um sie an dem Client weiterzugeben.

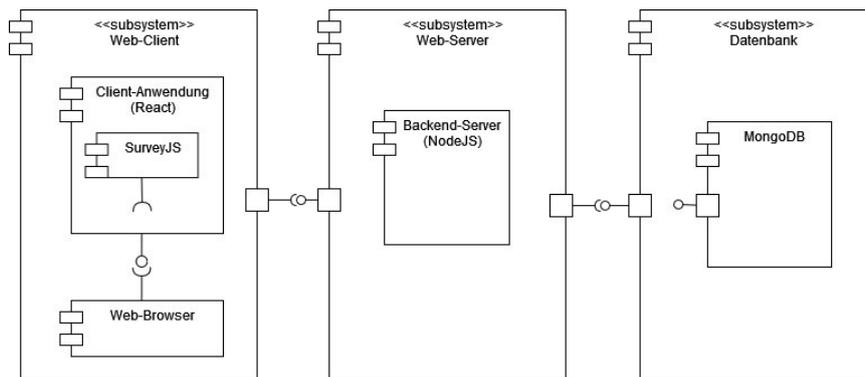


Abbildung 12: Design der Komponenten nach der Web-Architektur MVC-Pattern, Eigenes Werk

Die Datenbank unterstützt die vier Operationen auf dem persistenten Speicher. Datenbankobjekte können angelegt, gelesen, aktualisiert und gelöscht werden. Dabei werden die entsprechenden HTTP-Anfragen vom Server gesendet.

4.1.5 Zusammenfassung der Systemarchitektur

In diesem Kapitel wurde die Anwendung anhand einer Systemarchitektur vorgestellt. Die Anwendung benutzt das Web-Model, wodurch per HTTP-Protokoll über REST APIs kommuniziert wird. Das System ist als eine Drei-Schichten-Architektur aufgebaut und wird als Single-User-System implementiert, um Konflikte zwischen mehreren Nutzern zu vermeiden. Um mehrere Nutzer zu unterstützen, müsste eine passende Locking-Strategie ausgewählt werden. Dabei kümmert sich der Web-Server um die Umwandlung und Weiterleitung der JSON-Objekte zwischen der Datenbank und der Anwendung.

Für die Clients wurde als Struktur eine RIA Web-Architektur gewählt. Dabei wird die Anwendung zu einer browserbasierten Client-Anwendung. Der Web-Client enthält sowohl eine Benutzerschnittstelle als auch einen Teil der Logik. Der andere Teil der Logik liegt auf dem Server. Um das ganze System zu strukturieren, wird das Design/Architektur Pattern MVC benutzt. Mit der Nutzung von React wird die MVC-Architektur erreicht. Die Benutzung dieser Patterns können die nichtfunktionalen Anforderungen hinreichender erfüllen:

Tabelle 8: Erfüllte nicht-funktionale Anforderungen anhand der Architektur

<i>NF-Anforderung</i>	<i>Pattern</i>
Zuverlässigkeit	Aufgrund des verteilten Systems wirkt sich die Zuverlässigkeit einer Schicht nur begrenzt auf die anderen Schichten aus.
Benutzbarkeit	Durch das MVC-Pattern wird eine leichte Erweiterung um zusätzliche Ansichten und Eingaben über verschiedene Geräte unterstützt.
Weiterentwickelbarkeit	Durch eine Web-Architektur müssen keine zusätzlichen Programme installiert werden und wachsende Leistungsanforderungen können leicht angepasst werden.

4.2 Umsetzung der Anforderungen

4.2.1 Drag & Drop

Drag & Drop steht für das Ziehen und Ablegen der Benutzeroberflächen-Elemente. Diese Elemente können angeklickt werden und bei gedrückter Maustaste verschoben werden. Die häufigste Verwendung dieser Funktion ist das Verschieben, Kopieren, Öffnen und Hochladen von Dateien. Die Grundidee von Drag & Drop basiert auf der Visualisierung von Kopieren und Einfügen verschiedener Datenelemente, wobei statt der Zwischenablage der Systempuffer verwendet wird. Der Benutzer bekommt eine sofortige Rückmeldung, indem das Element markiert, die Visualisierung des Ziehens und Verlassen des Ursprungsortes angezeigt und das Anzeigen der Drop-Möglichkeit, sowie eine Rückmeldung beim Fallenlassen angezeigt wird. (Gauchat, 2018)

Das Drag & Drop wird in Web-Anwendungen durch die Benutzung von clientseitigen Skriptsprachen ermöglicht. Um diese Funktionalität in diese Arbeit zu integrieren, bietet React die Bibliothek „*react-beautiful-dnd*“ an. Mit dieser Bibliothek können Bereiche der Ansicht in einen Drag & Drop Kontext verpackt werden. Anzeige-Elemente können dabei ablegbar und ziehbar sein. In dem Dashboard sind Soundwalks abwerfbar und Fragebögen ziehbar. Wenn

ein Fragebogen auf einem Soundwalk gezogen wird, wird eine Kopie vom Fragebogen dem Soundwalk hinzugefügt.

Die Umsetzung von Drag & Drop beim Fragebogen-Editor wird analog wie beim Soundwalks und den Fragebögen realisiert. Hierbei wird die Liste der Fragetypen ziehbar sein. Der Bereich, wo die Fragen liegen, wird ziehbar und abwerfbar sein. Hiermit wird erreicht, dass die Fragetypen im Fragebogen gezogen werden können und die Reihenfolge der Fragen geändert werden kann.

Für die Entwicklung wurde React ausgewählt. Dadurch wird diese Funktionalität automatisch auf mobile Geräte übertragen, wenn sie implementiert wird. Die Funktionalität bleibt dabei gleich. Der Benutzer selektiert das Element und durch das Halten und Ziehen wird das Element bewegt und abgelegt. Durch die Verwendung von Drag & Drop wird die nicht-funktionale Anforderung Bedienbarkeit unterstützt.

4.2.2 Template-Mechanismus

Wie im Kapitel 2.1.1 beschrieben, handelt es sich bei einem Template um eine Vorlage oder Schablone für Fragebögen. In diese Arbeit wird diese Funktionalität umgesetzt, indem ein Fragebogen per Knopfdruck kopiert werden kann. Dadurch wird eine tiefe Kopie von den ausgewählten Fragebögen erzeugt und in der Datenbank abgelegt, so dass sich das Original und seine Kopie keine Attribute teilen. Der Header vom Fragebogen wird übernommen und der Titel auf „Kopie von Fragebogen #“ geändert. # repräsentiert den Titel des Originals. Da die Datenobjekte dieser Arbeit JSON-Objekte sind, können sie direkt mit den JSON-Funktionen `JSON.stringify()`, welche ein JSON-Objekt in einem JSON formatierter String konvertiert, und `JSON.parse()`, welcher ein JSON formatierter String in einem JSON-Objekt umwandelt, benutzt werden, um eine Ausgabe vom Original zu erzeugen. Um den Titel zu ändern kann `JSON.parse(JSON-String, replace())` mit einer Ersetzen-Funktion angewendet werden, wo die IDs und der Titel ersetzt werden können. Auf diese Kopien kann der Nutzer anschließend im Dashboard zugreifen. (contributors, 2020)

4.2.3 Gruppierung von Fragebögen

Bei keinem der untersuchten Anbieter in Kapitel 2 gibt es eine Möglichkeit Fragebögen in einer Sammlung zu gruppieren. Diese Sammlung wird in dieser Arbeit als Soundwalk benannt. Beim Soundwalk erstellen, erzeugt der Nutzer ein JSON-Objekt, welches ein Array für seine Fragebögen bereitstellt. Dieses Array kann anschließend vom Benutzer mit Fragebögen gefüllt werden, wodurch Kopien der ausgewählten Fragebögen in dem Fragebögen-Array gespeichert werden. Die Soundwalks können vom Benutzer ein oder ausgeschaltet werden, indem ein Status per Knopfdruck mit entweder „ACTIVE“ oder „INACTIVE“ gesetzt wird. Die aktiven Soundwalks können dann in der Ansicht vom Teilnehmer angezeigt werden, welcher anschließend daran teilnehmen kann.

4.2.4 Tonaufnahmen hinzufügen

Um Tonaufnahmen hinzuzufügen, wird der Frage-Typ „Datei“ benutzt. Dieser Frage-Typ erlaubt das Hochladen von verschiedenen Dateien u. a. Tonaufnahmen. Dabei wird eine POST-Anfrage mit dem Inhalt der Datei als JSON an dem Server gesendet. Diese Anfrage enthält alle Informationen über die Datei, sodass diese wiederzufinden ist. Sie wird in der Datenbank in Verknüpfung zum Fragebogen abgelegt.

4.2.5 Lokalisierung

Mittels die „React Leaflet“ Bibliothek kann ein Standort mit „OpenStreetsMap“ für die Fragebögen ausgewählt werden. Dabei kann der eigene Standort benutzt oder ein beliebiger Standort aus einer Karte ausgewählt werden. Wenn der Nutzer ein Punkt ausgewählt hat, werden die Koordinaten als JSON-Objekt extrahiert und in dem Header vom Fragebogen als JSON-String hinzugefügt. Der Nutzer kann dabei einen Radius eingeben. Dieser Radius kann bestimmen, ob ein Teilnehmer an einem Fragebogen teilnehmen darf. Wie die Anwendung prüft, ob sich der Teilnehmer innerhalb des gegebenen Radius befindet, ist nicht Teil dieser Arbeit. (Cam, 2022)

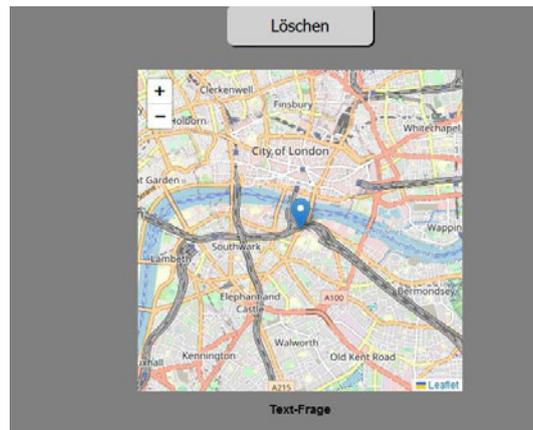


Abbildung 13: Implementierung der Lokalisierung, Eigenes Werk

4.2.6 Trennung der Administrator und Teilnehmer-Bereich

Der Administrator-Bereich wird mit einem Passwort geschützt. Durch die Verwendung von einer .htaccess- und einer .htpasswd-Datei wird eine Zugriffskontrolle auf dem Server für den Verzeichnis-Zweig ab der Position der Datei definiert. Dadurch können die Ansichten nur nach Eingabe eines Passworts aufgerufen werden.

4.3 Arbeitsablauf und Gestaltung der Benutzeroberfläche

Mit der Benutzung von React und der CSS-Funktionalitäten wird die Gestaltung der Anwendung auf den aktuellen Stand der Technik gebracht. Im Folgenden wird der Arbeitsablauf der Anwendung vorgestellt. Anschließend werden die einzelnen Ansichten und ihre Gestaltung präsentiert.

4.3.1 Arbeitsablauf

Der Web-Client erhält die gesamte Auswahl der Fragebögen und Soundwalks vom Web-Server. Bei dem Entwurf der Anwendung ist folgender Arbeitsablauf konzipiert worden.

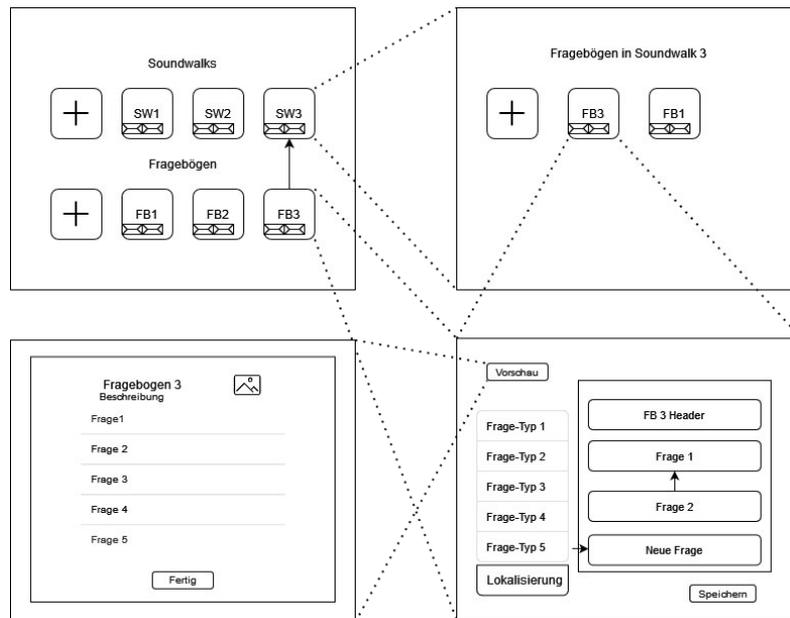


Abbildung 14: Skizze zum Arbeitsablauf der Anwendung, Quelle: Eigenes Werk

Die Kästchen in den Soundwalks und Fragebögen repräsentieren die Knöpfe für Löschen und Kopieren. Die gestrichelten Linien bezeichnen die Ansicht, die geöffnet wird, wenn das entsprechende UI Element selektiert wird. Die Pfeile bezeichnen die Drag & Drop Bewegungsmöglichkeiten in der Anwendung.

Nach dem Öffnen der Webanwendung auf dem Gerät, muss sich der Administrator zunächst mit einem Passwort einloggen. Ab diesem Moment wird die erste Ansicht der Anwendung, wo zwei Listen der aktuellen Soundwalks (SW) und Fragebögen (FB) stehen, geöffnet. Zudem kann der Nutzer einen Fragebogen per Drag & Drop in einem Soundwalk einfügen. Nach dem Klick auf einen Soundwalk wird eine neue Ansicht geöffnet, wo alle zugehörigen Fragebögen angezeigt werden. Durch das Klicken auf einen Fragebogen, entweder in der ersten Ansicht oder in der Soundwalk-Ansicht, wird ein Fragebogen-Editor eröffnet, wo der ausgewählte Fragebogen per Drag & Drop bearbeitet werden kann. Durch das Klicken vom Vorschau-Knopf wechselt die Ansicht auf die Anzeige des fertigen Fragebogens. Wenn einer der Ansichten größer als das Display ist, zum Beispiel, wenn ein Fragebogen mehrere Fragen beinhaltet, kann der Nutzer scrollen, um die ausgeblendeten Bereiche zu erreichen und sich durch die Ansicht navigieren.

Auf der Skizze ist die „Navbar“ nicht zu sehen. Eine Navbar ist ein Seitenelement, welches die Navigation in der Anwendung erleichtert. Hierdurch wird die Navigation hierarchisch organisiert. Die Ansichten sind so in einer Baumstruktur organisiert und der Nutzer kann Ansichten erreichen, indem er auch andere überspringt. So ist es möglich direkt von der Landing Page den Fragebogen-Editor für einen beliebigen Fragebogen zu öffnen, ohne dass die Dashboard-Ansicht erreicht wird. Wenn der Benutzer eine Ansicht erreicht, die nicht vorhanden ist, wird die Ansicht "Error 404" angezeigt. Hier kann der Benutzer zurück zur Startseite gehen oder die Navigationsleiste verwenden, um sein Ziel zu erreichen.

Nachdem der Arbeitsablauf entworfen ist, geht es an die grafische Umsetzung der einzelnen Oberflächenelemente. Die Navbar wird überall auf der Anwendung gezeigt. Sie ist die zentrale Navigationsmöglichkeit der Anwendung.

Die Benutzeroberfläche wird sich an die kleinen Bildschirme von Smartphones und die der größeren Tablets anpassen können. Daher wird die Anzahl der grafischen Elemente pro Ansicht gering gehalten. Bei der Anzeige der Oberflächen wird das Smartphone sowohl im Hoch-, als auch im Querformat betrachtet, wie auf den folgenden Seiten zusammengefasst wird. Um diese reaktionsfähige Eigenschaft zu erreichen, wurden die „Media Queries“ von CSS benutzt. Diese Abfragen prüfen welche Breite und Höhe, sowie welche Ausrichtung die benutzten Geräte bzw. Anzeigefenster haben. Dadurch können angepasste Style angewendet werden.

4.3.2 Landing Page

Die „Landing Page“ ist die erste Ansicht der Anwendung, welche der Nutzer zu sehen bekommt. Darauf abgebildet ist das HAW-Hamburg-Logo und ein Begrüßungstext, welcher der App einen guten ersten Eindruck verleihen und dem Nutzer erste Informationen über die App vermitteln soll. Im Hintergrund wird der Windpark Curslack angezeigt. Der Benutzer kann in der Navbar zwischen den verschiedenen Ansichten der Anwendung wechseln.

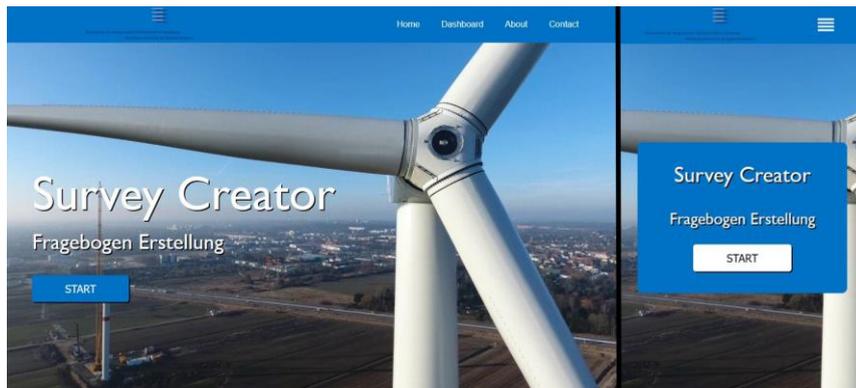


Abbildung 15: Landing Page der Anwendung auf dem Desktop-Browser (links) und Mobile-Browser (rechts), Quelle: Eigenes Werk

In der Mobile-Variante ändert sich die Navbar, um das Menu-Symbol anzuzeigen. Die Home, Dashboard, About und Contact Elemente wurden ausgeblendet, um sich an die neue Auflösung anzupassen. Sie sind durch das Klicken des Menu-Symbols trotzdem erreichbar. Dieses Verhalten bleibt bei jedem Ansicht-Wechsel erhalten.

4.3.3 Dashboard

Auf dem Dashboard stehen alle Soundwalks und Fragebögen in Form von zwei Listen zur Verfügung. Cards eignen sich hervorragend, um die Benutzeroberfläche dieser Web-Anwendung zu organisieren und zu verbessern.

Diese Seite bietet zwei unterschiedliche Listen. Oben befinden sich die angelegten Soundwalks und unten die Fragebögen. Beide werden als Cards dargestellt und mit einem Titel und einer Beschreibung ergänzt. Aus dieser Ansicht kann der Benutzer Fragebögen und Soundwalks erstellen, duplizieren oder löschen. Sie funktionieren bei Soundwalks und Fragebögen unterschiedlich. Das Bearbeiten wird durch Klicken auf die Cards aufgerufen. Das Löschen und Kopieren haben einen entsprechenden Knopf im Card-Element. Beim Kopieren wird ein neues Element mit Inhalt in der entsprechenden Liste eingefügt. Beim Löschen von Soundwalks wird der Soundwalk-Datenbankeintrag gelöscht. Beim Löschen von Fragebögen wird der Fragebogen-Datenbankeintrag entfernt und der Fragebogen und seine Fragen gehen verloren.

Die Reihenfolge wurde so ausgewählt, dass sie den Nutzern das Auswählen des Soundwalks vereinfacht. Das erste Element der Listen ist ein Additionsknopf, welcher einen neuen Soundwalk bzw. Fragebogen anlegt und anschließend in der zugehörigen Liste anzeigt.

Die Fragebogen-Cards werden mit einer „Pin“ versehen, wenn diese vom Fragebogen-Ersteller mit einem Lokalisierungspunkt verknüpft wurden. Außerdem werden die Fragebogen-Cards zusätzlich zum Titel und der Beschreibung mit einem Bild ergänzt. Diese Informationen werden den Inhalten aus dem Header der Fragebögen entsprechen.

Die Drag & Drop Funktionalität wurde implementiert und kann im Dashboard benutzt werden, um Fragebögen in Soundwalks hinzuzufügen.

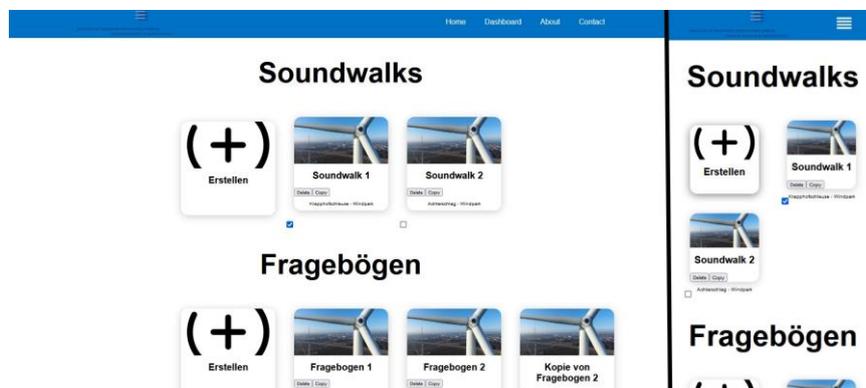


Abbildung 16: Dashboard Page der Anwendung auf dem Desktop-Browser (links) und Mobile-Browser (rechts), Quelle: Eigenes Werk

Um das Dashboard an mobile Geräte anzupassen, werden die Cards verkleinert und neu organisiert. Damit der Text nicht zu klein wird, wird die Beschreibung unter den Cards platziert.

4.3.4 Fragebogen-Editor

In der Fragebogen-Editor Ansicht, wird eine Liste der zur Verfügung stehende Fragebogen-Komponenten angezeigt. Diese Komponenten können per Drag & Drop in den Fragebogen-Bereich hineingezogen und dort bearbeitet werden. Dort kann auch per Drag & Drop ihre Reihenfolge geändert werden. Das letzte Element der Liste ist die Lokalisierungsfunktion.

Der Header von einem Fragebogen enthält einen Titel, eine Beschreibung, ein Logo bzw. ein Bild und optionale Geokoordinaten. Diese Informationen werden benötigt, um das Dashboard

mit den Fragebogen-Information zu befüllen. Der Benutzer kann diese Felder ändern, aber sie nicht aus dem Fragebogen entfernen. Anschließend folgen alle Fragen als Formulare, die bearbeitet werden können.

Der Vorschau Knopf bringt den Nutzer auf die Vorschau Ansicht, wo der fertige Fragebogen angezeigt wird und beantwortet werden kann.

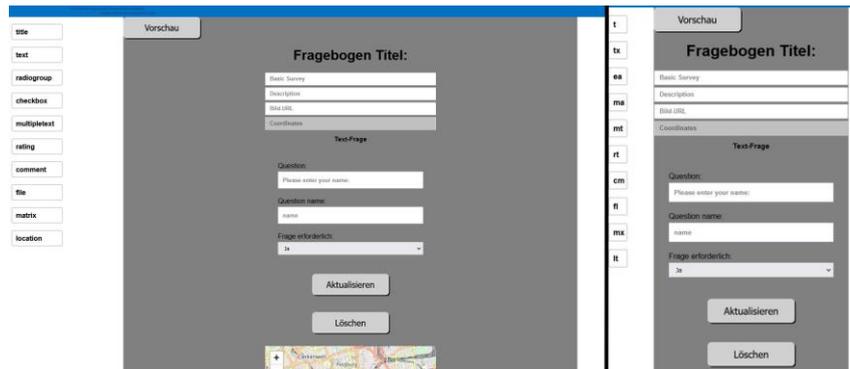


Abbildung 17: Fragebogen-Editor Page der Anwendung auf dem Desktop-Browser (links) und Mobile-Browser (rechts), Quelle: Eigenes Werk

Um der Fragebogen-Editor an mobile Geräte anzupassen, werden die Fragebogen-Komponenten durch Abkürzungen ersetzt. Dadurch verbessert sich die Benutzbarkeit.

Die SurveyJS-Library besitzt die in den Anforderungen formulierten Frage-Typen. Jede Frage in der SurveyJS-Bibliothek wird als ein JSON-Objekt definiert. Dadurch lassen sich die Frage-Typen mit der Nutzung dieser Bibliothek generieren. (SurveyJS, 2022)

4.3.5 Vorschau

In der Vorschau Ansicht wird der fertige Fragebogen angezeigt. Hier kann der Nutzer alle Felder bearbeiten und den Fragebogen ausfüllen. Dabei werden keine Daten gespeichert, da der Fragebogen für Testzwecke angezeigt wird. Das Rendern vom Fragebogen wird von der SurveyJS Library übernommen. Hierbei wird die Liste der Fragen im Fragebogen an die Vorschau weitergegeben. Die Vorschau zeigt anschließend die Fragen als einen fertigen Fragebogen an. Um die Fragen zu visualisieren, bietet SurveyJS mehrere Farbpaletten, die durch Parameter Eingaben geändert werden können. Um die Vorschau übersichtlich zu gestalten, wurden diese jedoch nicht angewendet.

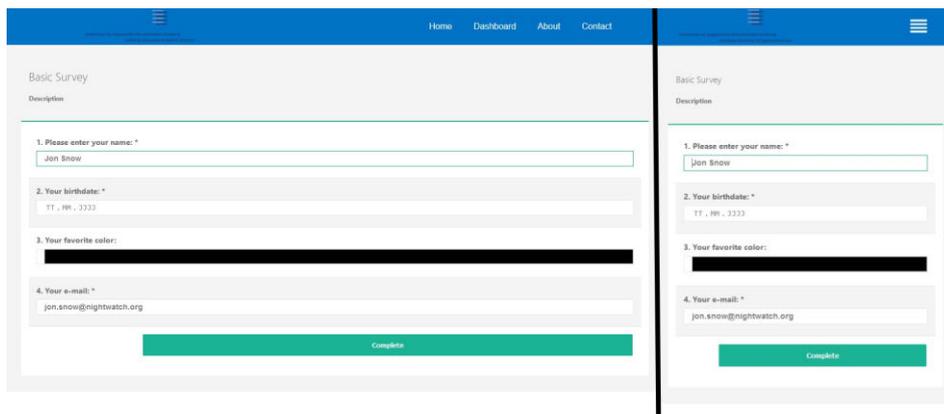


Abbildung 18: Vorschau vom Fragebogen auf dem Desktop-Browser (links) und Mobile-Browser (rechts), Quelle: Eigenes Werk

4.4 Persistenz

Diese Arbeit benutzt, wie im Teilkapitel 4.1 erwähnt, eine nicht-relationale Datenbank. Die Datenbank besteht aus JSON-Dokumenten, die Soundwalks, Fragebögen, sowie Fragen und Ergebnisse sein können.

Ein Soundwalk-Objekt enthält eine eindeutige Identifikationsnummer (ID). Außerdem verfügt ein Soundwalk über einen Titel, eine Beschreibung und eine Liste der enthaltenen Fragebögen. Dazu gehört ein aufzählbarer Status, der entweder ‚ACTIVE‘ oder ‚INACTIVE‘ gesetzt werden kann. Dieser Status bestimmt die Sichtbarkeit des Soundwalks für die Teilnehmer. Wenn ein Soundwalk gelöscht wird, werden die enthaltenen Fragebögen auch gelöscht. Sie stehen für andere Soundwalks in der Datenbank zur Verfügung.

Ein Fragebogen-Objekt enthält auch eine ID. Darüber hinaus enthält der Fragebogen einen Titel, einen Typ, um den Header zu generieren, eine Beschreibung, sowie ein Logo mit seinen Dimensionen und eine Liste der enthaltenen Fragen. Anschließend können optionale Geokoordinaten gesetzt werden. Diese Geokoordinaten bestimmen die Sichtbarkeit des Fragebogens für die Teilnehmer. Wenn ein Fragebogen gelöscht wird, werden die enthaltenen Fragen gelöscht. Mehrere Fragebogen-Objekte, aber nicht die gleichen, können in einem Soundwalk auftauchen.

Die Fragenobjekte enthalten verschiedene Parameter die abhängig vom jeweiligen Frage-Typ sind. Sie haben mindestens einen Namen, einen Titel und eine Beschreibung. Einen Frage-Typ kann kein bis beliebig viele-Male in einem Fragebogen auftauchen.

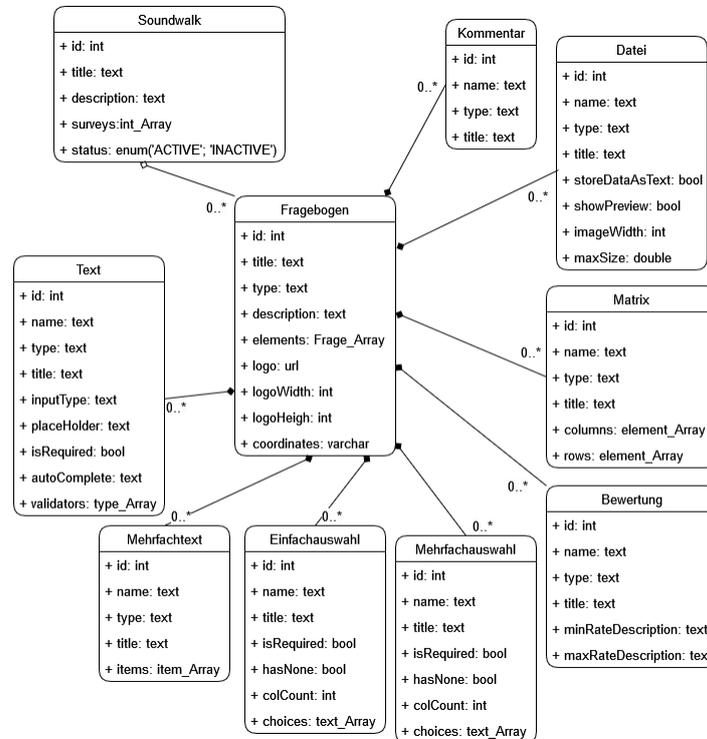


Abbildung 19: JSON-Objekte und deren Beziehungen, Quelle: Eigenes Werk

Bei dem Frage-Typ Datei existiert das Attribut „storeDataAsText“. Dieses Attribut wird verwendet um die Datei entweder als Text oder XML zu speichern. Dieser Frage-Typ besitzt mehrere interessante Attribute. Zum einen kann die maximale Größe der Datei eingestellt werden. Bei größeren Dateien wird das Hochladen von der SurveyJS-Library verweigert. Die Dateien können dann als mp3 Inhalt gespeichert werden und stehen anschließend im Server zur Verfügung.

Nach der Benutzer-Eingabe empfängt der Server die Anfragen. Der Server verarbeitet die Anfrage und sendet diese weiter an die Datenbank. Die Kommunikation erfolgt dabei über eine REST API. Dafür unterstützen der Server und die Datenbank die CRUD Operationen. Durch das Erstellen der Elemente, wie einem neuen Soundwalk oder einem neuen Fragebogen, wird eine POST HTTP-Anfrage gesendet. Um existierende Elemente, wie beispielsweise den Text

aus einer Frage zu bearbeiten, wird die PUT-Anfrage benutzt. Wenn der Nutzer nur Informationen bekommen soll, wie zum Beispiel beim Aufruf von allen Fragebögen im Dashboard, wird die GET-Anfrage benutzt. Um Elemente zu Löschen wird die DELETE-Anfrage verwendet.

4.5 Klassenkonzept

Die Anwendung ist nach dem MVC-Prinzip aufgebaut. Die Model-Klassen sind äquivalent zu den JSON-Objekten. So gibt es jeweils eine Klasse für Soundwalk, Fragebogen und Fragetypen. Alle Attribute haben entsprechende Getter und Setter. Sie werden für ein besseres Überblick nicht in den Klassen gezeigt. Es wird auch Beispielsweise nur ein Fragetyp gezeigt.

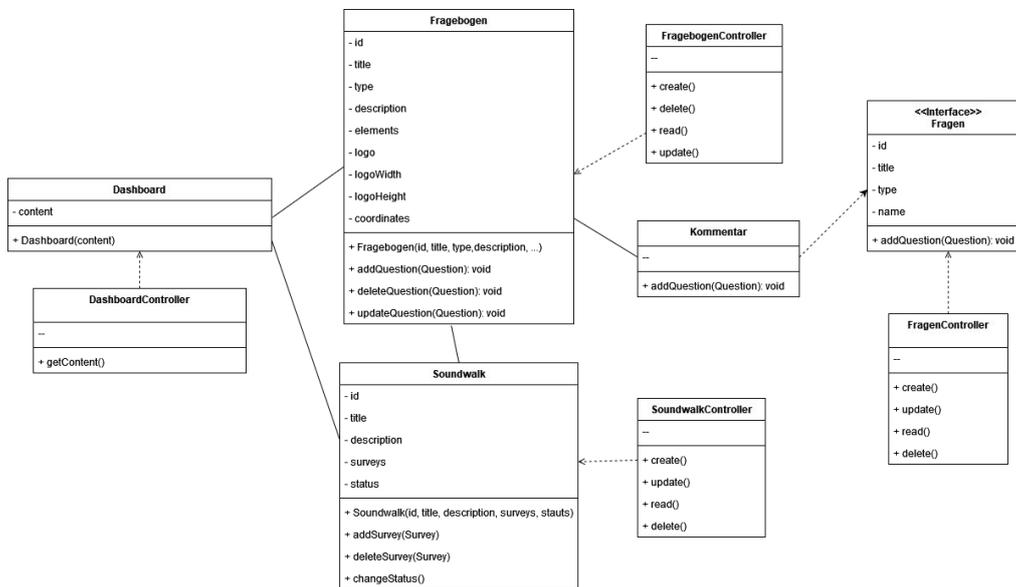


Abbildung 20: Klassenkonzept, Eigenes Werk

Die Controller-Klassen werden benutzt, um die Verbindung zu der Datenbank herzustellen und Zugriff auf jedes notwendige Objekt zu ermöglichen.

4.6 Zusammenfassung des Entwurfs

Durch diesen Entwurf ist eine Webanwendung entstanden, welche in Zukunft für den Anwender über das Internet erreichbar sein wird. Es wurde ein Konzept entworfen, wo eine Mehrschichten-Architektur benutzt wird. Um dies zu erreichen, wird die Web-Architektur mit dem MVC-Pattern verwendet.

Um die Drag & Drop Funktionalität zu erreichen, wurde eine REACT Bibliothek benutzt. Mit dieser Bibliothek lassen sich GUI-Elemente in einem Drag&Drop Kontext einpacken, um sie leicht mit dieser Funktionalität zu gestalten. Hierdurch können die gewünschten Funktionalitäten mit Drag & Drop implementiert werden.

Durch die Verwendung von REACT und die eingesetzte Design-Patterns konnten die nicht-funktionalen Anforderungen erfüllt werden.

Die Anwendung bietet die Funktionalität Soundwalks und Fragebögen zu erstellen. Dabei wurden die funktionalen Anforderungen aus Kapitel 3 umgesetzt und die dort ausgeführten nicht-funktionalen Anforderungen, durch die Wahl eines geeigneten Frameworks und den passenden Architektur-Patterns erfüllt.

Mit der Benutzung von React als Front-End Framework wurde die Benutzeroberfläche erstellt und Teile der Logik im Web-Client implementiert. Anschließend wurden die Datenbankobjekte dargestellt und die Funktionsweise der Datenbank erörtert.

Durch die Benutzung von einer Zugriffskontrolle auf dem Web-Server wird die Anwendung für unerlaubte Benutzer nicht sichtbar sein.

Die in dem Kapitel zur Anforderungsanalyse zusammengefassten Ansprüche an die Anwendung wurden wie in folgender Tabelle dargestellt, umgesetzt.

Tabelle 9: Überblick der umgesetzten Anforderungen

<i>Anforderung (AF)</i>	<i>Schlüsselwort</i>	<i>Akzeptanz</i>
AF01	Anzeigen	Siehe Teilkapitel 4.2.2 und 4.3.3
AF02	Anlegen	Siehe Teilkapitel 4.2.2, 4.2.3 und 4.3.3
AF03	Template	Siehe Teilkapitel 4.2.2, 4.2.3 und 4.3.3
AF04	D&D in Dashboard	Siehe Teilkapitel Fehler! Verweisquelle konnte nicht gefunden werden. und 4.2.24.3.3
AF05	Speicherung Fragebogen	Siehe Teilkapitel 4.4
AF06	D&D in Fragebogen	Siehe Teilkapitel Fehler! Verweisquelle konnte nicht gefunden werden. und 4.3.4
AF07	Header Fragebogen	Siehe Teilkapitel 4.2.2 und 4.3.4
AF08	Frage-Typen	Siehe Teilkapitel 4.3.4
AF09	Speicherung Fragen	Siehe Teilkapitel 4.4
AF10	Vorschau	Siehe Teilkapitel 4.3.5
AF11	Lokalisierung	Siehe Teilkapitel 4.2.5 und 4.3.4
AF12	Administrator Trennung	Siehe Teilkapitel 4.2.6

5 Zusammenfassung und Ausblick

In dieser Bachelorarbeit wurde eine Anwendung für die Verwaltung von Soundwalks und deren Fragebögen entwickelt. Diese Anwendung unterstützt Drag & Drop überall, wo es sinnvoll benutzt werden kann, um die Bedienbarkeit auf mobilen Geräten zu erhöhen. Einzelne Fragebögen lassen sich mittels Geokoordinaten an realen Wegpunkten orten. Um die Akustik an den einzelnen Stationen des Soundwalks nachzuvollziehen, ist es möglich Audio-Aufnahmen an den Stationen bzw. auf den Fragebögen zu hinterlegen.

In Kapitel 2 wurde eine Marktuntersuchung vorgenommen, wo unterschiedliche Fragebogen-Editoren vorgestellt wurden. Aus diesen Editoren wurden die wichtigsten Kriterien für mobile Fragebogen-Editoren zusammengefasst. Durch den Vergleich der unterschiedlichen Editoren wurde deutlich, dass keiner die Voraussetzungen für einen mobilen Soundwalk Fragebogen-Editor erfüllt und eine passende Anwendung entwickelt werden muss. Zum einen bietet keinen Anbieter eine Gruppierung von Fragebögen oder eine Geolokalisierung mit einem Radius kleiner als einen Kilometer. Zum anderen wird in ihren mobilen Versionen kein Drag & Drop unterstützt. Diese Funktionalitäten wurden in dieser Arbeit umgesetzt. Aus den untersuchten Editoren wurde SurveyJS als Baustein ausgewählt, weil er als einziger Editor eine kostenlose, öffentliche Bibliothek zur Verfügung stellt.

Anschließend wurden in Kapitel 3 die Funktionen der Anwendung beschrieben. Um diese zu erreichen, wurden die funktionalen und nicht-funktionalen Anforderungen, die für diese Arbeit relevant sind, formuliert. Aus den Ergebnissen aus Kapitel 2 und 3 wurde der Entwurf erstellt.

In Kapitel 4 wurde der Entwurf für das zu entwickelnde System vorgestellt und ein durchgängiger Prototyp entwickelt. Dabei wurde auf die Systemarchitektur, auf den Arbeitsablauf und die Gestaltung der Ansichten, sowie die Persistenz der Anwendung eingegangen. Es wurde auch ein Klassenkonzept formuliert und die Anforderungen aus Kapitel 3 umgesetzt. Die funktionalen Anforderungen, welche bei den sich auf dem Markt befindende Editoren nicht vorhanden waren, wurden nachgewiesen. Durch eine geeignete Wahl von Frameworks, konnte ein großer Teil der nicht-funktionalen Anforderungen abgedeckt werden.

5.1 Ausblick

In Hinblick auf die Weiterentwicklung dieser Anwendung steht zunächst der Mehrbenutzer-Betrieb im Vordergrund. Dies wurde jedoch nicht implementiert. Da der Mehrbenutzer-Betrieb Konflikte in der Datenspeicherung verursachen würde, soll eine Locking-Strategie angewendet werden. Für diese Anwendung könnte zum Beispiel das optimistische Locking angewendet werden.

Des Weiteren ist geplant, die Navigation zwischen den Ansichten auch durch die Benutzung von Gesten zu erweitern. Zum Beispiel könnte der Benutzer durch Wischen zwischen den Ansichten navigieren.

Um der Übertragungsgeschwindigkeit in Zonen ohne stabile Internetverbindung gerecht zu werden und den Benutzern eine bestmögliche Anzeige zu gewährleisten, sollte die Netzgeschwindigkeit vom Benutzer gemessen werden. Anschließend können die Bilder auf eine passende Qualität komprimiert und den Benutzern zur Verfügung gestellt werden.

Es ist auch denkbar, unterschiedliche Sprachausgaben einzuführen. Das bedeutet, dass Benutzer die Sprache des Textes in der Anwendung ändern können.

Eine Validierung der hier präsentierten Anwendung für die HAW-Hamburg ist in Planung und soll der Validierung der vorgestellten Lösung dienen.

Literaturverzeichnis

Balzert, Helmut. 2009. *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. 3. Heidelberg : Spektrum Akademischer Verlag, 2009.

—. **2011.** *Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb*. 3. Heidelberg : Spektrum Akademischer Verlag, 2011.

Cam, Paul Le. 2022. React Leaflet. [Online] 2022. [Zitat vom: 15. 05 2022.] <https://react-leaflet.js.org/>.

contributors, MDN. 2020. mdn web docs. [Online] 8. 12 2020. [Zitat vom: 10. 05 2022.] https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/JSON.

Döring, Nicola und Bortz, Jürgen. 2016. *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften* . 5. Berlin : Springer-Verlag Berlin Heidelberg, 2016.

Edenhofer, Ottmar, et al. 2011. *Renewable Energy Sources and Climate Mitigation*. Cambridge, UK : Cambridge University Press, 2011.

Forms.app. 2022. forms.app. [Online] 2022. [Zitat vom: 11. 04 2022.] <https://forms.app/de/preisgestaltung>.

Gauchat, J.D. 2018. *Drag and Drop API*. s.l. : J.D Gauchat, 2018.

Google. 2022. Google Forms. [Online] 2022. [Zitat vom: 20. 05 2022.] <https://docs.google.com/forms>.

Lethaus, Heinz. 2021. conciso. [Online] 2021. [Zitat vom: 28. 02 2022.] https://conciso.de/architekturentscheidungen-aus-nicht-funktionalen-anforderungen/#Was_sind_nicht-funktionale_Anforderungen.

Madasu, Vamsi Krishna, Venna, Trinadh Venkata Swamy Naidu and Eltaeib, Tarik. 2015. SOLID Principles in Software Architecture and Introduction to RESM Concept in OOP. *Journal of Engineering Science and Technology*. February 2015, pp. 1-3.

- Mardan, Azat. 2018.** *Full Stack JavaScript: Learn Backbone.js, Node.js, and MongoDB*. s.l. : Apress, 2018. 148423717X.
- Myck, Thomas und Wothge, Jördis. 2021.** *Infraschall von Windenergieanlagen*. Dessau : Umweltbundesamt, 2021.
- ReactJS. 2022.** Reactjs. *Reactjs*. [Online] React, 2022. [Zitat vom: 20. 05 2022.] <https://reactjs.org/>.
- Reimer, Nick. 2004.** Die älteste technisch genutzte regenerative Energie: Wasserkraft. *taz. die tageszeitung*. 24. 05 2004, S. 7.
- Starke, Gernot und Hruschka, Peter. 2009.** *Software Architektur kompakt*. Heidelberg : Spektrum Akademischer Verlag, 2009. ISBN 978-3-8274-2093-0.
- SurveyJS. 2022.** SurveyJS. [Online] 2022. [Zitat vom: 28. 03 2022.] <https://surveyjs.io/create-survey>.
- SurveyMonkey. 2022.** Survey Monkey. [Online] 2022. [Zitat vom: 05. 05 2022.] <https://www.surveymonkey.de/mp/aboutus/directors/>.
- Tenzer, F. 2022.** Statista. [Online] 2022. [Zitat vom: 29. März 2022.] <https://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenuutzer-in-deutschland-seit-2010/>.
- Waidner, Jannik. 2019.** 40,5 Grad – Neuer Hitzerekord in Deutschland. *Frankfurter Allgemeine Zeitung*. 24. Juli 2019.
- WHO.** *Leitlinien für Umgebungslärm für die europäische Region - Zusammenfassung*. Copenhagen, DK : Weltgesundheitsorganisation Regionalbüro für Europa.
- Yiou, P., et al. 2019.** *Analyses of the Northern European summer heatwave of 2018*. s.l. : Bulletin of the American Meteorological Society, 2019.
- Zammeti, Frank. 2020.** *Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker*. Pottstown : Apress, 2020. 9781484257388.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____ _____ _____
Ort Datum Unterschrift im Original