

# 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**

## **Zusammenfassung / Abstract**

### **Abstract**

# Inhaltsverzeichnis

Eidesstattliche Erklärung . . . . .	2
Danksagungen . . . . .	3
Zusammenfassung / Abstract . . . . .	4
<b>1 Einleitung</b>	<b>9</b>
1.1 Motivation . . . . .	9
1.2 Zielsetzung und Aufbau der Arbeit . . . . .	10
<b>2 Grundlagen</b>	<b>11</b>
2.1 Grundlegende Begriffe . . . . .	11
2.1.1 Workshop . . . . .	11
2.1.2 Brainstorming . . . . .	13
2.1.3 TCP/IP . . . . .	14
2.1.4 HTTP . . . . .	16
2.1.5 Ablauf einer HTTP-Verbindung . . . . .	17
2.1.6 AJAX . . . . .	18
2.1.7 Echtzeit . . . . .	19
2.1.8 WebSocket . . . . .	22
2.1.9 jQuery . . . . .	25
<b>3 Analyse</b>	<b>26</b>
3.1 Projektstruktur . . . . .	26
3.2 Ist-Analyse . . . . .	28
3.3 Anforderungsanalyse . . . . .	29
3.3.1 Funktionale Anforderungen . . . . .	29
3.3.2 Nicht-funktionale Anforderungen . . . . .	33
3.3.3 Muss- und Kann-Anforderungen . . . . .	35
3.3.4 Schematische Darstellung einer Bearbeitungshierarchie . . . . .	37
<b>4 Design</b>	<b>38</b>
<b>5 Implementierung</b>	<b>39</b>

<b>6 Fazit und Ausblick</b>	<b>40</b>
<b>Literaturverzeichnis</b>	<b>41</b>

# Abbildungsverzeichnis

2.1	Aufbau eines Datagramms . . . . .	14
2.2	Zeitüberschreitung und erneute Übertragung bei Verlust eines Pakets . . . . .	15
2.3	Die wichtigsten Protokolle im TCP/IP-Referenzmodell . . . . .	15
2.4	Das Client-Server-Modell . . . . .	16
2.5	Aufbau einer URL . . . . .	17
2.6	Klassisches HTTP Request-Response-Paradigma nach [Wöhr, 2004] . . . . .	17
2.7	synchrone und asynchrone Kommunikation . . . . .	19
2.8	Polling . . . . .	21
2.9	LongPolling . . . . .	21
2.10	Beispiel einer Client-Handshake-Anfrage . . . . .	23
2.11	WebSocket-Server Handshake . . . . .	23
2.12	Das WebSocket-Handshake . . . . .	24
3.1	Projektstrukturplan [Quelle: eigene Abbildung] . . . . .	27
3.2	Beispiel einer Mind-Map [Quelle: Learning Fundamentals: Student Study Techniques by Jane Genovese, Figure 4: Health mindmap. Online im Internet: URL: <a href="https://learningfundamentals.com.au/resources/">https://learningfundamentals.com.au/resources/</a> (Stand 11.07.2019)] . . . . .	29
3.3	Muss- und Kann-Anforderungen für die Startseite (S) . . . . .	35
3.4	Muss- und Kann-Anforderungen für die Workshop Controller-Seite (C) . . . . .	35
3.5	Muss- und Kann-Anforderungen für die Teilnehmer-Seite (T) . . . . .	35
3.6	Muss- und Kann-Anforderungen für die Präsentation-Seite (P) . . . . .	36
3.7	Muss- und Kann-Anforderungen für die Ergebnisse-Seite (Er) . . . . .	36
3.8	Sonstige Muss- und Kann-Anforderungen für Webanwendung . . . . .	36
3.9	Darstellung der Bearbeitungshierarchie in der Webanwendung [Quelle: eigene Abbildung] . . . . .	37

# **Tabellenverzeichnis**



# 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 bekannteste 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 neuen 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 6 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 sich 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. Die Teilnehmer 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 aktiver 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 stille Teilnehmer aktivieren sowie die dominante 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, können der Moderator je nach Dauer des Workshops folgenden 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 Problemlösung, neue Produkten, 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 Thema 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 behandelte Frage 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 stillere 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 dokumentierte 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 Vorschläge entstehen.
- **Nachbereitung:**  
Ein Brainstorming fördert nur die Kreativität. Die Vorschlägen 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 die Fantasie freien Lauf. Ungewöhnliche Ideen sind erwünscht.

### 2.1.3 TCP/IP

Das Transmission Control Protocol (TCP) und das Internet Protocol (IP) bilden 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 von 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])

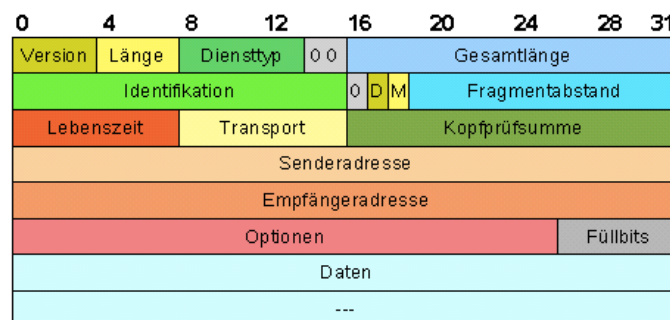


Abbildung 2.1: Aufbau eines Datagramms

**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 Datenmenge in kleinere Pakete zu zerlegen und die empfangene Datenpakete über Ports an den korrekten Anwendungen weiterzuleiten. Da sich TCP 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 nach dem Erhalt der Daten dem Sender 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 kommt, wird das Senden solange wiederholt, bis eine positive Antwort beim Sender eingegangen ist (**Abbildung 2.2**). (vgl. [Holtkamp, 2001])

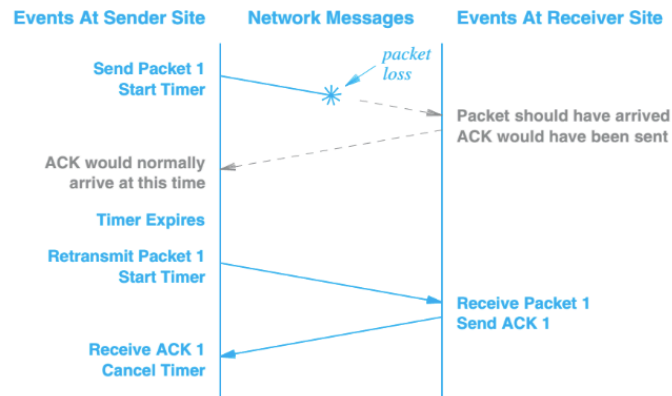


Abbildung 2.2: Zeitüberschreitung und erneute Übertragung bei Verlust eines Pakets

**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 Process Anwendungsschicht	Anwendungen und Nutzdaten interaktive Protokolle	HTTP, SMTP, FTP, DNS POP3, IMAP, Telnet SNMP, NTP	Stream	7 6 5
3 Host-to-Host Transportschicht	Verbindung zweier Netzteilnehmer zur Datenübermittlung	TCP, UDP, TLS (SSL), SCTP	Segment	4
2 Internet Internetschicht	Paketvermittlung Wegewahl (Routing)	IP mit ICMP EIGRP (IGRP) RIP, BGP, OSPF	Datagramm	3
1 Network Access Netzzugangsschicht	Techniken zur Datenübertragung in div. Netztypen	Ethernet mit CSMA/CD Token-Passing, FDDI PPP, ARP, BOOTP (RARP)	Frame	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**).

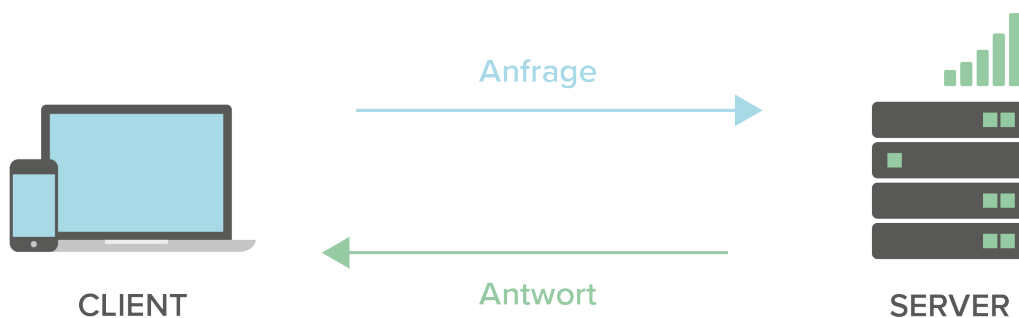


Abbildung 2.4: Das Client-Server-Modell

**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>:

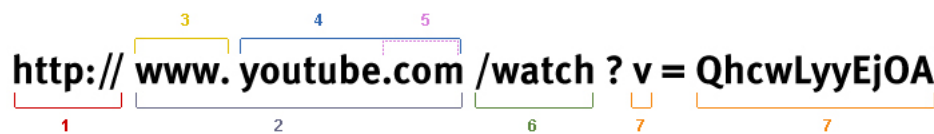


Abbildung 2.5: Aufbau einer URL

**Quelle:** <https://webdesignerinfuehrung.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:

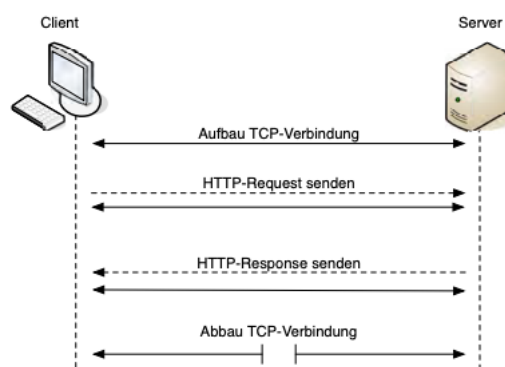


Abbildung 2.6: Klassisches HTTP Request-Response-Paradigma nach [Wöhr, 2004]

<sup>5</sup>vgl. <https://webdesignerinfuehrung.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 von dem 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

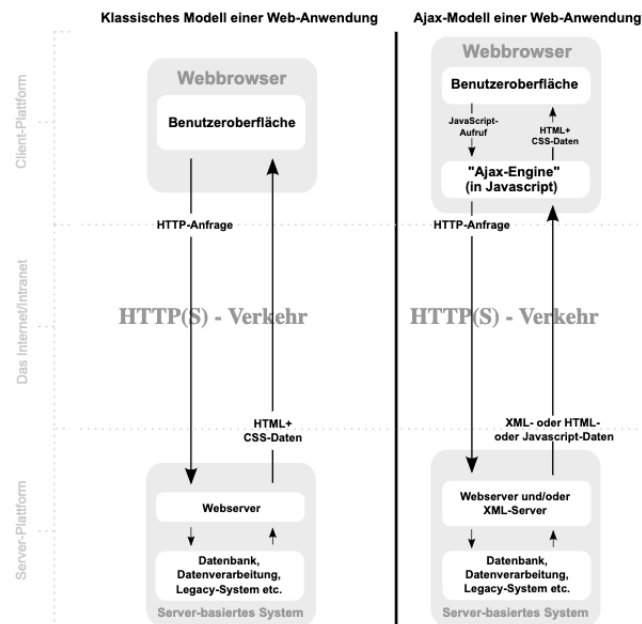


Abbildung 2.7: synchrone und asynchrone Kommunikation

**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])

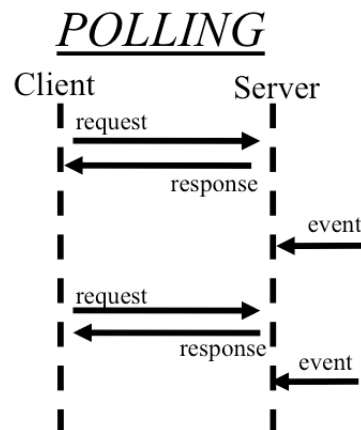


Abbildung 2.8: Polling

**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])

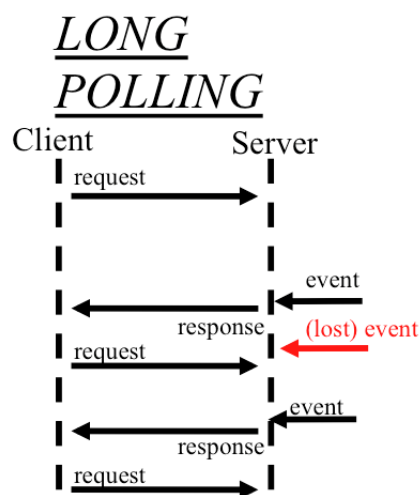


Abbildung 2.9: LongPolling

**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, 2019]

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 einen **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]

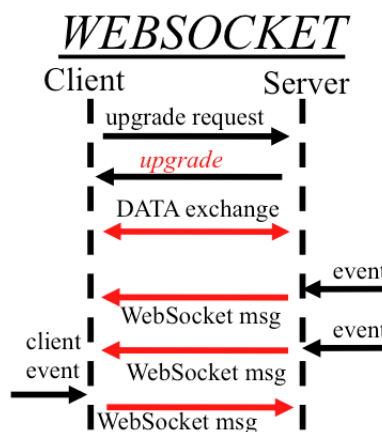


Abbildung 2.12: Das WebSocket-Handshake

**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 das neue URL-Schema. Statt wie gewohnt das „http:“ oder für sichere HTTP-Verbindungen das „https:“ als Protokoll anzugeben, wird bei einer WebSocket-Verbindung das „ws:“ verwenden. Für sichere Verbindungen steht das „wss:“ zur Verfügung.

Zusammenfassend lässt sich sagen, dass WebSocket ein geeigneter 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 zu-



künftiger Webanwendungen steht damit den Desktop-Anwendungen in nichts nach.“  
[Weßendorf, 2011]

### **2.1.9 jQuery**

## 3 Analyse

Dieses Kapitel befasst sich mit der Anforderungsanalyse und Konzeption der Webanwendung. Dazu werden zunächst eine allgemeine Struktur festgelegt, wie das Projekt systematisch aufgebaut sein soll.

Anschließend wird der aktuelle Zustand (Ist-Analyse) des Projektes ermittelt. Anhand dieser Ist-Analyse erfolgen die funktionalen und nicht funktionalen Anforderungen an die zu entwickelnde Webanwendung. Am Ende wird die Architektur der Webanwendung erklärt.

### 3.1 Projektstruktur

Nach Definition der DIN (Deutsches Institut für Normung e.v., 2009) 69901-5:2009 ist der Projektstrukturplan die “ [...] *vollständige hierarchische Darstellung aller Elemente (Teilprojekte, Arbeitspakete) der Projektstruktur als Diagramm oder Liste.*”

Auf der obersten Ebene steht das Projekt. Eine Ebene darunter die Teilprojekte oder Teilaufgaben, darunter schließlich die Arbeitspakete. Der Projektstrukturplan (Abbildung 2.1) entspricht dem typischen sequentiellen Vorgehensmodell zur Softwareentwicklung einschließlich der Entwicklung der Webanwendung.

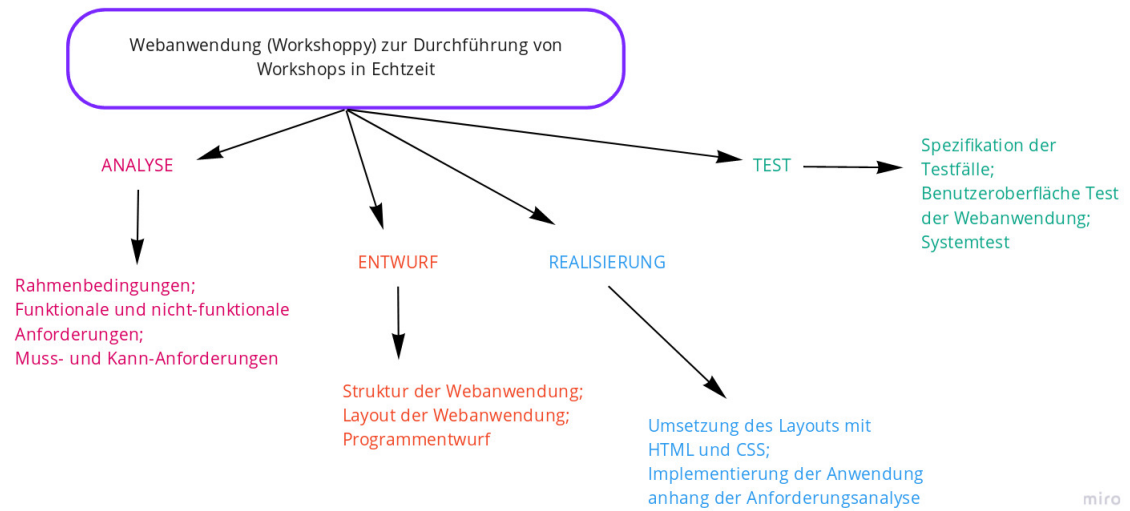


Abbildung 3.1: Projektstrukturplan [Quelle: eigene Abbildung]

Eine Analyse von funktionalen und nicht-funktionalen Anforderungen sowie die Muss- und Kann-Anforderungen wird in der ersten Phase untersucht. Anhand dieser Analyse wird die Struktur und ein passendes Layout der Webanwendung erstellt. Der daraus entstehende Entwurf wird technisch in eine Webanwendung umgesetzt und am Ende getestet.

## 3.2 Ist-Analyse

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

Neben dem Brainstorming wird häufig das Mind-Mapping als Methode zur Ideenfindung, Problemlösung und Kreativitätssteigerung eingesetzt. Eine Mindmap<sup>1</sup> oder auch Mind-Mapping genannt, ist in der Regel mit dem Brainstorming verwandt. In der Abbildung 3.2 wird diese Methode grob 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.

Bis jetzt existieren bereits mehrere webbasierte Mindmapping-Tools, mit denen man kostenlos Mindmaps erstellen, visualisieren und mit anderen in Echtzeit kollaborieren kann. Bei der Durchführung von Workshops gilt das Mindmapping aktuell als das idealste Format zum Brainstormen.

Nach der Betrachtung der aktuellen Lösung fällt das Fazit des Unternehmens folgendermaßen aus: Die Mindmaps sehen auf den ersten Blick unübersichtlich und verschachtelt aus. Sie können sehr schnell ihre Übersichtlichkeit verlieren, wenn verschiedene Schlüsselwörter in Beziehung stehen. 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 wird eine alternative Lösung benötigt, um dieses Problem benutzerfreundlicher und vor allem die Durchführung von Workshops effektiver zu gestalten.

---

<sup>1</sup>auf deutsch: die Gedankenlandkarte

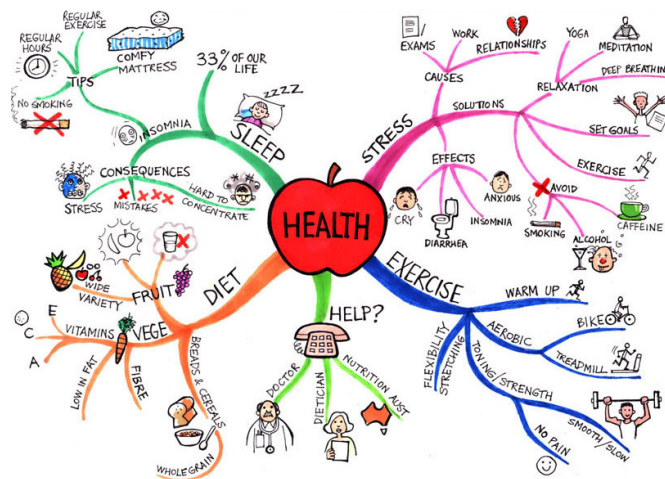


Abbildung 3.2: Beispiel einer Mind-Map

[Quelle: Learning Fundamentals: Student Study Techniques by Jane Genovese, Figure 4: Health mindmap. Online im Internet: URL: <https://learningfundamentals.com.au/resources/> (Stand 11.07.2019)]

### 3.3 Anforderungsanalyse

Dieses Kapitel umfasst die grundlegende Anforderungen dieser Bachelorarbeit. Die Anforderung wird in funktionale und nicht- funktionale 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-funktionale Anforderungen sind Anforderungen, die für die Nutzung des Systems wichtig sind. Außerdem wird eine Muss- und Kann-Anforderung formuliert, welche für das Projekt oberste Priorität hat und welche eher zweitrangig ist.

#### 3.3.1 Funktionale Anforderungen

Aus den Unternehmensanforderungen lassen sich folgende funktionale Anforderungen ableiten.

##### Startseite

Die Startseite der Webanwendung ist die Eingangsstelle für eine moderierende Person, nachdem diese sich bei der Anwendung angemeldet hat. Er kann auf dieser Seite neue Workshops erstellen, sie bearbeiten sowie löschen. Jeder Workshop hat einen Titel und wird in einer Datenbank gespeichert. Die erstellten Workshops werden

nacheinander aufgelistet. Auf der Startseite soll außerdem eine Liste der beendeten Workshops anzeigen. In dieser Liste sind die Workshops, die von der moderierenden Person vorher beendet wurden. Das Datum und die Uhrzeit, an dem die Workshops beendet wurden, soll ebenfalls ausgegeben werden. Außerdem soll jeder beendete Workshop einen Button besitzen, der die Ergebnisse des Workshops anzeigt.

## **Controller-Seite**

Der ausgewählte Workshop soll zur Controller-Seite führen. Diese Seite beinhaltet unter anderem den Titel vom ausgewählten Workshop und eine Liste der Sessions. Die Sessions können vom Moderator erstellt werden. Er soll sie auch bearbeiten und löschen können. Beim Erstellen einer neuen Session soll neben den Titel auch eine Frage, die behandelt wird, angegeben werden. Die Frage wird als Pflichtfeld gekennzeichnet. Die Sessions werden ebenfalls wie bei den Workshops in einer Datenbank gespeichert. Beim Bearbeiten einer Session soll der Moderator neben Titel- und Fragenänderung auch als Option Kategorien zu dieser Session hinzufügen können.

Eine Session versteht sich als eine Sitzung zur Ideenfindung und -sammlung, um Lösungen für eine Problemstellung zu finden. Der Moderator kann zu jedem Workshop mehrere Sessions erstellen.

Jede Session auf der Controller-Seite eines ausgewählten Workshops muss drei Buttons enthalten, die der moderierenden Person folgende Funktionen anbieten:

- Session starten-Button:
  - um Lösungen und Ideen für eine Problemstellung zu sammeln, muss die Session gestartet werden. Hat die moderierende Person eine Session gestartet, soll automatisch die Präsentation-Seite aufgerufen werden. Während eine Session läuft, sollte dieser Button bei den anderen Sessions deaktiviert sein.
- Eingabe beenden-Button:
  - beendet die Funktion zur Dateneingabe seitens der Teilnehmer. Auf der Präsentation-Seite soll nach dem Betätigen dieses Buttons ein Button zur Erstellen von Kategorien freigeschaltet werden.
- Session beenden-Button:
  - beendet die gestartete Session. Die Ergebnisse sollen anschließend in einer Datenbank gespeichert werden.
  - reaktiviert die zuvor deaktivierten Session starten-Buttons.

Folgenden Buttons müssen ebenfalls auf der Controller-Seite zur Verfügung stehen:

- Client-Button:
  - zeigt die Teilnehmer-Seite. Sie stellt den Teilnehmern die Funktionen für Dateneingabe bereit.
- Präsentation-Button:
  - zeigt die Präsentation-Seite. Die Präsentation-Seite wird über dem Beamer angezeigt und präsentiert die eingegebenen Daten von allen Teilnehmern in Echtzeit.
- Ergebnisse-Button:
  - ruft die Ergebnisse-Seite auf. Die Ergebnisse von allen Sessions eines ausgewählten Workshops werden auf der Seite in Form einer Tabelle präsentiert.
- Workshop Beenden-Button:
  - beendet den ausgewählten Workshop und führt den Moderator zu Startseite zurück. Der Workshop soll sich anschließend in der Liste der beendeten Workshops befinden.

Die Controller-Seite eines ausgewählten Workshops soll außerdem dem Moderator die Funktion anbieten, die es ihm erlaubt, den Teilnehmer eine Einladungsmail zur Teilnahme am Workshop zu senden.

## **Teilnehmer-Seite**

Die Teilnehmer-Seite soll jedem Teilnehmer am Workshop die Funktion zur Dateneingabe zu einer gestarteten Session bereitstellen. Der Teilnehmer muss die Möglichkeit haben, sich mit seinem Namen einloggen zu können. Wenn keine Session gestartet ist, soll auf der Teilnehmer-Seite ein Texthinweis wie z.B. "Bitte Warten" eingeblendet werden. Bei einer gestarteten Session steht als Überschrift die Frage der Session und das Eingabefeld wird angezeigt. Falls bereits von der moderierenden Person Kategorien erstellt wurden, sollen die erstellten Kategorien ebenfalls als ein Auswahlménü (Dropdown-Liste) eingeblendet werden. Der Teilnehmer soll seine Ideen nach Kategorien zuordnen können.

Im Eingabe-beenden-Prozess wird die Frage der laufenden Session sowie das Eingabefeld und das Auswahlménü von Kategorien ausgeblendet und stattdessen auf der Teilnehmer-Seite ein Texthinweis wie etwa "Bitte Warten" angezeigt. Der Benutzername und die Funktion zum Ausloggen soll in eine Navigationsleiste positioniert werden.

## **Präsentation-Seite**

Die Präsentation-Seite soll, wie bereits erwähnt, alle Eingaben aller Teilnehmer eines Workshops in Echtzeit präsentieren können. Die Frage der laufenden Session muss gut erkennbar dargestellt werden. Wenn die Session nicht läuft, wird der QR-Code zur Teilnahme am Workshop angezeigt.

Die Präsentation-Seite muss folgende Buttons beinhalten:

- Vollbildmodus-Button:
  - passt die Seite im Vollbildmodus auf dem gesamten Bildschirm an.
- QR-Code anzeigen-Button:
  - blendet den QR-Code zur Teilnahme am Workshop ein.
- QR-Code ausblenden-Button:
  - nur sichtbar, wenn der QR-Code anzeigen-Button getätigt wird.
  - schaltet den angezeigten QR-Code wieder aus.
- Kategorie erstellen-Button:
  - nur sichtbar, nachdem der Eingabe beenden-Button auf der Controller-Seite getätigt wurde.
  - Kategorien für die Sortierung der Ideen werden erstellt. Jede Kategorie hat einen Titel. Der Moderator muss den Titel bearbeiten sowie die Kategorien löschen können.



## Sortierung von Daten nach Kategorien

Der Moderator kann die Daten auf der Präsentation-Seite nach Kategorien sortieren. Die Sortierung soll per Drag & Drop<sup>2</sup> erfolgen. Die Daten, welche nicht sortiert wurden, sollen sich in der Kategorie "unsortiert" befinden. Die Kategorien selbst sollen nicht sortierbar sein. Nach dem Löschen einer nicht leeren Kategorie, müssen die darin befindlichen Daten automatisch nach Kategorie "unsortiert" geordnet werden.

## Ergebnisse-Seite

Nach Ausführen des Ergebnisse-Buttons auf der Controller-Seite eines ausgewählten Workshops soll die Ergebnisse-Seite alle Daten inklusive Kategorien von allen Sessions von diesem ausgewählten Workshop in Form einer Tabelle wiedergeben. Die Seite soll außerdem ein Button besitzen, über diesem der Moderator die Ergebnisse als eine PDF-Datei herunterladen kann.

### 3.3.2 Nicht-funktionale Anforderungen

Im oberen Unterkapitel wurden die funktionalen Anforderungen aufgelistet. In diesem Kapitel werden die nicht-funktionalen Anforderungen formuliert, welchen 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 sollen in unterschiedlichen Farben entsprechend der Funktionalität gestaltet werden.
- Anzeigen von Bestätigungsdialogen beim Löschen von Workshops, Sessions und Kategorien sowie beim Beenden von Workshops.

---

<sup>2</sup>Ziehen und Ablegen

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

### **Plattformübergreifend**

Die Webanwendung soll unabhängig der Plattform funktionieren. Deshalb sollte die Webanwendung so gestaltet werden, dass das Layout der Webseite auf dem Computer, Tablet und Smartphone eine gleichbleibende Benutzerfreundlichkeit anbietet. 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.

### **Browser und Betriebssysteme Unabhängigkeit**

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

### **Anzeigen von Fehlermeldungen und Deaktivieren von Buttons bei nicht vorhandenen Verbindung zwischen Client und Server**

Bei nicht vorhandenen bzw. unterbrochenen Verbindungen zwischen Client und Server soll dem Nutzer einen Hinweistext bereitgestellt werden und folgende Buttons sollten dabei deaktiviert werden:

- Start session-Button
- Ergebnisse-Button
- Workshop Beenden-Button
- Eingabe beenden- sowie Session beenden-Button

Die deaktivierten Buttons sollen bei wiederkehrender Verbindung automatisch reaktiviert werden.

### **Performance**

Die eingegeben Daten seitens der Teilnehmer sollen rechtzeitig und ohne Verzögerung auf der Präsentation-Seite geliefert werden.

### 3.3.3 Muss- und Kann-Anforderungen

Die funktionale sowie nicht-funktionale Anforderungen wurden bereits in Unterkapitel 3.3.1 und 3.3.2 dargestellt. In diesem Kapitel 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.

Merkmal	Anforderung	Priorität
S	Anlegen, Bearbeitung und Entfernen von Workshops	Hoch
S	Anzeigen von beendete Workshops	Hoch
S	Datum und Zeit, wann die Workshops beendet wurden	Hoch
S	Ergebnisse von den beendeten Workshops darstellen	Hoch
S	Benutzerdialog beim Löschen von Workshops	Hoch
S	Hinzufügen von Agenda beim Erstellen von Workshops	Niedrig
S	Plattformübergreifend	Niedrig

Abbildung 3.3: Muss- und Kann-Anforderungen für die Startseite (S)

Merkmal	Anforderung	Priorität
C	Anlegen, Bearbeitung und Entfernen von Sessions	Hoch
C	Benutzerdialog beim Löschen von Sessions	Hoch
C	Hinweistext, wenn die Session nicht läuft	Hoch
C	Aufruf von Teilnehmer-Seite	Hoch
C	Aufruf von Präsentation-Seite	Hoch
C	Aufruf von Ergebnisse-Seite	Hoch
C	Workshop beenden-Funktion	Hoch
C	Start Session-Funktion	Hoch
C	Session beenden-Funktion	Hoch
C	Benutzerdialog beim Beenden von Workshops	Hoch
C	Eingabe beenden-Funktion	Niedrig
C	Agenda anzeigen	Niedrig
C	Einladungsmail zur Teilnahme am Workshop	Niedrig
C	Plattformübergreifend	Niedrig

Abbildung 3.4: Muss- und Kann-Anforderungen für die Workshop Controller-Seite (C)

Merkmal	Anforderung	Priorität
T	Einloggen mit Benutzernamen	Hoch
T	Hinweistext, falls die Session nicht läuft	Hoch
T	Dateneingabe-Funktion	Hoch
T	Plattformübergreifend	Hoch
T	Ausloggen-Funktion	Niedrig

Abbildung 3.5: Muss- und Kann-Anforderungen für die Teilnehmer-Seite (T)

Merkmal	Anforderung	Priorität
P	Anzeigen von Daten in Echtzeit	Hoch
P	Vollbildmodus	Hoch
P	QR-Code zur Teilnahme am Workshop einschalten	Hoch
P	QR-Code ausblenden	Hoch
P	Kategorien erstellen	Hoch
P	Sortierung der Daten per Drag & Drop nach Kategorie	Hoch
P	Bearbeitung und Löschen von Kategorien	Hoch
P	Benutzerdialog beim Löschen von Kategorien	Hoch
P	Plattformübergreifen	Niedrig

Abbildung 3.6: Muss- und Kann-Anforderungen für die Präsentation-Seite (P)

Merkmal	Anforderung	Priorität
Er	Ergebnisse zum Workshop darstellen	Hoch
Er	Ergebnisse als PDF-Datei herunterladen	Hoch
Er	Plattformübergreifend	Niedrig

Abbildung 3.7: Muss- und Kann-Anforderungen für die Ergebnisse-Seite (Er)

Merkmal	Anforderung	Priorität
Sonstige	Benutzerfreundlichkeit	Hoch
Sonstige	Die Buttons sollen in unterschiedliche Farbe entsprechend der Funktionalität gestaltet werden	Hoch
Sonstige	Darstellung von Hilfeleistung in Form von Hilfetexten und Tooltips	Hoch
Sonstige	Einheitliche Gestaltung nach Unternehmensvorgabe	Hoch
Sonstige	Browser und Betriebssystem Unabhängigkeit	Hoch
Sonstige	Deaktivieren von Buttons bei nicht vorhandenen Verbindung zwischen Client und Server	Hoch
Sonstige	Navigation zur Startseite	Hoch

Abbildung 3.8: Sonstige Muss- und Kann-Anforderungen für Webanwendung

### 3.3.4 Schematische Darstellung einer Bearbeitungshierarchie

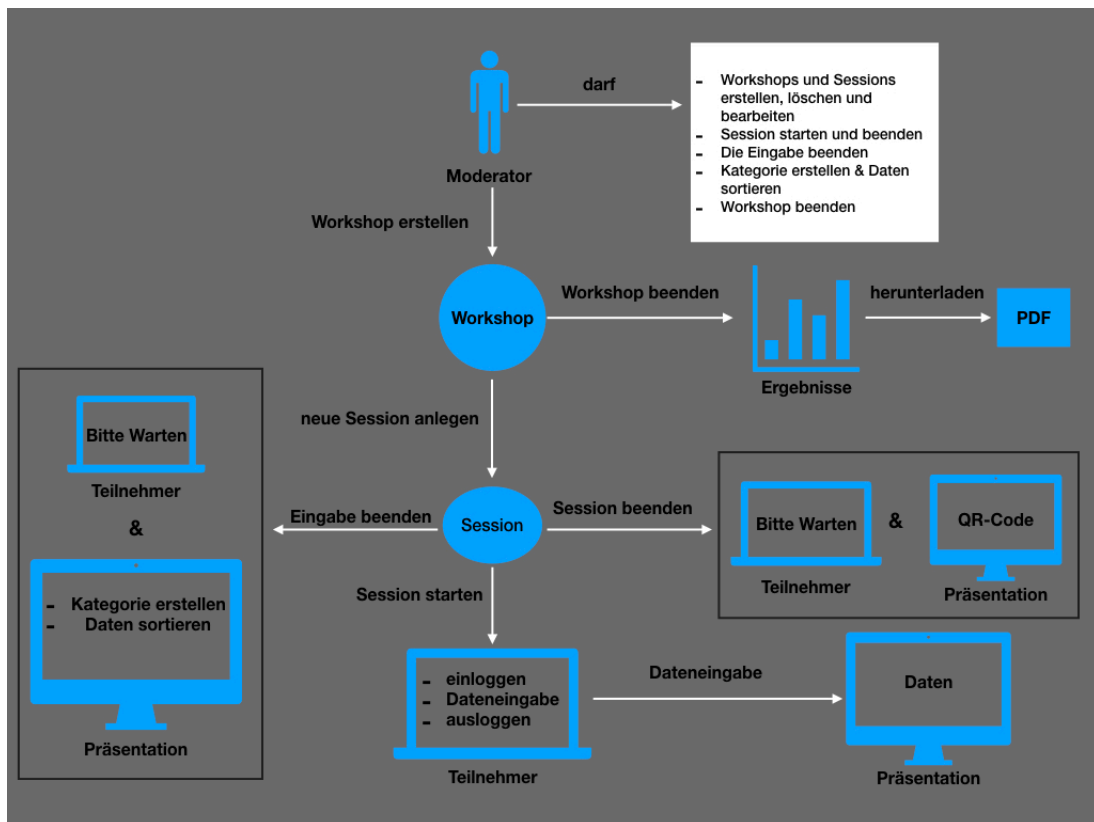


Abbildung 3.9: Darstellung der Bearbeitungshierarchie in der Webanwendung  
[Quelle: eigene Abbildung]

## **4 Design**

## **5 Implementierung**

## **6 Fazit und Ausblick**



# 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).
- [Balzert, 2010] Balzert, H. (2010). *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. 3. Auflage, Spektrum Akademischer Verlag.
- [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).
- [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).
- [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-kompodium.de, URL: <https://www.elektronik-kompodium.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).
- [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).
- [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-Kreativ%C3%A4tstechnik-Ideenfindung.html> (abgerufen am 18.07.2018).
- [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, 2019] Wikipedia (2019). "WebSocket". URL: <https://de.wikipedia.org/wiki/WebSocket> (abgerufen am 02.08.2019).
- [Wöhr, 2004] Wöhr, H. (2004). "*Web-Technologie*". 1.Auflage, dpunkt.verlag, ISBN: 978-3898642477, S.219.