

# ENTWICKLUNG EINER WEBANWENDUNG (WORKSHOPPY) ZUR DURCHFÜHRUNG VON WORKSHOPS IN ECHTZEIT

## **Abschlussarbeit**

zur Erlangung des akademischen Grades  
Bachelor of Science (B.Sc)

an der

Hochschule für Technik und Wirtschaft Berlin  
Fachbereich 4  
Studiengang Angewandte Informatik

vorgelegt von

**Alongkorn Kiatmontri**

(eingereicht am )

**Erstprüfer:** Herr Prof. Jung, Th.

**Zweitprüfer:** Herr Andreas Flack (LB)

## **Eidesstattliche Erklärung**

Hiermit versichere ich, die vorliegende Abschlussarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Sowohl inhaltlich als auch wörtlich entnommene Inhalte wurden als solche kenntlich gemacht. Die Arbeit hat in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegen.

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_

## **Danksagungen**

## **Kurzfassung**

Ziel dieser Bachelorarbeit ist es, eine Webanwendung zu entwickeln, über welche Workshops in Echtzeit durchgeführt werden können. Dazu werden Workshops angelegt und in einer Datenbank gespeichert. Das Erarbeiten von Ergebnissen erfolgt in einer Sitzung, welche ebenfalls in einer Datenbank gespeichert werden. Mit Hilfe des WebSockets und des Web Application Messaging Protocols soll die Datenübermittlung von mehreren Personen in Echtzeit geprüft werden. Das Hauptaugenmerk dieser Arbeit wird auf die Datenausgabe von mehreren Personen in Echtzeit und die digitale Datenzusammenfassung gelegt. Der Beamer wird dabei als Ausgabemedium verwendet. Außerdem passt sich diese Anwendung automatisch an die Auflösung und Darstellung des jeweiligen Endgerätes (Smartphone, Tablet und PC) an.

# Inhaltsverzeichnis

Eidesstattliche Erklärung . . . . .	2
Danksagungen . . . . .	3
Kurzfassung . . . . .	4
<b>1 Einleitung</b>	<b>12</b>
1.1 Motivation . . . . .	12
1.2 Zielsetzung und Aufbau der Arbeit . . . . .	13
<b>2 Grundlagen</b>	<b>14</b>
2.1 Grundlegende Begriffe . . . . .	14
2.1.1 Workshop . . . . .	14
2.1.2 Brainstorming . . . . .	16
2.1.3 TCP/IP . . . . .	17
2.1.4 HTTP . . . . .	19
2.1.5 Ablauf einer HTTP-Verbindung . . . . .	20
2.1.6 AJAX . . . . .	21
2.1.7 Echtzeit . . . . .	22
2.1.8 WebSocket . . . . .	25
2.1.9 WAMP - Web Application Messaging Protocol . . . . .	28
2.1.10 jQuery . . . . .	30
2.2 Responsive Webdesign . . . . .	31
2.3 Web 2.0 . . . . .	32
2.4 Rich Internet Applications . . . . .	35
2.5 Thick Client und Thin Client . . . . .	37
<b>3 Analyse</b>	<b>38</b>
3.1 Stand der Technik . . . . .	38
3.1.1 IdeaBoardz . . . . .	38
3.1.2 Miro-RealtimeBoard . . . . .	41
3.1.3 MindMap . . . . .	42
3.2 Erkennbare Stärken und Schwächen der Konkurrenz-Tools . . . . .	44
3.2.1 IdeaBoardz: . . . . .	44

3.2.2	Miro-RealtimeBoard:	45
3.3	Projektstruktur	47
3.4	Ist-Analyse	48
3.5	Unternehmensanforderungen	48
3.6	Anforderungsanalyse	49
3.6.1	Funktionale Anforderungen	49
3.6.2	Nicht-funktionale Anforderungen	51
3.6.3	Muss- und Kann-Anforderungen	52
3.7	Beispielszenario	54
<b>4</b>	<b>Design</b>	<b>55</b>
4.1	Gestaltungsrichtlinie	55
4.1.1	Farben	55
4.1.2	Schriftgestaltung	55
4.1.3	Icons	56
4.2	Konzeption	57
4.2.1	Mockup der Willkommenseite	57
4.2.2	Mockup der Hauptseite	58
4.2.3	Mockup der Controller-Seite	60
4.2.4	Mockup der Teilnehmer-Seite	63
4.2.5	Mockup der Präsentation-Seite	64
4.2.6	Mockup der Ergebnisse-Seite	66
4.2.7	Zusammenfassung der Konzeption	67
<b>5</b>	<b>Implementierung</b>	<b>68</b>
5.1	Verwendete Webtechnologien	68
5.1.1	MidCom CMS	68
5.1.2	HTML - Hypertext Markup Language	68
5.1.3	CSS - Cascading Stylesheets	69
5.1.4	Bootstrap	69
5.1.5	PHP - Hypertext Preprocessor	69
5.1.6	phpMyAdmin	70
5.1.7	JQuery UI	70
5.1.8	Ratchet und Autobahn JS	70
5.2	Datenmodellierung	71
5.2.1	ER-Modell	71
5.2.2	Datenbankschema	73
5.3	Architektur der Webanwendung	74
5.4	Serverseitige Implementierung	77

5.5	Clientseitige Implementierung . . . . .	78
5.5.1	Session starten . . . . .	80
5.5.2	Dateneingabe auf der Teilnehmer-Seite . . . . .	84
5.5.3	Darstellung von Daten in Echtzeit . . . . .	85
5.5.4	Daten in Kategorien zusammenfassen . . . . .	87
5.5.5	QR-Code anzeigen . . . . .	91
<b>6</b>	<b>Bewertung</b>	<b>93</b>
<b>7</b>	<b>Schluss</b>	<b>95</b>
7.1	Zusammenfassung . . . . .	95
	<b>Literaturverzeichnis</b>	<b>96</b>

# Abbildungsverzeichnis

2.1	Aufbau eines Datagramms . . . . .	17
2.2	Zeitüberschreitung und erneute Übertragung bei Verlust eines Pakets . . . . .	18
2.3	Die wichtigsten Protokolle im TCP/IP-Referenzmodell . . . . .	18
2.4	Das Client-Server-Modell . . . . .	19
2.5	Aufbau einer URL . . . . .	20
2.6	Klassisches HTTP Request-Response-Paradigma nach [Wöhr, 2004] . . . . .	20
2.7	synchrone und asynchrone Kommunikation . . . . .	22
2.8	Polling . . . . .	24
2.9	LongPolling . . . . .	24
2.10	Beispiel einer Client-Handshake-Anfrage . . . . .	26
2.11	WebSocket-Server Handshake . . . . .	26
2.12	Das WebSocket-Handshake . . . . .	27
2.13	Publish-Subscribe Pattern . . . . .	29
2.14	Remote Procedure Call . . . . .	29
2.15	DOM - Elementenbaum einer Webseite . . . . .	31
2.16	Das Konzept Web 2.0 nach einer Brainstorming-Sitzung . . . . .	33
2.17	Die Bedeutung von Web 2.0 . . . . .	34
2.18	Kommunikationsmodell einer klassischen Webanwendung . . . . .	35
2.19	Kommunikationsmodell einer RIA am Beispiel einer AJAX-Anwendung . . . . .	36
2.20	Vergleich der Desktopanwendung, klassischen Webanwendung und RIAs . . . . .	37
3.1	Erstellen eines eigenen IdeaBoards . . . . .	39
3.2	Darstellung von Sektionen . . . . .	39
3.3	Darstellung einer der Sektionen . . . . .	40
3.4	Zusammenführen und Zuordnen von Daten . . . . .	40
3.5	Realisieren eines Brainstormings mit Hilfe von Miro . . . . .	42
3.6	Health mindmap . . . . .	43
3.7	Projektstrukturplan . . . . .	47
4.1	Verwendete Icon Fonts . . . . .	56
4.2	Mockup für die Anmeldung . . . . .	57
4.3	Mockup für den Anmeldeungsfehler . . . . .	57



---

4.4	Mockup der Hauptseite . . . . .	58
4.5	Mockup für das Erstellen eines neuen Workshops . . . . .	59
4.6	Mockup der Controller-Seite . . . . .	60
4.7	Mockup für das Navigation-Tab „Teilnehmer“ . . . . .	61
4.8	Mockup für das Erstellen einer neuen Session . . . . .	61
4.9	Mockup für die aktive Session . . . . .	62
4.10	Mockup für Eingabe der Benutzernamen . . . . .	63
4.11	Mockup für die Anzeige der Infotext . . . . .	63
4.12	Mockup für die Eingabefunktion . . . . .	63
4.13	Mockup für das Anzeigen eines QR-Codes . . . . .	64
4.14	Mockup für das Anzeigen eines QR-Code und einer Agenda . . . . .	64
4.15	Mockup für die Darstellung der Dateneingabe auf der Präsentation-Seite . . . . .	65
4.16	Mockup für das Zusammenfassen von Daten auf der Präsentation-Seite . . . . .	66
4.17	Mockup für die Darstellung der Ergebnisse des Workshops . . . . .	66
4.18	Zusammenfassung der Konzeption der zu entwickelnden Webanwendung . . . . .	67
5.1	Ratchet-Workflow . . . . .	70
5.2	ER-Modell der Webanwendung . . . . .	72
5.3	Datenbankschema . . . . .	73
5.4	Model-View-Controller Entwurfsmuster . . . . .	75
5.5	Das MVC-Entwurfsmuster angepasst an die zu realisierenden Anwendung . . . . .	76
5.6	Architektur der Webanwendung . . . . .	76
5.7	Workflow auf der Controller-Seite . . . . .	78
5.8	PubSub-Protokoll . . . . .	79
5.9	Controller-Seite . . . . .	80
5.10	Teilnehmer-Seite . . . . .	84
5.11	Präsentation-Seite . . . . .	86
5.12	Controller-Seite, wenn die Session gestartet ist . . . . .	87
5.13	QR-Code auf der Präsentation-Seite . . . . .	92
6.1	Ergebnisse aus den Fragebogen . . . . .	94
6.2	Ergebnisse aus den Fragebogen . . . . .	94

# Tabellenverzeichnis

3.1 Funktionsüberblick der beiden Konkurrenzen. . . . .	46
3.2 Muss- und Kann-Anforderungen . . . . .	53

# Listings

5.1	WebSocket-Server . . . . .	77
5.2	Verbindungsaufbau zum WebSocket-Server - php . . . . .	79
5.3	Publish-Muster auf der Controller-Seite - JavaScript . . . . .	81
5.4	Subscribe-Muster auf der Teilnehmer-Seite - JavaScript . . . . .	82
5.5	Stellt die Eingabefunktion auf der Teilnehmer-Seite bereit - JavaScript . . . . .	83
5.6	HTML-Struktur der Teilnehmer-Seite . . . . .	83
5.7	Funktion für das Abschieken von Daten auf der Teilnehmer-Seite - JavaScript . . . . .	84
5.8	Subscribe-Muster auf der Präsentation-Seite - JavaScript . . . . .	85
5.9	Funktion zur Visualisieren der Daten auf der Präsentation-Seite - JavaScript . . . . .	86
5.10	HTML-Struktur der Präsentation-Seite . . . . .	88
5.11	Kategorie umsetzen - JavaScript . . . . .	88
5.12	Sortierfunktion auf der Präsentation-Seite - JavaScript . . . . .	89
5.13	Importiere das verwendeten JavaScript-Pluing für das Generieren des QR-Codes . . . . .	91
5.14	HTML-Struktur für das Anzeigen des QR-Codes . . . . .	91
5.15	Generieren des QR-Codes - JavaScript . . . . .	91

# 1 Einleitung

Im ersten Kapitelabschnitt der Bachelorarbeit, wird auf die Motivation und die Zielsetzung eingegangen. Zusätzlich wird ein Überblick über den Aufbau der Arbeit aufgezeigt.

## 1.1 Motivation

Beim Suchen und Finden von Lösungen, ungewöhnlichen Geschäftsideen, Innovationen oder um einzelne Projekte erfolgreicher zu machen, bereichert viele Menschen der Begriff Kreativität. Um die Kreativität zu fördern, braucht es Kreativitätstechniken, die dabei helfen, Ideen zu generieren und Einfälle zu sammeln.

Der Klassiker und eine der bekanntesten unter allen Kreativitätstechniken ist das klassische Brainstorming. Sie wurde vom Amerikaner Alex Faickney Osborn erfunden und von Charles Hutchison Clark zur Ideenfindung innerhalb von Gruppen weiterentwickelt. (vgl. [van Aerssen, oJ]) „Er benannte das Brainstorming nach der Idee dieser Methode, nämlich using the brain to storm a problem (wörtlich: Das Gehirn verwenden zum Sturm auf ein Problem).“ [Holzer, 2012]

Die Kreativitätstechnik Brainstorming gilt als eine der beliebtesten Methoden zur Ideenfindung und -sammmlung neuer Geschäftsideen, Ideen für ein Projekt/Produkt oder auch zu einer vorhandenen bzw. gegebenen Problemstellung.

Ziel des Brainstormings ist es, Denkblockaden auf der Suche nach neuen Ideen zu beenden. Diese Kreativitätstechnik wird häufig in Seminaren und Workshops angewendet, um die Gruppenarbeit effektiver und effizienter zu gestalten. Bei einer Brainstorming-Sitzung in einem Workshop kann jeder Teilnehmer auf die Ideen des anderen aufbauen und anknüpfen. Dadurch werden die Teilnehmer gegenseitig durch Ihre Ideen zu neuen Ideen angeregt, wodurch mehr Ergebnisse, als tatsächlich gebraucht, produziert werden.

Eine häufig angewendete Methodik für die Ausarbeitung des Brainstormings in den Workshops ist es, sich Karteikarten oder Notizzettel zu nehmen, seine Ideen und Gedanken darauf zu schreiben und an eine Pinnwand (Flipchart, Whiteboard) anzubringen. Haben alle Teilnehmer Ihre Karteikarten an der Pinnwand angebracht, wird anschließend analysiert und darüber diskutiert. Am Ende der Besprechung werden die gesammelten Daten bewertet und anschließend von dem Moderator dokumentiert. Mit herkömmlichen analogen Workshops bedeutet das für den Moderator, dass er die Karteikarten auf der

Pinnwand abtippen oder abfotografieren muss, um eine Dokumentation erstellen zu können. Da wir uns heutzutage in einem digitalen Zeitalter befinden und uns dieser neuen Welt nicht mehr entziehen können, gilt es, diesen Wandel als Chance zu begreifen, solche analogen Workshops zu digitalisieren, um dem Moderator eine Möglichkeit anzubieten, die Daten digital zusammenzufassen.

## 1.2 Zielsetzung und Aufbau der Arbeit

Im Rahmen dieser Bachelorarbeit soll eine dynamische Webanwendung (Workshoppy) zur Durchführung von Workshops in Echtzeit entwickelt werden, die das klassische Brainstorming digitalisieren und effektiver machen soll. Die Webanwendung soll künftig in den Workshops genutzt werden und muss die Funktionen bieten, welche mehrere Personen (Teilnehmer) über ihre Endgeräte (Smartphone, Laptop oder Tablet) ihre Ideen abgeben können. Dabei werden die eingebrachten Ideen der Teilnehmer in Echtzeit auf einer großen Leinwand (Beamer) präsentiert. Der Moderator soll anschließend die Möglichkeit erhalten, die Ergebnisse digital zusammenzufassen. Die Zusammenfassung soll auch als PDF-Datei exportiert werden können. Bei der Konzeption der Webanwendung ist zu beachten, dass eine benutzerfreundliche Darstellung für die Anwender gewährleistet ist.

Die vorliegende Arbeit ist wie folgt aufgebaut: Das Kapitel 2 stellt vorab ein Überblick über einige grundlegende Begriffe vor. Der Begriff Responsive Webdesign, AJAX-Technologie und Rich Internet Applications (RIA) werden besprochen. Anschließend wird der Thin Client und Thick Client beschrieben. Das Kapitel 3 beschäftigt sich zunächst mit dem Stand der Technik. Die Anforderung zur Webanwendung wird dabei analysiert und konzipiert. In diesem Kapitel werden vor allem die funktionale, nicht-funktionale Anforderungen sowie die Muss- und Kann- Anforderungen ermittelt. Aufbauend auf den Ergebnissen der Anforderungsanalyse erfolgt in Kapitel 4 eine ausführliche Beschreibung über den Entwurf der Benutzeroberfläche (GUI) der Webanwendung. Danach wird das Design der GUI entworfen. Im Kapitel 5 wird zunächst die zu verwendenden Webentwicklungswerkzeuge vorgestellt. Anschließend beschäftigt sich dieses Kapitel hauptsächlich mit der Implementierung der Webanwendung. Zum Schluss wird es im Kapitel ?? die erarbeiteten Ergebnisse zusammengefasst, sowie Ideen für zukünftigen Erweiterungen der entwickelten Webanwendung diskutiert.

## 2 Grundlagen

Dieses Kapitel behandelt die für diese Arbeit nötigen Grundlagen. Zunächst wird ein Überblick über grundlegende Begriffe vorgestellt. Das Responsive Webdesign und einige Merkmale für eine responsive Webseite in Bezug auf die zu entwickelnde Webanwendung werden erläutert. Dann werden das Web 2.0 und der entstandene Ausdruck *Rich Internet Application* beschrieben. Anschließend gibt es die Unterschiede zwischen Thin Client und Thick Client.

### 2.1 Grundlegende Begriffe

#### 2.1.1 Workshop

Workshop<sup>1</sup> ist eine Veranstaltung, bei der eine bestimmte Anzahl von Personen teilnimmt, um außerhalb der Routinearbeit Fragen, Probleme und Themen zu bearbeiten. Jeder Workshop wird von einem Moderator geleitet. Bei größeren Gruppen (mehr als 15 Teilnehmer) ist der Einsatz von weiteren Moderatoren zu empfehlen. Bei den Teilnehmern handeln es sich in der Regel um Spezialisten oder Betroffene, die Ihr Fachwissen zu der behandelten Aufgabe einfließen lassen. Das Ziel ist dabei: Lösungsvorschläge für Aufgaben- oder Problemstellung zu generieren und Maßnahmenplan für die Umsetzung zu entwickeln.

Der Moderator ist der aktive Dienstleister der Gruppe. Er ist für die Vorbereitung sowie Organisation verantwortlich und soll die Gruppe am Ende zum Ziel führen. Seine Aufgaben bestehen unter anderem, Fragestellung gezielt zu formulieren, den Ablaufplan zu erstellen, Denkprozesse anzuleiten, Zeitplan einzuhalten und Ergebnisse zu dokumentieren. Er muss außerdem die stillen Teilnehmer aktivieren sowie die dominanten bremsen und darauf achten, dass die Gruppe bei Diskussionsrunden das Ziel nicht aus den Augen verliert.

Nach Ansicht des Autors [phil. Tomas Bohinc, 2016] können Workshops in folgenden Projektphasen eingesetzt werden:

- Kick-Off-Veranstaltung
- Projektplanung-Prozess
- Problemlösung

---

<sup>1</sup>bedeutet so viel wie „Arbeitskreis oder -gruppe“ .

- Entscheidungsfindung
- Informationsaustausch
- Teamentwicklung
- Scrum
- Projektabschluss

Die Gestaltung von Workshops spielt bei der Qualität der Ergebnisse eine große Rolle. Bei einem unstrukturierten Workshop kann dazu führen, dass er keine Motivation bei den Teilnehmer erregt, um sich an dem Workshop einzubringen und Ergebnisse zu erarbeiten. Um dagegen vorzugehen, kann der Moderator je nach Dauer des Workshops folgende kreative Workshop-Methoden anwenden, um Workshops effektiv und interaktiv zu gestalten:

- World Cafe
- Open Space
- Six Thinking Hats
- Fishbowl
- Lego Serious Play

Die genauen Beschreibungen zu den oben genannten Workshop-Methoden können im Blogpost von [Chornaya, 2018] verfolgt werden.

Wenn es darum geht, neue Ideen für Problemlösungen, neue Produkte, neue Geschäftsideen oder Innovationen zu erzeugen, werden Kreativitätstechniken eingesetzt. Denn durch Kreativität werden Ideen generiert. Viele moderne Kreativitätstechniken haben sich im Laufe der Jahre etabliert. Dem Moderator steht deshalb eine Vielzahl von Kreativitätstechniken zur Verfügung. Der Klassiker und eine der beliebtesten unter allen Kreativitätstechniken ist wie bereits im Unterkapitel **1.1** erwähnt, das klassische Brainstorming. Da die vorliegende Arbeit eine Webanwendung zur Durchführung von Workshops behandelt, die das klassische Brainstorming digitalisieren soll, werde ich deshalb nicht auf die anderen vorhandenen Kreativitätstechniken eingehen.

### 2.1.2 Brainstorming

Wie bereits im Unterkapitel **1.1** benannt, werden beim Brainstorming anhand eines konkreten Themas bzw. Problems Ideen, Einfälle und Vorschläge gesammelt. Es kommt dabei nicht auf die Qualität der Ideen an, sondern zunächst, dass möglichst viele Ideen generiert werden. Beim Brainstorming zählt die Quantität vor Qualität. Die Teilnehmer in der Gruppe sollen ihre Gedanken öffentlich frei äußern. Durch diesen öffentlichen Austausch, können mehr Ergebnisse produziert werden.

Die Gruppengröße bei einer Brainstorming-Sitzung sollte nicht zu groß und zu klein sein. „Je nach Fachliteratur ist von Gruppengröße von 5 bis maximal 20 Personen die Rede“. [Holzer, 2012]

Nach [Reich, 2007] läuft eine Brainstorming-Sitzung in folgenden Phasen ab:

- **Vorbereitung:**  
Der Moderator stellt in dieser Phase die zu behandelten Fragen und die Regeln vor. Bei Notwendigkeit kann ein oder mehrere Protokollant/en bestimmt werden.
- **Ideen sammeln:**  
Die Teilnehmer dürfen Ideen und Vorschläge frei äußern. Der Moderator muss in dieser Phase vor allem die stilleren Teilnehmer motivieren und ermuntern. Die Kritik ist in dieser Phase untersagt. Die Ergebnisse werden dabei protokolliert. In der Regel werden die Ideen auf eine Notizzettel geschrieben und an die Wand gepinnt.
- **Zusammenfassung und Auswertung:**  
Das Brainstorming ist nun beendet und der Moderator wird die Gruppe zunächst die dokumentierten Ergebnisse präsentieren. Anschließend werden die Ideen gemeinsam mit der Gruppe ausgewertet, sortiert und geordnet. In dieser Phase ist Kritik erlaubt und darf geäußert werden. Am Ende dieser Phase soll eine Liste mit den gut bewerteten Ideen und Vorschlägen entstehen.
- **Nachbereitung:**  
Ein Brainstorming fördert nur die Kreativität. Die Vorschläge müssen danach umgesetzt und realisiert werden. Sonst helfen die Ideen nicht, wenn nichts daraus gemacht wird.

Damit eine Brainstorming-Sitzung erfolgreich verlaufen ist, sollten dabei folgenden Regeln eingehalten werden:

- Unabhängig wie verrückt jede einzelne Idee ist, keine Kritik in der Sammlungsphase.
- Quantität vor Qualität, je mehr Ideen desto besser.
- Entwicklung oder Verbesserung von fremden Ideen ist willkommen.
- Lass der Fantasie freien Lauf. Ungewöhnliche Ideen sind erwünscht.



### 2.1.3 TCP/IP

Das Transmission Control Protocol (TCP) und das Internet Protocol (IP) bilden die Grundlage für die gesamte Netzwerkkommunikation und legen demnach die grundlegende Technologien für das Internet dar.

TCP nutzt für die Übertragung der Datenpakete das Übertragungsprotokoll IP, welches zur Vermittlungsschicht im TCP/IP-Referenzmodell gehört. Die Aufgabe vom IP-Protokoll ist, die Datenpakete an den richtigen Rechner im Netzwerk zu transportieren. Die Datenpakete sind nicht anderes als Datagramme. Ein IP-Datagramm enthält unter anderem die IP-Adresse des Absenders und des Empfängers sowie weitere spezifische Übertragungsparameter (**Abbildung 2.1**). Ob alle versendeten Datagramme erfolgreich beim Empfänger angekommen sind, kann das IP-Protokoll jedoch nicht sicherstellen. Solche Fehlerbehandlungen, wie z.B. ob Pakete beim Empfänger tatsächlich angekommen sind, stellt die Transportschicht, allem voran TCP, sicher. (vgl. [Karadeniz, oJ])

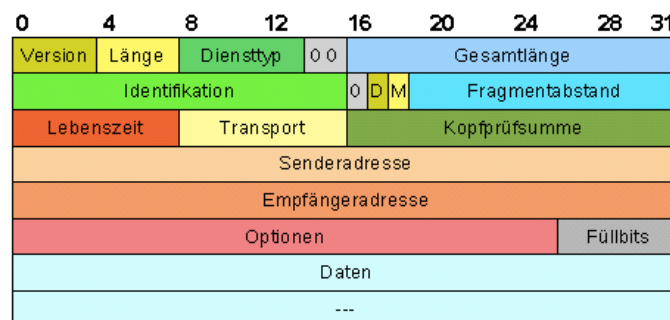


Abbildung 2.1: Aufbau eines Datagramms

**Quelle:** <http://einstein.informatik.uni-oldenburg.de/rechnernetze/diagramm.htm>

TCP ist eines der wichtigsten Protokolle der Transportschicht im TCP/IP-Referenzmodell und ist ein zuverlässiges, verbindungsorientiertes und paketvermittelndes Transportprotokoll, welches das Ziel hat, Datenverluste bei der Datenübertragung zu unterbinden, größere Datenmengen in kleinere Pakete zu zerlegen und die empfangenen Datenpakete über Ports an die korrekten Anwendungen weiterzuleiten. Da sich es bei TCP um ein verbindungsorientiertes Protokoll handelt, definiert das TCP-Protokoll eine Ende-zu-Ende-Verbindung zwischen den Kommunikationspartnern im Netzwerk. (vgl. [o.V., 2019])

Das TCP-Protokoll verwendet dabei das Verfahren namens *Positive Acknowledgement (ACK) with Retransmission*<sup>2</sup>, um die Zuverlässigkeit der Datenübertragung sicherzustellen. Dies hat zu bedeuten, dass der Empfänger dem Sender nach dem Erhalt der Daten mit einer positiven Nachricht quittiert wird. Mit einer positiven Nachricht weiß der Sender, dass das Paket den Empfänger erreicht hat. Sollte von

<sup>2</sup>auf deutsch: positive Bestätigung mit erneuter Übertragung

seitens der Empfänger keine positive Nachricht kommen, wird das Senden solange wiederholt, bis eine positive Antwort beim Sender eingegangen ist (**Abbildung 2.2**). (vgl. [Holtkamp, 2001])

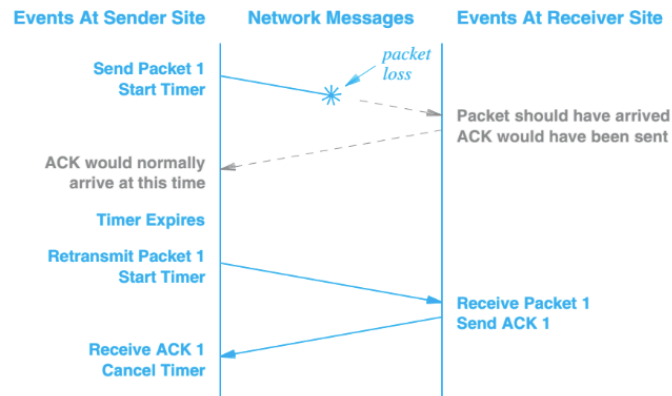


Abbildung 2.2: Zeitüberschreitung und erneute Übertragung bei Verlust eines Pakets

**Quelle:** [http://lemoncisco.blogspot.com/2014/06/internetnetworking-with-tcpip-notes\\_25.html](http://lemoncisco.blogspot.com/2014/06/internetnetworking-with-tcpip-notes_25.html)

Die folgende Abbildung (**Abbildung 2.3**) zeigt die wichtigsten Protokolle im TCP/IP-Referenzmodell. Über der TCP- und IP-Schicht im TCP/IP-Referenzmodell befindet sich die Anwendungsschicht. Diese Schicht beinhaltet alle Protokolle auf Anwendungsebene, die auf TCP oder UDP aufsetzen. Die Anwendungsschicht stellt den Anwendungsprogrammen Dienste zur Verfügung. Das bekannteste Protokoll auf der Anwendungsschicht ist wohl das Hypertext Transfer Protocol (HTTP), welches den Zugriff auf die Webseiten ermöglicht.

Schicht	DoD-Schichtmodell theoretisches Arbeitsmodell	TCP/IP- Protokollsammlung praktische Ausführung		OSI- Schicht Nr.
4 <b>Process Anwendungsschicht</b>	Anwendungen und Nutzdaten interaktive Protokolle	HTTP, SMTP, FTP, DNS POP3, IMAP, Telnet SNMP, NTP	<b>Stream</b>	7 6 5
3 <b>Host-to-Host Transportschicht</b>	Verbindung zweier Netzteilnehmer zur Datenübermittlung	TCP, UDP, TLS (SSL), SCTP	<b>Segment</b>	4
2 <b>Internet Internetschicht</b>	Paketvermittlung Wegewahl (Routing)	IP mit ICMP EIGRP (IGRP) RIP, BGP, OSPF	<b>Datagramm</b>	3
1 <b>Network Access Netzzugangsschicht</b>	Techniken zur Datenübertragung in div. Netztypen	Ethernet mit CSMA/CD Token-Passing, FDDI PPP, ARP, BOOTP (RARP)	<b>Frame</b>	2 1
↔ physikalisches Medium, Übertragungspfad, Kupfer- und Glasfaserleitungen, Funkstrecken ↔				

Abbildung 2.3: Die wichtigsten Protokolle im TCP/IP-Referenzmodell

**Quelle:** <https://www.elektroniktutor.de/internet/tcpip.html>

### 2.1.4 HTTP

Das Hypertext Transfer Protocol (HTTP) ist ein zustandsloses und unidirektionales Datenübertragungsprotokoll in einem Netzwerk. Es wird hauptsächlich eingesetzt, um die Dateien vom Server anzufordern und sie in den Browser zu laden und darzustellen. Bei HTTP handelt es sich um eine unverschlüsselte Kommunikation. Dies hat zur Folge, dass alle Informationen im Klartext gesendet werden. Für die verschlüsselte Verbindung bietet sich das sichere HyperText-Übertragungsprotokoll HTTPS<sup>3</sup> an. HTTP arbeitet nach dem Client-Server-Modell (**Abbildung 2.4**).

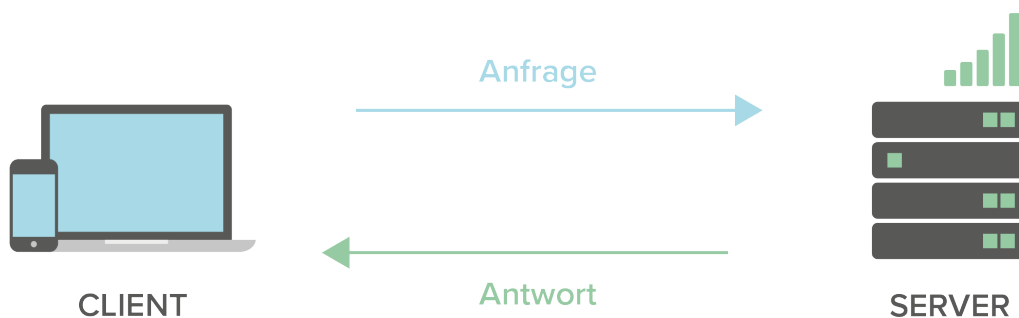


Abbildung 2.4: Das Client-Server-Modell

**Quelle:** <https://www.placetel.de/ratgeber/client>

Der Client (Webbrowser) sendet eine HTTP-Anfrage an den Port 80 des Servers (HTTP-Server). Dieser erledigt die Anfrage vom Client und schickt ihm eine Antwort zurück. Diese Kommunikation verläuft im Textformat. Die Anfrage- sowie die Antwortnachrichten bestehen aus einem Header und Daten. Der Header beinhaltet Steuerinformationen. Der Datenteil enthält den eigentlichen Inhalt der Seite. Nach Abarbeitung der Anfrage wird die Verbindung zwischen Client und Server abgeschlossen. Der Server steht also für die Bearbeitung von neuen Anfragen zur Verfügung. Um dem Server mitzuteilen, was er genau dem Client schicken soll, adressiert der Client bei der Anfrage eine Datei, die sich auf dem Server befindet muss. Dazu verwendet der Client eine URL<sup>4</sup>. Ist diese Datei vom Client nicht vorhanden, antwortet der Server mit der Fehlermeldung (Error 404) zurück. Für eine zuverlässige Kommunikation verwendet HTTP das verbindungsorientierte Transportprotokoll TCP. (vgl. [Stefan Luber, 2018])

<sup>3</sup>Hypertext Transfer Protocol Secure

<sup>4</sup>Uniform Resource Locator

Eine URL ist wie folgt aufgebaut<sup>5</sup>:

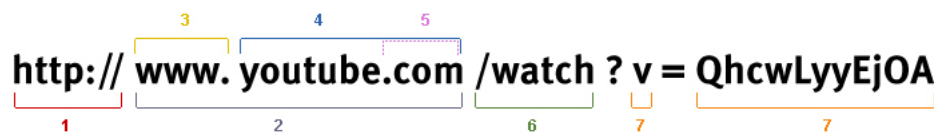


Abbildung 2.5: Aufbau einer URL

**Quelle:** <https://webdesigneinfuehrung.files.wordpress.com/2013/10/url-aufbau.jpg>

1. Das verwendete Protokoll (HTTP). Andere Protokolle könnten ebenfalls verwendet werden, wie HTTPS, FTP.
  2. Es handelt sich um den Host oder Hostnamen.
  3. Die Subdomain: www (World Wide Web).
  4. Die Domain oder der Domainname. Dieser Name ist einmalig, wie eine Postanschrift.
  5. beschreibt die Top-Level-Domain und bezieht sich auf das Ursprungsland der Webseite.
  6. Der Pfad. Dieser verweist auf eine bestimmte Ressource (Datei, Verzeichnis) auf dem Server.
  7. Parameter und Wert: v (Parameter), QhewLyyEjOA (Wert).
- Nach dem Pfad folgt in dem Beispiel ein URL-Parameter. Er wird durch ein Fragezeichen getrennt.

### 2.1.5 Ablauf einer HTTP-Verbindung

Der Ablauf einer HTTP-Verbindung wird mit dem Beispiel eines Aufrufes einer Webseite im Webbrowser dargestellt. Das Aufrufen einer Webseite im Browser erfolgt hauptsächlich in vier Schritten:

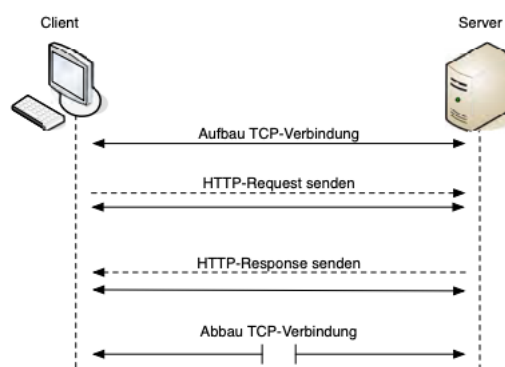


Abbildung 2.6: Klassisches HTTP Request-Response-Paradigma nach [Wöhr, 2004]

<sup>5</sup>vgl. <https://webdesigneinfuehrung.wordpress.com/tag-8/wie-ist-eine-url-aufgebaut/>

1. Der Client baut eine TCP-Verbindung zum Server auf.
2. Der Client, in diesem Fall der Benutzer gibt z.B. eine Adresse (URL) in das Adressfeld seines Webbrowsers ein. Diese Adresse wird als HTTP-Request an der Server gesendet.
3. Der Server bearbeitet die Anfrage vom Benutzer (Client) und antwortet ihm mit einer HTTP-Response zurück.
4. Nach dem Response baut der Server die Verbindung wieder ab.

### 2.1.6 AJAX

AJAX<sup>6</sup> ermöglicht, dass sich die Daten zwischen Browser und Server im Hintergrund austauschen können, ohne die Seite komplett neu zu laden. Man spricht von einer asynchronen Datenübertragung zwischen Client und Server.

Dabei ist das XMLHttpRequest<sup>7</sup>-Objekt in JavaScript für die Durchführung dieser asynchronen Datenübertragung zwischen Client und Server verantwortlich. XHR ist eine Schnittstelle zwischen JavaScript und Daten auf dem Server. Das XMLHttpRequest sendet eine HTTP-Anfrage an einen Webserver. Die Rückgabe vom Server kann ein JavaScript direkt per DOM<sup>8</sup> und CSS<sup>9</sup> in das Dokument ergänzen oder verändert, ohne die Seite neu laden zu müssen. Die statischen Inhalte bleiben erhalten, während nur veränderliche Information ergänzt werden. Das spart vor allem Zeit, reduziert den Trafficverbrauch und ermöglicht dem Nutzer interaktiv mit dem Server zu kommunizieren.

Nach [o.V., 2017] unterstützt XHR neben XML-Dokumente auch alle Textformate und kann eine Anfrage ebenfalls über HTTPS übermitteln. Ein typisches Beispiel für die AJAX-Anwendung ist die Autovervollständigung von Google. Sobald der Nutzer die Daten im Suchfeld auf der Google Webseite eingibt, wird dabei automatisch die passende Vorschläge geliefert.

---

<sup>6</sup>Asynchronous JavaScript and XML

<sup>7</sup>kurz: XHR

<sup>8</sup>Document Object Modal

<sup>9</sup>Cascading Style Sheets

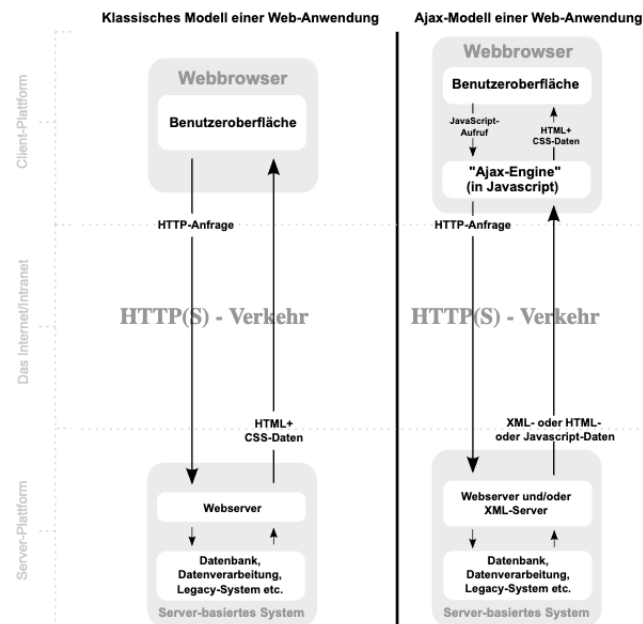


Abbildung 2.7: synchrone und asynchrone Kommunikation

**Quelle:** By I, DanielSHaischt, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=2223689>

### 2.1.7 Echtzeit

Der Begriff „Echtzeit“ rückt immer mehr insbesondere bei bestimmten Webanwendungen in den Vordergrund auf. Was genau steckt hinter diesem Begriff? Lutz Schmitt hat in seiner Diplomarbeit diesen Begriff folgendermaßen definiert: „Echtzeit beschreibt die Ausführung eines Prozesses in einem so kurzen Intervall, dass für die menschliche Wahrnehmung keine Zeit vergangen ist. Echtzeit beschreibt also das Phänomen der Reduzierung eines (Maschinen-)Prozesses auf einen Zeitpunkt.“ Ein Grund dafür, dass der Begriff Echtzeit in den letzten Jahren so viel an Bedeutung gewonnen hat, liegt darin, „[...] dass viele Informationsverarbeitungsprozesse, die bis vor wenigen Jahren noch eine wahrnehmbare Dauer in Anspruch nahmen, so stark beschleunigt worden sind, dass sie eben nicht mehr wahrzunehmen sind. [...] Aus der Rechenzeit, die ein Computer für eine bestimmte Aufgabe benötigt, wird die Prozessverarbeitung in Echtzeit, die sofortige Erledigung ohne Verzögerung. [...] Anstatt auf die Maschine warten zu müssen, kann der Mensch unmittelbar weiterarbeiten.“ [Schmitt, 2006]

In der heutigen Zeit, in der das Internet nicht mehr aus unserem Alltag wegzudenken ist, wurde der Begriff Echtzeit in den letzten Jahren so populär, vor allem bei Webanwendungen, die auf eine schnelle und latenzfreie Datenübertragung abhängig sind, wie z.B. Online-Spiele, Chat-Anwendungen oder kollaborative Webseite.

Ein latenzfreier Informationsaustausch zwischen zwei Teilnehmern in einem Netzwerk ist mit dem bekannten Übertragungsprotokoll HTTP nicht gewährleistet. Dieses Protokoll arbeitet, wie bereits bekannt, nach dem Client-Server-Modell (**Abbildung 2.4**). Das hat zu bedeuten, dass nur der Client die Verbindung zum Server aufbaut, nie umgekehrt. Erst dann wenn die Verbindung zum Server erfolgreich hergestellt ist, folgt das Abschicken von Request- und Response-Nachrichten zwischen Client und Server. Man spricht hier von synchroner Übertragung. Nach dem Absenden der Antwortnachricht baut der Server die Verbindung anschließend wieder ab.

Das Übertragungsprotokoll HTTP war in der Vergangenheit die perfekte Lösung für viele klassische Webanwendungen, um Kommunikation oder auch Interaktion zwischen zwei Kommunikationspartnern im Internet zu realisieren. Im Sinne der Echtzeit-Webanwendungen erfüllt dieses Übertragungsprotokoll jedoch nicht alle Anforderungen. Da es sich bei HTTP um eine synchrone Datenübertragung handelt, wird dabei die Benutzeraktivität unterbrochen, bis der Client die Antwort vom Server erhalten hat. Dieser Mangel kann durch AJAX mit der sogenannten asynchronen Datenübertragung behoben werden. Die Daten werden bei diesem Kommunikationsmodell im Hintergrund ausgetauscht, ohne dass die komplette Seite neu geladen werden muss. Trotz der Anwendung von AJAX bleibt das Hauptproblem weiterhin bestehen. Der Server kann bei HTTP nur auf Anfragen eines Clients reagieren, d.h. er wartet passiv auf Anforderungen. Eine Echtzeit-Anwendung soll durch die Interaktion vom Benutzer nicht unterbrochen werden. Häufig werden Echtzeit-Anwendungen durch Hacks (Polling oder Long Polling) simuliert. (vgl. [Herb, oD])

Beim Polling wird der Server in regelmäßigen Abständen (z.B. alle zwei Sekunden) vom Client angefragt, ob er neue Daten hat. Falls neue Daten vorliegen, wird der Server diese ohne Verzögerung an dem Client senden. Im Fall, dass keine Daten für die Anfrage vorliegen, wird dem Client vom Server mit einer leeren Nachricht geantwortet (**Abbildung 2.8**). (vgl. [Aderhold, 2013])

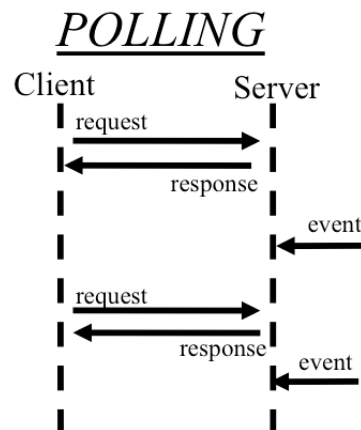


Abbildung 2.8: Polling

**Quelle:** <https://www.heise.de/developer/imgs/06/6/7/6/2/3/3/Polling-61cb54a128001c08.png>

Beim Long Polling wird der Server ebenfalls angefragt. Anders als beim Polling wird der Server diesmal bei nicht vorhandenen Daten solange warten, bis er sie an dem Client liefern kann. Das heißt, der Server hält die Verbindung solange offen, bis neue Daten für den Client verfügbar sind. Nachdem der Client die Daten erhalten hat, sendet er wieder eine Anfrage an den Server, um auf weitere Daten zu warten (**Abbildung 2.9**). (vgl. [Aderhold, 2013])

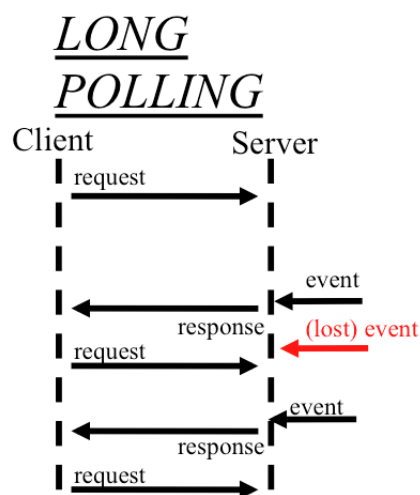


Abbildung 2.9: LongPolling

**Quelle:** <https://www.heise.de/developer/imgs/06/6/7/6/2/3/3/LongPolling-616183343d043825.png>



Damit der Client und Server mit möglichst geringen Latenzen kommunizieren können, wird dafür eine bidirektionale Kommunikation benötigt. Mit dieser Art der Kommunikation können Daten in beide Richtungen gleichzeitig übertragen werden. Man bezeichnet diese Kommunikationsart als Vollduplex. Im Gegensatz zu Vollduplex erlaubt das Halbduplex-Verfahren keine gleichzeitige Kommunikation in beide Richtungen (vgl. [Wikipedia, 2018]).

„HTTP ist von Natur aus ‘nur’ halbduplex. Das bedeutet, dass für die bidirektionale Kommunikation zwischen Browser und Server ein separater HTTP Request für jede Richtung benötigt wird. Das erzeugt natürlich einen Menge Overhead. [...] HTTP Request/Response Header können schnell ein paar Hundert Bytes veranschlagen. Hinzu kommt ab und an die eigentlich wertlose Information, dass es keine Änderungen am Zustand des Servers gab.“ [Weßendorf, 2011]

Um dieses Problem zu lösen, wurde deshalb der Kommunikationsstandard namens WebSocket entwickelt.

### 2.1.8 WebSocket

WebSocket wurde 2008 entwickelt. „Chrome war 2009 der erste Browser, der WebSocket unterstützte; nach und nach folgten alle großen Wettbewerber. Seit 2011 ist WebSocket ein W3C<sup>10</sup>-Standard.“ [o.V., oJ]

WebSocket ist ein bidirektionaler und vollduplexer Kommunikationsstandard, der entwickelt wurde, „[...] um eine bidirektionale Verbindung zwischen einer Webanwendung und einem WebSocket-Server bzw. einem Webserver, der auch WebSockets unterstützt, herzustellen.“ [Wikipedia, 019a]

Mit WebSocket werden Daten in beide Richtungen über einen Kommunikationskanal übertragen. Client und Server können gleichzeitig miteinander „reden“, sobald eine WebSocket-Verbindung besteht. WebSocket verwendet den gleichen Port wie HTTP, nämlich den Port 80. Es wird dabei ein WebSocket-Protokoll namens **Handshake** benutzt, um die Verbindung zwischen Client und Server aufzubauen. (vgl. [Weßendorf, 2011])

---

<sup>10</sup>World Wide Web Consortium

Das Handshake-Verfahren funktioniert wie folgt:

**Client:**

```
GET /chatService HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 8
```

Abbildung 2.10: Beispiel einer Client-Handshake-Anfrage

**Quelle:** <https://www.heise.de/developer/artikel/WebSocket-Annaeherung-an-Echtzeit-im-Web-1260189.html?seite=all>

Im Prinzip ist es ein Aufsatz, der praktisch auf dem HTTP-Protokoll läuft. Wie in der **Abbildung 2.10** zu sehen ist, schickt der Client eine normale GET-Anfrage an den Server und sagt dementsprechend auf der Serverseite, was er genau haben will. Mit dem **Upgrade** sagt der Client, dass er auf das WebSocket-Protokoll wechseln möchte. Dafür wird für den Verbindungsaufbau ein **Sec-WebSocket-Key** zum Server übermittelt. Bei diesem Key handelt es sich um eine Base64-encodierte Zeichenkette, welche vom Server benutzt wird, um den Verbindungsaufbau zu akzeptieren. (vgl. [Weßendorf, 2011])

**WebSocket-Server:**

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: superchat
```

Abbildung 2.11: WebSocket-Server Handshake

**Quelle:** <https://www.heise.de/developer/artikel/WebSocket-Annaeherung-an-Echtzeit-im-Web-1260189.html?seite=all>

Der WebSocket-Server (**Abbildung 2.11**) bearbeitet die Anfrage und antwortet mit dem HTTP-Status code 101 Switching Protocols. Dabei liefert er dem Client die Informationen mit, dass er das Upgrade akzeptiert hat (**Sec-WebSocket-Accept**). (vgl. [Weßendorf, 2011])

„Zusätzlich gibt der Server an, dass er das ‘superchat’-Protokoll kennt. Das hat den Vorteil, dass die Browseranwendung direkt gegen dieses Protokoll beziehungsweise diese API geschrieben wird, statt gegen die WebSocket-API. Entwickler, die mit der Programmierschnittstelle beziehungsweise dem anwendungsspezifischen Protokoll vertraut sind, brauchen keine neue API erlernen, um WebSocket-Anwendungen zu erstellen. Die clientseitige Schnittstelle des ‘superchat’-Protokolls kapselt die eigentliche Kommunikation mit dem WebSocket-Server.“ [Weßendorf, 2011]

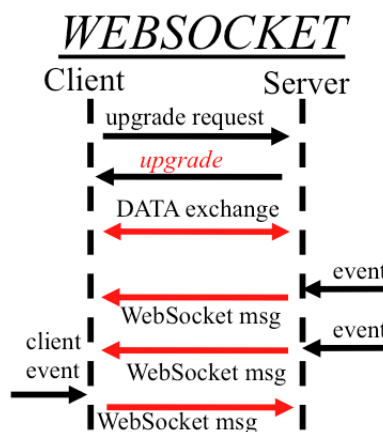


Abbildung 2.12: Das WebSocket-Handshake

**Quelle:** <https://www.heise.de/developer/imgs/06/6/7/6/2/3/3/WebSocket-a70195c3f57b1308.png>

Nach dem Handshake besteht eine persistente Verbindung zwischen Client und Server und beide können jederzeit mit dem Senden von Daten beginnen (**Abbildung 2.12**). Eine WebSocket-Verbindung erkennt man an dem neuen URL-Schema. Statt wie gewohnt „http:“ oder für sichere HTTP-Verbindungen „https:“ als Protokoll anzugeben, wird bei einer WebSocket-Verbindung „ws:“ verwendet. Für sichere Verbindungen steht „wss:“ zur Verfügung.

Zusammenfassend lässt sich sagen, dass WebSocket ein geeignetes Protokoll ist, um

„[...] die komplexen Probleme des ‘Echtzeit-Web’, wie Latenz oder Netzverkehr, anzugehen. WebSocket wird jedoch nicht als ein ‘besseres AJAX’ entwickelt. Ebenfalls stellt WebSocket keinen 1:1-Ersatz für HTTP dar, sondern bietet vielmehr einen effizienten, bidirektionalen Kommunikationskanal an. Die Integration von ‘Echtzeit’ innerhalb von Webanwendungen ist nicht mehr an Hacks und Workarounds gebunden, sondern erfolgt auf Basis eines standardisierten, effizienten und bidirektionalen Protokolls. Wichtig ist hierbei, dass man sämtliche TCP/UDP-Protokolle auf Basis von

WebSocket zum Browser bringen kann. Der Abstraktionsgrad zukünftiger Webanwendungen steht damit den Desktop-Anwendungen in nichts nach.“ [Weßendorf, 2011]

### 2.1.9 WAMP - Web Application Messaging Protocol

Das Web Application Messaging Protocol (WAMP) ist ein standardisiertes und unterstütztes WebSocket-Unterprotokoll, welches Entwicklern die richtige Semantik bietet, die sie benötigen, um Kommunikation zwischen Komponenten in verteilten Anwendungen auf bequeme Art und Weise zu handhaben. WAMP vereinigt zwei grundlegende Kommunikationsmuster zu einem Protokoll: das Publisher-Subscriber-Muster (**PubSub**) und Remote Procedure Calls (**RPC**). WAMP ist ein geroutetes Protokoll, bei dem alle Komponenten eine Verbindung zu einem WAMP-Router aufbauen. Der WAMP-Router, wird auch als Broker bezeichnet, übernimmt die zentrale Aufgabe des Annehmens und Verteilens von Anfragen. Eine Komponente steht für ein Programm, dass eine Verbindung zum Router aufbaut und anschließend Anfragen über RPC und PubSub stellen kann.(vgl. [WAMP, oD])

„WAMP basierte ursprünglich auf dem WebSocket-Protokoll, welches eine bidirektionale Verbindung zwischen zwei Teilnehmern mit geringen Latenzen ermöglicht. Heute ist WAMP von WebSocket unabhängig, auch wenn beide Protokolle noch häufig gemeinsam genutzt werden.“<sup>11</sup>

#### ▪ Publish-Subscribe

Das PubSub-Muster besteht aus einem Publisher (Verteiler) und vielen Subscribers (Abonnenten). Die Abonnenten haben die Möglichkeit, alle an sie gerichteten Nachrichten zu empfangen oder bestimmte Nachrichten zu abonnieren, die als Themen (Topics) bezeichnet werden. Das PubSub-Muster hat vor allem den Vorteil, dass ein Broker (Vermittler) zur Trennung von Verteiler und Abonnenten verwendet wird (**Abbildung 2.13**). Aus dem Grund benötigt der Verteiler und die Abonnenten keine gegenseitige Kenntnis. Dies vereinfacht die Implementierung und verbessert die Skalierbarkeit. Es gibt drei Abonnement-Ansätze, die eingesetzt werden können. In einer Web-Architektur, die einen Broker hat, übernimmt der Broker selbst das Nachrichten-Routing anstelle des Servers. In einer brokerlosen Architektur empfängt der Server die Themenanforderung und liefert die Nachrichten an diejenigen Clients, die eine Themenanforderung gestellt haben. Im dritten Ansatz empfangen die Clients alle Nachrichten vom Server und filtern die Nachrichten, die sie nicht benötigen. Der Nachteil bei diesem dritten Ansatz ist, dass er nicht skalierbar ist. (vgl. [eduonix, 2018])

---

<sup>11</sup><https://www.it-times.de/news/wamp-was-hinter-dem-web-application-messaging-protocol-steckt-118608/> (abgerufen am 26.08.2019)

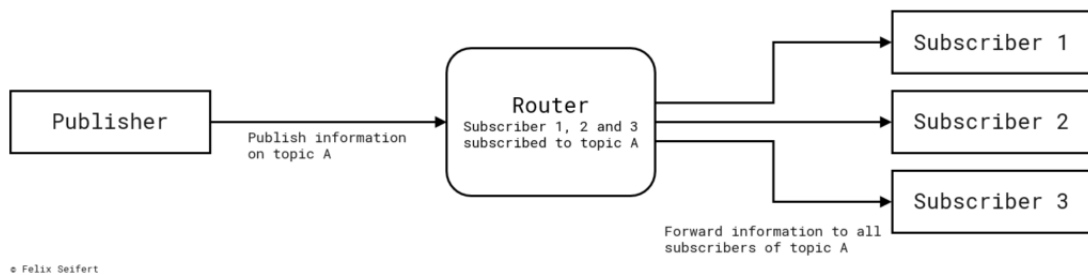


Abbildung 2.13: Publish-Subscribe Pattern

**Quelle:** <https://blog.felix-seifert.com/web-application-messaging-protocol/>

Dabei werden die Topics im WAMP Protokoll über eine sogenannte Uniform Resource Identifier (URI) beschrieben<sup>12</sup>, wie zum Beispiel:

- com.myapp.topicA
- com.myapp.topicB

Jene URIs dienen zur Identifizierung von Topics oder auch Remote Procedure Calls Methoden.

#### ▪ Remote Procedure Calls

Neben dem PubSub-Muster unterstützt das WAMP Protokoll auch das Remote Procedure Calls (RPC). Das RPC-Muster ermöglicht es, einem Programm mit einem anderen Programm in einem unterschiedlichen Netzwerk zu kommunizieren. Das RPC-Nachrichtenprotokoll besteht aus einer Anruf- und einer Antwortnachricht. In dieser Architektur sendet ein Client eine RPC-Anfrage an einen Server. Der Server sendet die Ergebnisse an den Client zurück. (vgl. [eduonix, 2018])

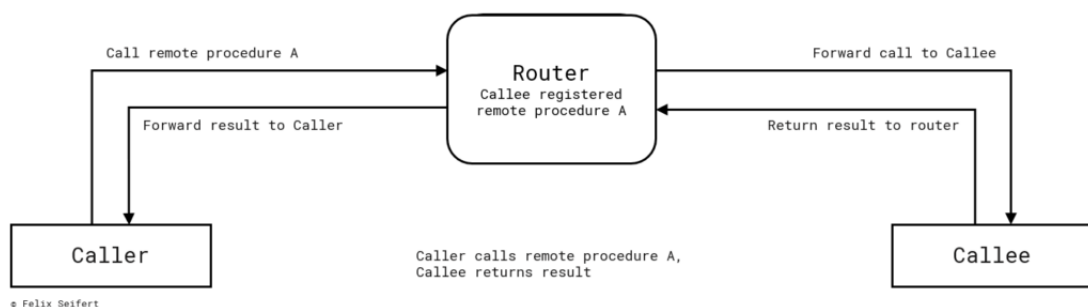


Abbildung 2.14: Remote Procedure Call

**Quelle:** <https://blog.felix-seifert.com/web-application-messaging-protocol/>

Die **Abbildung 2.14** beschreibt den Ablauf einer RPC-Anfrage im WAMP Protokoll. Dabei ist zu erkennen, dass auch beim RPC-Muster einen Router zum Einsatz kommt, der Anrufe entgegennimmt. Nachdem der Router einen Anruf erhalten hat, leitet er diesen Anruf an den entsprechenden

<sup>12</sup>vgl. <https://crossbar.io/docs/URI-Format/>

Angerufenen (Callee) weiter. Dieser Angerufene bearbeitet die empfangenen Anfragen und sendet die Ergebnisse an den Router, der sie direkt an den Anrufer (Caller) weiterleitet. Um dem Router die Adresse des Angerufenen mitzuteilen, muss eine entfernte Prozedur beim Router registriert werden. Somit müssen weder der Anrufer (Caller) noch der Angerufene (Callee) die Adresse des anderen Teilnehmers kennen.

Das in WAMP verwendete Standard-Transportprotokoll ist WebSocket. Mit WebSocket-Transport ist es möglich, eine dauerhafte bidirektionale Verbindung herzustellen. WAMP ist nicht auf die Unterstützung von Webanwendungen beschränkt. Es ist eine geeignete Lösung für verteilte Anwendungen. Ein besonderer Vorteil des WAMP ist, dass es die Trennung von Komponenten unterstützt, die miteinander kommunizieren müssen. Eine direkte Kommunikation ist nicht erforderlich, da der Router die Nachrichtenverteilung übernimmt. Außerdem ist WAMP sprachunabhängig, daher beschränkt es sich nicht auf eine bestimmte Programmiersprache. (vgl. [eduonix, 2018])

### **2.1.10 jQuery**

jQuery ist eine JavaScript-Bibliothek, die Klassen und Methoden zur Verfügung stellt, um die Arbeit mit JavaScript zu vereinfachen. jQuery ist nicht nur kompakter und komfortabler als JavaScript, sondern außerdem browserübergreifend, was bei JavaScript in der Vergangenheit nicht immer der Fall war. jQuery vereinfacht viele JavaScript-Funktionen, die bei der Webentwicklung oft verwendet wird. Wie JavaScript ermöglicht jQuery auch den Zugriff auf DOM-Elemente, die gezielt angesprochen und manipuliert werden können. (vgl. [Augsten, 2019])

Laut [Wikipedia, 019b] ist jQuery die meist verwendete JavaScript-Bibliothek und wird auf rund 70% der 10000 meistbesuchten Webseiten eingesetzt.

Das DOM versteht sich als die Schnittstelle für den Zugriff auf den Tags, Attribute sowie Inhalte von HTML- oder XML-Dokumenten und wird vom W3C definiert. Mithilfe von Selektoren und dem DOM können die HTML-Elemente aufgerufen, verändert, hinzugefügt und gelöscht werden. Die Elemente können über id- oder class-Attribute selektiert werden.

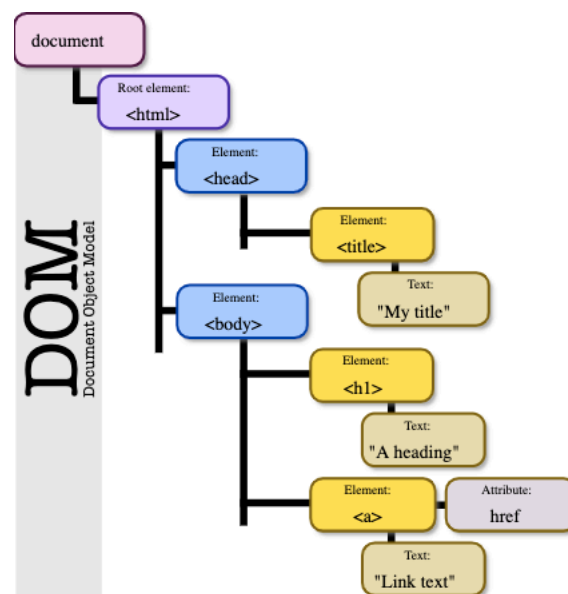


Abbildung 2.15: DOM - Elementenbaum einer Webseite

**Quelle:** By Birger Eriksson - Own work, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=18034500>

Weitere jQuery-Funktion:

- Event Handling
- Form Handling
- AJAX
- JSON
- Collect und Select
- Animationen

## 2.2 Responsive Webdesign

Um eine Webseite geräteübergreifend zu gestalten, benötigt sie ein „Responsive Webdesign“. Beim Responsive Webdesign handelt es sich um eine reaktions- und anpassungsfähige Weboberfläche, so dass diese ein einheitliches Anzeigen von Inhalten sowie den strukturellen Aufbau einer Webseite auf dem Desktop-Computer, Tablet und Smartphone bietet. Laut [Wikipedia, 019c] wurde der Begriff im Jahr 2010 vom amerikanischen Webdesigner Ethan Marcotte erfunden.

Da es zu jener Zeit kaum internetfähige Mobilgeräte auf dem Markt gab, waren die meisten Webseiten statisch und nur für Desktop-Computer entwickelt worden. Um eine Webseite für Mobilgeräte anzubieten, musste separat eine mobile Webseite entwickelt werden.

Erst mit der Einführung des iPhones von Apple im Januar 2007<sup>13</sup> war es für den Benutzer möglich, mobil ins Internet zu gehen. Erstmals verfügte ein Mobilgerät einen vollwertigen Webbrowser und war über einen Touchscreen verfügbar. Viele Firmen folgten dem Beispiel von Apple und wenige Jahre später brachten sie eine Menge an verschiedenen Mobilgeräten mit individuellen Displaygrößen auf dem Markt.

Seit der Einführung der Smartphones, Tablets und mit der steigenden mobilen Internetnutzung ist ein Responsive Webdesign heutzutage nicht nur ein nettes Feature sondern ein Pflichtprogramm für jeden Webseitenbetreiber. Für neue Webseiten liegt es auf der Hand, wie man vorgehen sollte, für bestehende statische Webseiten ist der Aufwand sehr groß, diese auf eine responsive Webseite umzustellen, da eventuell die komplette Seite neu entwickelt werden muss.

Die vorliegende Arbeit soll auf verschiedenen Geräten optimal angezeigt werden können. Als „responsive“ Webseite muss sie unter anderem folgende Eigenschaften haben.

- **flexibles Grid<sup>14</sup>-Layout**

Seiten und Elemente, wie Bilder oder Textblöcke, müssen sich der Bildschirmauflösung des mobilen Endgerätes anpassen. Hier werden für die Elemente und Seiten prozentuale statt fester Pixelwerte verwendet.

- **keine festen Schriftgrößen**

Fließtexte und Headlines müssen so angepasst werden, dass sie sowohl auf dem PC als auch auf Smartphones und Tablets gut lesbar sind. Hier ist eine feste Schriftgröße nicht geeignet. Es wird daher mit prozentualen Werten oder Maßen wie em gearbeitet.

## 2.3 Web 2.0

Das Web 2.0 wird als „Mitmach-Netz“ verstanden. Es ist eine Revolution hinsichtlich der Nutzung des World Wide Web, bei der Internetnutzer nicht mehr wie früher, nur die Inhalte des Internets konsumieren, sondern auch eigene Inhalte selbst produzieren, wie Videos und Fotos einstellen oder Texte schreiben. Durch seine Beteiligung im Web ist der Internetnutzer selbst ein Teil des Internets.

Beim Web 2.0 handelt es sich dabei nicht um eine neue Technologie, sondern vielmehr um die Weiterentwicklung des Internets. Dazu zählt beispielsweise der vermehrte Einsatz der AJAX-Technologie, die eine asynchrone Datenübertragung zwischen Client und Server ermöglicht. Oder der Einsatz von

---

<sup>13</sup>vgl. [https://de.wikipedia.org/wiki/IPhone\\_\(erste\\_Generation\)](https://de.wikipedia.org/wiki/IPhone_(erste_Generation))

<sup>14</sup>aus dem Englischen für das Gestaltungsraster bekannt



Abonnementdiensten (Web-Feeds), welche eine grundlegende Eigenschaft für die neue Generation des World Wide Webs sind. (vgl. [o.V., 2008])

Der Begriff Web 2.0 ist durch seinen Artikel „What is Web 2.0“ von amerikanischen Verleger Tim O'Reilly in 2005 bekannt geworden<sup>15</sup>.

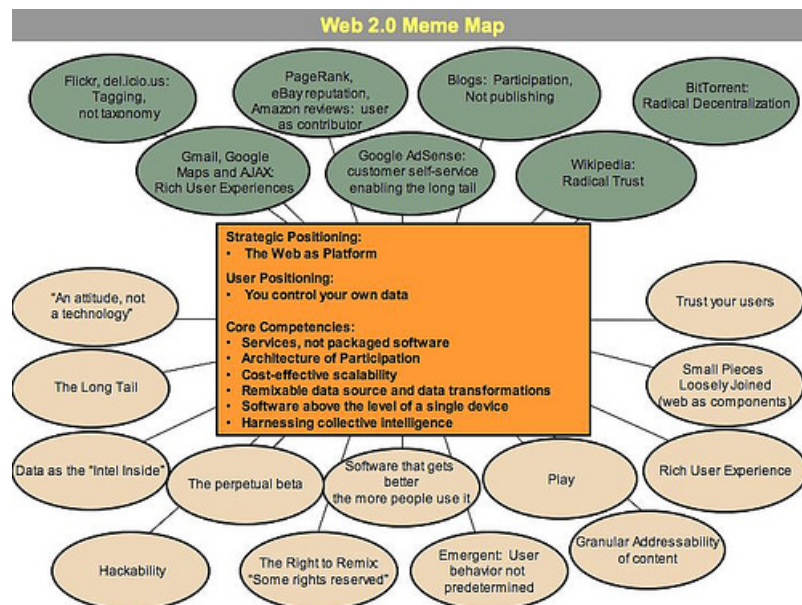


Abbildung 2.16: Das Konzept Web 2.0 nach einer Brainstorming-Sitzung

**Quelle:** <http://www.siliconbeat.com/entries/meme-map.jpg>

<sup>15</sup>vgl. <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>

Die **Abbildung 2.16** zeigt ein Konzept von Web 2.0, die mit einem Brainstorming zwischen O'Reilly und MediaLive International entwickelt wurde. Die **Abbildung 2.17** beschreibt einige Beispiele für die Bedeutung von Web 2.0, die im ersten Brainstorming zwischen O'Reilly und MediaLive International formuliert wurde.

Web 1.0		Web 2.0
DoubleClick	-->	Google AdSense
Ofoto	-->	Flickr
Akamai	-->	BitTorrent
mp3.com	-->	Napster
Britannica Online	-->	Wikipedia
personal websites	-->	blogging
evite	-->	upcoming.org and EVDB
domain name speculation	-->	search engine optimization
page views	-->	cost per click
screen scraping	-->	web services
publishing	-->	participation
content management systems	-->	wikis
directories (taxonomy)	-->	tagging ("folksonomy")
stickiness	-->	syndication

Abbildung 2.17: Die Bedeutung von Web 2.0

**Quelle:** <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=1#mememap>

Außerdem stellte O'Reilly sieben Merkmale vor, die kennzeichnend für Web 2.0 sind:

- **Das Web als Plattform:**
  - Das Web als Plattform ähnlich wie ein Betriebssystem.
- **Kollektive Intelligenz:**
  - Verlinkung der Daten und Seiten untereinander.
- **Daten als nächstes Intel Inside:**
  - Die gesammelten Daten sind die Basis einer Webanwendung und sind wichtiger und wertvoller als eine einzelne Anwendung.
- **Softwarelebenszyklus:**
  - Software wird nicht mehr als Produkt ausgeliefert, sondern als Service.
- **Lightweight Programming Models:**
  - Daten werden durch Web-Services bereitgestellt.
  - Die Daten werden über die Web-Services wie RSS oder REST-basierten Web-Services verteilt oder ausgetauscht.
- **Software über Gerätegrenzen hinaus:**

- Geräteunabhängige Anwendungen, z.B. nicht nur für den PC sondern auch mobile Geräte.
- **Rich User Experiences:**
  - Benutzerführung mit interaktiver Benutzeroberfläche, die sich kaum von einem Desktop-Programm unterscheiden.
  - AJAX-Technologie.

## 2.4 Rich Internet Applications

Die klassischen Webanwendungen schicken generell die Formulareingaben vom Nutzer an einen Webserver. Als Rückgabe vom Webserver bekommt der Client, in diesem Fall der Webbrowser, eine neu generierte Webseite gemäß dem zuvor eingegebenen Daten. Bei jeder Anfrage des Clients wird die komplette Webseite neu generiert und übertragen. Demzufolge werden solche Webanwendungen als wenig intuitiv bezeichnet und können sich auch nachteilig auf die Geschwindigkeit sowie schlechte User-Experiences führen.

Außerdem bieten klassische Webanwendungen nicht genügend Funktionalitäten und Interaktionsmöglichkeiten auf der Benutzeroberfläche. Das hat zur Folge, dass sie gegenüber den Desktop-Anwendungen im Nachteil sind.

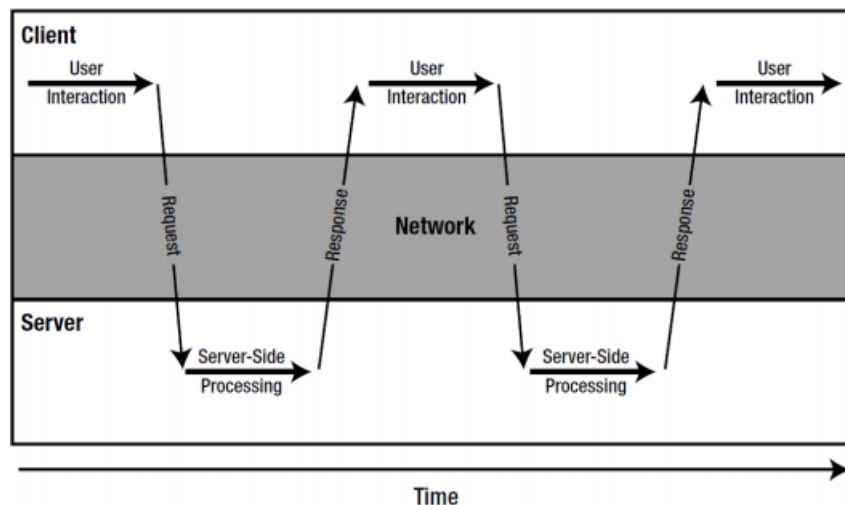


Abbildung 2.18: Kommunikationsmodell einer klassischen Webanwendung

**Quelle:** [Bankras/Smeets/Bram/Boness/Uri/Roald, 008a]

RIAs (Rich Internet Applications) sind webbasierte Anwendungen, deren Funktionalitäten und Eigenschaften denen einer Desktop-Anwendungen entsprechen. RIAs erkennt man unter anderem daran, dass sie plattformunabhängig sowie über das Internet zugänglich sind, wobei RIAs auch offline Anwendungen

bieten. Und sie sollten auch eine interaktive Benutzeroberfläche erzeugen können. Zu Interaktionsmöglichkeiten auf der Benutzeroberfläche zählen beispielsweise Drag-and-Drop-Fähigkeit oder Bedienbarkeit über Tastenkürzel.

Nach [Hartmann, 2007a] unterscheidet sich die RIA-Technologie in zwei Kategorien:

- **rein Browser-basierte Anwendungen:** Der Client verwendet ausschließlich einen Browser. In dieser Kategorie laufen alle AJAX-Anwendungen ohne weitere Browser-Plugins.
- **in plugin-basierte und Stand-Alone-Anwendungen:** Auf der Clientseite werden weitere zusätzliche Laufzeitumgebungen (Flash oder Java) verwendet.

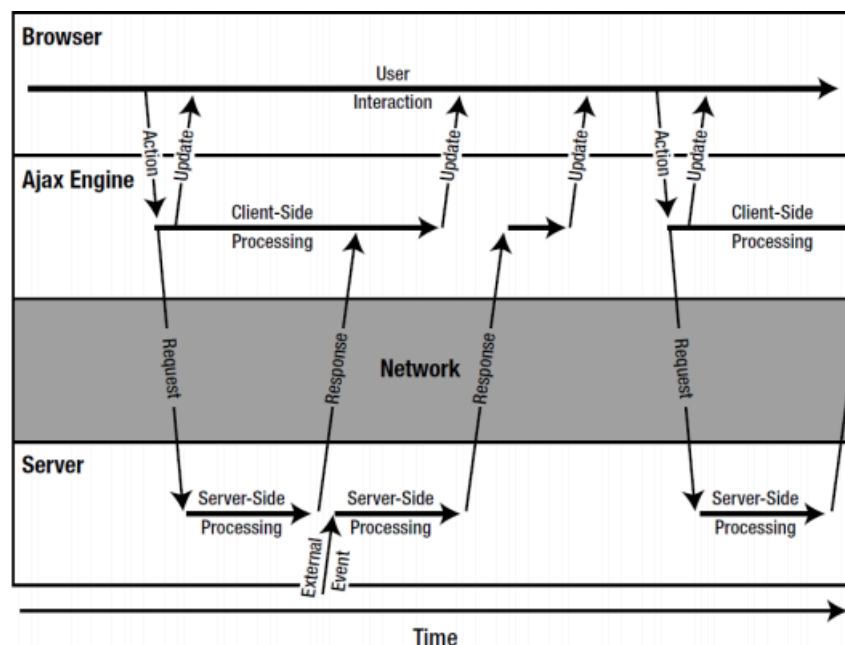


Abbildung 2.19: Kommunikationsmodell einer RIA am Beispiel einer AJAX-Anwendung

**Quelle:** [Bankras/Smeets/Bram/Boness/Uri/Roald, 008b]

Die Vorteile der RIA gegenüber herkömmlichen Webanwendungen können nach [Hartmann, 2007b] wie folgt beschrieben werden:

- **flüssigere Bedienung** im Vergleich zur herkömmlichen Webanwendungen, da nur notwendige Teile der GUI verändert werden.
- geringer Serverzugriff, da in RIAs **mehr Daten und Logik auf Clientseite** verarbeitet werden.
- geringer Serverzugriff und eine flüssigere Bedienung führen zu **weniger Netzwerklast**.

- **bekannte Benutzerschnittstelle**, da sie sich an Desktop-Anwendungen orientiert, ist dies für den Benutzer keine große Umgewöhnung.
- **einfacher Zugriff auf RIA**, da die Laufzeitumgebung, insbesondere der Browser in den meisten Fällen auf den Clients bereits installiert ist.
- **offline-Funktionalität**, wenn die Programmlogik der konzipierte Anwendung auf Clientseite abläuft, kann das Programm auch komplett ohne Serverzugriff verwendet werden, sobald es einmal geladen ist.

Feature	C/S, Desktop	Web	RIA
Universal client (browser)	YES	YES	YES
Client installation	Complex	Simple	Simple
Interaction capabilities	Rich	Limited	Rich
Server-side business logic	YES	YES	YES
Client-side business logic	YES	Limited	YES
Full page refresh required	NO	YES	NO
Frequent server round-trips	NO	YES	NO
Server-to-client communication	YES	NO	YES
Disconnected functioning	YES	NO	YES

Abbildung 2.20: Vergleich der Desktopanwendung, klassischen Webanwendung und RIAs

**Quelle:** [Bozzon/Comai/Fraternali/Carughi, 2006]

## 2.5 Thick Client und Thin Client

Ein Client ist ein Computer oder eine Software, die eine Netzwerkverbindung zu einem Server aufbaut, mit ihm kommuniziert und Dienste sowie Daten anfordert. Ein Client hat vor allem die Aufgabe, Anfragen der Anwender an den Server zu übermitteln und die übermittelten Informationen grafisch darzustellen, sodass der Anwender sie auf seinem Gerät wahrnehmen kann (**siehe Abbildung 2.4**).

Es gibt dabei zwei Arten von Clients zu unterscheiden:

- **Thick Client:**  
wird häufig auch als Fat Client bezeichnet, dies ist ein leistungsfähiger Desktop-Computer, der die Daten lokal selbst verarbeitet.
- **Thin Client:**  
ist in Abhängigkeit mit einem Server, um die Aufgaben verarbeiten zu können. Thin Client hat nur eine Rolle, nämlich Anfragen an den Server weiterzuleiten und die vom Server übermittelten Daten auszuwerten. Kein Installationsaufwand, da er über einen Webbrowser läuft.

## 3 Analyse

In diesem Kapitel der Arbeit wird zunächst dem gegenwärtigen Stand der Technik Rechnung getragen. Danach geht es um die Anforderungsanalyse der Webanwendung. Dazu wird eine allgemeine Struktur festgelegt, wie die Arbeit systematisch aufgebaut sein soll. Anschließend wird der aktuelle Zustand (Ist-Analyse) des Projektes ermittelt und anhand dieser Ist-Analyse erfolgen die funktionalen und nicht-funktionalen Anforderungen an die zu entwickelnde Webanwendung.

### 3.1 Stand der Technik

Bei der Suche nach öffentlich zugänglichen Tools für die Durchführung von Workshops wurden folgenden Ergebnisse gefunden:

#### 3.1.1 IdeaBoardz

IdeaBoardz<sup>1</sup> ist eine freie webbasierte Anwendung zum Brainstorming, Erstellen einer ToDo-Liste oder zur Retrospektive im agilen Projektmanagement. Mit diesem Tool ist eine Zusammenarbeit möglich. Die beteiligten Personen können entweder zeitgleich oder zu verschiedenen Zeiten ortsunabhängig auf das gemeinsame Dokument zugreifen und bearbeiten. In Echtzeit zusammenarbeiten, ist bei IdeaBoardz nicht realisierbar. Das hat zur Folge, dass bei der Veränderung des Zustands keine sofortige Aktualisierung der Benutzeroberfläche erfolgt. IdeaBoardz wird unter anderem bei der Brainstorming-Methode wie die 6-Hüte-Methode<sup>2</sup> von De Bono und auch für die Ideenbewertung bekannte SWOT<sup>3</sup>-Analyse angewendet. Die Registrierung ist optional, so dass der Nutzer auch IdeaBoardz verwenden kann, ohne sich anzumelden.

---

<sup>1</sup>vgl. <https://ideaboardz.com/>

<sup>2</sup>engl. Six Thinking Hats

<sup>3</sup>steht für Strengths-Weaknesses-Opportunities-Threats-Analyse

**IdeaBoardz** brainstorm, retrospect, collaborate

**How**

- Create my own IdeaBoard
- Share the url

Name \*

Description \*

Format  
2 Sections

Section Title \*

☐ Ich bin kein Roboter. reCAPTCHA  
Datenschutzerklärung · Nutzungsbedingungen

Create

Test Drive...  
Lets take it for a Spin!

Feedback  
• via Twitter  
• via Board

483931 Boardz

Best viewed on  
chrome, firefox or safari

Abbildung 3.1: Erstellen eines eigenen IdeaBoards

Die **Abbildung 3.1** zeigt, wie ein IdeaBoard zu erstellen ist. Neben dem Namen des Boards werden das Thema (Description) und Formate (Format) benötigt. Es können bis zu 10 Sektionen gewählt werden und es stehen außerdem noch weitere Formate zur Verfügung, wie Pro und Contra, ToDo-Liste, Six Thinking Hats und vieles mehr. Anschließend wird ein Titel für die jeweilige Sektion eingegeben.

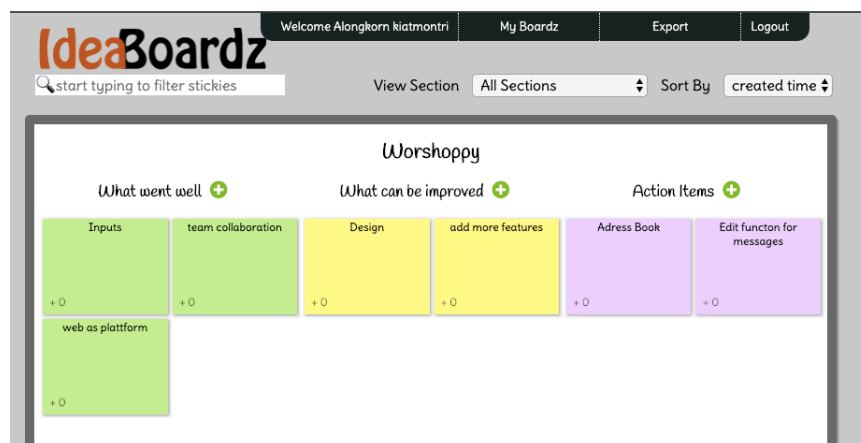


Abbildung 3.2: Darstellung von Sektionen

Wie in **Abbildung 3.2** zu sehen ist, sind die Eingaben in Sektionen strukturiert und farbig sortiert. Das Thema steht in der Mitte. Jede Sektion hat einen Titel und einen Plus-Button. Mit diesem Button können zu jeder Sektion neue Eingaben hinzugefügt werden. Die Eingaben werden als Karteikarten bzw. Notizzettel visualisiert. Der weiße Hintergrund kann wie ein Whiteboard oder eine Pinnwand gesehen werden. Die Eingaben können auch von beteiligten Personen abgestimmt werden. Es ist auch möglich,

die Daten nach Datum oder Abstimmungen sortieren zu lassen.

Als weiteres Feature lassen sich die Sektionen einzeln darstellen (**Abbildung 3.3**). Die Suche nach dem Eingabeinhalt und das Exportieren der Ergebnisse sowohl als PDF-Datei als auch in ein Excel-Dokument werden ebenfalls bei dieser Webanwendung angeboten. Die Daten können sowohl innerhalb als auch außerhalb der Sektion zusammengeführt (merge) werden. Ebenso können die Daten per Drag & Drop aus einer anderen Sektion zugeordnet werden, wie in **Abbildung 3.4** zu sehen ist. Mit dem Teilen der URL kann das jeweilige IdeaBoardz für die Zusammenarbeit freigegeben werden.

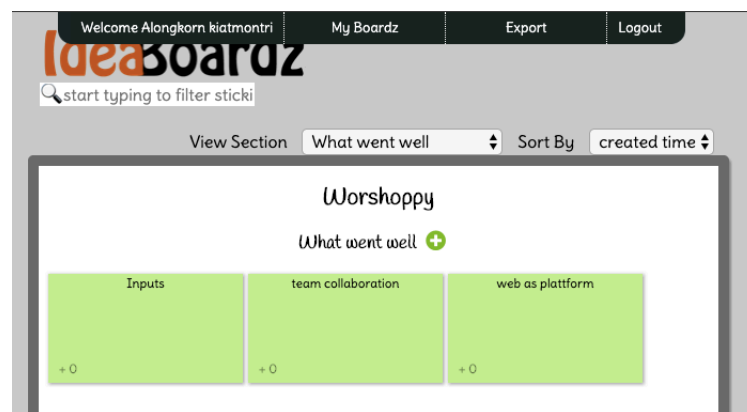


Abbildung 3.3: Darstellung einer der Sektionen

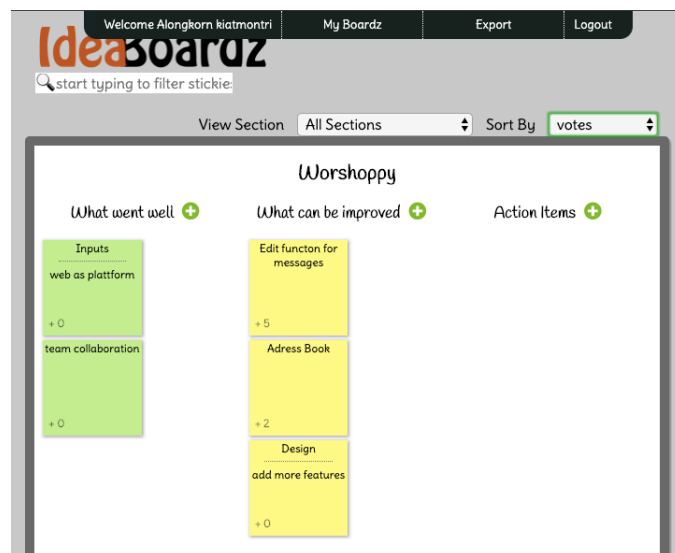


Abbildung 3.4: Zusammenführen und Zuordnen von Daten



Einige der oben dargestellten Features können für diese Arbeit übernommen werden. Zu nennen sind:

- Die Eingaben wie ein Notizzettel oder Karteikarten visualisieren.
- Die Ergebnisse als PDF-Datei exportieren.
- Eingaben in Sektion darstellen.
- Zuordnung von Daten (per Drag & Drop).

Mit welchen Webtechnologien IdeaBoardz entwickelt wurde, lässt sich anhand der Informationen auf der Webseite nicht erkennen. Man kann aber davon ausgehen, dass es sich bei IdeaBoardz um eine webbasierte Anwendung mit reichlich Interaktionen auf der Benutzeroberfläche handelt, d.h. es ist über einen Webbrowser nutzbar und der Nutzer muss nichts installieren. Dementsprechend gehört IdeaBoardz zu einer Thin Client-Anwendung und zählt auch zu Rich Internet Applications sowie Web 2.0-Anwendung. (siehe Kapitel 2).

### 3.1.2 Miro-RealtimeBoard

Miro<sup>4</sup> ist eine dynamische Webanwendung und es handelt sich dabei um ein kollaboratives Online-Whiteboard in Echtzeit. Um das Online-Whiteboard nutzen zu können, wird ein Account benötigt. Dafür muss man sich bei Miro registrieren. Miro bietet die kostenlose Version an, sie ist für bis zu drei Teammitglieder und drei Boards erlaubt.

Begonnen wird mit einer leeren Seite oder man verwendet eine von Miro bereitgestellten Vorlage. Zur Vorlage gehören unter anderem MindMap, Flowchart, Brainwriting und Concept Map. Einfügen neuer Dateien, Bilder und Dokumenten aus Google Drive oder vom Rechner ist auch möglich, um Informationen auszutauschen. Der Nutzer kann virtuelle Notizen erstellen. Die Notizen lassen sich nach Farbe unterscheiden und können per Drag & Drop über das komplette Board verschoben werden. Mit Hilfe von Share-Button vereinfacht Miro die Teilen-Funktion über eine URL oder einen Gmail-Account das ortsunabhängige und kollaborative Arbeiten in Echtzeit. Somit können die beteiligten Personen beispielsweise während des Brainstormings auf die Ideen der anderen eingehen und kommentieren. Außerdem können die Benutzer das Whiteboard in eine Präsentation umwandeln oder als eine PDF-Datei exportieren.

Miro ist ebenfalls gut geeignet zur Umsetzung eines Brainstormings (**Abbildung 3.5**). Die Ideen werden in Form von Notizen erstellt. Zusammenfassend können die Notizen per Drag & Drop nach Farben kategorisiert werden.

---

<sup>4</sup>vgl. <https://miro.com/>



Abbildung 3.5: Realisieren eines Brainstormings mit Hilfe von Miro

Folgende Erkenntnisse wurden bei der Analyse von Miro gefunden und werden für das Workshoppy-Projekt übernommen:

- AJAX-Anwendung
- Thin Client-Anwendung
- Web 2.0-Anwendung
- Rich Internet Applications

### 3.1.3 MindMap

MindMap<sup>5</sup> zählt auch zu den Favoriten unter den Kreativitätstechniken und wird häufig in vielen Workshops als Methode zur Ideenfindung und -strukturierung eingesetzt. Man kann sie beispielsweise auch für das Brainstorming, die Projektplanung oder Ideensammlung verwenden.

Bei einer Mindmap werden Begriffe und deren zugehörige Beziehungen grafisch dargestellt. Das Hauptthema oder das Schlüsselwort befindet sich als Knoten kreisförmig in der Mitte. Um das Thema herum wird alles in Form von Hauptästen notiert. Man schreibt auf jeden Hauptast ein Schlüsselwort auf. Verbunden werden sie zum Hauptthema mit Linien. Die Hauptäste bilden die ersten Gedankengänge. Von jedem Hauptast zweigen weitere Nebenäste mit Begriffen ab (**Abbildung 3.6**).

<sup>5</sup>wird häufig auch Mindmapping genannt und versteht sich als Gedankenlandkarte

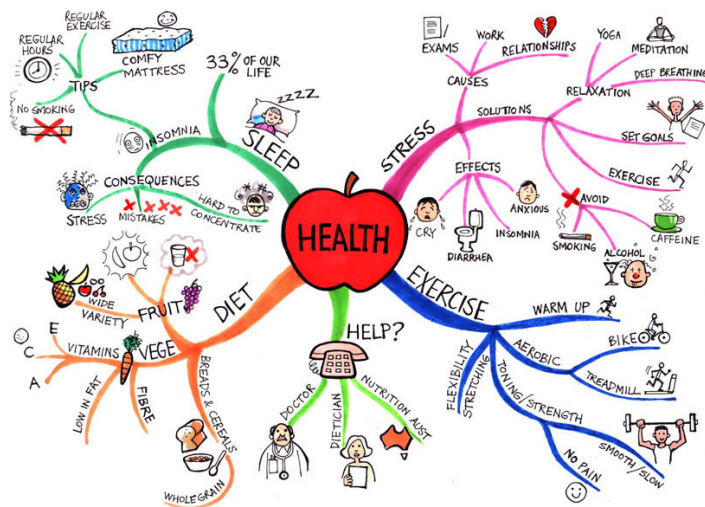


Abbildung 3.6: Health mindmap

**Quelle:** Learning Fundamentals: Student Study Techniques by Jane Genovese, Figure 4: Health mindmap. Online im Internet: URL: <https://learningfundamentals.com.au/resources/>

Es existieren heutzutage bereits mehrere webbasierte Mindmapping-Tools sowohl kostenlos als auch kostenpflichtig auf dem Markt. Einer von diesen ist, wie bereits im vorherigen Abschnitt vorgestellt, das **Miro** (siehe Abschnitt 3.1.2), mit dem man Ideen visualisieren und in Echtzeit zusammenarbeiten kann. Die Übersicht zu den anderen Mindmapping-Tools kann auf dieser Webseite<sup>6</sup> verfolgt werden. Neben diesen beiden dargestellten Tools, **IdeaBoardz** und **Miro**, gibt es keine weiteren nennenswerte Anwendungen. Zwar gibt es noch zahlreiche Brainstorming-Tools, die in diesem Artikel<sup>7</sup> aufgelistet. Jedoch sind sie meisten visuell gleich und unterscheiden sich im funktionalen Bereich wenig. Von daher geben sie keine Anreize für diese Arbeit.

<sup>6</sup><https://t3n.de/news/mind-mapping-online-tools-568258/>

<sup>7</sup><https://tallyfy.com/brainstorming-tools/>

## 3.2 Erkennbare Stärken und Schwächen der Konkurrenz-Tools

### 3.2.1 IdeaBoardz:

- **Stärken:**

- **Zusammenfügen von Daten per Drag & Drop:**

- Die Daten können sowohl innerhalb als auch außerhalb einer Sektion per Drag & Drop zusammengeführt werden.

- **Vote-Funktion:**

- Den Benutzern wird eine Schaltfläche geboten, mit der sie die Möglichkeit haben, Ihr Gefallen für Inhalte von anderen Usern oder von sich selbst auszudrücken. Vergleichbar mit einem Like-Button<sup>8</sup> auf Social Media Plattformen.

- **Exportieren:**

- Die Ergebnisse können sowohl als PDF- oder auch als Excel-Datei exportiert werden.

- **Benutzerfreundlichkeit:**

- Die Webanwendung ist übersichtlich dargestellt, hat eine klare Strukturierung. Sie bietet außerdem eine einfache und verständliche Navigation, hat keinen unnötigen Ballast, wie z.B. Bilder, lange Texte. Sie beinhaltet außerdem kontrastreiche Farben. Die Benutzer erreichen das Ziel mit wenig Aufwand (Klick, Zeit).

- **Kein Schulungsaufwand:**

- Die Webanwendung ist sehr verständlich und leicht zu bedienen. Der Benutzer kommt ohne Schulung gut an sein Ziel.

- **Ohne Registrierung und nicht kostenpflichtig:**

- IdeaBoardz ist eine kostenlose Webanwendung. Für die Anwendung ist keine Registrierung nötig.

- **Responsive Webdesign:**

- Das Layout der Webseite ist flexibel gestaltet, dass dieses auf dem Tablet und Smartphone eine gleichbleibende Benutzerfreundlichkeit bietet. Der Inhalt der Webseite wird auf dem mobilen Gerät einheitlich wie auf dem Laptop oder Desktop-Computer dargestellt.

- **Schwächen:**

- **Keine Möglichkeit in Echtzeit zusammenzuarbeiten:**

- Eine der größten Nachteile von dieser Webanwendung ist, dass sie die Daten nicht in Echtzeit liefern kann. Die Whiteboards können nicht in Echtzeit aktualisiert werden, somit verlaufen die Brainstorming-Sitzungen mit etwas Verzögerung.

---

<sup>8</sup>Gefällt-mir Knopf

- **Keine Möglichkeit Thema oder Titel zu editieren:**  
Das behandelte Thema und die Titel der Sektionen können nach dem Erstellen nicht mehr geändert werden.
- **Löschen eines erstellten IdeaBoards und von Sektionen nicht möglich:**  
Sektionen und das erstellte IdeaBoard können nicht gelöscht werden.
- **Daten können nur einmal zusammengeführt werden:**  
Beim ersten Zusammenführen sind die Daten fest geordnet, d.h. es ist unmöglich, sie einzeln wieder zu trennen oder mit anderen Daten zusammenzuführen.

### 3.2.2 Miro-RealtimeBoard:

- **Stärken:**

- **Benutzerfreundlichkeit:**  
Die Webanwendung hat eine klare Übersicht sowie ein modernes Layout. Sie hat eine klare Strukturierung und bietet eine einfache und verständliche Navigation. Die Werkzeuge sind gut erkennbar, gut strukturiert und verständlich. Die Benutzer erreichen ihr Ziel mit wenig Aufwand.
- **Kein Schulungsaufwand:**  
Die Webanwendung ist sehr verständlich und leicht zu bedienen. Die Benutzer kommen ohne große Bemühungen gut an ihr Ziel.
- **Exportieren:**  
Miro stellt dem Benutzer die Möglichkeit zur Verfügung, die Ergebnisse in verschiedene Formate zu exportieren. Die Ergebnisse können sowohl als PDF- oder auch als CSV- sowie als JPEG-Datei exportiert werden.
- **Zusammenarbeit in Echtzeit:**  
Mit Hilfe von Share-Button vereinfacht Miro die Teilen-Funktion über einen URL oder einen Gmail-Account das ortsunabhängige und kollaborative Arbeiten der Teammitglieder in Echtzeit.
- **Präsentationsmodus:**  
Die Ergebnisse können in einem Präsentationsmodus verwandelt werden.
- **Chatfunktion:**  
Die Teammitglieder können sich mittels einer eingebauten Chat-Funktion Nachrichten untereinander austauschen.
- **Hochladen von Dateien:**  
Einfügen neuer Dateien, Bilder und Dokumente aus Google Drive oder vom Rechner ist auch möglich.

- **Kommentar in Echtzeit hinzufügen:**

Durch der eingebauten Kommentarfunktion ist es möglich, das Feedback der Mitglieder in Echtzeit zu erhalten, um die Qualität der Inhalte verbessern zu können.

- **Verschiedene Vorlagen:**

Der Benutzer hat die Möglichkeit, verschiedene Vorlagen, wie Mindmapping, User Story-Map, Flowchart, Concept-Map, Brainwriting sowie Wireframing zu verwenden.

- **Responsive Webdesign:**

Das Layout der Webseite ist flexibel gestaltet. Das einheitliche Anzeigen von Inhalten wird auf allen Endgeräten (Laptop, Tablet, Smartphone) gewährleistet. Somit kann der Inhalt gänzlich und schnell vom Benutzer aufgenommen werden.

- **Schwächen:**

- **Registrierung notwendig:**

Bei dieser Webanwendung ist ein Account notwendig. Der Benutzer muss sich bei Miro registrieren.

- **Begrenzte Funktion bei der kostenlosen Version:**

Die kostenlose Version ist auf bis zu drei Mitglieder und drei Boards erlaubt. Die Funktionen ist bei der kostenlosen Version begrenzt. Ein Upgrade auf 40 \$ pro Monat bringt zwei weitere Teammitglieder, unbegrenzte Boards sowie Funktionen.

Die **Tabelle 3.1** stellt zusammenfassend die Funktionsüberblick der beiden Konkurrenz-Tools vor.

Funktion	IdeaBoardz	Miro
In Echtzeit zusammenarbeiten	Nein	Ja
Präsentationsmodus	Nein	Ja
Hochladen von Dateien	Nein	Ja
Vote-Funktion	Ja	Nein
Benutzerfreundlichkeit	Ja	Ja
kostenlos	volle Funktionen	begrenzte Funktionen und Mitglieder
Registrierung	Nein	Ja
Exportieren in andere Formate, z.B. in PDF-Datei	Ja	Ja
Responsive	Ja	Ja
Schulungsaufwand	Nein	Nein
Verschiedene Vorlagen, z.B. Mind Map	Nein	Ja

Tabelle 3.1: Funktionsüberblick der beiden Konkurrenzen.

### 3.3 Projektstruktur

Im folgenden wird der geplante Projektablauf in Form eines Projektstrukturplans<sup>9</sup> dargestellt:

Auf der obersten Ebene steht das Projekt. Eine Ebene darunter die Teilprojekte oder Teilaufgaben, darunter schließlich die Arbeitspakete. Der Projektstrukturplan (**Abbildung 3.7**) entspricht dem typischen sequentiellen Vorgehensmodell zur Softwareentwicklung einschließlich der Entwicklung der Webanwendung.

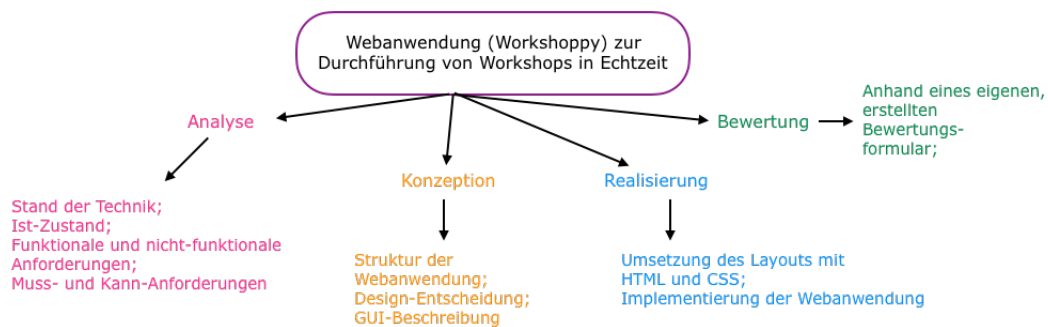


Abbildung 3.7: Projektstrukturplan

**Quelle:** eigene Abbildung

Die Konkurrenz-Tools und eine Analyse von funktionalen, nicht-funktionalen Anforderungen sowie die Muss- und Kann-Anforderungen werden in der ersten Phase untersucht. Durch diese Analyse wird die Struktur und ein passendes Layout der Webanwendung erstellt. Die daraus entstehende Designentscheidung wird technisch in eine Webanwendung umgesetzt und am Ende wird die Webanwendung anhand eines selbst erstellten Bewertungsformulars von den Mitarbeiter im Unternehmen (mindestens 2 Personen) bewertet.

<sup>9</sup>Nach Definition der DIN 69901-5:2009 ist der Projektstrukturplan die „[...] vollständige hierarchische Darstellung aller Elemente (Teilprojekte, Arbeitspakete) der Projektstruktur als Diagramm oder Liste.“

## 3.4 Ist-Analyse

Bei der Projektvorstellung wurde in der Firma zunächst über den Zustand der aktuellen Lösungen für die Durchführung von Workshops gesprochen.

Wie der Abschnitt **Stand der Technik (Abschnitt 3.1)** aufgezeigt hat, existieren bereits zahlreiche webbasierte Tools für die Durchführung von Workshops.

Nach der Betrachtung der aktuellen Lösungen fällt das Fazit des Unternehmens folgendermaßen aus: Die vorhandenen Tools reichen noch nicht aus, um Workshops effektiv durchzuführen. Während **IdeaBoardz 3.1.1** keine Zusammenarbeit in Echtzeit bieten kann, hat **Miro 3.1.2** bei der kostenlosen Version eine begrenzte Anzahl an Teammitgliedern, d.h. Workshops mit mehr als drei Teilnehmern muss deshalb die kostenpflichtige Version verwendet werden. Außerdem bereiten die aktuellen Lösungen viel Mühe in puncto Dateneingabe. Besonders auf dem Smartphone-Bildschirm ist die Dateneingabe sehr fummelig und nicht komfortabel genug. Das Smartphone scheint für eine Arbeit mit den aktuellen Lösungen noch nicht geeignet zu sein.

Als weitere und oft eingesetzte Methode zum Brainstormen in den Workshops ist das Mindmapping. Es ist eine Form, die beim Brainstorming entstehenden Ideen bildlich zu strukturieren. Jedoch hat die Mindmapping-Methode auch ihre Nachteile. Man muss sich zunächst an diese Form der Aufzeichnung gewöhnen. Denn MindMaps sehen auf den ersten Blick unübersichtlich und verschachtelt aus. Diese können sehr schnell ihre Übersichtlichkeit verlieren, wenn verschiedene Schlüsselwörter in Beziehung stehen. Demzufolge ist die Akzeptanz der Nutzer, die mit dieser Methode bzw. dieser Aufzeichnung nicht vertraut sind, eher gering. Es ist außerdem sehr zeitaufwendig, eine Mindmap exakt nach den Regeln zu erstellen. Mindmaps sind eher für den individuellen Gebrauch geeignet, da die verwendeten Schlüsselbegriffe und die Strukturierungen häufig für andere Personen unverständlich sind.

Es lässt sich nicht verhindern, dass eine neue Lösung benötigt wird, um das aktuelle Problem zu lösen und vor allem die Durchführung von Workshops effektiver zu gestalten.

## 3.5 Unternehmensanforderungen

In diesem Abschnitt werden die Unternehmensanforderungen an die zu entwickelnde Lösung besprochen.

Die neue Lösung soll eine benutzerfreundliche Webanwendung sein, die intuitiv bedienbar sein soll. Der Moderator übernimmt die Rolle des Administrators und ist hauptverantwortlich für die Steuerung der Webanwendung. Das Brainstorming soll in einer Sitzung (Session) durchgeführt werden. In dieser Sitzung wird versucht, möglichst viele Ideen für ein zuvor klar definiertes Problem zu produzieren. Ein Workshop



sollte sich nicht nur auf eine Sitzung beschränken. Es sollte möglich sein, in einem Workshop mehrere Brainstorming-Sitzungen abwickeln zu können.

Die teilnehmenden Personen wiederum sind nur für die inhaltlichen Beiträge zuständig. Die neue Lösung sollte so entwickelt werden, dass die Teilnehmer besonders auf Ihren Mobilgeräten ohne mühevollen Tipperei und ohne großen Aufwand Ihre Ideen abgeben können. Die Ideen sollten dann in Echtzeit für alle sichtbar dargestellt werden, welche nach der Sammlungsphase digital vom Moderator durch ein einfach zu bedienendes User Interface zusammengefasst werden können/müssen.

Die weiteren Hauptkriterien für die zu entwickelnde Webanwendung sind:

- Login-Bereich (Moderator)
- QR-Code zur Teilnahme am Workshop (Teilnehmer)
- Thin Client-Anwendung
- Rich Internet Applications
- Responsive Webdesign
- Browserunabhängigkeit
- Export der Ergebnisse in eine PDF-Datei

## 3.6 Anforderungsanalyse

Dieses Kapitel umfasst die grundlegenden Anforderungen dieser Bachelorarbeit. Die Anforderung wird in funktionalen und nicht-funktionalen Anforderungen aufgeteilt.

Eine funktionale Anforderung wird nach der Definition aus dem Buch [Balzert, 2010] die gewünschte Funktionalität des Systems bzw. eines Produkts beschrieben. Die nicht-funktionalen Anforderungen sind Anforderungen, die für die Nutzung des Systems wichtig sind. Außerdem werden Muss- und Kann-Anforderungen formuliert, welche für das Projekt oberste Priorität haben und welche eher zweitrangig sind.

### 3.6.1 Funktionale Anforderungen

Aus den Unternehmensanforderungen (**Abschnitt 3.5**) lassen sich folgende funktionale Anforderungen ableiten.

1. **Der Moderator soll sich über ein Anmeldeformular anmelden können.**

Eine dem Moderator bekannte URL führt auf die Willkommenseite der Webanwendung. Dort

wird er über ein Anmeldeformular aufgefordert, seinen Benutzernamen und Passwort einzugeben. Das System vergleicht die Eingabe mit der in der Datenbank angelegten Nutzerdaten. Gelingt die Anmeldung, wird auf die Hauptseite weitergeleitet. Wenn dem System die Anmeldung nicht bekannt ist, wird eine entsprechende Fehlermeldung angezeigt und die Willkommenseite verbleibt. Für die vorliegende Arbeit werden die Nutzerdaten manuell in einem Datenbanksystem erstellt. Der Registrierungsvorgang wird für die zukünftige Weiterentwicklung im letzten Kapitel festgehalten.

**2. Der Moderator soll sich mittels eines Navigationselementes abmelden können.**

Jede moderne Webanwendung bietet dem eingeloggten Nutzer die Möglichkeit, sich ordnungsgemäß auszuloggen. Mit dem Element „Ausloggen“ in der Navigationsleiste kann sich der Moderator abmelden und er wird zur Willkommenseite weitergeleitet.

**3. Der Moderator soll neue Workshops erstellen, sie bearbeiten und löschen können.**

Die moderierende Person soll die Möglichkeit haben, neue Workshops anzulegen. Die erstellten Workshops werden in eine Liste angezeigt und sollten von dem Moderator bearbeitet und gelöscht werden können. Die Workshops sollten in einer Datenbank gespeichert werden.

**4. In einem Workshop sollen eine oder mehreren Sitzungen (Sessions) für Ideenfindung und -sammlung erstellt werden können.**

Eine Session versteht sich als eine Sitzung, um Lösungen für Problemstellung zu generieren, verschiedene Themen aufzuarbeiten oder Entwicklung neuer Geschäftsideen sowie Innovationen zu fördern. Der Moderator soll in einem Workshop eine oder mehreren Sitzungen (Sessions) erstellt können. Er soll auch die erstellten Sessions auch bearbeiten und löschen können. Die Sessions sollten ebenso in einer Datenbank gespeichert werden.

**5. Die teilnehmenden Personen sollen über einen QR-Code oder eine Einladungsmail an dem jeweiligen Workshop mitwirken können.**

Zu Beginn des Workshops sollte ein QR<sup>10</sup>-Code mittels Beamer angezeigt werden, sodass die anwesenden Teilnehmer diesen mit ihren Mobilgeräten einscannen und an diesem Workshop mitwirken können. Der Moderator sollte auch die Möglichkeit haben, auch während einer Sitzung den QR-Code einblenden zu können. Die Einladung zur Teilnahme am Workshop sollte ebenfalls auch per Mail gesendet werden können.

**6. Die Teilnehmer sollen auf ihren Endgeräten Ihre Ideen abgeben können.**

Sobald eine Sitzung eines Workshops gestartet ist, sollte auf den Endgeräten der Teilnehmer zunächst ein Eingabefeld für den Benutzernamen erscheinen. Dort werden sie aufgefordert, ihren Benutzernamen einzugeben. Nach der Eingabe des Benutzernamens wird eine Textarea für die Dateneingabe freigeschaltet. Über diese sollen die Teilnehmer Ihre Gedanken und Vorschläge frei äußern können. Außerdem sollte es den Teilnehmern möglich sein, Ihren Benutzernamen zu ändern.

---

<sup>10</sup>englisch: Quick Response

**7. Die Dateneingabe der Teilnehmer sollen mittels Beamer in Echtzeit angezeigt werden können.**

Die eingegebenen Daten der Teilnehmer sollen in Echtzeit auf der Präsentation-Seite, welche parallel mittels Beamer läuft, präsentiert werden.

**8. Der Moderator soll die Daten in Kategorien zusammenfassen können.**

Die Ideensammelungsphase ist beendet und der Moderator wird anschließend mit der Gruppe die Ergebnisse auswerten und sortieren. Es soll dem Moderator erlaubt sein, die gesammelten Ergebnisse direkt auf der Präsentation-Seite per Drag & Drop in Kategorien zusammenzufassen. Die erstellten Kategorien sollten bearbeitet sowie gelöscht werden können. Sie sollten auch in einer Datenbank gespeichert werden.

**9. Beim Löschen von Kategorien sollen die darin befindlichen Daten nicht betroffen sein.**

Beim Löschen einer nicht leeren Kategorie, sollen die darin befindlichen Daten erhalten bleiben. Es wird nur die Kategorie gelöscht.

**10. Die Ergebnisse sollen als eine PDF-Datei exportiert und heruntergeladen werden können.**

Nach Beendigung des Workshops sollen die Daten digital von allen Sitzungen zusammengefasst und als PDF-Datei heruntergeladen werden können. Der beendete Workshop sollte dann in einer separaten Liste archiviert werden.

### **3.6.2 Nicht-funktionale Anforderungen**

Im oberen Unterkapitel wurden die funktionalen Anforderungen aufgelistet. In diesem Unterkapitel werden die nicht-funktionalen Anforderungen formuliert, welche zu diesem Projekt gehören sollen.

▪ **Layout, Handhabung und Benutzbarkeit**

Gemessen am Funktionsumfang sollte die zu entwickelnde Anwendung ein möglichst strukturiertes, einfaches und bedienerfreundliches Layout besitzen. Beim Entwurf und der Entwicklung der Anwendung sollten deshalb die folgenden Punkte beachtet werden:

- Die Verwendung der Webanwendung soll für Nutzer intuitiv sein. Der Nutzer soll mit wenigem Aufwand, ohne besondere Schulung und in kurzer Zeit durch die Webanwendung navigieren sowie sie verwenden und die wichtigen Funktionen der Webanwendung ausführen können.
- Bereitstellung von Hilfeleistung in Form von Hilfetexten und Tooltips zur Förderung der intuitiven Bedienbarkeit.
- Die Buttons sollten in unterschiedlichen Farben entsprechend der Funktionalität gestaltet werden.
- Anzeigen von Bestätigungsdialogen beim Löschen von Workshops, Sessions und Kategorien sowie beim Beenden von Workshops.

- Die Gestaltung der Webanwendung sollte einheitlich nach vorgegebenen Designvorlagen vom Unternehmen erfolgen.

- **Plattformübergreifend**

Die Webanwendung soll unabhängig der Plattform funktionieren. Deshalb sollte die Webanwendung nach **responsive Webdesign** gestaltet werden. Das bedeutet, die Inhalts- und Navigationselemente sowie der strukturelle Aufbau der Webanwendung sollten sich der Bildschirmauflösung aller Endgeräte anpassen. Somit ist es für den Nutzer möglich, diese Anwendung auf verschiedenen Endgeräten zu betreiben.

- **Browserunabhängigkeit**

Außer der Plattformunabhängigkeit sollte die Anwendung in unterschiedlichen Browsern, wie Firefox oder Chrome genutzt werden können.

- **Thin Client-Anwendung**

Die Webanwendung soll eine Thin Client-Anwendung sein, dies bedeutet, dass die Anwendung nicht mehr auf jedem Client installiert werden muss, sondern über einen Webbrowser nutzbar ist. Das Updaten von Programmen sowie applikationsspezifische Funktionalitäten werden von Server zur Verfügung gestellt. Da alles über den Webbrowser abläuft, ist eine Thin Client-Anwendung hardware-, sprach- und betriebssystemunabhängig. Für JavaScript- bzw. AJAX-Anwendungen müssen keine weiteren Plug-Ins installiert werden, da die meisten Browser JavaScript unterstützen.

- **Performance**

Die eingegeben Daten seitens der Teilnehmer sollten in Echtzeit auf der Präsentation-Seite angezeigt werden.

### 3.6.3 Muss- und Kann-Anforderungen

Die funktionalen Anforderungen sowie nicht-funktionalen Anforderungen wurden bereits im Unterkapitel 3.6.1 und 3.6.2 dargestellt. In diesem Abschnitt werden die Muss- und Kann-Anforderungen formuliert. Die Muss-Anforderung wird mit Priorität „Hoch“ gekennzeichnet, für die Kann-Anforderung wird die Priorität auf „Niedrig“ gesetzt.

<b>Merkmal</b>	<b>Anforderung</b>	<b>Priorität</b>
FA	Erstellen, Bearbeiten und Löschen von Workshops	<b>Hoch</b>
FA	Auflisten von Workshops	<b>Hoch</b>
FA	Archivieren und Anzeigen von beendeten Workshops	<b>Hoch</b>
FA	Anmeldeformular für die Moderation	<b>Hoch</b>
FA	Der Moderator muss sich ausloggen können	<b>Hoch</b>
FA	Erstellen, Bearbeiten und Löschen von Sessions	<b>Hoch</b>
FA	QR-Code für die Teilnahme am Workshop	<b>Hoch</b>
FA	Die Teilnehmer werden zu Beginn des Workshop aufgefordert, einen Benutzernamen einzugeben	<b>Hoch</b>
FA	Benutzernamen ändern	<b>Hoch</b>
FA	Dateneingabefunktion	<b>Hoch</b>
FA	Dateneingabe in Echtzeit auf der Präsentationsseite darstellen	<b>Hoch</b>
FA	Erstellen, Bearbeiten und Löschen von Kategorien	<b>Hoch</b>
FA	Beim Löschen von Kategorien sollen die darin befindlichen Daten erhalten bleiben	<b>Hoch</b>
FA	Daten in Kategorien zusammenfassen	<b>Hoch</b>
FA	Die Ergebnisse sollen in eine PDF-Datei exportiert und heruntergeladen werden können	<b>Hoch</b>
NFA	Responsive Webdesign	<b>Hoch</b>
NFA	Browserunabhängigkeit	<b>Hoch</b>
NFA	Performance	<b>Hoch</b>
NFA	Anzeigen von Bestätigungsdialogen beim Löschen von Workshops, Sessions und Kategorien sowie beim Beenden von Workshops	<b>Hoch</b>
NFA	Thin Client-Anwendung	<b>Hoch</b>
FA	Einladungsmail zur Teilnahme am Workshop	<b>Niedrig</b>
NFA	Bereitstellung von Hilfeleistung in Form von Hilfetexten und Tooltips zur Förderung der intuitiven Bedienbarkeit	<b>Niedrig</b>
NFA	Die Buttons sollen in unterschiedlichen Farben entsprechend der Funktionalität gestaltet werden	<b>Niedrig</b>
NFA	Die Gestaltung der Webanwendung soll einheitlich nach vorgegebenen Designvorlagen vom Unternehmen erfolgen	<b>Niedrig</b>

Tabelle 3.2: Muss- und Kann-Anforderungen  
 FA = Funktionale Anforderung  
 NFA = Nicht-funktionale Anforderung

## 3.7 Beispielszenario

In diesem Abschnitt wird die zu entwickelnde Webanwendung anhand eines Beispielszenarios näher beschrieben.

Man stelle sich folgende Situation vor: Sie führen ein Unternehmen und suchen für ein Problem eine Lösung. Hier sind also die Ideen gefragt. Sie laden alle Abteilungsleiter in den Besprechungsraum ein und führen dazu einen Workshop, um die Lösungsansätze zu erarbeiten. Beim Erarbeiten von Ergebnissen spielt dabei die hierarchische Position im Unternehmen keine Rolle. In einem Workshop ist jeder „gleich“.

Sie übernehmen die Rolle eines Moderators und erläutern den teilnehmenden Personen das zu behandelnde Problem, die Regeln sowie das Ziel des Workshops. Als Werkzeug für die Durchführung des Workshops steht Ihnen die zu entwickelnde Webanwendung zur Verfügung. Bevor Sie mit dem Workshop beginnen, legen Sie bei der Webanwendung einen Workshop und die dazugehörige Session an.

In dieser Session findet die Ideenfindungsphase statt. Jeder der im Besprechungsraum anwesenden Teilnehmer scannt den angezeigten QR-Code ein, um an diesem Workshop teilnehmen zu können. Danach kommt jeder dran und gibt auf seinem Endgerät Ideen für die Lösung des Problems ein. Jede eingebrachte Idee wird mittels Beamer in Echtzeit präsentiert. Die Ideenfindungsphase ist vorüber. Sie als Administrator beenden die Eingabephase auf der Webanwendung. Hier dürfen die anwesenden Teilnehmer keine Ideen mehr eingeben. Sie haben jetzt die Aufgabe, die gesammelten Ideen gemeinsam mit der Gruppe zu analysieren und anschließend zusammenzufassen. Am Ende des Workshops stehen Ihnen die Ergebnisse als digitale Dokumentation zur Verfügung.

## 4 Design

Aus den gesammelten Anforderungen vom **Kapitel 3** wird es in diesem Kapitel um den Entwurf der Weboberfläche (WebUI) gehen. Die grundlegende Gestaltungsrichtlinie darunter die einheitliche Verwendung von Icons, Farben und Schriftgestaltung werden dabei beschrieben. Außerdem wird der Aufbau der zu entwickelnden Webanwendung mithilfe von Mockups dargestellt.

„Gutes Design ist so wenig Design wie möglich Weniger Design ist mehr, konzentriert es sich doch auf das Wesentliche, statt die Produkte mit Überflüssigem zu befrachten. Zurück zum Puren, zum Einfachen!“ - Dieter Rams (vgl. 10 Thesen für gutes Design<sup>1</sup>)

### 4.1 Gestaltungsrichtlinie

Um der Webanwendung ein einheitliches, strukturiertes und benutzerfreundliches Design zu geben ist es erforderlich, feste Layoutvorgaben zu definieren. Sie sorgt für eine verständliche und intuitiv bedienbare Benutzeroberfläche, sodass eine positive Empfindung der Nutzer bei der Bedienung der Webanwendung hervorgerufen wird und ohne größere Einarbeitungszeit beherrscht werden kann. Diese Vorgabe wird auf alle Seiten der Webanwendung angewendet.

#### 4.1.1 Farben

Um die Webanwendung übersichtlich zu halten, wird auf die Verwendung mehrerer Farben verzichtet. Es werden grundsätzlich die Farben grau und weiß verwendet. Die Gestaltung von Buttons und Daten werden wiederum mit bunten Farben gestaltet, damit sie optisch auffallend sind.

#### 4.1.2 Schriftgestaltung

Bei der Schriftgestaltung, wie Schriftgröße und Schriftart, ist darauf zu achten, dass sie eine gute Lesbarkeit und ein modernes Aussehen bieten. Deshalb wird eine Schriftart verwendet, bei der es sich um eine serifenlose Schrift handelt.

---

<sup>1</sup><https://www.vitsoe.com/de/ueber-vitsoe/gutes-design>

### 4.1.3 Icons

Die Buttons werden mit Icons gestaltet, um Inhalte schneller zu verstehen und den Nutzen der Funktionen zu verdeutlichen. Hierbei handelt es sich um allgemein übliche und bei mobilen Anwendungen bekannte Icons. Sie sind so gewählt, dass sich der Nutzer bei den grundlegenden Funktionen auf der Webanwendung schnell zurechtfindet. Bei den Icons handelt es sich hierbei um keine Grafiken, sondern um eine sogenannte Icon Fonts, welche über eigenes Stylesheet geladen werden. Die Fonts haben vor allem den Vorteil, dass sie auf jede Bildschirmgröße skaliert werden können. Für diese vorliegende Arbeit werden die Icon Fonts von Font Awesome<sup>2</sup> verwendet.



Abbildung 4.1: Verwendete Icon Fonts

**Quelle:** <https://fontawesome.com/icons?d=gallery>

---

<sup>2</sup>vgl. <https://fontawesome.com/>



## 4.2 Konzeption

In diesem Abschnitt werden die erstellten Mockups, also die Wireframes der verschiedenen Seiten dargestellt und erläutert. Die definierte Gestaltungsrichtlinie sollte hierbei bei der Konzeption eingehalten werden.

### 4.2.1 Mockup der Willkommenseite

Wenn die richtige URL aufgerufen wird, erscheint die Willkommenseite mit der Aufforderung zur Eingabe von Benutzernamen und Passwort (**Abbildung 4.2**). Zur Erinnerung, die Benutzerdaten werden in dieser Arbeit manuell in der Datenbank angelegt. Der Registrierungsvorgang wird in dieser Arbeit nicht vorhanden sein. Dieser wird für die zukünftige Weiterentwicklung festgehalten.

Die Formulardaten werden über den Anmelde-Button verschickt, welcher ein Ereignis (Event) auslöst, um den richtigen User zu authentifizieren. Falls der User nicht existiert, wird eine Fehlermeldung auf der Seite ausgegeben (**Abbildung 4.3**).



The mockup shows a login form for 'Workshoppy'. At the top, it says 'Anmelden'. Below this is a blue instruction box: 'Bitte geben Sie Ihren Benutzernamen und Ihr Passwort ein'. There are two input fields: 'Benutzernamen' and 'Passwort'. At the bottom is a grey 'Anmelden' button.

Abbildung 4.2: Mockup für die Anmeldung



The mockup shows the login form with an error message. At the top, it says 'Anmelden'. Below this is a red error box: 'Benutzername oder Passwort waren ungültig.'. There are two input fields: 'Benutzernamen' (containing 'eeee') and 'Passwort' (containing 'dedededed'). At the bottom is a grey 'Anmelden' button.

Abbildung 4.3: Mockup für den Anmelungsfehler

### 4.2.2 Mockup der Hauptseite

Nach der erfolgreichen Anmeldung wird der Nutzer, der für die Durchführung des Workshops verantwortlich ist, auf die Hauptseite weitergeleitet (**Abbildung 4.3**). Auf dieser Seite sind drei grundlegende Bereiche (Nr.1, Nr.2 und Nr.3) zu sehen:

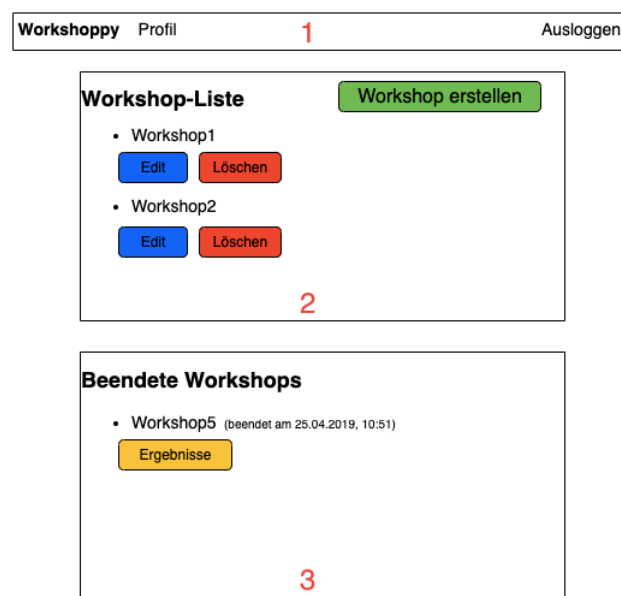


Abbildung 4.4: Mockup der Hauptseite

#### 1. Header:

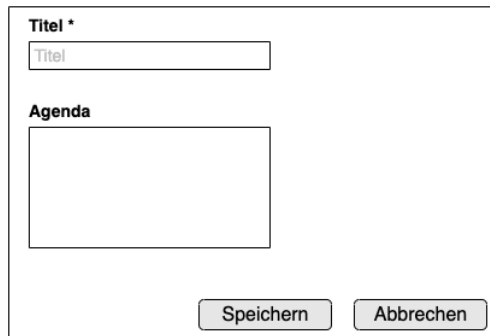
Im Header befindet sich die Navigationsleiste. In dieser sind drei Navigationselemente enthalten.

- **Workshopy:**  
Navigiert den Nutzer zur Hauptseite.
- **Profil:**  
Die angemeldete Person kann unter dem Profil seine Daten verwalten, wie z.B. seine Accountdaten (Benutzername, Passwort) ändern, den Account löschen und seine hinterlegten Personendaten anzeigen lassen.
- **Ausloggen:**  
Ermöglicht dem Nutzer, sich ordnungsgemäß von der Webanwendung auszuloggen.

#### 2. Workshop-Liste:

In diesem Bereich werden die erstellten Workshops aufgelistet. Mit dem „Workshop erstellen“-Button kann ein neuer Workshop erstellt werden. Durch das Anklicken des Edit- sowie Löschen-Buttons kann der Workshop gezielt bearbeitet und gelöscht werden.

Die **Abbildung 4.5** zeigt das Erstellen eines neuen Workshops. Der Titel ist ein Pflichtfeld und muss beim Erstellen angegeben werden. Als Option steht ein Textbereich für die Agenda zur Verfügung.



The mockup shows a form with a title field labeled 'Titel \*' and an agenda field labeled 'Agenda'. At the bottom are 'Speichern' and 'Abbrechen' buttons.

<b>Titel *</b> <input type="text" value="Titel"/>
<b>Agenda</b> <div></div>
<div>Speichern Abbrechen</div>

Abbildung 4.5: Mockup für das Erstellen eines neuen Workshops

Beim Titel des Workshops handelt es sich um ein Linktext. Beim Anklicken wird der Moderator zur **Controller-Seite** dieses Workshops geführt.

3. Beendete Workshops:

Die beendeten Workshops werden in diesem Bereich archiviert. Der Ergebnisse-Button führt zur **Ergebnisse-Seite** des archivierten Workshops.

### 4.2.3 Mockup der Controller-Seite

Jeder Workshop hat seine eigene Controller-Seite. Die **Abbildung 4.6** zeigt beispielsweise die Controller-Seite von „Workshops1“. Auf dieser sind der Titel des Workshops (Nr.1), die Navigation-Tabs (Nr.2) und die Session-Liste (Nr.3) zu sehen.



Abbildung 4.6: Mockup der Controller-Seite

Es sind drei Navigations-Tabs (Nr.2) vorhanden:

1. Das Navigation-Tab „WS-Controller“ beinhaltet vier folgende Buttons:

- Client-Button:  
öffnet die **Teilnehmer-Seite** als neues Browser-Tab. Auf dieser Seite können die Teilnehmer die Dateneingabe tätigen.  
**Anmerkung:** Der Client-Button wird in der zukünftigen Weiterentwicklung entfernt, da der Moderator nicht für die Dateneingabe beteiligt werden darf. Für diese Arbeit wird der Client-Button aufgrund des Funktionstests erstmal erhalten bleiben.
- Präsentation-Button:  
öffnet als neues Browser-Fenster die **Präsentation-Seite**. Mittels Beamer präsentiert sie den Teilnehmern die eingegebenen Daten in Echtzeit.
- Ergebnisse-Button:  
öffnet ein neues Browser-Tab und ruft die **Ergebnisse-Seite** auf. Die Ergebnisse des Workshops werden dargestellt. Der Ergebnisse-Button ist erst aktiviert, wenn die Ergebnisse vorhanden sind.

- Beenden-Button:  
beendet den laufenden Workshop und leitet den Moderator zur **Hauptseite** weiter. Der Workshop wird anschließend in „Beendete Workshops“ archiviert (**Abbildung 4.4**).
- 2. Die Agenda, falls sie vorhanden ist, wird im Navigation-Tab „Agenda“ dargestellt.
- 3. Neben dem Einscannen des QR-Codes auf der Präsentation-Seite (**Abbildung 4.13**) können die Teilnehmer im Navigation-Tab „Teilnehmer“ die Einladung per Mail senden lassen, um am Workshop teilzunehmen.

The mockup shows a user interface for a workshop. At the top, there's a header bar with 'Workshopy Profile' on the left and 'Ausloggen' on the right. Below this, the main content area is titled 'Workshop "Workshop1"'. Under the title, there are three tabs: 'WS-Controller', 'Agenda', and 'Teilnehmer', with 'Teilnehmer' being the active tab. The content of the 'Teilnehmer' tab is titled 'Einladungsmail zur Teilnahme am Workshop "Workshop1"'. It contains a text input field labeled 'Email Address' and a blue button labeled 'Abschicken'.

Abbildung 4.7: Mockup für das Navigation-Tab „Teilnehmer“

Das Brainstorming wird in der Session-Liste (Nr.3) in **Abbildung 4.6** durchgeführt. Zunächst muss der Moderator mit dem „Session Erstellen“-Button eine neue Session anlegen (**Abbildung 4.8**).

The mockup shows a form for creating a new session. It has two input fields: 'Titel' and 'Frage \*'. The 'Frage \*' field is marked as required. At the bottom of the form, there are two buttons: 'Speichern' and 'Abbrechen'.

Abbildung 4.8: Mockup für das Erstellen einer neuen Session

Die behandelte Frage, die auf der **Teilnehmer- und Präsentation-Seite** zu sehen sein wird, muss definiert werden. Als Option kann der Titel der Session angegeben werden. Wie viele Sessions in einem Workshop benötigt werden, das entscheidet der Moderator selbst. Er kann unbegrenzt viele Sessions erstellen.

Neben jeder Session sind in **Abbildung 4.6** drei Buttons zu sehen.

1. Edit-Button:

Mit diesem Button kann die Titel- sowie Fragenänderung durchgeführt werden.

2. Löschen-Button:

Der Löschen-Button löscht die Session inklusive ihrer zugehörigen Daten.

3. Starten-Button:

Es wird erst „gebrainstormt“, wenn die Session gestartet ist. Während die Session läuft, darf sie nicht bearbeitet und gelöscht werden. Alle Buttons von nicht aktiven Sessions werden auch in dieser Phase deaktiviert. Es kann nur eine Session gestartet werden. Außerdem kann der Workshop bei laufender Session nicht beendet werden. Der „Beenden“-Button in WS-Controller (**Abbildung 4.9**) wird deshalb deaktiviert.

Es gibt zusätzlich noch zwei weiteren Buttons, welche erst sichtbar werden, wenn eine Session gerade läuft. Das ist der „Eingabe beenden“- und „Session Beenden“-Button (**Abbildung 4.9**).

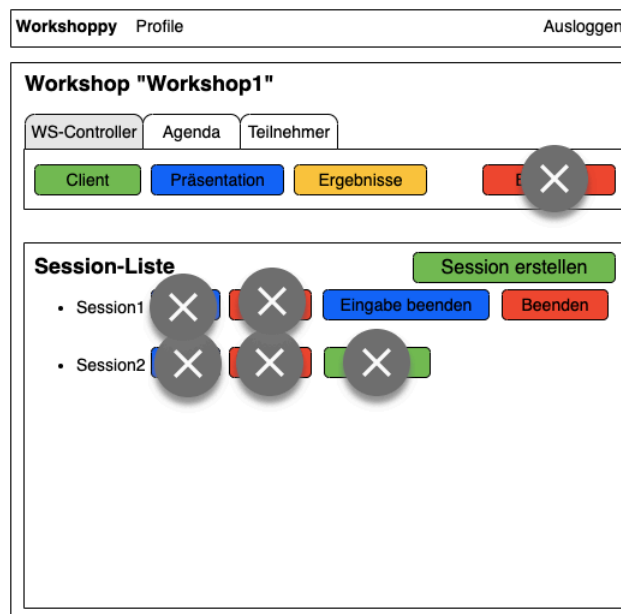


Abbildung 4.9: Mockup für die aktive Session

Der „Eingabe beenden“-Button stoppt die Eingabe auf der Teilnehmer-Seite (**Abschnitt 4.2.4**). Demzufolge können die Teilnehmer keine weiteren Daten mehr eingeben. Der „Session Beenden“-Button beendet die gerade laufende Session. Auf der **Teilnehmer-Seite** wird durch den Klick auf dem „Session Beenden“-Button der Infotext „Bitte Warten“ angezeigt und gleichzeitig wird

der QR-Code auf der **Präsentation-Seite** dargestellt (**Abbildung 4.13**). Erst nach dem Beenden einer laufenden Session werden alle zuvor deaktivierten Buttons wieder reaktiviert.

#### 4.2.4 Mockup der Teilnehmer-Seite

Um auf diese Seite zu kommen, müssen die Teilnehmer den QR-Code entweder über Ihre Mobilgeräte auf der Präsentation-Seite (**Abschnitt 4.2.5**) einscannen oder sie lassen sich per Mail die Einladung zur Teilnahme am Workshop zusenden. Die Teilnehmer-Seite stellt jedem Workshop-Teilnehmer die Dateneingabefunktion zu einer gestarteten Session bereit. Beim Aufrufen der Seite werden die Teilnehmer zunächst aufgefordert, ihren Benutzernamen einzugeben (**Abbildung 4.10**).

The mockup shows a rectangular frame with a title 'Willkommen bei Workshopy' at the top center. Below the title, the text 'Geben Sie Ihren Namen ein:' is displayed. Underneath this text is a text input field containing the placeholder 'Dein Name...'. To the right of the input field is a blue button with the text 'Speichern'.

Abbildung 4.10: Mockup für Eingabe der Benutzernamen

Nach Eingabe ihres Benutzernamens werden die Teilnehmer auf die Eingabefunktion weitergeleitet. Die **Abbildung 4.11** beschreibt, dass die Teilnehmer-Seite gerade auf das Kommando des Moderators wartet. Sobald er eine Session startet, werden die Teilnehmer für die Funktionen zur Dateneingabe freigeschaltet (**Abbildung 4.12**).

The mockup shows a rectangular frame with a header bar at the top containing the text '"Benutzername"' on the left and 'Ausloggen' on the right. The main area of the frame contains the text 'Bitte warten' centered.

Abbildung 4.11: Mockup für die Anzeige der Infotext

The mockup shows a rectangular frame with a header bar at the top containing the text '"Benutzername"' on the left and 'Ausloggen' on the right. The main area of the frame has a title 'Session-Frage' at the top. Below the title is a text input field with the placeholder 'Bitte gib deine Antwort ein...'. At the bottom of the main area is a blue button with the text 'Abschicken'.

Abbildung 4.12: Mockup für die Eingabefunktion

In der Navigationsleiste in **Abbildung 4.11** sowie in **Abbildung 4.12** sind zwei Navigationselemente zu sehen. Der Benutzername ist der Name des Teilnehmers, welcher zuvor eingegeben wurde. Das Ausloggen

erlaubt dem Teilnehmer, seinen Benutzernamen zu ändern. Die Session-Frage in **Abbildung 4.12** ist die eigentliche Frage, welche in dieser Session behandelt wird. Die Textarea dient der Dateneingabe. Beim Klick auf den Abschicken-Button wird die Dateneingabe in Echtzeit auf der **Präsentation-Seite** dargestellt.

#### 4.2.5 Mockup der Präsentation-Seite

Die Präsentation-Seite dient der Darstellung der Dateneingabe von allen Teilnehmern in Echtzeit. Die Seite hat zwei Zustände, nämlich passiv und aktiv.

- passiver Zustand:  
Dieser Zustand bedeutet, dass momentan keine Session läuft. Die Präsentation-Seite zeigt bei diesem Zustand den QR-Code zur Teilnahme am Workshop an. Bevor die Session tatsächlich beginnt, können die Teilnehmer den QR-Code über Ihre Mobilgeräte einscannen, um am Workshop teilzunehmen (**Abbildung 4.13**). Falls es eine Agenda zu dem Workshop gibt, wird sie neben dem QR-Code dargestellt (**Abbildung 4.14**).

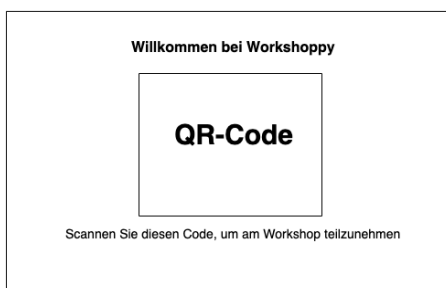


Abbildung 4.13: Mockup für das Anzeigen eines QR-Codes

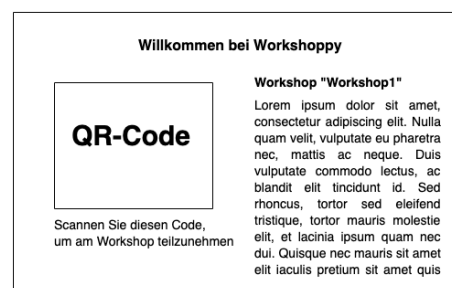


Abbildung 4.14: Mockup für das Anzeigen eines QR-Code und einer Agenda



- aktiver Zustand:

Die Präsentation-Seite befindet sich im aktiven Zustand, wenn die Session gestartet ist. Auf der Präsentation-Seite werden die Dateneingaben der Teilnehmer in Echtzeit präsentiert (**Abbildung 4.15**).



Abbildung 4.15: Mockup für die Darstellung der Dateneingabe auf der Präsentation-Seite

Die behandelte Frage (Nr.1) ist am oberen Inhaltsbereich zu sehen. Die Eingaben der Teilnehmer (Nr.2) werden wie ein Notizzettel visualisiert. Ein Notizzettel besteht aus zwei Teilen, dem Namen des Teilnehmers und seine Idee. Am unteren Bereich der Präsentation-Seite befinden sich zwei Buttons (Nr.3). Während die Session läuft, kann der QR-Code des Workshops mit dem Drücken des QR-Code-Buttons angezeigt werden. Wenn der QR-Code-Button getätigt wird, wandelt er sich anschließend im „QR-Code ausblenden“-Button um. Die Session wird dabei nicht beendet und kann mit dem Drücken des „QR-Code ausblenden“-Button wieder auf dem Zustand kommen, wie auf der **Abbildung 4.15** dargestellt ist. Der QR-Code kann sowohl im passiven Zustand (**Abbildung 4.13**) als auch im aktiven Zustand dargestellt werden. Der Vollbild-Button wandelt die Präsentation-Seite in den Vollbildmodus um.

Um die eingegebenen Daten auf der Präsentation-Seite zusammenfassen zu können, muss die Ideensammlungsphase beendet werden. Dafür klickt der Moderator auf den „Eingabe beenden“-Button auf der Controller-Seite, wie in **Abbildung 4.9** zu sehen ist. Daraus folgt, dass auf der **Teilnehmer-Seite** die Eingabe gestoppt und stattdessen der Infotext „Bitte Warten“ angezeigt wird. Durch den Klick auf den „Eingabe beenden“-Button auf der **Controller-Seite** wird der „Kategorie erstellen“-Button auf der Präsentation-Seite freigeschaltet, mit dem der Moderator Kategorien erstellen kann (**Abbildung 4.16**).

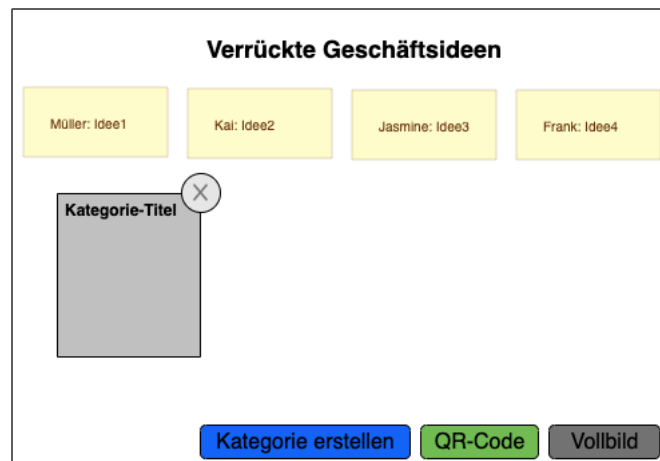


Abbildung 4.16: Mockup für das Zusammenfassen von Daten auf der Präsentation-Seite

Direkt auf der Präsentation-Seite kann der Moderator die Daten mit der Drag-&-Drop-Funktion in Kategorien zusammenfassen. Die Kategorien selbst lassen sich nicht verschieben. Das Löschen einer Kategorie erfolgt mit dem Klick auf dem X-Button. Es wird nur die Kategorie gelöscht, d.h. die darin befindlichen Daten bleiben auf der Präsentation-Seite erhalten.

#### 4.2.6 Mockup der Ergebnisse-Seite

Mit Klick auf den Ergebnisse-Button in **Abbildung 4.4** sowie in **Abbildung 4.6** wird die Ergebnisse-Seite des Workshops als neues Browser-Tab geöffnet. Auf dieser Seite befinden sich die Daten inklusive Kategorien des Workshops. Die Ergebnisse-Seite beinhaltet außerdem einen Button, mit dem die Ergebnisse als eine PDF-Datei heruntergeladen werden können (**Abbildung 4.17**).



Abbildung 4.17: Mockup für die Darstellung der Ergebnisse des Workshops

### 4.2.7 Zusammenfassung der Konzeption

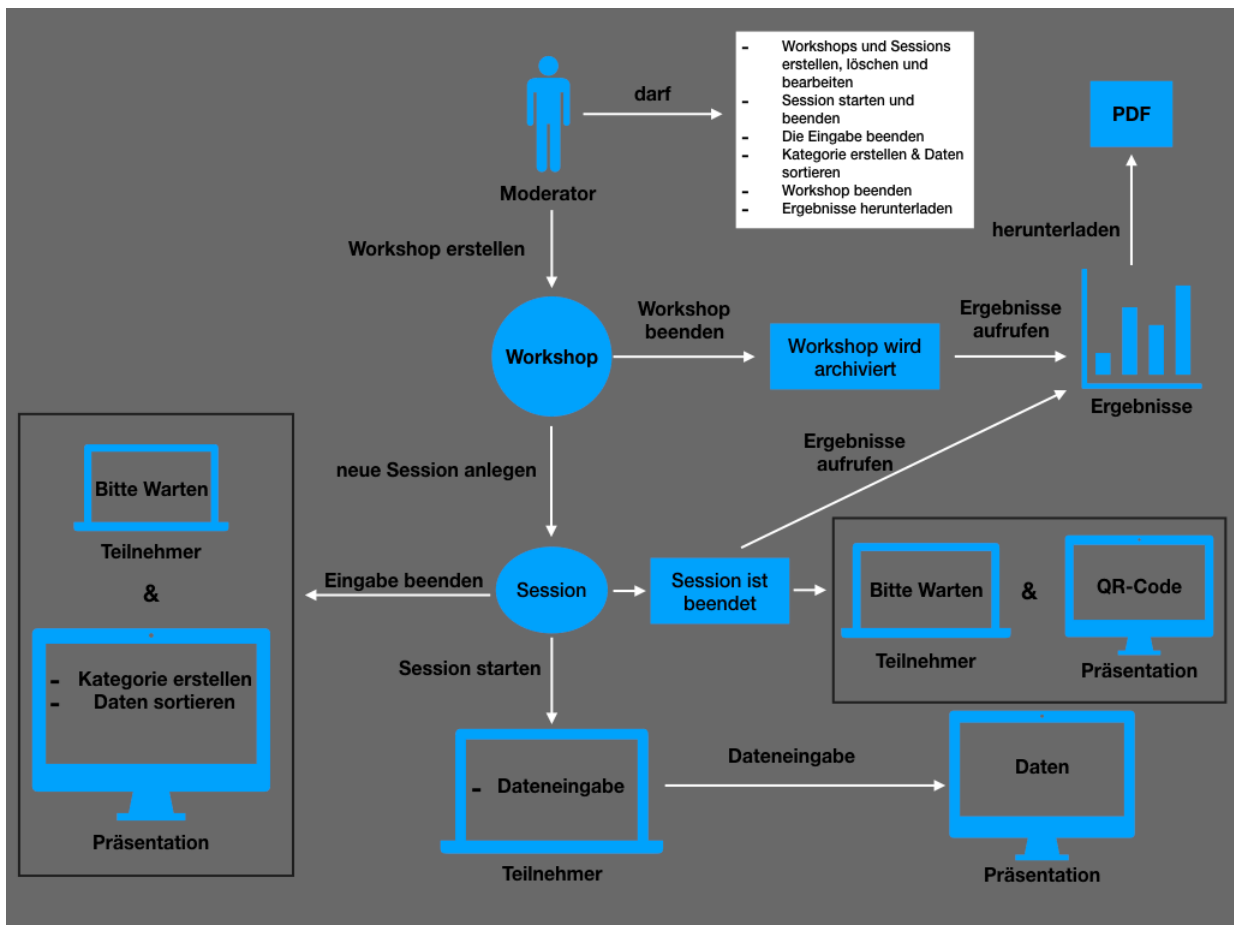


Abbildung 4.18: Zusammenfassung der Konzeption der zu entwickelnden Webanwendung

# 5 Implementierung

Dieses Kapitel beschreibt die Implementierung der zuvor konzipierten Webanwendung anhand einiger Codebeispiele. Zunächst werden die verwendeten Webtechnologien vorgestellt. Danach folgt der Abschnitt Datenmodellierung. In diesem werden das Datenbankschema und die Datenbankstruktur, welche durch ein ER-Modell festgelegt wird, dargestellt. Weiter geht es mit der Architektur und Struktur der Webanwendung. Dabei wird in diesem Abschnitt das MVC-Entwurfsmuster behandelt. Anschließend wird die eigentliche Implementierung, welche in serverseitig und clientseitig aufgeteilt sind, beschrieben.

## 5.1 Verwendete Webtechnologien

In diesem Unterkapitel werden die verwendeten Webtechnologien erläutert.

### 5.1.1 MidCom CMS

Symfony<sup>1</sup> ist eines der weltweit größten PHP Frameworks und wird von führenden Open-Source Projekten eingesetzt, wie zum Beispiel Drupal oder Wordpress. Das hauseigene Webframework MidCom CMS basiert auf der Symfony Umgebung, dies gibt die Architektur einer Webanwendung vor und stellt eine Vielzahl an fertig implementierten Modulen bereit, die den Programmieraufwand reduzieren.

### 5.1.2 HTML - Hypertext Markup Language

„Webseiten bestehen erst einmal aus reinem Text. Eine HTML-Seite wird nicht programmiert, es gibt keine Funktionen, die bei der Erfüllung bestimmter Bedingungen ausgeführt werden, wie das berühmte if-else in Programmiersprachen. Für eine Webseite wird erst einmal ein Text geschrieben und dieser Text wird dann mit HTML kodiert, das heißt, die logischen Bestandteile eines textorientierten Dokuments werden beschrieben. Dadurch werden die einzelnen Bestandteile des Textes zu HTML-Elementen (wie Überschriften, Absätze, List usw.). HTML beschreibt nicht, wie ein Element aussieht oder wo es platziert ist, HTML beschreibt, was ein Element ist - eine Überschrift, eine Liste, ein Bild, eine Tabelle usw. Es geht einzig und allein um die Struktur eines Dokuments, nicht um seine Präsentation“ [Schulz, 2008]

---

<sup>1</sup><https://symfony.com/>

HTML ist der grundlegende Bestandteil einer Webseite. Es verwendet "Markup" (Begriff für Textauszeichnung), um ein Dokument semantisch zu strukturieren. Um die Auszeichnung des Contents auf dem Webbrowser darzustellen, werden Markup-Elemente, so genannte Tags, verwendet. Diese Elemente sind innerhalb einer Auszeichnungssprache wie HTML standardisiert. Tags werden in spitzen Klammern notiert. Der Inhalt wird durch ein einleitendes und ein abschließendes Tag markiert.

### 5.1.3 CSS - Cascading Stylesheets

„Cascading Stylesheets (CSS) sind die Formatierungssprache für Webseiten. Der Einsatz von CSS bietet viele Vorteile, denn CSS ermöglicht eine Trennung von Inhalt und Layout: Damit lassen sich per CSS gestaltete Webseiten besser warten und leichter aktualisieren. Möchten Sie beispielsweise alle Überschriften eines Webprojekts in einer anderen Farbe ausgeben lassen, so müssen Sie bei einem konsequenten Einsatz von CSS die Änderung nur an einer Stelle, in einer von den (X)HTML-Dateien separaten Datei, vornehmen. Wird diese Formatierung hingegen direkt in (X)HTML realisiert, sind Änderungen bei jeder einzelnen Überschrift in den Dateien selbst durchzuführen.“  
[Maurice, 2011]

Während HTML den Aufbau bzw. die Struktur der Webseite bestimmt, ist CSS für das Aussehen der Elemente zuständig.

### 5.1.4 Bootstrap

Um ein einheitliches Erscheinungsbild für die Webanwendung zu erhalten, verwende ich das kostenlose und quelloffene Frontend-Framework namens Bootstrap. Das Framework stellt verschiedene HTML- und CSS-Gestaltungsvorlagen für Formulare, Buttons, Tabellen, Navigation, Typografie sowie das Grid-System für Layouts zur Verfügung. Durch ein JavaScript-Modul ist es möglich, die Interaktionen auf der Webanwendung einzubinden. Außerdem bietet Bootstrap alle Voraussetzungen für ein responsives Webdesign, um die Webanwendung auf allen Geräten (Desktop, Smartphone und Tablet) einheitlich und optimal darzustellen.

### 5.1.5 PHP - Hypertext Preprocessor

PHP ist eine weit verbreitete Programmiersprache speziell für die Entwicklung dynamischer Webanwendungen. PHP ist plattformübergreifend und lässt sich in HTML einbinden. Anders als JavaScript ist PHP eine serverseitige Programmiersprache. Demzufolge läuft der PHP-Code auf dem Server und die fertige Webseite wird vom Server an den Benutzer ausgeliefert. Die ausgelieferte Webseite enthält keinen Programmiercode, sondern nur HTML-Code, welcher vom Browser verarbeitet und dargestellt wird.<sup>2</sup>

---

<sup>2</sup>vgl. <https://www.php.net/manual/de/intro-what-is.php>

### 5.1.6 phpMyAdmin

phpMyAdmin ist eine kostenlose Software für das Verwalten von MySQL-Datenbanken. Es werden Datenbanken zur Verfügung gestellt, in denen Inhalte in Tabellen abgespeichert werden. phpMyAdmin stellt eine grafische Benutzeroberfläche bereit, um Datenbanken ohne MySQL-Kenntnisse zu verwalten.

### 5.1.7 JQuery UI

JQuery UI ist die Erweiterung von JQuery (**Abschnitt 2.1.10**) und bietet unabhängig vom Webbrowser die Funktionalität zur Gestaltung interaktiver Benutzeroberflächen. (vgl. [Augsten, 2019])

Unter anderem stellt JQuery UI folgende Interaktionen zur Verfügung:

- Sortable: Einträge innerhalb einer Liste sortieren.
- Draggable: Elemente innerhalb eines Bereiches verschieben.
- Droppable: Elemente verschieben und ablegen, wie Drag & Drop.
- Selectable: Ein Element oder mehrere Elemente selektieren.

Um die Daten auf der Präsentation-Seite in Kategorien zusammenfassen zu können, wird dafür die Funktion „Sortable“ angewendet.

### 5.1.8 Ratchet und Autobahn|JS

#### ▪ Ratchet - WebSockets for PHP:

Bei Ratchet<sup>3</sup> handelt es sich um eine PHP-Bibliothek für eine WebSocket-Anwendung. WebSocket ist eine bidirektionale und vollduplexe Kommunikation zwischen einem Webbrowser und Server (**Abschnitt 2.1.8**).

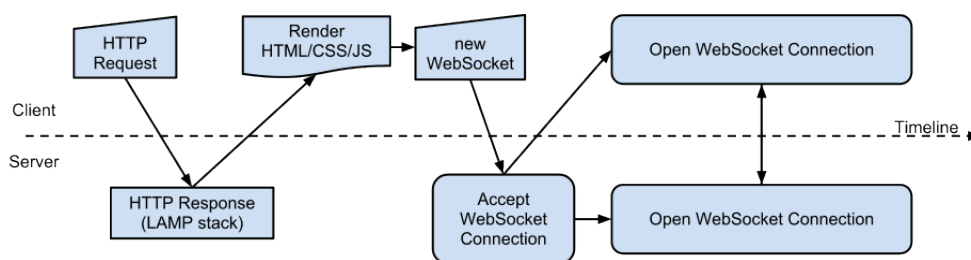


Abbildung 5.1: Ratchet-Workflow  
**Quelle:** <http://socketo.me/docs/flow>

<sup>3</sup><http://socketo.me/docs/>

Sobald eine WebSocket-Verbindung erfolgreich aufgebaut ist, können Server und Client gleichzeitig miteinander kommunizieren.

- **Autobahn|JS:**

Autobahn|JS ist ein Teilprojekt von Autobahn Projekt<sup>4</sup> und bietet eine Open-Source Implementierung von Web Application Messaging Protocol<sup>5</sup> (WAMP) in JavaScript.

Für die Umsetzung der Webanwendung werden sowohl Ratchet als auch Autobahn|JS zum Einsatz kommen. Ratchet wird als WebSocket-Server verwendet und Autobahn|JS als Client, der mit dem Ratchet kommuniziert.

## 5.2 Datenmodellierung

Ein gut strukturierter Datenbestand in Form einer relationalen Datenbank ist die Grundvoraussetzung der zu realisierenden Webanwendung. Als Werkzeug für die Verwaltung von Datenbanken kommt das freie Administration-Backend phpMyAdmin (**Abschnitt 5.1.6**) zum Einsatz. Als erster Schritt der Datenmodellierung wird die Datenbankstruktur mit einem ER-Modell festgelegt. Durch ein ER-Modell kann eine Datenbankstruktur vor der Programmierung geplant und verbessert werden.

### 5.2.1 ER-Modell

Ein ER-Modell besteht grob aus drei Teilen:

- **Entität:**  
Objekte aus der realen Welt und gleichzeitig der Tabellenname.
- **Attribut:**  
Eigenschaft der Entität. Das entspricht der Spalte in der Tabelle. Ein Attribut besteht aus einfachen Datentypen wie integer oder char.
- **Beziehung:**  
Beziehung zwischen Entitäten. Dabei gibt es 1:1, 1:n und n:m Beziehungen.

---

<sup>4</sup><https://crossbar.io/autobahn/>

<sup>5</sup>[urlhttps://github.com/crossbario/autobahn-js](https://github.com/crossbario/autobahn-js)

Die **Abbildung 5.2** visualisiert die Datenbanktabellen. Aus Gründen der Übersichtlichkeit wird bewusst auf die Darstellung der Attribute verzichtet.

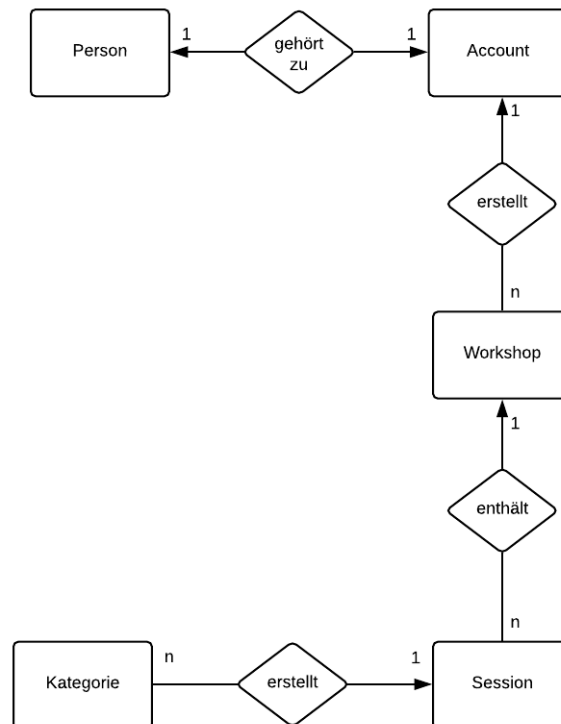


Abbildung 5.2: ER-Modell der Webanwendung

**Quelle:** eigene Abbildung

Eine Person gehört genau zu einem User-Account und ein User-Account gehört zu einer Person. Ein User-Account kann mehrere Workshops erstellen, aber ein Workshop kann höchstens von einem User-Account erstellt werden. Ein Workshop kann mehrere Sessions enthalten, wobei eine Session von genau einem Workshop zugeordnet wird. In einer Session können mehrere Kategorien erstellt werden, aber eine Kategorie wird höchstens von einer Session erstellt.



### 5.2.2 Datenbankschema

Vom ER-Modell (**Abbildung 5.2**) wird das Datenbankschema abgeleitet. Die **Abbildung 5.3** gibt eine Übersicht über die einzelnen Tabellen mit ihren Attributen sowie die Beziehungen. Für die Durchführung von Workshops werden drei Tabellen (workshop, session und category) benötigt. Der User-Account wird in die midgard\_user-Datenbank gespeichert, welche in einer Beziehung zu der Person-Tabelle steht.

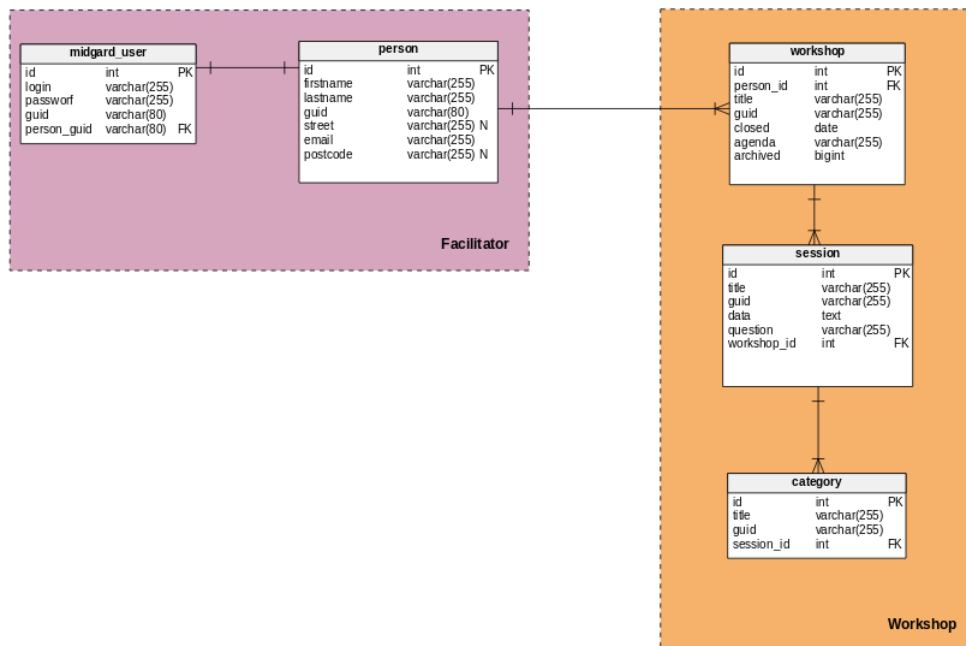


Abbildung 5.3: Datenbankschema

**Quelle:** eigene Abbildung

## 5.3 Architektur der Webanwendung

Die Architektur der zu entwickelnden Webanwendung ist angelehnt an das **Model-View-Controller (MVC)** Entwurfsmuster. Daher wird dieses in diesem Unterkapitel erläutert und dargestellt.

„Das MVC-Konzept wurde 1979 zunächst für Benutzeroberflächen in Smalltalk durch Trygve Reenskaug beschrieben (Seeheim-Modell), der damals an Smalltalk im Xerox PARC arbeitete. Es gilt mittlerweile aber als De-facto-Standard für den Grobentwurf vieler komplexer Softwaresysteme, teils mit Differenzierungen und oftmals mehreren jeweils nach dem MVC-Muster aufgeteilten Modulen.“ [Wikipedia, 019d]

Das Ziel dieses Entwurfsmusters ist es, eine interaktive Anwendung in drei Komponenten mit unterschiedlichen Aufgaben zu unterteilen. (vgl. [Stal, 1996])

- **Model (M)**

Das Model enthält die Anwendungsdaten, die dem Benutzer mithilfe der View repräsentiert werden. Bei einer Änderung des Models werden alle zugehörigen View über die Änderung benachrichtigt, so dass die Daten auch auf der View-Komponente aktualisiert werden.

- **View (V)**

Die View stellt dem Benutzer die Daten aus dem Model dar. Für das Model kann es mehrere Views geben. Dies hat vor allem den Vorteil, dass die Views andere Ausgaben generieren können.

- **Controller (C)**

Der Controller ist sowohl mit der View als auch mit dem Model verbunden. Sie reagiert auf Benutzeraktionen, manipuliert sie im Model und aktualisiert selbst in einigen Fällen auch die View. Jede Veränderung von Daten informiert der Controller dem Model und das Model wiederum setzt die View über die Datenänderung in Kenntnis, so dass die Daten aktualisiert werden können.

Die **Abbildung 5.4** stellt grafisch die Kommunikationsverbindung der drei Komponenten in einem MVC-Entwurfsmuster dar.

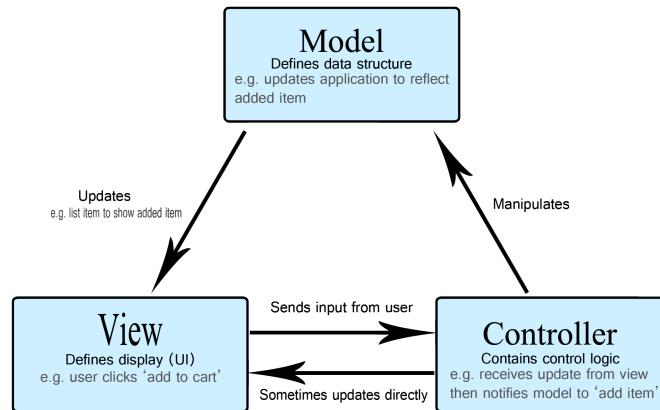


Abbildung 5.4: Model-View-Controller Entwurfsmuster

**Quelle:** <https://developer.mozilla.org/en-US/docs/Glossary/MVC>

Durch die nicht mehr so starke Kopplung der Komponenten, gewährleistet das MVC-Entwurfsmuster eine höhere Flexibilität im Programmdesign. Außerdem hat Herr Steffen D. in seinem Blog folgende Vorteile über das MVC-Entwurfsmuster aufgeführt:

„Vorteile finden sich zum einen in der Möglichkeit, dass ein und dasselbe Datenmodell in mehreren Ansichten repräsentiert werden kann. Des Weiteren sind alle Ansichten, die sich auf ein Modell beziehen, automatisch synchronisiert und es können Bedienelemente beliebig vertauscht werden. Außerdem ist es ohne Weiteres möglich auf einem vorhandenen Modell einen neuen View zu implementieren. Dadurch ist ein solches System nahezu beliebig skalierbar.“ [Steffen, 2018]

Die **Abbildung 5.5** stellt das MVC-Entwurfsmuster der zu realisierenden Webanwendung dar. Die WebSockets-Komponente kommuniziert sowohl mit der View als auch mit dem Controller. Im Controller wird die Anwendungslogik des WebSockets implementiert. Mit Hilfe des WebSockets werden auf der View-Komponente anschließend die Anwendungsdaten in Echtzeit präsentiert.

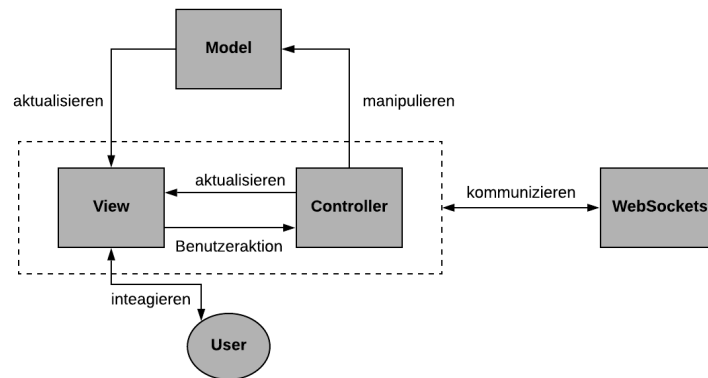


Abbildung 5.5: Das MVC-Entwurfsmuster angepasst an die zu realisierenden Anwendung

**Quelle:** eigene Abbildung

Die **Abbildung 5.6** zeigt die einzelnen Schichten der Model-, Controller-, und View-Module. In der View-Schicht befinden sich die Views der einzelnen Seiten der Webanwendung. Jede View hat einen eigenen Controller. Zusammen bilden sie ihre eigene Benutzeroberfläche der Webanwendung. Der WebSocket ist hauptverantwortlich für die bidirektionale Kommunikation zwischen der View- und Controller-Schicht. Die Model-Schicht definiert die Datenbankschnittstelle der Anwendung und stellt die Operationen zur Datenmanipulation bereit.

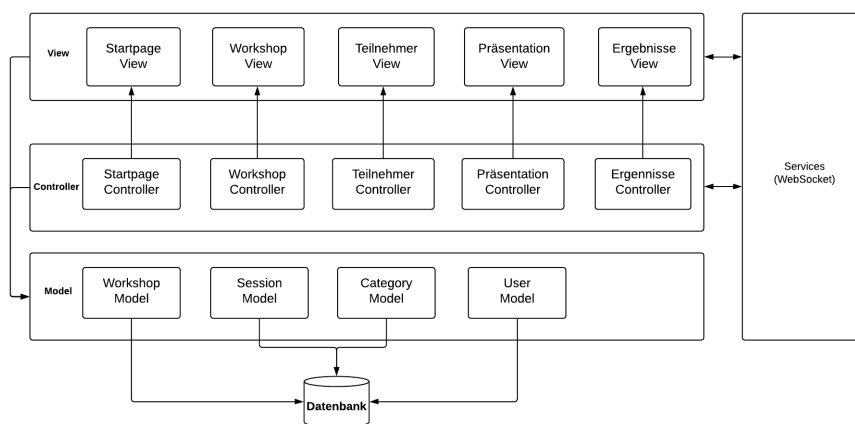


Abbildung 5.6: Architektur der Webanwendung

**Quelle:** eigene Abbildung

## 5.4 Serverseitige Implementierung

Die Serverapplikation wurde mit der WebSocket-Bibliothek namens Ratchet, welche schon im **Abschnitt 5.1.8** erläutert wurde, umgesetzt. Der WebSocket-Server fungiert in dieser Webanwendung als Router, der die Informationen von einem Client zu anderen Clients in Echtzeit transferiert. Das **Listing 5.1** stellt den WebSocket-Server dar, welcher in php implementiert wurde. Diese php-Datei muss vor jeder Anwendung ausgeführt werden, um den Server zu starten. Der Server ist über die IP-Adresse 127.0.0.1 und den Port 7070 aufrufbar. Damit die Clients die WebSockets-Verbindung aufbauen können, muss auf der Clientseite das Protokoll „ws“, die Adresse und der Port angegeben werden. Die Server-Adresse auf der Clientseite lautet somit „ws://127.0.0.1:7070“.

```
1 use Thruway\Peer\Router;
2 use Thruway\Transport\RatchetTransportProvider;
3
4 //Erzeugt neuer Router mit der Adresse 127.0.0.1 und den Port 7070
5 $router = new Router();
6 $transportProvider = new RatchetTransportProvider("127.0.0.1", 7070);
7 $router->addTransportProvider($transportProvider);
8 //Start den Router
9 $router->start();
```

Listing 5.1: WebSocket-Server

## 5.5 Clientseitige Implementierung

In diesem Unterkapitel wird die clientseitige Implementierung und die Vorgehensweise bei der Entwicklung der Webanwendung beschrieben.

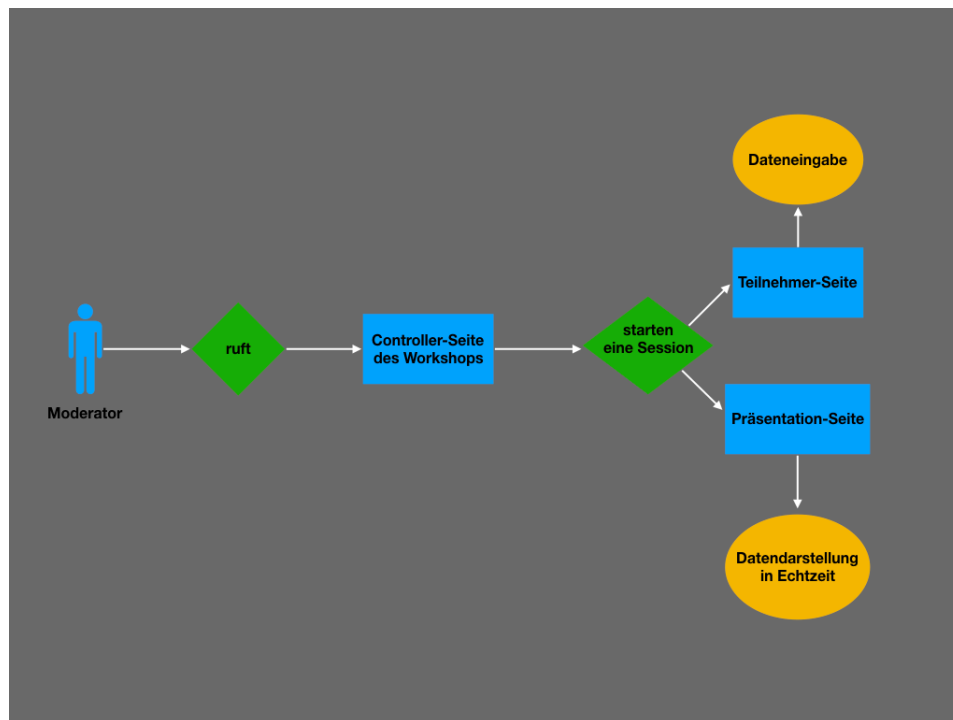


Abbildung 5.7: Workflow auf der Controller-Seite

**Quelle:** eigene Abbildung

Die **Abbildung 5.7** beschreibt den Arbeitsablauf auf der Controller-Seite. Nachdem der Moderator einen Workshop erstellt hat, ruft er diesen auf. Er wird anschließend auf die Controller-Seite des Workshops weitergeleitet. Der Moderator erstellt für die Ideenfindung eine neue Session und startet sie. Auf der Teilnehmer-Seite wird beim Starten der Session die Eingabefunktion freigeschaltet. Auf der Präsentation-Seite wird beim Starten der Session ein Bereich für die Darstellung von Daten bereitgestellt.

Um diesen beschriebenen Arbeitsablauf, wie auf der **Abbildung 5.7** dargestellt, zu realisieren, kommt das WAMP-Protokoll (**Abschnitt 2.1.9**) zum Einsatz. Dabei wird das Publish-Subscribe-Muster (PubSub-Muster) angewendet, bei dem die Abonnenten (Subscribers) die vom Verteiler (Publisher) veröffentlichte Nachrichten empfangen können. Der WebSocket-Server übernimmt beim PubSub-Muster das Nachrichten-Routing, welcher nur eine Aufgabe hat, die veröffentlichten Nachrichten an die Abonnenten zu verteilen (**Abbildung 5.8**).

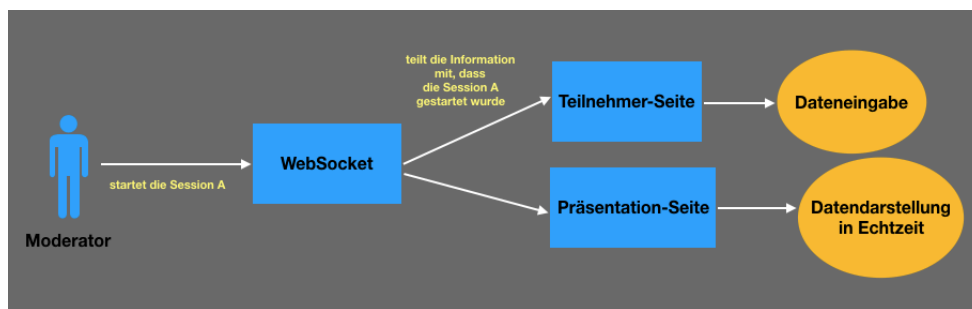


Abbildung 5.8: PubSub-Protokoll

**Quelle:** in Anlehnung an <https://blog.felix-seifert.com/web-application-messaging-protocol/>

Der WebSocket-Server wurde bereits im vorherigen Unterkapitel implementiert. Nun wird jeweils auf der Controller-, Teilnehmer- sowie Präsentation-Seite eine Verbindung zum WebSocket-Server aufgebaut. Als Werkzeug für die Implementierung kommt das AutobahnJS (**Abschnitt 5.1.8**) zum Einsatz.

Das **Listing 5.2** zeigt, wie eine Verbindung zum WebSocket-Server aufgebaut wird. Die URL gibt die Adresse zum WebSocket-Server an, welcher das Protokoll „ws“ verwendet. Der „realm“-Parameter legt eine Domäne oder einen Bereich für das Nachrichten-Routing fest. Es ist zu beachten, dass die Abonnenten die veröffentlichten Nachrichten empfangen können, wenn sie im selben Bereich wie der Herausgeber (Publisher) sind.

```
1 var connection = new autobahn.Connection({
2     url: "ws://127.0.0.1:7070",
3     realm: "workshoppy"
4 });
```

Listing 5.2: Verbindungsaufbau zum WebSocket-Server - php

### 5.5.1 Session starten

Die Session ist erst gestartet, wenn der „Session starten“-Button auf der Controller-Seite getätigt wird (**Abbildung 5.9**).

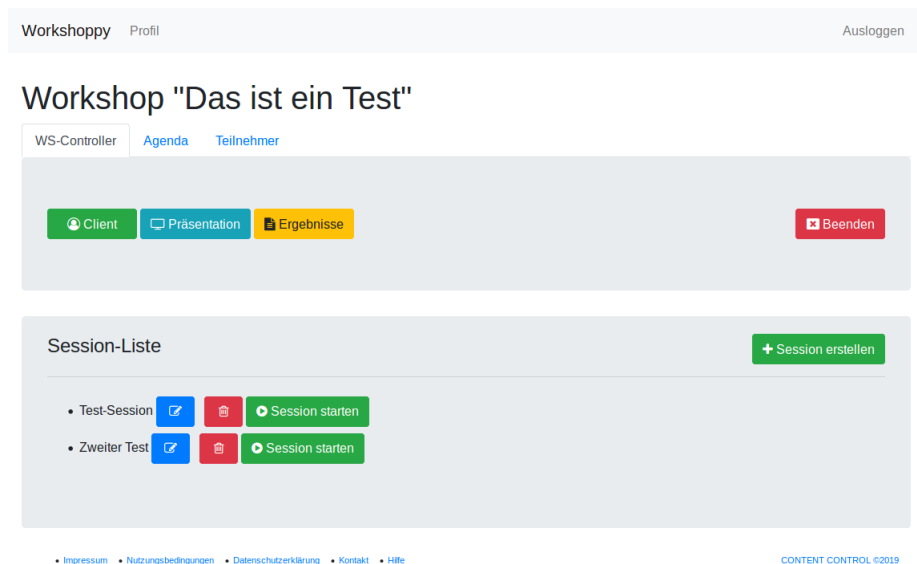


Abbildung 5.9: Controller-Seite

**Quelle:** eigene Abbildung

Die Implementierung auf der Controller-Seite erfolgt mittels JavaScript (**Listing 5.3**). Die Variable „controller\_topic“ beschreibt die URI des Workshops, über die eine Ressource identifiziert wird. Innerhalb der „add\_controls“-Funktion wurde ein Event-Handler für den „Session starten“-Button implementiert. Ein Event (Ereignis) wird ausgelöst, wenn der Button geklickt wird. Das id-Attribut „session-list“ gibt den Bereich an, in dem sich der Button befindet. Das Button-Element selbst wird über das class-Attribut „session-start“ angesprochen. Mit dem onclickt-Event wird dabei eine Funktion aufgerufen. In dieser Funktion wurde das Publish-Muster für die Verteilung der Nachrichten implementiert. Es wird dabei ein Objekt mit zwei Werten an die Abonnenten gesendet. Der Schlüssel „stage“ mit dem Wert „session“ gibt an, dass die Session gestartet ist. Auf die Session-Frage kann über den Schlüssel „msg“ zugegriffen werden. Das Event-Listener „connection.onopen“ indiziert, dass die Verbindung mit dem WebSocket-Server bereit ist, so dass man Informationen versenden und empfangen kann. Bei einem Verbindungsfehler oder wenn die Verbindung zum WebSocket geschlossen ist, wird auf der Konsole eine Warnung ausgegeben.



```
1 //URI von Controller-Seite
2 var controller_topic = 'de.ccb.workshoppy.controller.' + window.workshoppy.
   guid;
3 //Verbindung ist geöffnet
4 connection.onopen = function(session, details) {
5     add_controls(session);
6 };
7 //Verbindung ist geschlossen
8 connection.onclose = function (reason, details) {
9     //Warnungsausgabe
10    console.warn('WebSocket connection closed: ' + reason);
11 };
12 function add_controls(session) {
13     //clickt-Event für den Session starten Button
14     $('#session-list').on('click', '.session-start', function(e) {
15         //Abonniert die Controller-Seite, um Nachrichten zu empfangen
16         session.publish(controller_topic, [{
17             //Übermittelte Daten
18             stage: 'session',
19             msg: $(this).attr('data-question');
20         }]);
21     });
22 }
```

Listing 5.3: Publish-Muster auf der Controller-Seite - JavaScript

Um die Daten vom Publisher (Controller-Seite) empfangen zu können, muss der Empfänger (Teilnehmer-Seite) das Publisher-Event „controller\_topic“ abonnieren. Diese Vorgehensweise wird mit dem Subscribe-Muster umgesetzt. (**Listing 5.4**).

```
1 //Controller-Event
2 var controller_topic = 'de.ccb.workshoppy.controller.' + window.workshoppy.
   guid,
3 //Teilnehmer-Event
4   client_topic = 'de.ccb.workshoppy.client.' + window.workshoppy.guid;
5 //Stellt eine Verbindung zum WebSocket-Server her
6 var connection = new autobahn.Connection({
7   url: "ws://127.0.0.1:7070",
8   realm: "workshoppy"
9 });
10 //Verbindung ist geöffnet
11 connection.onopen = function(session, details) {
12   //Abonniere der Controller-Seite, um Nachrichten zu empfangen
13   session.subscribe(controller_topic, function(args) {
14     //Aktualisiere die Eingabefunktion auf der Teilnehmer-Seite
15     update_stage(args[0]);
16   });
17   //Funktion für das Abschicken der Daten
18   add_controls(session);
19 }
20 //Verbindung ist geschlossen
21 connection.onclose = function (reason, details) {
22   //Warnungsausgabe
23   console.warn('WebSocket connection closed: ' + reason);
24 };
```

Listing 5.4: Subscribe-Muster auf der Teilnehmer-Seite - JavaScript

Die Funktion „update\_stage“ im Subscribe-Muster im **Listing 5.4** stellt die Eingabefunktion auf der Teilnehmer-Seite (Abbildung 5.x: Teilnehmer-Seite) zur Verfügung, wenn das empfangene Objekt (data) den Zustand „session“ aufweist (**Listing 5.5**).

```

1 function update_stage(data) {
2     //Das empfangene Objekt wird geprüft
3     if (data.stage === 'session') {
4         //Wählt ein Element mit dem angegebenen id-Attribut aus und
           fügt die Session-Frage hier ein.
5         $('#question').text(data.msg);
6         //Gib der Bereich zur Dateneingabe frei
           $('#stage-session').show();
7     }
8 }
9

```

Listing 5.5: Stellt die Eingabefunktion auf der Teilnehmer-Seite bereit - JavaScript

Das **Listing 5.6** beschreibt die HTML-Struktur der Teilnehmer-Seite (**Abbildung 5.10**). Innerhalb des <form>-Tags befindet sich das gesamte Formular (Textarea und Button) für die Dateneingabe. Die Session-Frage, welche eine Überschrift zweiter Ordnung ist, wurde mittels Heading-Tag (<h2></h2>) definiert.

```

1 <div class="container stage" id="stage-session">
2     <h2 id="question" class="text-center"></h2>
3     <hr class="my-4">
4     <form id="answer">
5         <div class="form-group row">
6             <div class="col-sm-12">
7                 <textarea rows="4" cols="56" id="message_input" required
8                     class="form-control"
9                     placeholder="Bitte gib deine Antwort ein..."></textarea>
10            </div>
11        </div>
12        <button class="btn btn-primary btn-block" type="submit">
13            <i class="fa fa-comment" aria-hidden="true"></i>
14            Abschicken
15        </button>
16    </form>
17 </div>

```

Listing 5.6: HTML-Struktur der Teilnehmer-Seite

## Test-Session

Abbildung 5.10: Teilnehmer-Seite

**Quelle:** eigene Abbildung

### 5.5.2 Dateneingabe auf der Teilnehmer-Seite

Das **Listing 5.7** beschreibt die „add\_controls“-Funktion im **Listing 5.4**. Diese Funktion sorgt auf der Teilnehmer-Seite dafür, dass die Daten nach dem Tätigen des Abschicken-Buttons bei der Präsentation-Seite ankommen. Das Eingabeformular wird in einem form-Tag definiert (**Listing 5.6**). Das Button-Element ist vom typ-Attribut „submit“, welches beim Klicken auf den Button ein submit-Event auslöst. Innerhalb der Funktion des submit-Events wird das Publish-Muster für die Verbreitung der eingegebenen Daten implementiert. Über die Variable „client\_topic“ im **Listing 5.4** wird die Teilnehmer-Seite (Publisher) identifiziert. Die Daten werden als ein Objekt (cardData) übermittelt. In diesem Objekt sind der Name des Absenders und seine eingegebenen Daten enthalten.

```

1 function add_controls(session) {
2     //submit-Event
3     $('#answer').on('submit',function(e) {
4         e.preventDefault();
5         //wenn der Benutzername leer ist, gibt eine Meldung aus, dass
6           der Benutzername werden muss.
7         if (typeof username === 'undefined'){
8             alert("Username muss gesetzt werden");
9         }
10        return;
11        //Die Variable 'msg' wird mit den eingegebenen Daten
12          initialisiert.
13        var msg = $(this).find('textarea').val(),
14        //Daten sowie Benutzernamen werden in einem Objekt gekapselt.
15        cardData = {
16            'user_id': username,
17            'msg': msg
18        },

```

```

18         //Verbreite diese Information an seine Abonnenten
19         session.publish(client_topic, [cardData]);
20         $(this).find('').val('').focus();
21         $(this).find('textarea').val('').focus();
22     });
23 }

```

Listing 5.7: Funktion für das Abschicken von Daten auf der Teilnehmer-Seite - JavaScript

Die Teilnehmer-Seite übernimmt im Prinzip zwei Rollen:

1. der Subscriber, der die Nachrichten von der Controller-Seite empfängt.
2. der Publisher, der die eingegebenen Daten der Teilnehmer an die Präsentation-Seite weiterleitet.

### 5.5.3 Darstellung von Daten in Echtzeit

Die Präsentation-Seite ist das Herzstück der Webanwendung. Auf dieser werden die eingegebenen Daten von den Teilnehmern in Echtzeit dargestellt und anschließend werden die gesammelten Daten in Kategorien zusammengefasst. Damit die Präsentation-Seite die Daten in Echtzeit darstellen kann, muss zunächst eine Verbindung zum WebSockets-Server aufgebaut werden (vgl. **Listing 5.2**).

Die Präsentation-Seite muss den „client\_topic“ abonnieren, um die Nachrichten der Teilnehmer empfangen zu können (**Listing 5.8**).

```

1 //Verbindung zum WebSocket ist geöffnet
2 connection.onopen = function(session, details) {
3     //Abonniere die Teilnehmer-Seite, um die eingegebenen Daten zu erhalten
4     session.subscribe(client_topic, function (args) {
5         //Funktion zur Darstellung der Daten
6         add_card(args[0]);
7     });
8 };

```

Listing 5.8: Subscribe-Muster auf der Präsentation-Seite - JavaScript

Die Funktion „add\_card“, welche im **Listing 5.8** zu sehen ist, wird im **Listing 5.9** detailliert vorgestellt. Die empfangenen Daten werden wie eine Karte auf der Präsentation-Seite dargestellt. Auf dieser Karte sind der Benutzername (user\_id) und dahinter die Nachricht (msg) zu sehen. Als Gestaltungsvorlage wird die Klasse „alert“ und „alert-warning“ vom Bootstrap verwendet. Die Daten werden in einem Array namens „cards“ gespeichert. Mit Hilfe der JQuery-Methode „append“ werden die Daten im Bereich „message\_output“ auf der Präsentation-Seite dargestellt (**Abbildung 5.11**).

```
1 function add_card(data) {  
2     var card = $('<div data-index="' + index + '" class="alert alert-  
        warning"><p><b>' + htmlDecode(data.user_id) + ':</b> ' +  
        htmlDecode(data.msg) + '</p></div>'),  
3     //Speichert die Daten in einem Arrey namens cards  
4     cards.push(data);  
5     $('#message_output').append(card);  
6 }
```

Listing 5.9: Funktion zur Visualisieren der Daten auf der Präsentation-Seite - JavaScript

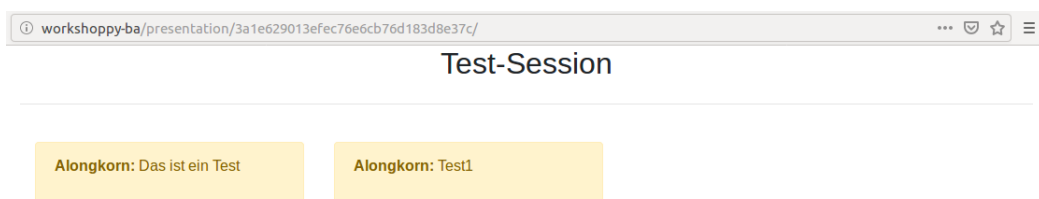
[QR-Code anzeigen](#)

Abbildung 5.11: Präsentation-Seite  
**Quelle:** eigene Abbildung

### 5.5.4 Daten in Kategorien zusammenfassen

Die auf der Präsentation-Seite befindlichen Daten können erst in Kategorien zusammengefasst werden, wenn die Ideensammlungsphase vorüber ist. Das bedeutet, dass der „Eingabe beenden“-Button auf der Controller-Seite getätigt werden muss (**Abbildung 5.12**). Durch das Klicken auf diesen Buttons wird auf der Präsentation-Seite ein neuer Button namens „Kategorien erstellen“ eingeblendet und die Eingabe auf der Teilnehmer-Seite wird dabei gestoppt. Es können demzufolge keine weiteren Daten mehr eingegeben werden.

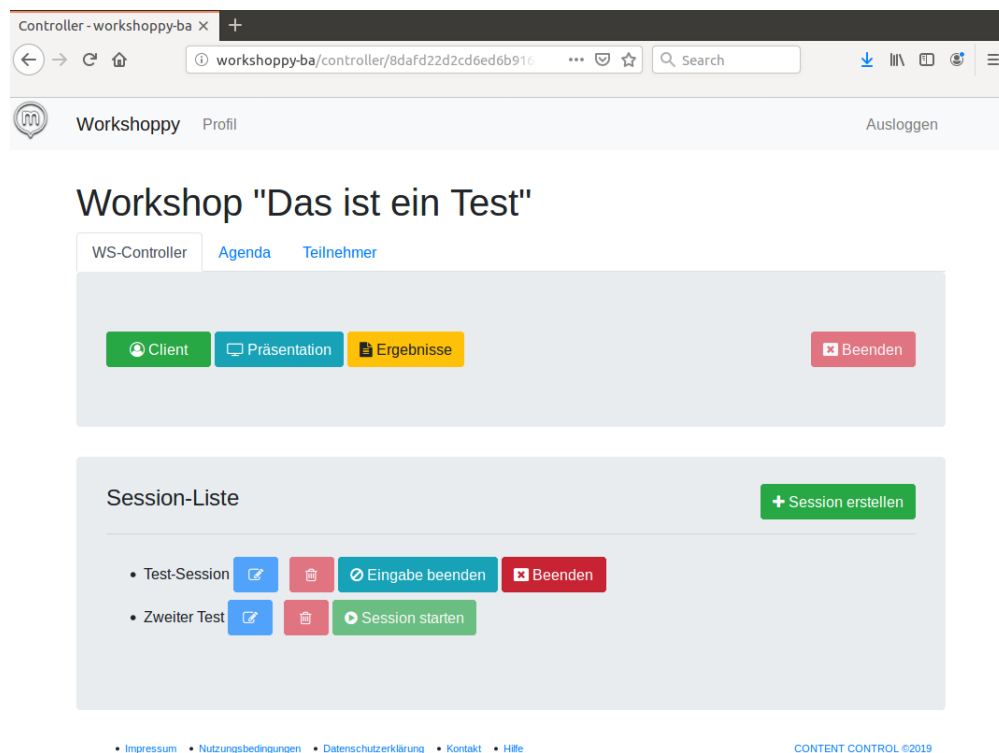


Abbildung 5.12: Controller-Seite, wenn die Session gestartet ist

**Quelle:** eigene Abbildung

Das **Listing 5.10** stellt den Aufbau des Inhaltsbereiches auf der Präsentation-Seite dar. Bei der Session-Frage handelt es sich, wie bei der Teilnehmer-Seite, um eine Überschrift zweiter Ordnung. Die eingegebenen Daten befinden sich innerhalb des div-Elements mit der Id „message\_output“. Das div-Element ist ein Bereich, in welchem Elemente eingeschlossen werden. Die erstellten Kategorien werden im Bereich „category\_output“ dargestellt. Der Titel einer Kategorie kann durch das Anklicken auf den Bereich „Kategorie-Titel“ verändert werden. Die Kategorie wurde mit Hilfe von JavaScript umgesetzt, wie im **Listing 5.11** zu sehen ist. Gespeichert werden sie in einem Array namens „categories“. Über das Id-Attribut werden die Kategorien angesprochen.

```

1 <div class="container-fluid stage" id="stage-session">
2     <h2 id="question" class="text-center"></h2>
3     <hr class="my-4">
4     <div id="message_output"></div>
5     <div id="category_output"></div>
6 </div>

```

Listing 5.10: HTML-Struktur der Präsentation-Seite

```

1 function add_category(data) {
2     //Kategorie-Titel
3     var categoryTitle = $('<div class="row"><div id="categoryTitle-' +
4         data.id + '" class="col categoryTitle" data-category="' + data.id
5         + '>' + htmlDecode(data.title) + '</div></div>'),
6     //Kategorie-Body
7     categoryBox = $('<div data-category="' + data.id + '" class="
8         connectedSortable sortedMessages"></div>'),
9     //Close-Button
10    closeButton = $('<a class="boxclose" data-dialog="delete" data-form-id
11        ="confirm-delete"></a>'),
12    //Trennlinie zwischen Kategorie-Titel und Kategorie-Body
13    lineal = $('<hr class="style1">'),
14    //Setzt die Kategorie-Elemente zusammen
15    $('#category_output').append(categoryBox);
16    categoryBox.append(categoryTitle);
17    categoryBox.append(closeButton);
18    categoryBox.append(lineal);
19    //Speichert die Ids von Kategorien in einem Array namens categories
20    categories.push(data);
21    //Sortable Funktion
22    initSortable(data.id);
23 }

```

Listing 5.11: Kategorie umsetzen - JavaScript



Innerhalb der „add\_category“-Funktion im **Listing 5.11** ist eine Funktion namens „initSortable“ zu sehen. In dieser Funktion wird das Sortable Widgets von JQuery UI implementiert, um die Daten in Kategorien zusammenfassen zu können (**Listing 5.12**).

```
1 function initSortable(id) {
2     if (!$('#message_output').hasClass('connectedSortable')) {
3         // Mache aus der Liste mit der Id 'message_output' eine
4         // Sortable-Liste.
5         $('#message_output').sortable({
6             //Definiere eigenes update-Event. Dieses Event wird ausgelöst,
7             //wenn der Benutzer die Sortierung beendet und die DOM-
8             //Position geändert hat.
9             update: function(event, ui) {
10                 ui.placeholder.css({visibility: 'visible', border: '2
11                 px solid yellow'});
12                 //Hier beginnt die Iteration durch dem 'cards'-Array,
13                 //wo die Elemente gespeichert sind.
14                 cards.forEach(function (item) {
15                     //Eine neue Eigenschaft (Property) namens '
16                     //category' mit dem Wert null wird im Array
17                     //hinzugefügt. Das bedeutet, dass die
18                     //Elemente, die sich in dieser Liste '
19                     //message_output' befinden, keine Kategorien
20                     //zugeordnet werden.
21                     item.category = null;
22                 });
23             },
24             //Gibt an, dass die Elemente aus der 'alert'-Klasse sortiert
25             //werden darf.
26             items: '.alert',
27             //Gibt an, dass die Elemente aus der 'disable-sort'-Klasse
28             //nicht sortiert werden darf.
29             cancel: '.disable-sort',
30             //Um Elemente beispielweise von List A nach List B oder
31             //umgekehrt zu sortieren, müssen die Option 'connectWith'
32             //angewendet werden. Die Option 'connectWith' sorgt dafür,
33             //dass die Listen miteinander verbunden werden.
34             connectWith: '.connectedSortable'
35         }).disableSelection();
36         //Ordnet die Liste 'message_output' eine neue Klasse zu.
37         $('#message_output').addClass('connectedSortable');
38     }
39 }
```

```
25      //Mache die Kategorie mit der 'Id' aus der Klasse 'sortedMessages'
      eine Sortable-Liste.
26      $('[data-category="' + id + '"]').sortable({
27          //Gibt an, dass die Elemente aus der 'alert'-Klasse sortiert
          werden darf.
28          items: '.alert',
29          //Der Kategorie-Titel darf nicht sortiert werden.
          cancel: '.categoryTitle',
30          //Jede erstellte Kategorie wurde zwei Klassen ('sortedMessages'
          ' und 'connectedSortable') zugeordnet (vgl. Listing 5.11)
          Die Option 'connectWith' verbindet die Listen aus der
          Klasse 'connectedSortable' miteinander. Somit können die
          Elemente von Liste 'message_output' nach Liste '
          sortedMessages' sortiert werden und umgekehrt.
32          connectWith: '.connectedSortable',
33          //Definiere eigenes update-Event. Dieses Event wird ausgelöst,
          wenn der Benutzer die Sortierung beendet und die DOM-
          Position geändert hat.
34          update: function(event, ui) {
35              ui.placeholder.css({visibility: 'visible', border: '2
                  px solid yellow'});
36              //Hier beginnt die Iteration durch dem 'cards'-Array,
                  wo die Elemente gespeichert sind.
37              cards.forEach(function (item) {
38                  //Jedes Element in einer Kategorie bekommt die
                  Id von der entsprechenden Kategorie als
                  neue Werte gespeichert.
39                  item.category = $(event.target).attr('data-
                      category');
40              });
41          }
42      }).disableSelection();
43  }
```

Listing 5.12: Sortierfunktion auf der Präsentation-Seite - JavaScript

### 5.5.5 QR-Code anzeigen

Für das Generieren des QR-Codes wurde als Hilfsmittel das frei verfügbare JavaScript-Plugin `qrcode.js`<sup>6</sup> von Kazuhiko Arase verwendet und nach Anforderungen angepasst. Um dieses Plugin nutzen zu können, muss es in das entsprechende HTML-Dokument importiert werden.

```
1 <script type="text/javascript" src="qrcode.js"></script>
```

Listing 5.13: Importiere das verwendete JavaScript-Plugin für das Generieren des QR-Codes

Das **Listing 5.14** beschreibt den strukturellen Aufbau für das Anzeigen des QR-Codes auf der Präsentationsseite. Der ganze Inhaltsbereich wird in einem `div`-Tag eingeschlossen. Bei dem QR-Code handelt es sich hierbei um ein Canvas und wird im `div`-Bereich mit dem `Id`-Attribut „qrcode“ angezeigt. Unter dem QR-Code befindet sich die Beschreibung, die in einem Absatz geschrieben wird.

```
1 <div class="container-fluid stage" id="stage-welcome">
2     <h1 class="text-center" id="welcome-message">Willkommen bei Workshopy
3     </h1>
4     <div class="col-sm-12 text-center" id="qrcode"></div>
5     <p class="text-center" style="font-size: 20px;">Scannen Sie diesen
6     Code, um am Workshop teilzunehmen.</p>
7 </div>
```

Listing 5.14: HTML-Struktur für das Anzeigen des QR-Codes

Das **Listing 5.15** beschreibt, wie der QR-Code mit Hilfe des JavaScripts auf der Präsentation-Seite generiert wird.

```
1 //jQuery integrierte Funktion, die verwendet wird, wenn das Browserfenster
2   seine Größe ändert.
3 $(window).on('resize', function() {
4     //Diese Variable gibt die Url der Teilnehmer-Seite mit dem
5     //entsprechenden GUID des Workshops an.
6     var url = '<?php echo $url_client; ?>/client/' + window.workshopy.
7     guid + '/';
8     //Entfernt zunächst alle untergeordneten Knoten und Inhalte aus den
9     //ausgewählten Elementen.
10    $('#qrcode').empty();
11    //Passt der QR-Code auf die Größe des Browserfensters an.
12    var size = Math.min($('#qrcode').width(), $(window).height() - ($('#qrcode').offset().top + 55));
```

<sup>6</sup><https://github.com/kazuhikoarase/qrcode-generator/tree/master/js>

```
9 //In div-Bereich mit dem Id-Attribut 'qrcode' wird der QR-Code anhand  
    der berechneten Größe generiert und gerendert.  
10 $('#qrcode').qrcode({width: size, height: size, text: url});  
11 });
```

Listing 5.15: Generieren des QR-Codes - JavaScript



Abbildung 5.13: QR-Code auf der Präsentation-Seite

**Quelle:** eigene Abbildung

## 6 Bewertung

In diesem Kapitel wird die Webanwendung anhand eines Fragebogens insgesamt von vier Personen bewertet. Alle vier Personen haben eine Position im Unternehmen inne. Die Information über den Namen vom jeweiligen Bewerter sowie seine Position im Unternehmen wird aus Datenschutzgründen nicht angegeben. Mit Hilfe des Online Tools AttrakDiff<sup>1</sup> wurde der Fragebogen erstellt.

Allgemein betrachtet sind die Ergebnisse aus dem Fragebogen zufriedenstellend ausgefallen. Besonders positiv ausgefallen sind die Ergebnisse in puncto Farbigkeit. Vor allem die eingesetzten Farben wurden von den Bewertern sehr positiv bewertet. In Bezug auf die Vielseitigkeit blieb die Bewertung hinter den Erwartungen zurück. Die Seite wirkt nicht abwechslungsreich genug und auch die Seitengestaltung ist zu eintönig. Ebenfalls in Sachen Geschicklichkeit wurde ein positives Feedback von den Bewertern erwartet. Leider blieben die Ergebnisse hinter den Erwartungen zurück.

Aus dem Feedback wurden einige Verbesserungsvorschläge herausgearbeitet. Zum Beispiel soll überlegt werden, wie das Layout zukünftig verbessert werden kann, so dass es bei den Anwendern einen guten Eindruck hinterlässt.

---

<sup>1</sup><http://attrakdiff.de/>

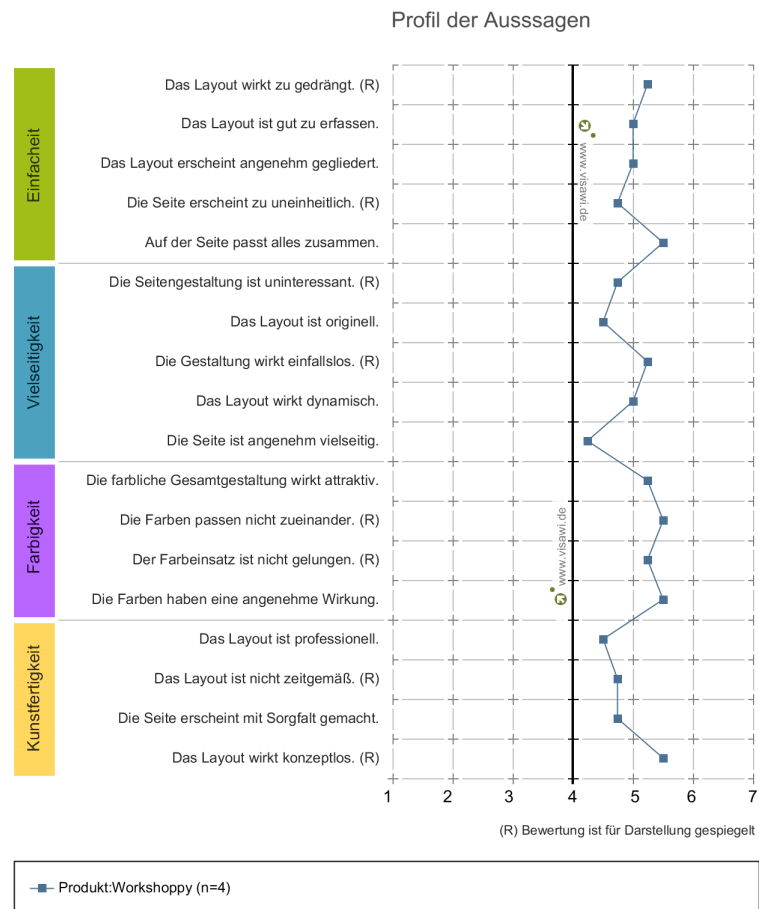


Abbildung 6.1: Ergebnisse aus den Fragebogen

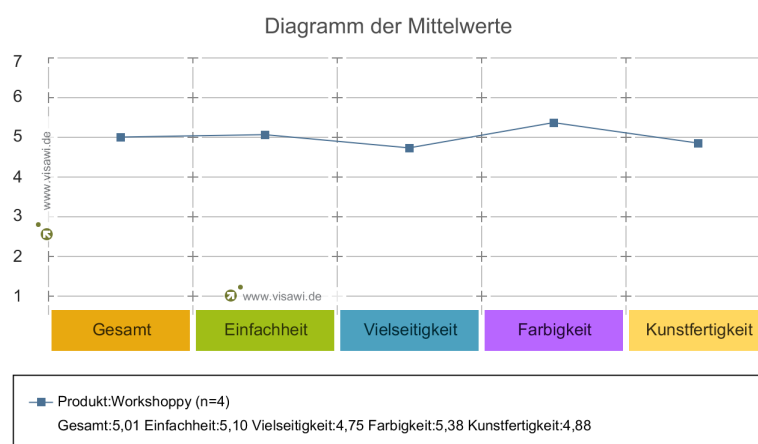


Abbildung 6.2: Ergebnisse aus den Fragebogen

# 7 Schluss

Dieses Kapitel beschäftigt sich zunächst mit der Zusammenfassung der erarbeiteten Ergebnisse. Im Anschluss werden die Ideen für die zukünftige Erweiterung der Webanwendung vorgestellt.

## 7.1 Zusammenfassung

Ziel dieser Bachelorarbeit war es, eine benutzerfreundliche, intuitiv bedienbare und einfach anzuwendende Webanwendung zur Durchführung von Workshops in Echtzeit zu entwickeln, die das klassische Brainstorming digitalisiert. Die zu entwickelnde Webanwendung soll dabei nicht die herkömmlichen Workshops ersetzen, sondern sie effektiver und einfacher machen, in dem die Daten nicht mehr abgetippt oder abfotografiert werden, sondern digital zusammengefasst werden.

Dazu wurden zu Anfang im Kapitel 2 zunächst die für diese Arbeit nötigen Grundlagen erläutert. Im Kapitel 3 wurden zunächst die Konkurrenten analysiert. Daraus wurden dann die erkennbaren Stärken und Schwächen der Konkurrenten entwickelt. Danach ging es innerhalb dieses Kapitels weiter um die Anforderungsanalyse, welche aus funktionalen und nicht-funktionalen Anforderungen bestehen. Zum Schluss des Kapitels wurden die Muss- sowie Kann-Anforderungen festgelegt.

Aus den gesammelten Anforderungen konnte in Kapitel 4 mit der Konzeption begonnen werden. Zuerst wurden die Gestaltungsvorgaben, wie die Verwendung von Farben, die Schriftgestaltung sowie die Festlegung der verwendeten Icons erarbeitet. Danach wurde die Webanwendung anhand der Erstellung von Mockups entworfen. Das darauffolgende Kapitel 5 wurde mit der Implementierung der Webanwendung begonnen. Dafür wurden als erstes die verwendeten Werkzeuge, die für die Implementierung angewendet wurden, vorgestellt. Des Weiteren wurden die Datenmodellierung und die Architektur der Webanwendung untersucht. Anschließend folgte die eigentliche Implementierung, welche in serverseitig und clientseitig aufgeteilt wurde. Während es bei der serverseitigen Implementierung vor allem um die Umsetzung des WebSocket-Servers ging, wird in der clientseitigen Implementierung beschrieben, wie die Nachrichten der Teilnehmer in Echtzeit dargestellt und wie die Ergebnisse in Kategorien zusammengefasst werden können. Die Bewertung der Webanwendung erfolgte im Kapitel 6. Für die Bewertung wurde dabei ein Fragebogen verwendet, das von vier Personen ausgefüllt wurde.

## 7.2 Ausblick

Im Rahmen einer zukünftigen Weiterentwicklung könnte die Webanwendung um die im Folgenden beschriebenen Bereiche erweitert werden.

- **Registrierungsfunktion:**

Bisher wird ein Benutzerkonto manuell in der Datenbank angelegt. Es gibt auf der Webanwendung noch keine Möglichkeit, sich zu registrieren. Von daher ist es wünschenswert, eine Registrierungsfunktion zu haben, so dass sich die moderierende Person zukünftig direkt bei der Webanwendung registrieren kann.

- **Vote-Funktion:**

Während der Auswertungsphase sollte der Moderator zukünftig die Möglichkeit erhalten, die Ideen zu votieren. Eine Möglichkeit wäre, dass bei jeder Idee ein Gefällt-mir-Button implementiert wird, so dass der Moderator durch jedes Klicken auf diesen Button viele Likes abgeben kann. Dementsprechend sollten die Ideen auch nach Anzahl der Likes sortiert werden können.

- **Kategorien sollten nicht nur auf der Präsentation-Seite erstellt werden können:**

Bisher können Kategorien nur auf der Präsentation-Seite erstellt werden, wenn die Ideensammelungsphase beendet ist. Als zukünftige Weiterentwicklung wäre es sinnvoll, dass der Moderator beim Erstellen der Session Kategorien anlegen können. Dementsprechend könnten die Teilnehmer bei der Abgabe ihrer Ideen einer Kategorie zuordnen.

- **Eingegebene Daten bearbeiten:**

Die eingegebenen Daten können bisher nicht bearbeitet werden. Somit wäre es für den Moderator optimal, die eingegebenen Daten bei Tippfehlern oder bei Unklarheiten editieren zu können. Die bearbeiteten Daten sollten sowohl auf der Präsentation-Seite als auch auf dem Ergebnisprotokoll ein Symbol haben, dass sie als bearbeitet anzeigt.

- **Im Ergebnisprotokoll sollte der Name vom Moderator stehen:**

Als möglicher Erweiterungspunkt könnte der Name des Moderators im Ergebnisprotokoll aufgeführt werden.

- **Eingabe wieder aufnehmen:**

Während die Session läuft, gibt es auf der Controller-Seite einen Button, der die Eingabe auf der Teilnehmer-Seite stoppt. Dies hat zur Folge, dass die Teilnehmer in diesem Moment keine Daten mehr eingeben können. Um in dieser Session Daten wieder eingeben zu können, muss die Session neu gestartet werden. Somit sollte in der Zukunft eine Funktion implementiert werden, dass sie die Eingabe auf der Teilnehmer-Seite erneut aufnimmt.



# Literaturverzeichnis

- [Aderhold, 2013] Aderhold, M. (2013). „Skalierbares HTTP Long Polling“. In: accso.de, URL: <https://accso.de/magazin/skalierbares-http-long-polling/> (abgerufen am 02.08.2019).
- [Augsten, 2019] Augsten, S. (2019). „Was ist jQuery?“. In: dev-insider.de, URL: <https://www.dev-insider.de/was-ist-jquery-a-782237/> (abgerufen am: 16.07.2019).
- [Balzert, 2010] Balzert, H. (2010). *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. 3. Auflage, Spektrum Akademischer Verlag, ISBN: 978-3-8274-1705-3 , S.455-474.
- [Bankras/Smeets/Bram/Boness/Uri/Roald, 008a] Bankras/Smeets/Bram/Boness/Uri/Roald (2008a). „*Beginning Google Web Toolkit from Novice to Professional*“. Apress Verlag, ISBN 978-1-4302-1032-0, S. 6.
- [Bankras/Smeets/Bram/Boness/Uri/Roald, 008b] Bankras/Smeets/Bram/Boness/Uri/Roald (2008b). „*Beginning Google Web Toolkit from Novice to Professional*“. Apress Verlag, ISBN 978-1-4302-1032-0, S. 7.
- [Bozzon/Comai/Fraternali/Carughi, 2006] Bozzon/Comai/Fraternali/Carughi (2006). „*Conceptual modeling and code generation for rich internet applications*“. Proceedings of the 6th international conference on Web engineering, ISBN:1-59593-352-2, S.353-360.
- [Chornaya, 2018] Chornaya, J. (2018). „5 effektive Workshop Methoden die außerdem Spaß machen“. In: eventbrite.de, URL: <https://www.eventbrite.de/blog/effektive-workshop-methoden/#workshopmethodefishbowl> (abgerufen am 18.07.2019).
- [eduonix, 2018] eduonix (2018). „Introduction to Web Application Messaging Protocol (WAMP)“. In: eduonix.com, URL: <https://blog.eduonix.com/web-programming-tutorials/web-application-messaging-protocol/> (abgerufen am 26.08.2019).
- [Hartmann, 2007a] Hartmann, L. (2007a). „Rich Internet Applications“. Hochschule für Angewandte Wissenschaften Hamburg, Master Seminararbeit, URL: <https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/hartmann/bericht.pdf>, S.3, (abgerufen am 22.07.2019).
- [Hartmann, 2007b] Hartmann, L. (2007b). „Rich Internet Applications“. Hochschule für Angewandte Wissenschaften Hamburg, Master Seminararbeit, URL: <https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/hartmann/bericht.pdf>, S.4 f., (abgerufen am 22.07.2019).

- [Herb, oD] Herb, M. (o.D.). „Data Synchronisation zwischen mehreren Clients in Echtzeit“. In: doag.org, URL: [https://www.doag.org/formes/pubfiles/2262550/91-2010-K-JAVA-Herb-Data\\_Synchronisation.pdf](https://www.doag.org/formes/pubfiles/2262550/91-2010-K-JAVA-Herb-Data_Synchronisation.pdf) (abgerufen am 2.08.2019).
- [Holtkamp, 2001] Holtkamp, H. (2001). „TCP/IP im Detail“. In: rvs.uni-bielefeld.de, URL: [http://www.rvs.uni-bielefeld.de/~heiko/tcpip/tcpip\\_html\\_alt/kap\\_2\\_4.html](http://www.rvs.uni-bielefeld.de/~heiko/tcpip/tcpip_html_alt/kap_2_4.html) (abgerufen am 30.07.2019).
- [Holzer, 2012] Holzer, P. (2012). *Diplomarbeit: Kreativitätstechniken zur Ideenfindung*. Bachelor + Master Publishing, ISBN: 978-3-95549-501-5, S.11.
- [Karadeniz, oJ] Karadeniz, B. (o.J.). „TCP/IP - Haussprache des Internet“. In: netplanet.org, URL: <https://www.netplanet.org/aufbau/tcpip.shtml> (abgerufen am 31.07.2019).
- [Maurice, 2011] Maurice, C. W. H. (2011). *„Das Website Handbuch. Programmierung und Design“*. Markt + Technik Verlag, ISBN 978-3-8272-4696-7, S.84.
- [o.V., 2008] o.V. (2008). „Im Blickpunkt Web 2.0“. In: imblickpunkt.grimme-institut.de, URL: <https://imblickpunkt.grimme-institut.de/wp/wp-content/uploads/2014/12/IB-Web20.pdf> (abgerufen am 21.07.2019).
- [o.V., 2017] o.V. (2017). „Javascript XMLHttpRequest (XHR) und Ajax“. In: mediaevent.de, URL: <https://www.mediaevent.de/javascript/XMLHttpRequest.html> (abgerufen am 20.07.2019).
- [o.V., 2019] o.V. (2019). „TCP - Transmission Control Protocol“. In: elektronik-kompendium.de, URL: <https://www.elektronik-kompendium.de/sites/net/0812271.htm> (abgerufen am 30.07.2019).
- [o.V., oJ] o.V. (o.J.). „WebSocket: So funktioniert die Verbindung zwischen einer Webanwendung und einem Webserver“. In: it-times.de, URL: <https://www.it-times.de/news/websocket-so-funktioniert-die-verbinding-zwischen-einer-webanwendung-und-einem-webserver-118921/> (abgerufen am 02.08.2019).
- [phil. Tomas Bohinc, 2016] phil. Tomas Bohinc, D. (2016). „Workshop“. In: projektmagazin.de, URL: <https://www.projektmagazin.de/methoden/workshop> (abgerufen am 17.07.2019).
- [Reich, 2007] Reich, P. D. K. (2007). „Brainstorming“. In: methodenpool.uni-köln.de, URL: <http://methodenpool.uni-koeln.de/download/brainstorming.pdf> (abgerufen am 18.07.2019).
- [Schmitt, 2006] Schmitt, L. (2006). „Die Wahrnehmung von Zeit und der Einfluss von Design“. Köln International School of Design, S.55, URL: [https://lutzschmitt.com/content/3-publications/lutz\\_schmitt-wahrnehmung\\_zeit\\_einfluss\\_design.pdf](https://lutzschmitt.com/content/3-publications/lutz_schmitt-wahrnehmung_zeit_einfluss_design.pdf) (abgerufen am 01.08.2019).
- [Schulz, 2008] Schulz, R. G. (2008). *„HTML und CSS Praxisbuch. Einführung in strukturiertes Web-design“*. 1.Auflage, Verlag: mitp, ISBN 978-3-8266-1775-1, S.43.

- [Stal, 1996] Stal, F. B. M. R. S. (1996). „*Pattern-Oriented Software Architecture Volume 1: A System of Patterns*“. Wiley, ISBN 9780471958697, S.123 ff.
- [Stefan Luber, 2018] Stefan Luber, A. D. (2018). „Was ist HTTP (Hypertext Transfer Protocol)?“. In: ip-insider.de, URL: <https://www.ip-insider.de/was-ist-http-hypertext-transfer-protocol-a-691181/> (abgerufen am: 30.07.2019).
- [Steffen, 2018] Steffen (2018). „Model-View-Controller (MVC)“. In: methodenpark.de, URL: <https://www.methodpark.de/blog/model-view-controller-mvc/> (abgerufen am 22.08.2019).
- [van Aerssen, oJ] van Aerssen, B. (o.J.). „Brainstorming im Florence Innovation Project des verrocchio Institute“. In: ideenfindung.de, URL: <https://www.ideenfindung.de/Brainstorming-Kreativit%C3%A4tstechnik-Ideenfindung.html> (abgerufen am 18.07.2018).
- [WAMP, oD] WAMP (o.D.). „Web Application Messaging Protocol “. In: wamp-proto.org, URL: <https://wamp-proto.org/intro.html> (abgerufen am 26.08.2019).
- [Weßendorf, 2011] Weßendorf, M. (2011). „WebSocket: Annäherung an Echtzeit im Web“. In: heise.de, URL: <https://www.heise.de/developer/artikel/WebSocket-Annaeherung-an-Echtzeit-im-Web-1260189.html?seite=all> (abgerufen am 2.08.2019).
- [Wikipedia, 2018] Wikipedia (2018). „Duplex (Nachrichtentechnik)“. URL: [https://de.wikipedia.org/wiki/Duplex\\_\(Nachrichtentechnik\)](https://de.wikipedia.org/wiki/Duplex_(Nachrichtentechnik)) (abgerufen am 2.08.2019).
- [Wikipedia, 019a] Wikipedia (2019a). „WebSocket“. URL: <https://de.wikipedia.org/wiki/WebSocket> (abgerufen am 02.08.2019).
- [Wikipedia, 019b] Wikipedia (2019b). „jQuery“. URL: <https://de.wikipedia.org/wiki/JQuery> (abgerufen am 20.07.2019).
- [Wikipedia, 019c] Wikipedia (2019c). „Responsive Webdesign“. URL: [https://de.wikipedia.org/wiki/Responsive\\_Webdesign](https://de.wikipedia.org/wiki/Responsive_Webdesign) (abgerufen am 20.07.2019).
- [Wikipedia, 019d] Wikipedia (2019d). „Model View Controller“. URL: [https://de.wikipedia.org/wiki/Model\\_View\\_Controller](https://de.wikipedia.org/wiki/Model_View_Controller) (abgerufen am 24.08.2019).
- [Wöhr, 2004] Wöhr, H. (2004). „*Web-Technologie*“. 1.Auflage, dpunkt.verlag, ISBN: 978-3898642477, S.219.