

HuddleUp

HuddleUp - Webplattform für Videotreffen

Bachelorarbeit

Studiengang Informatik
OST - Otschweizer Fachhochschule
Campus Rapperswil-Jona

Frühlingssemester 2021

AUTOREN

Flavio Böni
Severin Hauser

BETREUER

Prof. Frank Koch

EXPERTE

Prof. Hansjörg Huser

GEGENLESER

Dr. Thomas Bocek

Abstract

Pandemie-bedingte Massnahmen wie Home-Office und Fernstudium haben eine Reduktion von persönlichen Kontakten zur Folge. Während Videokonferenz-Plattformen wie Zoom diese Kontakte teilweise ersetzen können, beschränkt sich dies auf bereits bestehende Bekanntschaften. Die Möglichkeiten um neue Menschen kennen zu lernen sind jedoch begrenzt.

Ziel dieser Bachelorarbeit war das Erstellen einer Webapplikation zur Durchführung von Videokonferenzen für einander unbekannt Personen. Sie soll ermöglichen, Menschen mit gemeinsamen Interessen zu finden und zu treffen. Im Rahmen der Arbeit sollten die genauen Anforderungen erfasst, eingesetzte Technologien evaluiert und anschliessend umgesetzt werden.

Auf Basis einer Recherche zu bestehenden Lösungen und Technologien wurden die genauen Anforderungen an die Applikation erfasst. Ein wichtiges Anliegen war, dass die Plattform frei verfügbar ist. Somit wurde darauf geachtet, dass nur Open Source-Bibliotheken und -Technologien eingesetzt werden. Weiter sollte die Applikation durch den Betreiber konfigurierbar sein. Die erarbeiteten funktionellen und nicht-funktionellen Anforderungen bildeten die Grundlage für die Implementation. Während der Entwicklung wurde die Funktionalität anhand automatisierter Unit Tests sichergestellt und die Benutzbarkeit durch Usability Tests überprüft.

Das Endprodukt der Arbeit ist «HuddleUp», eine Webapplikation zur Durchführung von Online-Gruppentreffen. Benutzer können Treffen organisieren, oder an ausgeschriebenen Treffen teilnehmen. Die Suche nach Treffen wird durch die Verwendung von Tags erleichtert. Sobald der Startzeitpunkt eines Treffens erreicht ist, kann der Organisator die Videokonferenz starten. Die Konferenz und die Zugangsdaten werden durch «HuddleUp» automatisch erstellt, wodurch die Durchführung des Treffens erleichtert wird. Als Videokonferenz-Provider wurde Jitsi Meet integriert. Es handelt sich dabei um eine Referenzimplementation und kann in Zukunft durch weitere Provider ergänzt werden.

Im aktuellen Stand bietet «HuddleUp» eine überschaubare Auswahl an Funktionen an und ist in einem kleinen Rahmen einsatzbereit. Es besteht eine Vielzahl von weiteren Features, welche durch den Fokus auf Erweiterbarkeit und Modularität gut umsetzbar sind.

Management Summary

Ausgangslage

Problemstellung Regulärer zwischenmenschlicher Kontakt ist ein Grundbedürfnis von Menschen. Pandemie-bedingte Massnahmen wie Social Distancing, Lockdowns, Home Office, oder Fernunterricht schränken dies stark ein. Teilweise wird der persönliche Kontakt durch Videokonferenzen wie Zoom ersetzt. Doch ein Ersatz für den richtigen Kontakt ist dies nicht. Der virtuelle Kontakt schränkt sich auf bereits bestehende Beziehungen, oder berufliche und schulische Bereiche ein. Es entfällt die Möglichkeiten neue Bekanntschaften zu machen.

Ziel der Arbeit Das Ziel dieser Arbeit war die Erstellung einer Applikation zur Durchführung von Videokonferenzen für einander unbekannte Personen. Benutzer sollten einander auf Grund von Interessen und Themen finden können. Die Treffen sollten durch die Einbindung eines Videokonferenz-Anbieters direkt aus der Applikation gestartet werden können. Die Ausarbeitung der genauen Form dieser Applikation war Bestandteil der Arbeit.

Vorgehen und Technologien

Vorgehen Die Arbeit teilte sich in zwei Phasen auf. In der ersten Phase wurde auf Basis einer Recherche von bestehenden Lösungen die Vision für die Applikation festgelegt. In der zweiten Phase wurden die genauen Anforderungen an die Applikation erfasst und anschliessend umgesetzt.

Bei der Umsetzung wurde ein agiles Vorgehen gewählt, bei welchem die Phasen aus dem **Unified Process** und die iterative Entwicklung aus **Scrum** angewendet wurden. Zur Erfassung der funktionellen Anforderungen wurden Use Cases verwendet, welche zu User Stories heruntergebrochen die Grundlage für Arbeitspakete bildeten. Für die Integration der Videokonferenz-Funktionalität wurden bestehende Provider evaluiert und eine Lösung zur Implementation gewählt.

Technologien Zur Umsetzung wurde eine Unterteilung in zwei Teilapplikationen gewählt: Das Frontend als im Webbrowser ausgeführte **Single Page Application** und eine serverseitige **API**. Beide Teilapplikationen wurden in der Programmiersprache **TypeScript** umgesetzt. Zum Einsatz kommen React¹ und Material-UI² für das User-Interface, NestJS³ für die API und PostgreSQL⁴ als Datenbank. Die Kommunikation zwischen den beiden Applikationen basiert auf GraphQL⁵. Nach der Evaluation bestehender Provider wurde Jitsi Meet⁶ für die Integration von Videokonferenzen gewählt.

¹<https://reactjs.org>

²<https://material-ui.com/>

³<https://nestjs.com>

⁴<https://www.postgresql.org>

⁵<https://graphql.org/>

⁶<https://jitsi.org/>

Ergebnisse

Nach 17 Wochen und 777 Stunden Aufwand wurde mit *HuddleUp* das Ergebnis dieser Arbeit umgesetzt. *HuddleUp* ist eine Webapplikation zur Durchführung von Online-Gruppentreffen. Benutzer können Treffen organisieren, oder an ausgeschriebenen Treffen teilnehmen. Die Suche nach Treffen wird durch die Verwendung von Tags erleichtert. Sobald der Startzeitpunkt eines Treffens erreicht ist, kann der Organisator die Videokonferenz starten. Die Konferenz und die Zugangsdaten werden dabei automatisch erstellt, wodurch die Durchführung des Treffens erleichtert wird.

Das Ziel bei der Umsetzung war, dass *HuddleUp* frei zugänglich und an die Bedürfnisse der Benutzer und des Betreibers anpassbar ist. Als Konsequenz ist das Projekt Open Source und lässt den Benutzern Freiheit bei der Durchführung und Gestaltung der Treffen. Die Applikation ist so ausgelegt, dass möglichst viele Aspekte konfigurier- oder austauschbar sind. Während der Umfang von *HuddleUp* noch überschaubar ist, ist es dennoch in einem kleinen Rahmen einsatzbereit.

Ausblick

Funktionen Es besteht bereits eine Vielzahl an Ideen für weitere, oder den Ausbau bestehender Funktionen. Als besonders wichtig sehen wir folgende Funktionen:

- Bessere Integration des Videokonferenz-Providers
- Implementation von weiteren Videokonferenz-Providern
- Implementation eines Administratoren-Bereichs
- Bessere Integration des Identity Providers

Vision Während *HuddleUp* eingesetzt werden kann, so fehlt zur Zeit jedoch ein Anwender. Es fehlt deshalb noch ein Abgleich zwischen der erarbeiteten Funktionalität und den Bedürfnissen eines richtigen Anwenders, oder Betreibers. Kurzfristig gibt es viele Möglichkeiten für die Verbesserung von *HuddleUp*. Wir denken jedoch, dass langfristige konkrete Benutzer benötigt werden, damit die Richtung einer Weiterentwicklung an deren Bedürfnissen ausgerichtet werden kann.

Inhaltsverzeichnis

I	Technischer Bericht	5
1	Technischer Bericht	6
1.1	Einführung	6
1.2	Bestehende Lösungen	7
1.3	Zielsetzung	10
1.4	Umsetzung	12
2	Resultate	16
2.1	Ergebnisse	16
2.2	Schlussfolgerungen	23
II	Projektdokumentation	25
3	Projektplan	26
3.1	Einführung	26
3.2	Projektübersicht	26
3.3	Projektorganisation	27
3.4	Projektablauf	28
3.5	Risikomanagement	30
3.6	Arbeitspakete	31
3.7	Infrastruktur	31
3.8	Qualitätsmassnahmen	31
3.9	MVP	34
4	Anforderungsspezifikation	35
4.1	Use Cases	35
4.2	Nicht-funktionale Anforderungen	49
4.3	Wichtige Abläufe	50
5	Domänenanalyse	53
5.1	Domänenmodell	54
5.2	Konzepte	54
6	Architektur und Design Spezifikation	56
6.1	Vorlagen	56
6.2	Technologien	56
6.3	Building Block View	58

6.4	Runtime View	63
6.5	Deployment View	65
6.6	Crosscutting Concepts	66
6.7	Architekturentscheidungen	69
7	Risikomanagement	72
7.1	Einführung	72
7.2	Umgang mit Risiken	72
7.3	Identifizierte Risiken	73
	Glossar	77
	Akronyme	80
	Literaturverzeichnis	81
	Abbildungsverzeichnis	82
	Tabellenverzeichnis	82
A	Anhang	84
A.1	Aufgabenstellung	84
A.2	Installationsanleitung und Benutzerhandbuch	90
A.3	Evaluation Videokonferenz-Provider	90
A.4	Systemtests	97
A.5	Systemtest Protokolle	99
A.6	Kommentare zu den Tests	100
A.7	Usability Tests	100
A.8	Usability Test Protokolle	104
A.9	Revisionen der Risikoanalyse	113
A.10	Wireframes	115

Teil I

Technischer Bericht

Kapitel 1

Technischer Bericht

1.1 Einführung

1.1.1 Problemstellung

Regulärer zwischenmenschlicher Kontakt ist ein Grundbedürfnis von Menschen. Pandemiebedingte Massnahmen wie Social Distancing, Lockdowns, Home Office, oder Fernunterricht schränken dies stark ein. Teilweise wird der persönliche Kontakt durch Videokonferenzen wie Zoom ersetzt. Doch ein Ersatz für den richtigen Kontakt ist dies nicht. Der virtuelle Kontakt schränkt sich auf bereits bestehende Beziehungen, oder berufliche und schulische Bereiche ein. Es entfällt die Möglichkeiten neue Bekanntschaften zu machen.

Die Einschränkungen zeigen sich am Beispiel von Studierenden. Der Fernunterricht schränkt die Interaktion mit Mitstudierenden und Dozierenden auf den Studieninhalt ein. Der Austausch in der Pause, oder Anlässe und Aktivitäten von Studentenvereinen finden nicht statt. Es entfällt also eine Grosszahl an Möglichkeiten, Studierende anderer Studiengänge, oder Jahrgänge kennenzulernen. Gerade für Erstsemstrige wird so das Einleben in den Studienalltag erschwert.

Es gibt jedoch Lösungen, um auch online Gesprächspartner zu finden. Eine Vielzahl von Plattformen bieten verschiedene Formen an um mit anderen Menschen in Kontakt zu kommen. Für den Anwender stellen sich dann aber gewisse Herausforderungen. Die Plattformen sind entweder generisch, anonym, oder bieten keine einfache Möglichkeit an, das Videogespräch durchzuführen. Vor allem bieten diese keinen richtigen Ersatz für alltägliche Kontakte. Ein Erstsemstriger weiss so etwa nicht, wo er seine Kommilitonen finden kann.

1.1.2 Ausgangslage

Die Themenausschreibung auf welcher diese Arbeit basiert, beschrieb das Ziel, eine Videokonferenzlösung für einander Unbekannte zu erstellen. Benutzer sollten einander auf Grund von Interessen und Themen finden können. Die Treffen sollten durch die Einbindung eines Videokonferenz-Anbieters direkt aus der Applikation gestartet werden können.

Das Thema war bewusst weit gefasst. Die Absicht war, dass die genaue Form der Applikation - und somit auch die Aufgabenstellung - durch die Studierenden erarbeitet werden.

1.1.3 Vorgehen

Die Durchführung dieser Arbeit lässt sich in zwei Phasen unterteilen. In einer ersten Phase wurden bestehende Lösungen recherchiert, Ideen gesammelt und auf Basis der Erkenntnisse die Aufgabenstellung definiert. In einer zweiten Phase wurde die erarbeitete Aufgabenstellung umgesetzt.

1.1.4 Rahmenbedingungen

Die Rahmenbedingungen für diese Arbeit basieren auf der Aufgabenstellung (Abschnitt A.1) sowie den Guidelines für Studien- und Bachelorarbeiten im Studiengang Informatik an der OST[3].

1.2 Bestehende Lösungen

Zur Erfassung der Anforderungen an die zu entwickelnde Applikation, war es wichtig bestehende Lösungen zu evaluieren. Der folgende Abschnitt beschreibt diese Recherche, die gefundenen Lösungen und unsere daraus gezogenen Erkenntnisse.

1.2.1 Recherche

Ziel

Unsere Absicht bei der Recherche war, einen Überblick zu bestehenden Angeboten zu erhalten, welche dem Ausschreibungsthema entsprechen. Neben der Inspiration für mögliche Funktionen sollte dies auch helfen, nicht bediente Bedürfnisse zu finden und Nachteile bestehender Lösungsansätze zu erkennen.

Umfang

Die Suche nach bestehenden Lösungen war weit gefasst. Zum einen wurden Lösungen gesucht, welche klar dem Ausschreibungsthema entsprechen. Dies umfasst Applikationen, welche themenorientierte Videokonferenzen, oder eine interessenbasierte Benutzervermittlung anbieten. Zusätzlich wurden auch Applikationen gesucht, die zwar nicht explizit Treffen anbieten, aber trotzdem Interessengruppen den Austausch ermöglichen.

1.2.2 Ergebnisse

Unsere Suche zeigte, dass bereits eine grosse Anzahl bestehender Lösungen existiert. Diese unterscheiden sich jedoch stark in der Ausrichtung und Popularität. Die gefundenen Lösungen haben wir in grobe Kategorien unterteilt, auf welche wir nachfolgend eingehen.

Kennenlernen von Einzelpersonen

Unter dieser Kategorie verstehen wir Applikationen, welche sich auf die Interaktion von Einzelpersonen fokussieren. Das Ziel dabei ist, Gespräche zwischen einzelnen Benutzern zu vermitteln.

Vermittlung von Benutzern Die Vermittlung geschieht entweder zufällig, oder auf der Basis von Benutzerprofilen und dort angegebenen Interessen. Während viele der gefundenen Applikationen die Vermittlung auf Basis des Benutzerprofils anpreisen, ist deren Funktion beschränkt. Die Möglichkeiten das eigene Profil anzupassen sind oft auf Alter, Ort, und ein Textfeld zur Angabe der Interessen begrenzt. Die meisten der gefundenen Applikationen können dieser Kategorie zugewiesen werden.

Popularität Die Popularität der gefundenen Applikationen ist schwierig abzuschätzen. Generell erscheinen die Applikationen oft leer, auch wenn hunderte Benutzer aktiv sind. Dies liegt zum Teil daran, dass die Applikationen einen weltweiten Fokus haben. Die Benutzer sind deshalb entweder auf Länder- oder Sprach-spezifische Bereiche verteilt, oder wenig aktiv, da aufgrund von Sprachunterschieden kaum Gesprächspartner vorhanden sind.

Qualität Die Qualität der gefundenen Applikationen ist durchzogen. Während diese zwar technisch oft gut umgesetzt sind, ist der Inhalt oft bedenkenswert. Selbst wenn die Anwendungen eine Registration verlangen, sind die Benutzer grösstenteils anonym. Viele Inhalte sind deshalb vulgär, oder anstössig. Dies wird entweder von den Plattformen gewünscht, toleriert, oder wegen ungenügender Moderation nicht entfernt.

Beispiele Eine Auswahl von Applikationen in dieser Kategorie:

- FaceFlow¹
- Camfrog²
- Emerald Chat³
- PalTalk⁴
- StrangerMeetup⁵
- Nearby⁶

Gruppenorientiert

Unter dieser Kategorie verstehen wir Applikationen, welche sich auf die Bildung und Vermittlung von Interessengruppen fokussieren. Das Ziel dabei ist, dass Benutzer Gruppen für gemeinsamen Interessen bilden können, oder die Treffen unter mehreren Benutzern stattfinden.

Vermittlung von Benutzern Die Vermittlung von Benutzern wird hauptsächlich themenorientiert gelöst. Ein Treffen wird zu einem Thema ausgeschrieben, an welchem Benutzer teilnehmen können. Alternativ wird eine Gruppe zu einem Thema erstellt, in deren Rahmen mehrere Treffen durchgeführt werden. Der Benutzer sucht sich also seinen Interessen entsprechend Gruppen, um an diesen teilzunehmen.

¹<https://www.faceflow.com/>

²<https://www.camfrog.com/en/>

³<https://www.emeraldchat.com/>

⁴<https://www.paltalk.com/>

⁵<https://strangermeetup.com/>

⁶<https://www.wnmlive.com/>

Popularität Diese Kategorie hat weniger Ergebnisse als die vorherige, beinhaltet aber mit *Meetup* die mit Abstand populärste Applikation. Gemäss eigenen Angaben sind über 52 Millionen Mitglieder auf *Meetup* registriert, mit mehr als 100'000 veranstalteten Treffen pro Woche^[10].

Qualität Die Applikationen in dieser Kategorie sind technisch und inhaltlich qualitativ hochstehend. Dies liegt unter anderem daran, dass die meisten schon seit Jahren existieren. Im Vergleich zur vorherigen Kategorie bieten jedoch nur wenige in die Applikation integrierte Videokonferenzen an. In diesen Fällen müssen Konferenzen durch die Benutzer organisiert und der Zugang auf der Applikation hinterlegt werden.

Beispiele Eine Auswahl von Applikationen in dieser Kategorie:

- Meetup⁷
- Living Room Conversations⁸
- Junto⁹ (noch in geschlossener Beta)

1.2.3 Weitere

Diese Kategorie umfasst Applikationen, welche nicht explizit Treffen zwischen Benutzern vermitteln, aber trotzdem den Austausch zwischen Personen mit gemeinsamen Interessen ermöglichen.

Beispiele Eine Auswahl von Applikationen in dieser Kategorie:

- Internetforen (z.B. Discourse¹⁰)
- Soziale Netzwerke
 - Facebook Gruppen
 - Mastodon¹¹
- Discord¹²

1.2.4 Erkenntnisse

Vermittlung von Einzelpersonen ist schwierig

Beobachtung Unsere Suche hat gezeigt, dass eine grosse Anzahl Lösungen bereits existiert. Gerade im Bereich der Vermittlung von Einzelpersonen besteht ein grosses Angebot, wenn auch mit bescheidener Qualität. Keine dieser Applikationen hat eine überzeugende Vermittlungsfunktion. Gründe dafür sehen wir in der Komplexität eines Vermittlungs-Algorithmus und der dafür benötigten Datengrundlage. Eine zufällige Vermittlung als Alternative ist ebenfalls nur bedingt überzeugend.

⁷<https://www.meetup.com>

⁸<https://livingroomconversations.org/>

⁹<https://about.myjunto.app/>

¹⁰<https://www.discourse.org/>

¹¹<https://joinmastodon.org/>

¹²<https://discord.com/>

Erkenntnis Aufgrund der geschilderten Probleme und des begrenzten Zeitrahmens der Arbeit ist ein Fokus auf die Vermittlung von Einzelpersonen schwer umsetzbar.

Kosten

Beobachtung Die meisten gefundenen Applikationen haben entweder kostenpflichtige Funktionen, oder sind durch Spenden finanziert. Dies ist wenig überraschend, denn der Unterhalt einer Applikation verursacht Kosten. Während diese für eine kleine Anzahl Benutzer noch tragbar sind, stellt sich mit steigender Benutzerzahl zunehmend die Finanzierungsfrage.

Erkenntnis Da es keinen Auftraggeber für diese Arbeit gibt, werden die finanziellen Ressourcen für den Betrieb begrenzt sein. Wir müssen deshalb darauf achten, dass unsere Lösung ohne grossen finanziellen Aufwand betrieben werden kann.

Moderation ist wichtig

Beobachtung Mehrere der gefundenen Applikationen hatten nur begrenzte, oder keine Moderation. Kombiniert mit der Anonymität des Benutzers führt dies zu einer unfreundlichen Atmosphäre. Bei den grösseren Applikationen wird dies durch Moderatoren und automatische Systeme abgefangen. Die Implementation solcher Systeme ist jedoch komplex. In vielen Applikationen bestehen jedoch auch Lösungen zur Selbstmoderation. In *Meetup* kann der Organisator zum Beispiel Mitglieder aus der Gruppe entfernen, wenn diese sich unangepasst verhalten.

Erkenntnis Die Moderation von Benutzern muss in unserer Lösung berücksichtigt werden. Wenn möglich sollte die Applikation so ausgelegt werden, dass ein Mass von Selbstmoderation durch die Benutzer selbst gemacht werden kann.

Die Benutzer machen die Plattform

Beobachtung Eine Applikation zur Vermittlung von Benutzern *ohne* Benutzer ist nutzlos. Keine - oder nur wenige - Benutzer haben zur Folge, dass kein Anreiz für neue Benutzer besteht.

Eine weiterer Grund für die Attraktivität einer Plattform ist, wie sich diese den Bedürfnissen der Nutzer anpasst. Werden Themen und Strukturen zu stark vorgegeben, kann dies als zu restriktiv empfunden werden. Ein gutes Beispiel hierfür ist *Meetup*: Die Plattform stellt die Struktur und generelle Themen zur Verfügung. Genaue Themen, die Struktur, der Ablauf und die Durchführung der Treffen ist komplett den Benutzern überlassen.

Erkenntnis Wir müssen einen Anreiz schaffen, dass Benutzer unsere Applikation ausprobieren. Damit die Benutzer auch bleiben, sollte die Applikation nicht zu restriktiv sein, sondern den Nutzern Freiraum bei der Gestaltung lassen.

1.3 Zielsetzung

Ausgehend von der Themenausschreibung, den Erkenntnissen aus unserer Recherche und unseren eigenen Erfahrungen und Ideen, haben wir die Aufgabenstellung (Abschnitt A.1)

für diese Arbeit erfasst. Der folgende Abschnitt dient als Ausführung dazu.

1.3.1 Einführung

Unserer Meinung nach haben wir die meisten Menschen, mit welchen wir bedeutungsvolle Kontakte hatten nicht per Zufall kennengelernt. Wir haben sie in einem bestehenden, *geteilten Rahmen* kennen gelernt. Dieser *bestehende Rahmen* kann eine Hochschule, der Arbeitsplatz, ein Verein, das Militär, oder ein Online-Computerspiel sein. Durch diesen Rahmen entsteht ein gemeinsamer Nenner und somit eine Grundlage, um darauf aufbauend andere Menschen besser kennen zu lernen. Anstatt einander komplett unbekannte Menschen miteinander in Kontakt zu bringen, wollen wir deshalb eine Plattform zur Verfügung stellen, um andere Menschen innerhalb eines geteilten Rahmens besser kennen zu lernen.

Unsere Recherche hat gezeigt, dass die Entwicklung einer Applikation zur Vermittlung von Einzelpersonen für uns ungeeignet ist. Die Anforderungen an die Moderation, den Vermittlungsalgorithmus und die Grösse der Benutzerbasis sind hoch und unserer Meinung nach im Rahmen einer Bachelorarbeit nur ungenügend erfüllbar. Wir haben uns deshalb entschlossen, uns auf die Entwicklung einer gruppenorientierten Applikation zu konzentrieren.

Diese Ausrichtung soll sich auch in dem von uns für das Projekt gewählten Namen zeigen: *HuddleUp* leitet sich aus dem Englischen «to huddle up» ab und bedeutet «to gather in a close-packed group»[4].

1.3.2 Vision

Das Ziel unserer Arbeit soll die Entwicklung einer Applikation sein, welche bestehende *geteilte Rahmen* nicht ersetzt, sondern ergänzt. Am Beispiel von Studierenden soll zusätzlich zu den bestehenden Kommunikationskanälen eine Plattform zum Austausch unter Mitstudenten geboten werden. Die Kernaufgaben der Applikation sind dabei die Funktionalität zur Organisation und Teilnahme an Video-Treffen. Die Applikation soll an die Bedürfnisse der Benutzer anpassbar und erweiterbar sein: Keine vorgegebene Themen und Strukturen für Treffen, Betrieb auf eigener Infrastruktur und Open Source.

1.3.3 Funktionsumfang

Die Funktionalität dreht sich um das Organisieren von und die Teilnahme an Treffen. Die Hauptfunktionen sind somit das Erstellen, die Suche sowie die Durchführung von Treffen und die Teilnahme daran. Tags werden zur Kategorisierung von Treffen eingesetzt und helfen somit den Benutzer bei der Suche und Einschätzung von Treffen. Um nicht nur einen Namen zu sehen, können Benutzer in ihrem Profil ihre für alle Benutzer sichtbare Biografie erfassen. Weiter soll die Applikation durch den Betreiber konfigurierbar sein um sie seinen Bedürfnissen anpassen zu können.

1.3.4 Anforderungen

Die Offenheit der Applikation stellt mehrere Anforderungen damit sie auch von möglichst vielen Benutzern verwendet werden kann:

Mehrsprachigkeit Damit die Applikation möglichst vielen Benutzer zugänglich ist, soll diese übersetzbar sein.

Responsive Design Die Applikation soll sowohl auf Desktop- wie auch Mobile-Geräten verwendbar sein.

Austauschbarkeit Die Lösung zur Authentifikation und der Videokonferenz-Provider sollen austauschbar sein.

Konfiguration Die Applikation soll durch den Betreiber konfiguriert werden können.

1.3.5 Einschränkungen

Durch die zeitliche Limitierung der Bachelor-Arbeit besteht die Einschränkung, dass nur ein Videokonferenz-Provider integriert werden soll. Die Implementation soll jedoch eine zukünftige Ergänzung um zusätzliche Provider ermöglichen.

1.4 Umsetzung

Der folgende Abschnitt bietet einen Überblick über die Methodik und das Vorgehen während der Arbeit. Mehr Informationen zum Vorgehen können dem Projektplan (Kapitel 3) entnommen werden.

1.4.1 Phasen

Für die Umsetzung der Arbeit orientierten wir uns an der Methodik von SCRUM+[7], wobei es sich um eine Kombination der Projektphasen des **Unified Process (UP)** und der iterativen Entwicklung von **Scrum**. Die Methodik ist uns aus dem von Prof. Daniel Keller unterrichteten Modul Software Engineering 2 bekannt. Zusätzlich zu Phasen und Iterationen wurden Meilensteine verwendet um wichtige Zeitpunkte im Projektverlauf zu definieren.

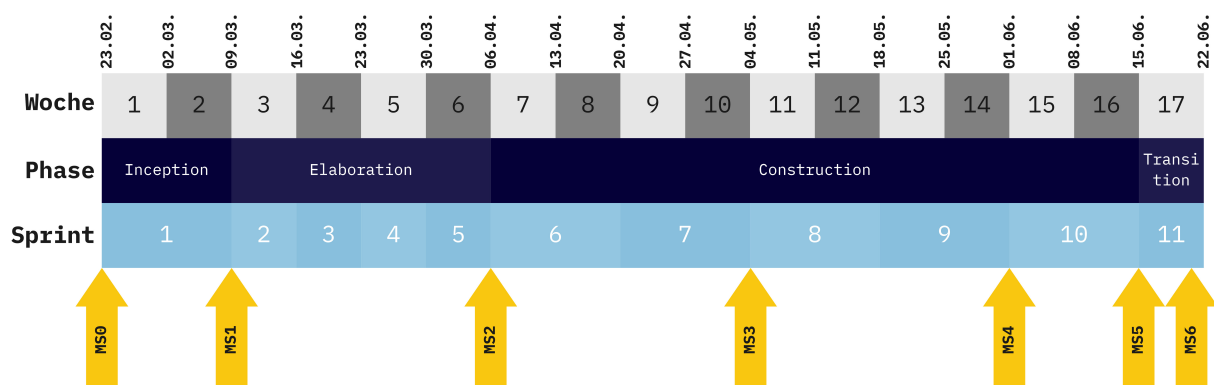


Abbildung 1.1: Übersicht über Phasen, Sprints und Meilensteine

Die vier UP-Phasen teilten sich in folgende Arbeit auf:

Inception Recherche bestehender Lösungen, Erarbeiten der Aufgabenstellung

Elaboration Evaluation von Technologien, Erfassen von Anforderungen, Wireframes-Entwürfe, Prototyping

Construction Implementation und Überprüfung

Transition Abschluss der Arbeit

Wir hatten mit dieser Methodik bereits während dem Engineering Projekt und der Studiarbeit Erfahrungen gemacht. Diese hatten uns jedoch gezeigt, dass die strikte Durchführung von Sprints nach Scrum abhängig von der Phase nur bedingt effektiv war. Als Folge wurden nur die Iterationen in der Phase *Construction* strikt nach Scrum umgesetzt.

1.4.2 Anforderungen

Funktionale Anforderungen

Die funktionalen Anforderungen erfassten wir in der Form von **Use Cases**. Um deren Umsetzungsreihenfolge festzulegen, wurde diese zusätzlich priorisiert. Use Cases mit erster Priorität wurden im *Fully Dressed*-Format erfasst. Use Cases mit zweiter Priorität wurden zu Beginn im *Casual*-Format erfasst und vor der effektiven Umsetzung ins *Fully Dressed*-Format erweitert. Use Cases mit dritter Priorität wurden nur im *Brief*-Format erfasst. Die vollständigen funktionalen Anforderungen können in der Anforderungsspezifikation (Abschnitt 4.1) gefunden werden.

Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen wurden anhand von **FURPS** kategorisiert. Die ausführlichen nicht-funktionalen Anforderungen können in der Anforderungsspezifikation (Abschnitt 4.2) gefunden werden.

1.4.3 Planung

Arbeitspakete

Zur Planung haben wir die Use Cases zu **User Storys** verfeinert, die während der Entwicklung als Arbeitspakete dienen. Aufgaben, die keinen direkten Bezug zu einem Use Case, oder einer User Story hatten, wurden als separate Tasks erfasst.

Tooling

Die gesamte Planung wurde auf **YouTrack** verwaltet. Die starke Ausrichtung auf agile Entwicklungsmethoden und die hohe Konfiguration konnte es gut an unsere Bedürfnisse angepasst werden. Die integrierte Zeiterfassung erlaubte uns im Vergleich zu vorherigen Arbeiten nicht nur die Planung, sondern auch die Schätzung, Erfassung und das Controlling des Zeitaufwands in der gleichen Applikation durchführen.

1.4.4 Design

Wireframes

Zur Definition des Benutzer-Interfaces wurden während der *Elaboration*-Phase Wireframes erstellt. Durch die Verwendung von **Figma** konnten wir die Ansichten nicht nur entwerfen,

sondern dank der Protoyping-Funktion auch interaktiv testen.

Die Wireframes dienten als Grundlage für die Implementation, das exakte Design wurde jedoch nicht vorgegeben. Während dies zwar zu einem erhöhten Aufwand während der Implementation führte, ersparte es uns den Aufwand für hochwertige Designvorlagen sowie erlaubte es uns schneller und flexibler Anpassungen vorzunehmen. Eine vollständige Übersicht erstellter Wireframes können im Anhang (Abschnitt A.10) gefunden werden.

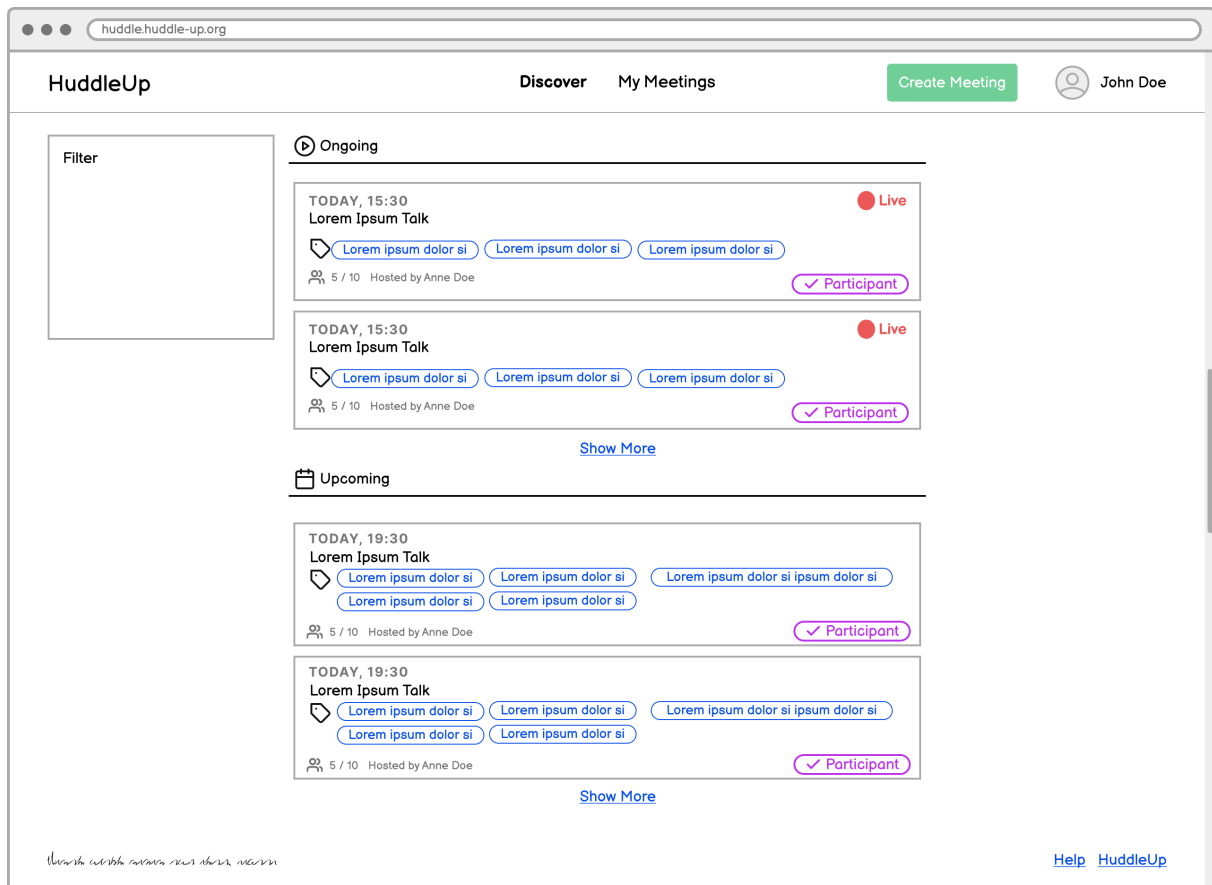


Abbildung 1.2: Wireframe für die Übersicht aller Treffen

Material Design

Da die Erstellung einer eigenen Design-Sprache mit hohem Zeitaufwand verbunden ist, haben wir uns an **Material Design** zu orientiert und versucht möglichst viele Komponenten daraus zu verwenden.

1.4.5 Entwicklung

Versionskontrolle

Aufgrund unserer bisherigen Erfahrungen mit GitHub entschlossen wir uns dies auch für die Entwicklung von *HuddleUp* zu verwenden. Dass GitHub äusserst populär für Open Source-Projekte ist, war ein ein weiterer Grund für den Entschluss. Mit GitHub Actions¹³ konnte zusätzlich die automatische Ausführung von Tests sichergestellt werden.

¹³<https://github.com/features/actions>

Zur Verwaltung des Applikations-Codes haben wir uns für ein **Monorepository** entschlossen. Die Verwaltung von Front- und Backend im gleichen Repository hat uns mehrere Vorteile geboten:

Schichtenübergreifende Änderungen Features können für Front- und Backend im gleichen Repository getätigt werden. Pull Requests und Code Reviews sind so leichter durchzuführen.

Teilen von Logik und Konfiguration Grund-Konfigurationen müssen nur an einem Ort verwaltet werden und das Einbinden von geteilter Logik ist vereinfacht.

Release Management Die Versionen von Teil-Projekten müssen nicht über mehrere Repositories abgeglichen werden.

Vorgehen

Für die Entwicklung richteten wir uns an den GitHub-Flow^[1]. Bei diesem Vorgehen befindet sich der aktuelle Stand auf dem *Main*-Branch und Änderungen werden auf *Feature*-Branches erfasst. *Feature*-Branches werden über **Pull Requests** in den *Main*-Branch zurück geführt, auf dem *Main*-Branch werden keine direkten Änderungen gemacht. Pull Requests sind dabei auch der Zeitpunkt, an welchem Code Reviews durchgeführt und die Test in der **CI** ausgeführt werden.

Testing

Die Funktionalität und Benutzbarkeit der Applikation haben wir auf verschiedene Weisen überprüft und sichergestellt:

Unit Tests Unit Tests dienen der Sicherstellung der korrekten Funktion von Komponenten. Diese wurden während der Entwicklung geschrieben und bei Pull Requests automatisch durch die GitHub Actions **CI** ausgeführt. Da Änderungen im Frontend oft grösser sind und mit grösserem Aufwand für Tests verbunden sind, wurden Unit Tests primär für das Backend geschrieben.

Usability Tests Am Ende des zweitletzten Sprints der *Construction*-Phase wurden Usability Tests durchgeführt. Dies diente uns zur Überprüfung, ob Endanwender die Applikation verstehen und bedienen können. Die Erkenntnisse aus diesen Tests konnten im letzten Sprint zur Korrektur der grössten Probleme umgesetzt werden.

Systemtests Um die Applikation in ihrer Gesamtheit zu überprüfen wurde am Ende der *Construction*-Phase ein Systemtest durchgeführt.

Kapitel 2

Resultate

2.1 Ergebnisse

2.1.1 Erreichte Funktionalität

Alle definierten nicht-optionalen Funktionen wurden während der 10-wöchigen Entwicklungszeit umgesetzt. Die nachfolgenden Abschnitte bieten einen Überblick über die umgesetzten Funktionen von *HuddleUp*.

Startseite

Die Startseite ist der erste Berührungspunkt eines neuen Benutzers mit *HuddleUp*. Es dient als Ausgangspunkt für die Anmeldung und bietet dem Benutzer die Möglichkeit sich über die Applikation zu informieren.

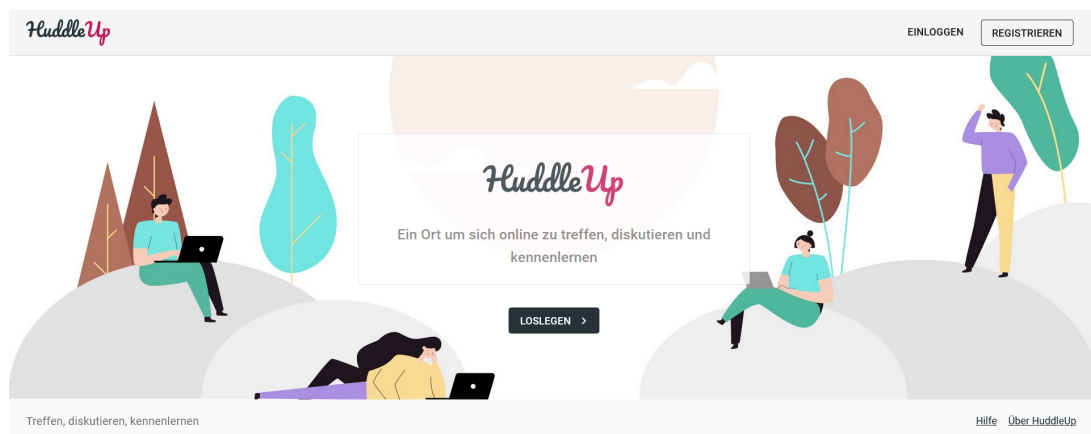


Abbildung 2.2: Startseite

Benutzerprofil

Bei der ersten Anmeldung wird für den Benutzer ein Benutzerprofil erstellt. Das eigene Benutzerprofil kann jederzeit durch den angemeldeten Benutzer bearbeitet und falls gewünscht auch wieder gelöscht werden. Über öffentliche Benutzerprofile können Benutzer mehr über die Mitglieder und den Organisator eines Treffens erfahren.

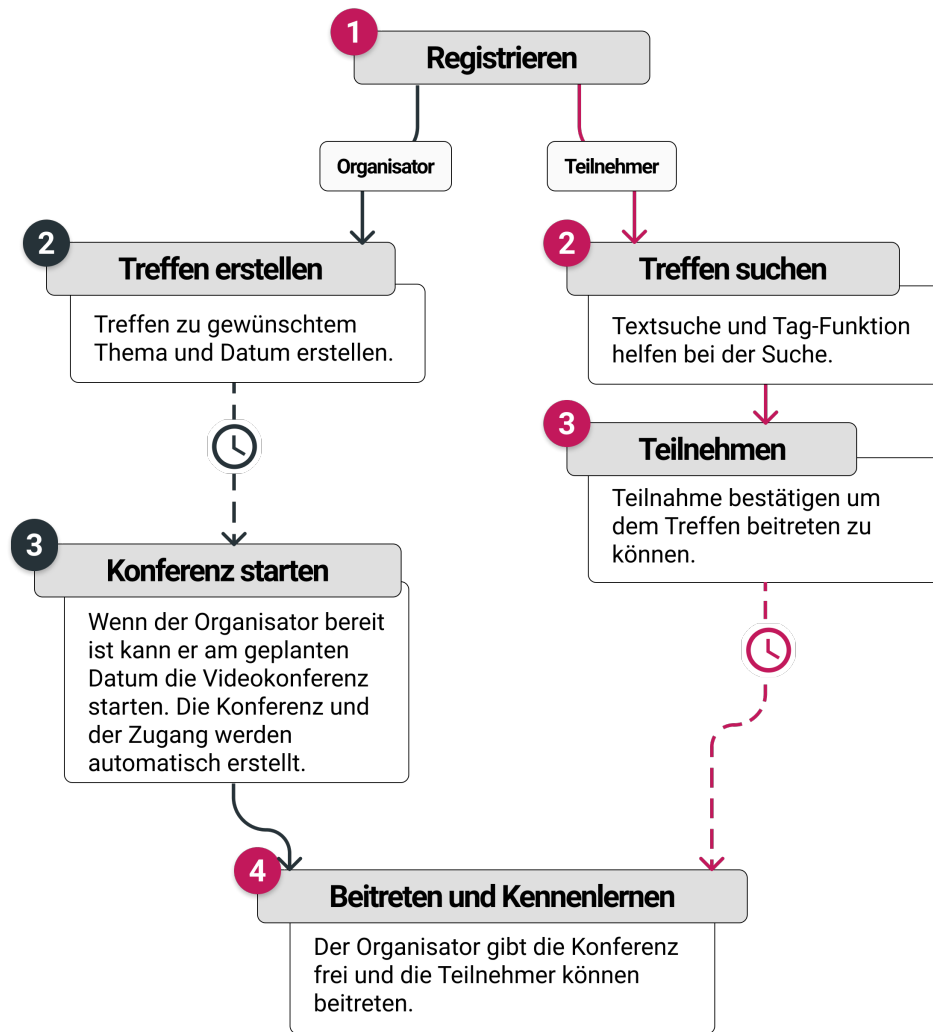


Abbildung 2.1: Vereinfachter Ablauf für Benutzer

Teilnahme an Treffen

Nach der Anmeldung sieht der Benutzer eine Übersicht über bereits laufende und anstehenden Treffen. Bei der Übersicht wurde darauf geachtet, das Finden und Entdecken von Treffen zu vereinfachen.

Treffen suchen Zur Suche von Treffen dient eine Filter-Komponente mit den folgenden Optionen:

- Such-Feld für Textsuche nach Titel oder Beschreibung eines Treffens
- Tag-Feld für die Suche nach Kategorien in Form von Tags
- Datum-Feld für die Suche nach Treffen ab einem gewissen Zeitpunkt
- Datum-Feld für die Suche nach Treffen bis zu einem gewissen Zeitpunkt

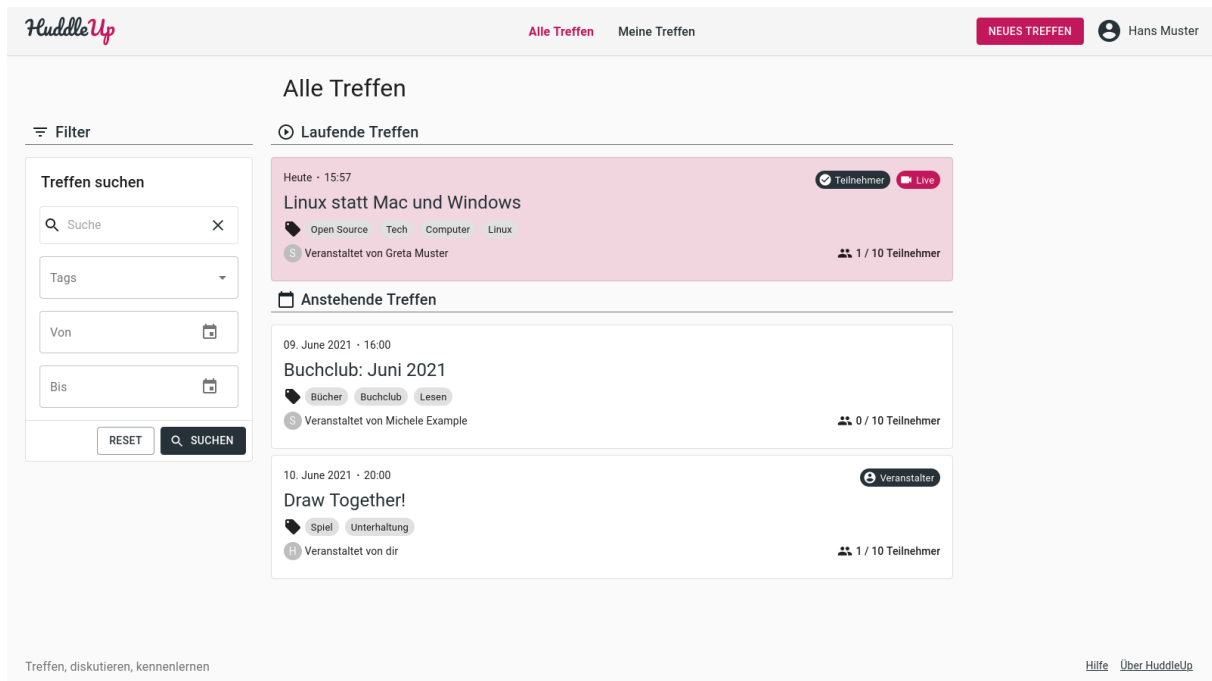


Abbildung 2.3: Übersicht von laufenden und anstehenden Treffen

Teilnahme an Treffen Ein Benutzer kann sich bei einem laufenden, oder anstehenden Treffen als Teilnehmer registrieren. Die Teilnahme ist jedoch nur möglich solange die maximale Teilnehmeranzahl eines Treffens noch nicht erreicht ist. Sobald der Organisator die Videokonferenz freigegeben hat, kann der Teilnehmer dieser über einen automatisch generierten Zugang beitreten.

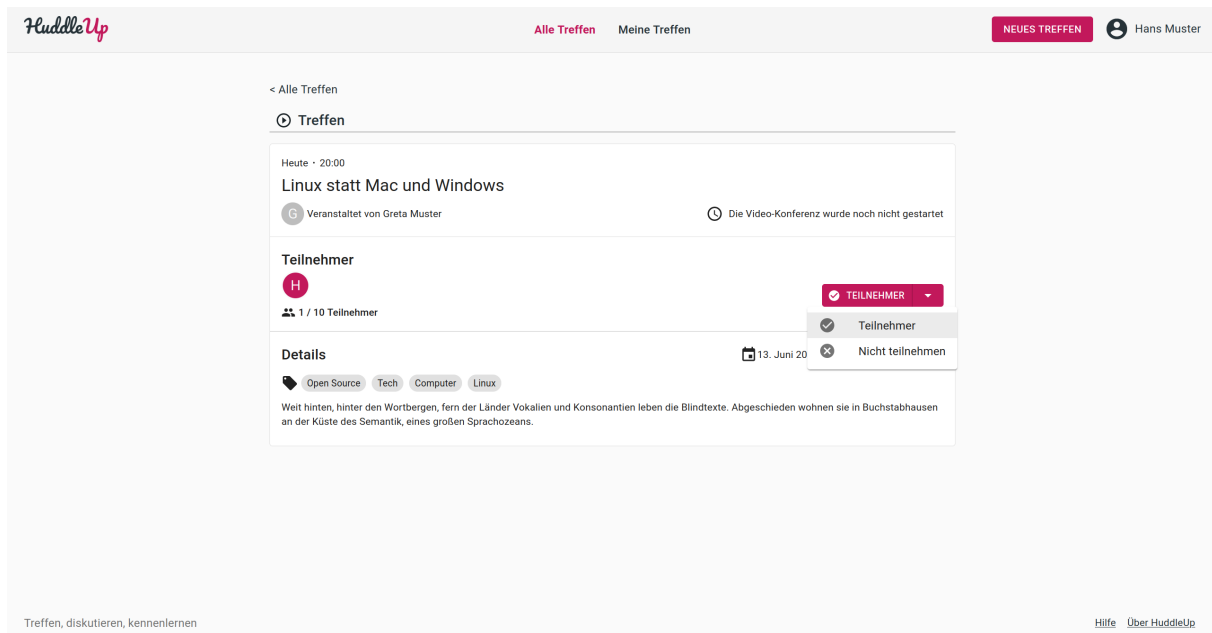


Abbildung 2.4: Detailansicht eines Treffens als Teilnehmer, mit Teilnahmeselektion

Erstellen von Treffen

Jeder Benutzer kann eigene Treffen erfassen. Ein Treffen hat Titel, Beschreibung, Start- und geplantes End-Datum und kann durch die Erfassung eigener Tags, oder der Auswahl von bestehenden Tags, kategorisiert werden. Weiter kann die maximale Anzahl Teilnehmer angegeben werden. Bis zur Durchführung kann der Organisator das Treffen bearbeiten und absagen. Alle Treffen bei der ein Benutzer Teilnehmer, oder Organisator ist, werden unter der Ansicht *Meine Treffen* dargestellt.

Durchführung des Treffens

Die Videokonferenz wird mit dem ersten Zugriff des Organisators erstellt. Die Videokonferenz läuft nicht direkt im Treffen der Webplattform, sondern wird in einem neuen Fenster gestartet. Im Falle von *Jitsi* wäre eine Integration über eine Iframe möglich, würde aber zusätzlichen Provider-spezifischen Code im Frontend benötigen. Da wir uns für einen generischen Ansatz entschieden haben, wird die Videokonferenz deshalb in einem neuen Fenster geöffnet.

Der Ablauf zur Durchführung eines Treffens ist in vier Schritte aufgeteilt. Damit der Prozess für den Benutzer verständlich ist, wird dieser in einer Schritte-Ansicht dargestellt.

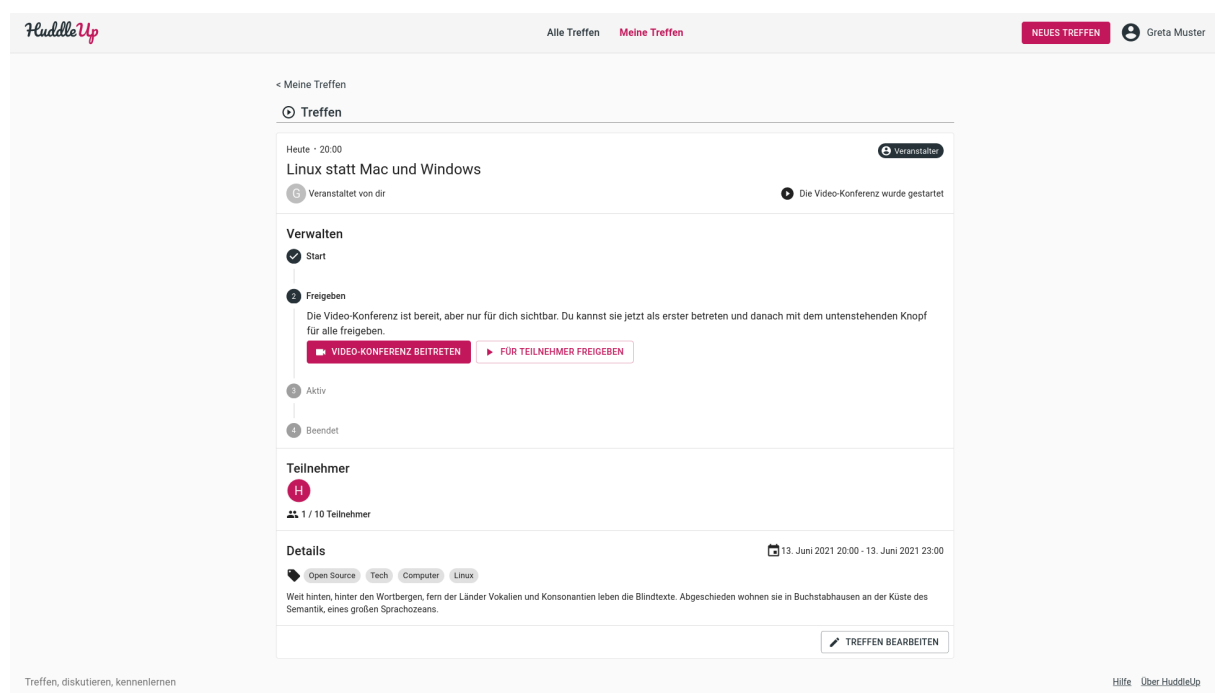


Abbildung 2.5: Detailansicht eines gestarteten Treffens als Organisator

Videokonferenz starten Kurz vor dem Startdatum kann der Organisator den Prozess für die Durchführung der Videokonferenz starten.

Videokonferenz freigeben Der Organisator kann im nächsten Schritt den Teilnehmern den Zugang freischalten. Erst ab diesem Zeitpunkt ist die Videokonferenz für alle als laufend sichtbar. Ein laufendes Treffen wird für alle Benutzer rot hervorgehoben und als *Live* gekennzeichnet. Die registrierten Teilnehmer können nun der Videokonferenz beitreten.

Videokonferenz aktiv Die Videokonferenz ist solange aktiv wie es Teilnehmer in der Videokonferenz hat. Da wir nicht bei jedem Videokonferenz-Provider die aktiven Teilnehmer auslesen können, müssen wir das Beenden des Treffens in unserer Webplattform separat durchführen. Dies ist auch der Fall bei dem von uns eingesetzten Jisti Videokonferenz-Provider. Der Veranstalter muss deshalb das Treffen in der Webapplikation nach der Durchführung selbst auf beendet setzen. Damit wird der Zugriff über die Webplattform nicht mehr ermöglicht. Spätestens zum geplanten Enddatum wird der Zugang über die Webapplikation automatisch verhindert. Ein erneuter Zugriff ist dann über die Webplattform nicht mehr möglich, auch wenn die Videokonferenz noch aktiv ist.

Videokonferenz beendet Das Treffen wird als beendet gekennzeichnet und es ist keine Teilnahme mehr möglich.

2.1.2 Technologie

Sprachen und Frameworks

Für die Implementation von *HuddleUp* haben wir uns für eine Separation von Benutzer-Interface und Serverlogik entschieden. Das Benutzerinterface wurde deshalb als Single Page Application umgesetzt, welches über eine **GraphQL**-Schnittstelle mit dem Backend kommuniziert. In beiden Teilapplikationen wurde *TypeScript*¹ als Programmiersprache eingesetzt.

Backend Zur Implementation des Backends wurde *NestJS*² als Framework gewählt. Die ausführliche Dokumentation, der erstklassige *TypeScript*-Support, die modulare Architektur und die grosse Anzahl unterstützter Technologien waren dabei die ausschlaggebenden Punkte. Als Datenbank kommt *PostgreSQL* zum Einsatz mit *TypeORM*³ als Schnittstelle zur Applikation.

Frontend Für die Implementation des Frontends wurde *React*⁴ als Framework gewählt. Neben der Popularität und dem umfangreichen Ökosystem waren die vorhandenen Erfahrungen beider Autoren mit dem Framework der Grund für die Wahl. Da wir uns für das Design an **Material Design** orientierten, setzten wir *Material-UI*⁵ als Komponenten-Bibliothek ein.

GraphQL Für die Kommunikation zwischen Front- und Backend stellte sich die Frage zwischen **REST** und *GraphQL*. Trotz fehlender Erfahrungen entschieden wir uns für die Umsetzung mit *GraphQL*, wobei *Apollo*⁶ in beiden Teilapplikationen als Client, bzw. Server zum Einsatz kam. Gründe für die Wahl waren die Komplexität der erwarteten Abfragen, die Caching- und State Management-Kapazitäten von *Apollo* sowie die Möglichkeit zur Synchronisation der Schema-Defintion zwischen Front- und Backend.

¹<https://www.typescriptlang.org/>

²<https://www.nestjs.com>

³<https://typeorm.io>

⁴<https://reactjs.org>

⁵<https://material-ui.com/>

⁶<https://www.apollographql.com/docs/>

Während die Wahl von *GraphQL* viele Vorteile geboten hat, so zeigte sich dennoch die fehlende Erfahrung. Vor allem die richtige Konfiguration des Caches von *Apollo* verursachte unerwartete Verzögerungen.

Anmeldemechanismus

Die Anforderung, dass der Anmeldemechanismus austauschbar sein soll, war schwieriger umsetzbar als erwartet. Zu Beginn war unsere Idee, dass verschiedene Identity Provider und Protokolle möglich sein sollen (z.B. **OAuth**, **SAML**, lokale Anmeldung mit E-Mail und Passwort). Die Protokolle stellen jedoch unterschiedliche Anforderungen, wie eigene Datenbankschemata (z.B. für das Speichern von Benutzernamen und Passwörtern), die Verwendung von Cookies und serverseitigen Sessions, oder das serverseitige Generieren von HTML-Fragmenten (z.B. für SAML). Diese Anforderungen stehen entweder im Konflikt mit bestehenden, oder erhöhen die Komplexität der Implementation massiv. Eine Lösung für das Problem wäre eine separate Teilapplikation, deren spezifische Implementation durch eine vorgegebene **REST**-Schnittstelle abstrahiert wird. Dies wäre im Rahmen dieser Arbeit aus Zeitgründen jedoch nicht umsetzbar gewesen.

Wir haben uns deshalb entschlossen, den Anmeldungmechanismus auf **OpenID Connect** zu beschränken. OpenID Connect basiert auf OAuth, ist weit verbreitet und wird von populären Identity Providern wie Google[11] unterstützt. Alternativ zu bestehenden Providern kann auch ein kompatibler Service wie Auth0⁷ verwendet werden, welcher neben einer Passwort-basierten Authentifizierung auch Integrationen für andere Protokolle wie z.B. SAML anbietet[12].

Authentifikation

Um den verwendeten Anmeldemechanismus von der restlichen Applikation zu separieren, wird die Authentifikation mit selbst ausgestellten **JWT** durchgeführt. Der Ablauf ist somit, dass im Frontend die Authentifikation mit dem konfigurierten OpenID Connect Provider durchgeführt wird. Der vom Provider erhaltene Identity Token wird dann zur Identifikation bei unserem Backend verwendet, welches als Antwort ein generiertes Access Token zurücksendet. Dieses Token wird dann zur Authentifikation bei Aufrufen an das Backend verwendet.

Bei den Access Token fehlt zur Zeit noch die Möglichkeit diese zu erneuern. Läuft ein Access Token ab, muss der Benutzer deshalb sich erneut über den Identity Provider anmelden. Dies kann durch die Verwendung von langlebigen Refresh Tokens gelöst werden, welche zur Erneuerung der Access Tokens verwendet werden. Diese Funktionalität war in unserem Backlog erfasst, konnte jedoch aufgrund von Zeitmangel nicht umgesetzt werden.

Einbindung Videokonferenz-Provider

Jitsi Meet Nach einer Evaluation von Videokonferenz-Providern (A.3) wurde **Jitsi Meet** für die Integration gewählt. Die Wahl fiel auf Jitsi Meet, weil es weit verbreitet und Open Source ist. Ein weiterer Grund war, dass es ein kostenfreie öffentliche Instanz⁸ gibt, welche explizit auch für die Einbindung in Applikationen verwendet werden darf[6]. Kombiniert

⁷<https://www.auth0.com>

⁸<https://meet.jit.si>

mit der einfachen Installation für eine selber betriebene Instanz, war Jitsi Meet für uns deshalb die Lösung, die die Bedürfnisse der meisten Benutzer abdecken sollte.

Einbindung Im Vergleich zu anderen Providern bietet Jitsi Meet keine API zur Verwaltung von Konferenzen. Ein Meeting wird erstellt, wenn dessen URL aufgerufen wird. Dies ermöglichte zwar eine einfache Einbindung, erschwerte aber auch die Kontrolle über ein Treffen. Uns waren folgende Eigenschaften bei der Einbindung wichtig:

- Nur Teilnehmer eines Treffens sollten dessen Konferenz beitreten können
- Es muss sichergestellt sein, dass der Organisator Moderatorenrechte erhält
- Teilnehmer sollten in der Konferenz identifizierbar sein
- Es soll für unser Applikation möglich sein, den Status eines Treffens abzufragen

Bis auf den letzten Punkt konnten wir alle Eigenschaften erfüllen. Dank der Modularität von Jitsi Meet konnte durch die Verwendung der JWT-Funktionalität⁹ die Zugriffsrechte und die Identität der Teilnehmer sichergestellt werden. Die Moderatorenrechte konnten garantiert werden, indem der Organisator dem Treffen als erstes beitrifft und somit automatisch die Rolle des Moderators erhält. Diese Funktionen sind jedoch nur auf einer selbst gehosteten Jitsi-Instanz möglich, da die öffentliche Instanz diese nicht aktiviert hat.

Weitere Provider Im Rahmen dieser Arbeit wurde nur eine Integration für Jitsi Meet umgesetzt. Bei der Implementation wurde darauf geachtet, dass weitere Provider mit wenig Aufwand integriert werden können. Dies wurde sichergestellt, indem der Jitsi-spezifische Service nicht direkt, sondern über ein generisches Interface aufgerufen wird. Ob diese Separation ausreichend ist um auch weitere Provider integrieren zu können, wird sich jedoch erst bei einer entsprechenden Implementation zeigen.

Konfiguration

Mehrere Aspekte von «HuddleUp» können über Konfigurations-Dateien angepasst werden. Dabei handelt es sich entweder um Environment- (.env), oder JSON-Dateien. Neben den grundlegenden Attributen wie Datenbank-Zugänge können hier auch Identity- und Videokonferenz-Provider, sowie die Default-Sprache konfiguriert werden.

Während die wichtigsten Eigenschaften konfigurierbar sind um «HuddleUp» in einer eigenen Installation in Betrieb nehmen zu können, so sind die Möglichkeiten noch begrenzt. Zum einen kann der Betreiber nicht beeinflussen, wer und wann Treffen veranstalten kann und darf. Weiter ist die Personalisierung nur begrenzt, oder umständlich möglich. Der Betreiber kann zum Beispiel keine eigenen Logos, oder Texte einfügen. Diese Funktionen und eine einheitlichere Konfiguration konnten aus Zeitgründen leider nicht umgesetzt werden.

2.1.3 Nicht erreichte Funktionalität

Die in der Aufgabenstellung definierten Anforderungen konnten alle umgesetzt werden. Das Ergebnis der umgesetzten Funktionen genügt somit unseren gesteckten Zielen und den Anforderungen eines kleinen Betreibers. Es gibt jedoch noch viele weitere Funktionen und zusätzliche Anforderungen, welche in dieser Arbeit nicht abgedeckt wurden. Aufgrund

⁹<https://jitsi.github.io/handbook/docs/devops-guide/secure-domain>

der Absenz eines konkreten Endanwenders war es uns nur begrenzt möglich die Produktionstauglichkeit abzuschätzen. Es ist deshalb zu erwarten, dass bei einem produktiven Einsatz von uns nicht berücksichtigte Anforderungen ersichtlich werden.

2.2 Schlussfolgerungen

2.2.1 Bewertung der Ergebnisse

Nach 17 Wochen und 777 Arbeitsstunden konnte diese Bachelorarbeit mit der Umsetzung von *HuddleUp* abgeschlossen werden. In dieser Zeit haben wir ausgehend von der Themenschreibung «Videokonferenzen für einander Unbekannte» die Idee und Anforderungen an die Applikation ausgearbeitet und anschliessend umgesetzt. Während das weit gefasste Thema viel Raum für unsere eigenen Ideen und Entscheidungen liess, war dies in gewissen Bereichen auch negativ zu spüren.

Um ein realistisches Ziel zu setzen, versuchten wir bei der Ausarbeitung der Aufgabenstellung nicht zu viele Funktionen als zwingend erwartet zu erfassen. Dass wir am Ende der Arbeit alle erwarteten Funktionen umsetzen konnten, aber keine der optionalen, zeigt dass wir in unserer Vorsicht berechtigt waren.

Unsere Qualitätsanforderungen betreffend Austauschbarkeit und Konfiguration wurden hingegen zu einer Herausforderung. Sie stellten zu Beginn weg hohe Anforderungen in der Implementation: Diese sollte nicht nur funktionieren, sondern auch durch den Benutzer austauschbar sein. Aus Zeitgründen setzten wir jeweils nur einen Provider pro Funktion um. Es war deshalb schwierig diese so zu gestalten, dass mit den definierten Schnittstellen auch weitere Provider umsetzbar sind. Ein gefundenes Fressen für den Code Smell «Speculative Generality»[13].

Dennoch sind wir mit dem Ergebnis der Arbeit zufrieden. Für die aus unseren Anforderungen entstandenen Herausforderungen konnten wir jeweils eine Lösung finden, auch wenn diese weniger umfangreich ist als die ursprünglich vorgesehene. Uns war es dabei jeweils wichtig, dass das Endprodukt auch benutzbar ist und priorisierten deshalb eine gute Basisfunktionalität. Die gesetzten Ziele von Modularität und Erweiterbarkeit konnten somit sichergestellt werden.

Ein weiteres wichtiges Anliegen für uns war, dass die Applikation benutzerfreundlich ist. Dies zeigt sich zum Beispiel auch am User Interface, welches in gewissen Bereichen mehrere Iterationen durchlief und in Usability Tests überprüft wurde.

2.2.2 Risikomanagement

Das Risikomanagement hat sich als ein nützliches Instrument herausgestellt und einige Risiken mit erhöhtem Schaden aufgezeigt. Aufgrund der während der Arbeit relevanten Pandemie war das Risiko *Ausfalls eines Teammitglieds* ein stark berücksichtigtes. Durch regelmässige Pushs von Code auf GitHub und einem kontinuierlichen Wissensaustausch, konnte das Risiko mitigiert werden.

2.2.3 Fazit

Trotz unseren zu Beginn hohen Ambitionen und den daraus entstandenen Herausforderungen, konnten wir die gesetzten Ziele dieser Arbeit erfüllen. Sie stellte uns Anforderungen in mehreren Disziplinen und erlaubte uns somit, gelerntes Wissen aus vergangenen Semestern vielfältig anzuwenden. Die von uns gewählten Technologien haben sich bewährt und erlaubten uns eine modulare und erweiterbare Applikation umzusetzen. Der Fokus auf eine benutzbare und benutzerfreundliche Applikation hat ermöglicht, dass *HuddleUp* in kleinem Rahmen verwendbar ist.

Rückblickend würden wir die Ziele vermutlich etwas weniger hoch setzen, und statt mehrere Lösungen von Beginn weg zu unterstützen, uns auf eine zu konzentrieren. Dennoch bildet der aktuelle Stand eine gute Grundlage, um das Projekt weiterzuentwickeln.

2.2.4 Ausblick

Für die Weiterentwicklung von *HuddleUp* sehen wir verschiedene Möglichkeiten. Zum einen besteht eine Vielzahl von angedachten Funktionen, wie sie zum Beispiel der Aufgabenstellung zu entnehmen sind. Hervorzuheben ist dabei das Implementation eines Administratoren-Bereichs, da dessen Möglichkeiten derzeit begrenzt sind. Andererseits gibt es für die bestehenden Funktionen auch noch Erweiterungs- und Verbesserungsmöglichkeiten:

- Ausbau der Videokonferenz-Integration
- Implementation eines weitere Videokonferenz-Providers
- Verbesserung des Anmeldemechanismus um das Onboarding zu vereinfachen
- Implementation von Refresh Tokens um regelmässiges Anmelden zu vermeiden
- Ausbau der Customizing-Möglichkeiten
- Vereinheitlichung der Konfiguration
- Vereinfachte Installation durch Verwendung von Docker-Containern

Neben weiteren Funktionen denken wir jedoch, dass das Projekt eine genauere Vision benötigt um effektiv weiter entwickelt werden zu können. Eine Voraussetzung dafür wäre, wenn die Applikation zu einem Einsatz kommen würde, oder ein Interessent für eine spezifische Anwendung meldet. Erwartungen und Feedback von richtigen Benutzern und Betreibern sind wichtig, um die weitere Stossrichtung und somit die Vision für das Projekt festlegen zu können.

Teil II

ProjeKtdokumentation

Kapitel 3

Projektplan

3.1 Einführung

3.1.1 Zweck

Das vorliegende Dokument bietet eine Übersicht über Planung, Umfang, Ziel, Organisation und Prozesse der Bachelorarbeit (BA) zum Thema "Videokonferenzen für einander Unbekannte".

3.1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokumentes erstreckt sich über die gesamte Laufzeit der Bachelorarbeit. Sofern nicht anders angegeben beziehen sich alle folgenden Datumsangaben auf das Jahr 2021.

3.1.3 Referenzen

Dieses Dokument wurde mit dem Wissen aus den Modulen Software Engineering 1 und 2 sowie der gemachten Erfahrungen im Rahmen der Studienarbeit erstellt.

3.2 Projektübersicht

Im Rahmen dieser Bachelorarbeit soll gemäss der Aufgabenstellung (Abschnitt [A.1](#)) eine Web-Plattform für Video-Treffen erstellt werden. Die Plattform soll sich einander unbekanntes Personen eine Möglichkeit zum Kennenlernen und Austausch über gemeinsame Interessen bieten.

3.2.1 Lieferumfang

Die Abgabe am Ende der Arbeit umfasst die folgenden Dokumente:

- Abstract
- Aufgabenstellung
- Projektplan

- Projektdokumentation
- Installationsanleitung
- Bedienungsanleitung
- Einverständniserklärung
- Erklärung zur Urheberschaft
- Persönliche Berichte
- Protokolle
- Source Code
- A0-Plakat

3.3 Projektorganisation

3.3.1 Personen und Rollen

Team

Das Team setzt sich aus den folgenden Mitgliedern zusammen:

- Flavio Böni (flavio.boeni@ost.ch)
- Severin Hauser (severin.hauser@ost.ch)

Alle Mitglieder arbeiten primär in der Rolle als Entwickler. Im Team existieren keine hierarchische Strukturen. Wesentliche Entscheidungen werden im Plenum getroffen, wobei alle Mitglieder gleichgestellt sind.

Externe Schnittstellen

- Prof. Frank Koch (Dozent Studiengang Informatik): Betreuer, Stakeholder
- Prof. Hansjörg Huser: Experte
- Dr. Thomas Bocek: Gegenleser

3.3.2 Besprechungen

Aufgrund der aktuellen Situation bezüglich COVID-19 finden Besprechungen vor allem Online via Microsoft Teams statt. Das BA-Team trifft sich dabei mehrere Male pro Woche, etwa um das Vorgehen festzulegen, sich auszutauschen oder zur Zusammenarbeit.

Bei Erreichen von Meilensteinen und Fertigstellung wichtiger Projektartefakte werden diese mit dem Betreuer der Arbeit in einem Review-Meeting besprochen. Alle Review-Meetings sowie wesentliche Punkte der Teambesprechungen werden protokolliert.

3.4 Projektablauf

3.4.1 Zeitaufwand

Die Bachelorarbeit beginnt in der Kalenderwoche 8 (23. Februar 2021) und erstreckt sich total über eine Dauer von 17 Wochen. Die Abgabe der Arbeit wird in der Kalenderwoche 24 (14. – 18. Juni) erfolgen. Das Zeitbudget ergibt sich aus den für die Durchführung der Arbeit erhaltenen 12 ECTS-Punkten, wobei ein ECTS-Punkt 30 Stunden Arbeitsaufwand entspricht.

$$12 \text{ ECTS} \times 30 \text{ h} \times 2 \text{ Studenten} = 720 \text{ h}$$

Über 17 Wochen verteilt ergibt dies somit ein wöchentliches Pensum von 21 Stunden pro Student.

Start	23.02.2021
Ende	18.06.2021
Dauer	17 Wochen
Zeitbudget	720 Stunden

Tabelle 3.1: Zeitliche Rahmenbedingungen

3.4.2 Vorgehen

Für das Vorgehen im Projekt orientieren wir uns an SCRUM+[7]. Bei SCRUM+ handelt es sich um eine Kombination der Projektphasen des **Unified Process** (UP) und der iterativen Entwicklung von SCRUM. Die Methodik ist uns aus dem von Prof. Daniel Keller unterrichteten Modul Software Engineering 2 bekannt. Zusätzlich zu Phasen und Iterationen werden Meilensteine verwendet um wichtige Zeitpunkte im Projektverlauf zu definieren.

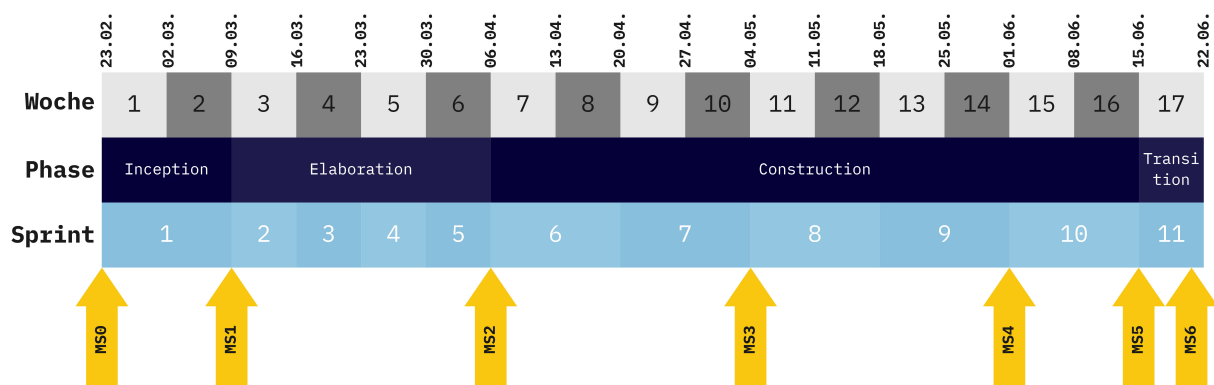


Abbildung 3.1: Übersicht über Phasen, Sprints und Meilensteine

Phasen

Die aus **UP** entnommenen Phasen *Inception*, *Elaboration*, *Construction* und *Transition* sind wie folgt geplant:

Phase	Kalenderwoche	Datum	Anzahl Wochen
Inception	8 - 9	23.02. - 10.03	2
Elaboration	10 - 13	11.03. - 05.04.	4
Construction	14 - 23	06.04. - 13.06	10
Transition	24	14.06. - 18.06.	1

Tabelle 3.2: Projektphasen

Iterationen

Innerhalb der Phasen wird in Iterationen, bzw. in **Sprints** gearbeitet. Unsere Erfahrung während dem Engineering Projekt und der Studienarbeit hat gezeigt, dass Sprints ausserhalb der Construction-Phase nur bedingt erfolgreich sind. Die Dauer eines Sprints und wie stark dieser nach SCRUM umgesetzt wird, ist deshalb davon abhängig in welcher Phase dieser sich befindet. Es wird deshalb zwischen folgenden Formen der Umsetzung unterschieden:

keine Keine Umsetzung nach SCRUM. Ad-Hoc Planung und Erfassung grober Arbeitspakete.

partiell Pflegen eines Backlogs, Erfassen klarer Arbeitspaketen mit Schätzungen zu Beginn des Sprints. Veränderungen am Sprint-Backlog sind möglich.

komplett Umsetzung gemäss SCRUM. Ein Sprint beginnt mit der Sprintplanung und endet mit einem kurzen Sprintreview und ggf. einer Retrospektive. Der Sprint Backlog wird im Verlauf eines Sprints nicht verändert.

Phase	Sprint-Dauer	Umsetzungsform
Inception	2 Wochen	keine
Elaboration	1 Woche	partiell
Construction	2 Wochen	komplett
Transition	1 Woche	keine

Tabelle 3.3: Umsetzung von Sprints

Meilensteine

Folgende Meilensteine wurde gesetzt:

MS0 Kickoff (23. Februar)

- Projektbeginn

MS1 Projektplan (10. März)

- Projektplan erstellt und mit dem Betreuer besprochen
- Risikoanalyse erstellt

MS2 End of Elaboration (5. April)

- Use Cases, funktionale und nichtfunktionale Anforderungen erfasst
- Domainanalyse erstellt
- High-Level Architektur erstellt
- Prototyp erstellt
- Repository, CI, Entwicklungsumgebung bereit
- UI-Mockups der wichtigsten Funktionen erstellt
- Video-Konferenz Provider evaluiert
- Definition des Minimum Viable Product (MVP) erfasst
- Zwischenpräsentation mit Betreuer, Experte und Gegenleser durchgeführt

MS3 Alpha: Lauffähiges MVP (4. Mai)

- Detaillierte Architektur wurde vollständig definiert

MS4 Beta: Feature Freeze (1. Juni)

- Usability Test
- Systemtest-Spezifikation ist erfasst
- Qualitätssicherung in Dokumentation erfasst

MS5 Release Candidate: Code Freeze, finale getestete Version (13. Juni)

- Systemtest durchgeführt und protokolliert

MS6 Übergabe (18. Juni)

- Schlussbericht abgegeben

3.5 Risikomanagement

Die identifizierten Risiken und der damit verbundene Umgang sind in der Risikoanalyse (Kapitel 7) zu finden.

3.5.1 Umgang mit Risiken

Alle während der Durchführung erwarteten Risiken werden in der Risikoanalyse erfasst. Bei Erreichen jedes Meilensteins werden diese neu bewertet, neu aufgetretene Risiken erfasst und Massnahmen wenn nötig angepasst.

Die Ergebnisse aus der Analyse fließen in die Planung weiterer Sprints und die Priorisierung von Features ein. In der letzten Iteration der Construction Phase wird ein Puffer eingeplant um Verzögerungen durch das Eintreten eines solchen Risikos zu kompensieren. Tritt ein schwerwiegendes Risiko ein dessen Auswirkungen nicht durch den Puffer kompensiert werden können, werden Features entfernt um die Fertigstellung sicher zu stellen.

3.6 Arbeitspakete

Arbeitspakete werden im Backlog erfasst und priorisiert. Je höher die Priorität eines Arbeitspakets, desto ausführlicher sollte dessen Beschreibung sein. Die Verwaltung der Arbeitspakete findet in **YouTrack** statt.

3.7 Infrastruktur

Für das Projekt werden verschiedene physische und virtuelle Tools eingesetzt.

3.7.1 Physisch

PC / Laptop Die Entwicklung findet auf den persönlichen Geräten der Teammitglieder statt.

3.7.2 Virtuell

Server Als Staging-Server wird ein von der OST zur Verfügung gestellter Linux-Server (Ubuntu 20.04) eingesetzt.

GitHub (Git) Versionskontrolle, Code Reviews, Continuous Integration (**CI**)

YouTrack Projektmanagement, Zeiterfassung

Notion Meeting-Protokolle, Text-Entwürfe

Overleaf Dokumentation

Miro Brainstorming

Figma UI-Design

Visual Studio Code Empfohlene IDE für alle Entwickler

Microsoft Teams Kommunikation, Meetings, Reviews

3.8 Qualitätsmassnahmen

3.8.1 Dokumentation

Die Dokumentation des Projekts wird in \LaTeX erstellt. Als Editor und zur Versionierung wird ein Projekt auf Overleaf¹ verwendet. Alle Änderungen an der Dokumentation werden durch ein weiteres Teammitglied gegengelesen. Dies stellt sicher, dass die Dokumentation der Meinung des gesamten Teams entspricht.

3.8.2 Entwicklung

Der Quellcode wird auf Git-Repositories verwaltet und befindet sich auf GitHub. Zur Organisation der Repositories wird die GitHub-Organisation «huddle-up»² verwendet.

¹<https://www.overleaf.com>

²<https://github.com/huddle-up>

Zur Ausführung von automatisierten Tests wird GitHub Actions³ eingesetzt.

Vorgehen

Das Vorgehen während der Entwicklung orientiert sich am GitHub-Flow[1]. Dabei befindet sich der aktuelle Stand auf dem *Main Branch*, während die Entwicklung von Features auf separaten *Feature-Banches* erfolgt. Das Vorgehen besteht aus den folgenden Schritten:

1. Erstellen eines Feature Branches, basierend auf dem Main Branch.
2. Committen von Änderungen auf den Feature Branch. Bei jedem gepushten Commit werden automatisch die Tests durch GitHub Actions ausgeführt.
3. Erstellen eines *Draft Pull Requests*, sobald erstes Feedback gewünscht ist.
4. Weitere Implementierungen.
5. Wechsel des Draft Pull Requests zu einem regulären Pull Request um den Request als 'ready to review' zu markieren.
6. Review des Pull Requests durch mindestens ein weiteres Teammitglied.
7. Bei positivem Review und erfolgreichem Durchlaufen der GitHub Actions wird der Pull Request akzeptiert und somit die Änderungen des Feature Branches in den Main Branch gemerged.

Direkte Änderungen am Main-Branch werden blockiert um das Einhalten dieses Ablaufs sicherzustellen.

Code Reviews

Code Reviews werden durchgeführt um die Codequalität sicherzustellen und den Wissenstand des gesamten Teams zu verbessern. Reviews werden bei jedem Pull Request durchgeführt und stellen so sicher, dass nur akzeptable Änderungen in den Main Branch gemerged werden.

Code Reviews für Pull Requests werden jeweils durch mindestens ein weiteres Teammitglied durchgeführt.

Auf folgende Punkte werden während einem Code Review geachtet:

- Ist die Funktionalität gemäss Arbeitspaket implementiert?
- Werden alle Style Guidelines eingehalten?
- Wurden entsprechende Tests erfasst bzw. ergänzt?
- Läuft die Test-Pipeline erfolgreich durch?

Code Style Guidelines

Formatierung Als Formatierungs-Guideline wird Prettier⁴ verwendet. Die Abweichungen des Default-Presets werden im Source-Code in der Prettier Konfigurations-Datei fest-

³<https://github.com/features/actions>

⁴<https://prettier.io/>

gehalten. Durch die Verwendung des Prettier-Plugins für VS Code⁵ kann die automatische Formatierung beim Speichern sichergestellt werden.

Syntax und Style Ergänzend zu Prettier wird ESLint⁶ mit dem Airbnb-Preset⁷ als Style- und Syntax-Guideline verwendet. Durch Verwendung des ESLint-Plugins in VS Code⁸ kann das Feedback des Linters direkt in der IDE angezeigt werden.

Git-Commits Commit-Messages werden entsprechend der Vorgabe von Conventional Commits geschrieben⁹.

Überprüfung Die Einhaltung der Guidelines wird in der Test-Pipeline mit ESLint und Husky¹⁰ geprüft und durchläuft nur, wenn keine Fehler gefunden werden.

Software Tests

Unit und Integration Tests werden parallel zu den zu testenden Features geschrieben und befinden sich im selben Repository wie der Quellcode.

Folgende Frameworks und Bibliotheken kommen bei den automatisierten Tests zum Einsatz:

Jest (Test Framework) Jest¹¹ ist eines der populärsten Testing-Frameworks für JavaScript. Es eignet sich für das Projekt, da sowohl Nest (Backend) wie auch React (Frontend) Jest als Test Framework empfehlen.

React Testing Library (Bibliothek) Die React Testing Library¹² bietet Hilfsfunktionen zum Testen von React Komponenten an und wird von React empfohlen.

Unit und Integration Tests Um die Funktionalität des Programmes zu gewährleisten und Regressionen vorzubeugen werden Unit und Integration Tests durchgeführt. Diese sind Bestandteil von jedem Feature und werden somit in der Zeitschätzung bereits eingerechnet. Die Tests laufen lokal und werden in jedem Fall auch automatisch in der CI-Pipeline ausgeführt.

Backend Unit und Integration Tests werden für alle wichtige Funktionen geschrieben. Ziel ist eine Testabdeckung von 75%.

Frontend Auf Integration Tests wird im Frontend verzichtet. Unit Tests werden nur für wichtige Kernfunktionen geschrieben.

⁵<https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode>

⁶<https://eslint.org/>

⁷<https://www.npmjs.com/package/eslint-config-airbnb>

⁸<https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint>

⁹<https://www.conventionalcommits.org/en/v1.0.0/>

¹⁰<https://www.npmjs.com/package/husky>

¹¹<https://jestjs.io/>

¹²<https://testing-library.com/docs/react-testing-library/intro/>

Usability Tests Um die Benutzbarkeit der Applikation sicherzustellen werden im Verlauf der Arbeit Usability Tests durchgeführt. Die Durchführung der Tests findet vor Erreichen des Meilensteins 4 (Beta: Feature Freeze) statt. Dies erlaubt die Erkenntnisse aus den Tests vor Ende der Construction-Phase umzusetzen.

3.9 MVP

Der **MVP** wird die Grundfunktionalität aus den Use Cases mit Priorität 1 abdecken. Die umgesetzten Funktionen lauten folglich:

- Registration von Benutzern
- Erstellen von Treffen
- Suchen von Treffen
- Durchführen von Treffen durch den Organisator
- Teilnahme an Treffen durch Benutzer

Kapitel 4

Anforderungsspezifikation

4.1 Use Cases

Die funktionellen Anforderungen werden durch Use Cases nach Larman beschrieben. Es gibt die drei Detaillierungsgrade *brief* (zwei bis drei Sätze), *casual* (ca. eine halbe Seite) und *fully dressed* (ausführlicher und div. weitere Attribute), welche je nach Priorität und Komplexität angewendet wurden. Für die Identifikation wird jedem umgesetzten Use Case zusätzlich zum Titel eine eindeutige Nummer zugewiesen.[9]

Priorisierung Die erfassten Use Cases sind nicht alle im Verlauf der Arbeit umsetzbar. Sie werden deshalb in drei Prioritätsstufen unterteilt. Use Cases mit der höchsten Priorität werden als erstes bearbeitet.

4.1.1 Akteure

Akteur	Beschreibung
Benutzer	Registrierter Anwender der Applikation
Organisator	Hat ein Treffen erstellt und ist durch die Erstellung der Organisator und moderiert die Videositzung. Er ist ein Benutzer der Applikation.
Teilnehmer	Nimmt an einem Treffen teil und ist ein Benutzer der Applikation
Administrator	Ist der Betreiber und muss kein Benutzer der Applikation sein

4.1.2 Use Case Diagram

Das Use Case Diagramm gibt einen Überblick über die relevanten Anwendungsfälle. Um die Übersichtlichkeit zu verbessern sind die Prio 3 Use Cases im Diagramm nicht enthalten.

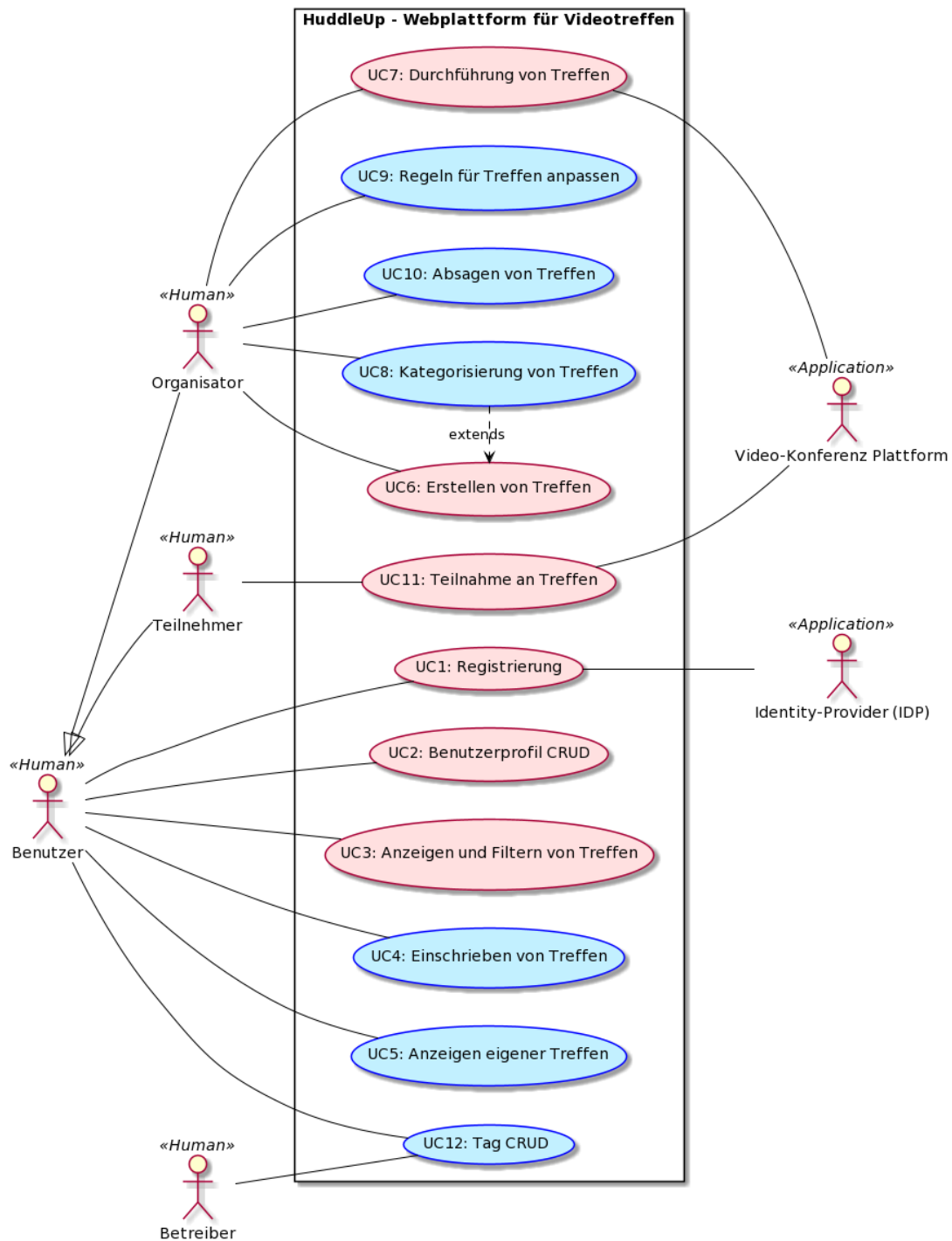


Abbildung 4.1: Use-Case-Diagramm

Farbe	Bedeutung
Rot	Use Cases mit Priorität 1 werden zuerst bearbeitet
Blau	Use Cases mit Priorität 2

4.1.3 Use Case Beschreibungen

UC 1: Registrierung

Beschreibung	
Um die Webplattform zu nutzen muss eine Registrierung durchgeführt werden.	
Primärer Akteur	Benutzer
Interessenten	<ul style="list-style-type: none"> • Benutzer: Möchte ein eigenes Benutzerkonto besitzen und beim nächsten Besuch wieder erkannt werden. • Betreiber: Möchte das Benutzer sich nur über den von ihm vorkonfigurierten Identity Provider (IDP) registrieren können.
Vorbedingungen	<ul style="list-style-type: none"> • Der vom Betreiber konfigurierte IDP ist funktionsfähig eingerichtet und erreichbar.
Erfolgsgarantie	<ul style="list-style-type: none"> • Der Benutzer wird bei der Registrierung an den vorkonfigurierten IDP weitergeleitet. • Der Benutzer wird anhand des IDP authentifiziert und bei der Webplattform registriert.
Hauptszenario	<ol style="list-style-type: none"> 1. Der Benutzer ruft die Webplattform über seinen Browser auf. 2. Es wird eine Registrierung via vorkonfigurierten IDP angeboten. 3. Der Benutzer authentifiziert sich über den IDP. 4. Die Authentifizierungsdaten (z.B. Token) und Benutzerdaten (z.B. E-Mail, Name) werden vom IDP übernommen um die Registrierung abzuschliessen. 5. Der Benutzer wird an der Webplattform angemeldet.
Alternativen	<ol style="list-style-type: none"> 2a. Der Benutzer ist bereits registriert und nach der Authentifizierung durch den IDP direkt in der Applikation angemeldet.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.3: UC 1: Registrierung

UC 2: Benutzerprofil CRUD

Beschreibung	
Der Benutzer kann direkt nach der Registrierung oder zu einem späteren Zeitpunkt sein Benutzerprofil bearbeiten. Das Benutzerprofil umfasst optionale für andere Benutzer ersichtliche Daten. Im minimum ist eine Biographie enthalten, des weiteren wäre ein Benutzername und eine Auswahl eines Profilbildes möglich. Das Benutzerprofil kann jederzeit vom Benutzer editiert und werden.	
Primärer Akteur	Benutzer
Hauptszenario	<ol style="list-style-type: none"> 1. Ein Benutzer öffnet nach der Registrierung sein Benutzerprofil über sein Benutzerkonto. 2. Der Benutzer erfasst eine Biographie und evtl. weitere spezifische Daten bzw. editiert sie. 3. Nachdem der Benutzer die Eingabe speichert sind die erfassten Daten des Benutzerprofils für andere Benutzer ersicht-lich.
Alternativen	<ol style="list-style-type: none"> 2a. Der Benutzer entfernt erfasste Daten aus dem Benutzerprofil. 2b. Die Daten aus dem Benutzerprofil sind für andere Benutzer nicht mehr ersichtlich.
Erweiterungen	<ol style="list-style-type: none"> 1a. Dem Benutzer wird direkt nach der Registrierung im Rahmen des Onboarding-Prozesses die Möglichkeit gegeben sein Benutzerprofil zu erstellen.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.4: UC 2: Benutzerprofil CRUD

UC 3: Anzeigen und Filtern von Treffen

Beschreibung	
Benutzer sehen eine Auflistung von laufenden und geplanten Treffen. Damit Benutzer ihren Interessen entsprechende Treffen finden können, kann die Auflistung nach verschiedenen Kriterien gefiltert und sortiert werden.	
Primärer Akteur	Benutzer

Interessenten	<ul style="list-style-type: none"> • Benutzer: Möchte geplante und laufende Treffen die ihn interessieren finden können. • Organisator: Möchte seine Treffen im Überblick haben und schnell wieder finden. • Teilnehmer: Möchte die Treffen bei denen er sich registriert hat schnell wieder finden.
Erfolgsgarantie	<ul style="list-style-type: none"> • Der Benutzer sieht geplante, laufende sowie seine erstellten bzw. registrierten Treffen übersichtlich dargestellt. • Es werden nur Treffer angezeigt, welche den Filterkriterien des Benutzers entsprechen. • Die Treffen werden zur Übersichtlichkeit in vorsortierten und seitennummerierten (paged) Ansichten unterteilt.
Hauptszenario	<ol style="list-style-type: none"> 1. Ein Benutzer erhält auf der Startseite direkt eine Auflistung von laufenden und geplanten Treffen. 2. Die Treffen werden nach verschiedenen Kriterien vorsortiert bzw. unterteilt dargestellt. 3. Der Benutzer kann die Treffen anhand einer Filterung von verschiedenen Kriterien eingrenzen (z.B. Name, Tags). 4. Bei Auswahl eines Treffens wird dem Benutzer in einer Detailansicht die Beschreibung des Treffens sowie dessen Organisator angezeigt.
Alternativen	2a. Der Benutzer kann die Sortierung bzw. Unterteilung beeinflussen.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.5: UC 3: Anzeigen und Filtern von Treffen

UC 4: Einschrieben von Treffen

Beschreibung	
Der Benutzer kann sich in ein geplantes Treffen einschreiben und die Teilnahme auch wieder zurücknehmen.	
Primärer Akteur	Teilnehmer

Vorbedingungen	<ul style="list-style-type: none"> • Das Treffen wurde von einem anderen Benutzer erstellt. • Das Treffen wurde noch nicht durchgeführt oder abgesagt. • Das Treffen hat nicht die maximale Anzahl Teilnehmer erreicht.
Erfolgsgarantie	<ul style="list-style-type: none"> • Alle Teilnehmer sind im Treffen ersichtlich.
Hauptszenario	<ol style="list-style-type: none"> 1. Der Benutzer öffnet ein geplantes oder laufendes Treffen. 2. Der Benutzer wählt den <i>teilnehmen</i> Button aus. 3. Das Treffen wird für den Teilnehmer als <i>Teilnehmer</i> markiert
Alternativen	<ol style="list-style-type: none"> 2a. Der Teilnehmer wählt den <i>nicht teilnehmen</i> Button aus. 3a. Der Benutzer wird nicht mehr als Teilnehmer markiert.
Erweiterungen	<ol style="list-style-type: none"> 2a. Ein interessierter Benutzer kann das öffentliche Profil der Teilnehmer anschauen.
Auftrittshäufigkeit	Die Kombination von <i>teilnehmen</i> und <i>nicht teilnehmen</i> kann beliebig oft durchgeführt werden.

Tabelle 4.6: UC 4: Einschrieben von Treffen

UC 5: Anzeigen eigener Treffen

Beschreibung	
Benutzer will sich anstehende und vergangene Treffen anzeigen bei denen er Teilnehmer bzw. Organisator ist. Es wird eine separate Ansicht mit den eigenen Treffen dargestellt.	
Primärer Akteur	Benutzer
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer hat bereits eigene Treffen erstellt oder sich als Teilnehmer registriert.
Erfolgsgarantie	<ul style="list-style-type: none"> • Der Benutzer sieht seine laufenden, anstehenden und vergangenen Treffen. • Der Benutzer sieht pro Treffen ob er Teilnehmer oder Organisator ist.

Hauptszenario	<ol style="list-style-type: none"> 1. Der Benutzer öffnet die Ansicht für seine Treffen. 2. Der Benutzer sieht gruppierte Ansichten über seine laufenden, anstehenden und vergangenen Treffen. 3. Der Benutzer kann seine Treffen anhand der Filterkriterien weiter eingrenzen.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.7: UC 5: Anzeigen eigener Treffen

UC 6: Erstellen von Treffen

Beschreibung	
<p>Jeder Benutzer kann Treffen erstellen und diese bis zur effektiven Durchführung bearbeiten. Bei der Erstellung werden folgenden Attribute erfasst:</p> <ul style="list-style-type: none"> • Titel und Beschreibung des Treffens • Start/End Datum und Uhrzeit des Treffens • Maximale Anzahl Teilnehmer 	
Primärer Akteur	Benutzer
Erfolgsgarantie	<ul style="list-style-type: none"> • Das Treffen ist mit allen angegebenen Informationen für alle anderen Benutzer ersichtlich. • Der Benutzer kann das Treffen bis zur Durchführung bearbeiten.
Hauptszenario	<ol style="list-style-type: none"> 1. Ein registrierter Benutzer möchte über die Applikation ein neues Treffen erstellen. 2. Der Benutzer öffnet die Ansicht für die Erstellung eines neuen Treffens. 3. Das Treffen wird mittels den in der Beschreibung deklarierten Attributen erfasst. 4. Der Benutzer speichert das Treffen ab. 5. Der Benutzer kann das Treffen weiter bearbeiten.
Erweiterungen	5a. Eine geplantes Treffen kann nur eingeschränkt editiert werden (z.B. nur Beschreibung)
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.8: UC 6: Erstellen von Treffen

UC 7: Durchführung von Treffen**Beschreibung**

Der Organisator startet das Treffen. Die Videokonferenz wird erstellt und für die Teilnehmer freigegeben.

Primärer Akteur Organisator

Interessenten

- Organisator: Möchte das geplantes Treffen starten und der Videokonferenz beitreten.
 - Benutzer: Möchte das Treffen finden und sehen ob das Treffen läuft.
 - Teilnehmer: Möchte der der Videokonferenz beitreten.
 - Betreiber: Möchte das die Videokonferenz über den vorkonfigurierten Provider durchgeführt wird.
-

Vorbedingungen

- Der Organisator ist an der Applikation angemeldet.
 - Das Treffen wurde vom Organisator anhand des **UC 6: Erstellen von Treffen** erstellt.
 - Das Treffen wurde noch nicht durchgeführt oder abgesagt.
 - Der vorkonfigurierte Videokonferenz Provider ist für die Durchführung bereit.
-

Erfolgsgarantie

- Der Organisator ist der Videokonferenz über die Applikation beigetreten.
 - Die Teilnehmer sehen das laufende Treffen über die Applikation.
 - Die Teilnehmer erhalten über das Treffen in der Applikation die Zugangsdaten und können sich darüber mit der Videokonferenz verbinden und an der Videokonferenz teilnehmen.
 - Der Organisator kann die Durchführung beenden, worauf kein Zugang für die Teilnehmer mehr möglich ist.
-

Hauptszenario	<ol style="list-style-type: none"> 1. Der Organisator möchte das Treffen kurz vor dem geplanten Zeitpunkt starten. 2. Der Organisator öffnet das Treffen aus der Liste eigener Treffen aus. 3. Der Organisator startet die Videokonferenz aus dem Treffen in der Applikation. 4. Die Applikation erstellt die Zugangsdaten (z.B. Links zur Teilnahme) zur Videokonferenz und zeigt diese dem Organisator an. 5. Der Organisator verbindet sich über die Zugangsdaten mit der Videokonferenz. 6. Der Organisator gibt das Treffen frei für die Teilnehmer.
Alternativen	<ol style="list-style-type: none"> 3a. Die Videokonferenz wird über die Applikation zu einem spezifischen Zeitpunkt generiert. 6a. Die Videokonferenz wird freigegeben sobald der Organisator sie erstellt hat.
Erweiterungen	<ol style="list-style-type: none"> 5a. Der Organisator richtet die Videokonferenz nach seinen Bedürfnissen ein.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.9: UC 7: Durchführung von Treffen

UC 8: Kategorisierung von Treffen

Beschreibung	
Die Kategorisierung von Treffen wird durch die Auswahl von Tags gemacht. Die bereits verfügbaren und neuen Tags können vom Benutzer einem Treffen zugewiesen werden.	
Primärer Akteur	Organisator
Erfolgsgarantie	<ul style="list-style-type: none"> • Die Tags werden in den Listansichten und in der Detailansicht des gespeicherten Treffens dargestellt.
Hauptszenario	<ol style="list-style-type: none"> 1. Der Organisator wählt bei der Erstellung oder Bearbeitung eines Treffens einen oder mehrere existierenden Tags aus. 2. Der Organisator speichert das Treffen.

Alternativen	<ol style="list-style-type: none"> 1a. Der Organisator gibt einen oder mehrere neue noch nicht existierenden Tags ein. 1a. Der Organisator löscht einen oder mehrere bestehende Tags. 2a. Der Organisator bricht ab und speichert das Treffen nicht.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.10: UC 8: Kategorisierung von Treffen

UC 9: Regeln für Treffen anpassen

Beschreibung	
Der Organisator kann Regeln für das Treffen anpassen. Er kann im Minimum die maximale Anzahl Teilnehmer pro Treffen definieren.	
Primärer Akteur	Organisator
Erfolgsgarantie	<ul style="list-style-type: none"> • Die Anzahl maximaler Teilnehmer ist falls vorhanden pro Treffen ersichtlich. • Die Teilnahme ist nur möglich, wenn die Anzahl maximaler Teilnehmer noch nicht erreicht wurde.
Hauptszenario	<ol style="list-style-type: none"> 1. Der Organisator gibt bei der Erstellung oder Bearbeitung eines Treffens eine maximale Anzahl Teilnehmer an. 2. Der Organisator speichert das Treffen. 3. Das Treffen wird mit der maximalen Anzahl Teilnehmer gekennzeichnet.
Erweiterungen	1a. Die Anzahl maximaler Teilnehmer ist vordefiniert und kann vom Betreiber gesetzt werden.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.11: UC 9: Regeln für Treffen anpassen

UC 10: Absagen von Treffen

Beschreibung	
Der Organisator kann sein geplantes Treffen vor der Durchführung absagen.	
Primärer Akteur	Organisator

Erfolgsgarantie	<ul style="list-style-type: none"> Das Treffen kann bis zur Durchführung vom Organisator abgesagt werden. Laufende oder bereits durchgeführte Treffen können nicht durch den Organisator nicht abgesagt werden.
Hauptzenario	<ol style="list-style-type: none"> Der Organisator möchte ein ein geplantes Treffen absagen. Der Organisator wählt das gewünschte Treffen aus der Liste aus. Der Organisator sagt das Treffen durch eine Einstellung im Treffen ab. Nach einer Bestätigung wird das Treffen bei den Teilnehmern als Abgesagt gekennzeichnet und es ist keine Teilnahme mehr möglich.
Alternativen	<ol style="list-style-type: none"> Abgesagte Treffen werden nicht mehr dargestellt und gelöscht.

Tabelle 4.12: UC 10: Absagen von Treffen

UC 11: Teilnahme an Treffen

Beschreibung	
Ein bei einem Treffen eingeschriebener Benutzer tritt diesem bei.	
Primärer Akteur	Benutzer
Interessenten	<ul style="list-style-type: none"> Benutzer/Teilnehmer: Sucht ein geplantes bzw. laufendes Treffen und möchte diesem beitreten. Organisator: Wartet auf Teilnehmer und limitierte evtl. den Zutritt für Teilnehmer. Betreiber: Möchte das der Videokonferenz über den vorkonfigurierten Provider beigetreten wird.
Vorbedingungen	<ul style="list-style-type: none"> Der Benutzer ist an der Applikation angemeldet. Das Treffen wurde vom Organisator anhand des UC 7: Durchführung von Treffen gestartet. Das Treffen wurde noch nicht durchgeführt oder abgesagt. Der vorkonfigurierte Videokonferenz Provider ist für die Durchführung bereit.

Erfolgsgarantie	<ul style="list-style-type: none"> • Der Teilnehmer ist der gewünschten Videokonferenz beigetreten. • Der Teilnehmer können zu jedem Zeitpunkt während die Videokonferenz läuft beitreten bzw. sie verlassen.
Hauptszenario	<ol style="list-style-type: none"> 1. Der Benutzer öffnet das Treffen welchem er beitreten will aus der Treffenliste. 2. Ist das Treffen durch den Organisator freigegeben, werden dem Benutzer die Zugangsdaten zur Videokonferenz angezeigt. 3. Der Teilnehmer verwendet die erstellten Zugangsdaten um sich mit der Videokonferenz zu verbinden. 4. Abhängig von der vorkonfigurierten Videokonferenz Plattform wird die Teilnahme direkt über einen generierten Link ermöglicht. 5. Die Videokonferenz wird in einem eigenen Bereich / Fenster geöffnet.
Alternativen	<ol style="list-style-type: none"> 4a. Spezifische Zugangsdaten werden vom Organisator bereitgestellt. 5a. Die Teilnahme ist nur möglich wenn eine gesetzte Teilnehmergrenze noch nicht erreicht ist.
Erweiterungen	<ol style="list-style-type: none"> 5a. Teilnehmer landen zuerst in einem Warteraum und werden durch den Organisator freigegeben.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.13: UC 11: Teilnahme an Treffen

UC 12: Tag CRUD**Beschreibung**

Der Betreiber kann vordefinierte Tags zu Verfügung stellen, welche die Benutzer verwenden können. Ein Benutzer kann neue Tags erfassen, falls diese noch nicht erfasst wurden.

Primärer Akteur	Betreiber
-----------------	-----------

Erfolgsgarantie	<ul style="list-style-type: none"> • Die vordefinierten Tags können von den Benutzern verwendet werden. • Der Organisator kann zu einem Treffen neue Tags erfassen.
Hauptszenario	<ol style="list-style-type: none"> 1. Der Betreiber definiert Tags in der Konfiguration. 2. Die vordefinierten Tags werden bei der Kategorisierung durch den Organisator verwendet. 3. Der Benutzer gibt zusätzlich eigene Tags an. 4. Die Tags werden falls sie noch nicht existiert beim Speichern des Treffens erstellt.
Erweiterungen	<ul style="list-style-type: none"> • Der Betreiber kann bestehende Tags bearbeiten oder löschen.
Auftrittshäufigkeit	Beliebig oft

Tabelle 4.14: UC 12: Tag CRUD

4.1.4 Weitere optionale Use Cases

Die hier aufgeführten Use Case haben Prio 3 und sind im *brief* Format gehalten.

Bewertung erstellen

Teilnehmer eines Treffen können dem Treffen eine Bewertung abgeben.

Bewertungskatalog anpassen

Der Admin kann den Bewertungskatalog anpassen.

Anschauen der Bewertungen eines Organisators

Ein Benutzer möchte die Bewertung eines Organisators ansehen.

Freunde / Folgen anderer Benutzer

Um über deren Aktivität informiert zu bleiben, können Benutzer andere Benutzer als Freund hinzufügen, oder diesen folgen.

Kommentarfunktion

Benutzer können zu Treffen Kommentare schreiben um Fragen zu stellen, oder Feedback zu geben.

Gruppe erstellen

Benutzer können Gruppen erstellen, unter welchen Treffen organisiert werden (z.B. ein Studentenverein hat einen Gruppen-Account). Treffen können nur Gruppenmitgliedern zugänglich gemacht werden.

Kalenderexport

Benutzer können geplante Treffen in externe Applikationen (Bsp. Google Kalender) exportieren um diese nicht zu vergessen.

Benachrichtigungen

Implementation von E-Mail-Benachrichtigungen über demnächst startende Treffen, neue Treffen mit gewissen Tags, oder anderer Benutzern welchen der Benutzer folgt.

Private Treffen durchführen

Ein Benutzer kann direkt ein anderen Benutzer auswählen und mit ihm ein persönliches Treffen durchführen.

Anschauen der Bewertungen eines Treffens

Ein Organisator möchte seine Bewertung seiner Treffen ansehen.

Teilnehmer verwalten

Der Organisator kann Teilnehmer eines Treffens bestätigen, oder aussperren.

Einladungen zu Treffen

Der Organisator kann Einladungen zu Treffen verschicken.

Kategorisierung CRUD

Die Erstellung, Bearbeitung und Löschung von Kategorien.

Administration der Applikation

Administratoren steht ein Interface zur Verfügung zur Verwaltung, Moderation und Konfiguration der Applikation.

Erstellen persistenter Treffen

Treffen die immer aktiv und ohne Moderation sind, z.B. im Sinne einer Kaffee-Ecke.

Unterstützen mehrerer Videokonferenz-Provider

Bei der Erstellung eines Treffens kann aus mehreren Videokonferenz-Plattformen ausgewählt werden, auf welcher das Treffen stattfinden soll.

4.2 Nicht-funktionale Anforderungen

Die erfassten nicht-funktionalen Anforderungen wurden anhand des **FURPS** Model[8] kategorisiert.

Kategorie	Anforderung
Functionality	Die Videokonferenzen werden nicht durch die Applikation selbst, sondern durch eine externe Plattform, wie zum Beispiel Jitsi, oder Zoom durchgeführt.
Usability	Nicht alle Benutzer haben immer einen Desktop-Computer zur Verfügung. Die Webapplikation soll deshalb sowohl auf Desktop- wie auch Mobilgeräten nutzbar sein. Um die Applikation auch für nicht Englisch sprechende Benutzer nutzbar zu machen, soll die Applikation übersetzbar sein.
Performance	Um bei einem hohen Benutzervolumen die Verfügbarkeit der Applikation sicherzustellen, sollen mehrere Instanzen der Server-Komponente laufen können.
Supportability	Um die Applikation möglichst vielen Nutzern zugänglich zu machen, soll diese als Open Source verfügbar sein. Wichtige Funktionen der Applikation sollen durch den Betreiber konfigurierbar sein. Die Module für die Authentifizierung und den Videokonferenz-Provider sollen austauschbar sein. Die Applikation soll nur frei verfügbare Bibliotheken und Services verwenden. Die Applikation soll ohne grosse Aufwände erweitert werden können. Dies impliziert die Anwendung von Software Engineering Praktiken, die Verwendung einer durchdachten Architektur und Dokumentation.

Tabelle 4.15: Nicht-funktionale Anforderungen

4.3 Wichtige Abläufe

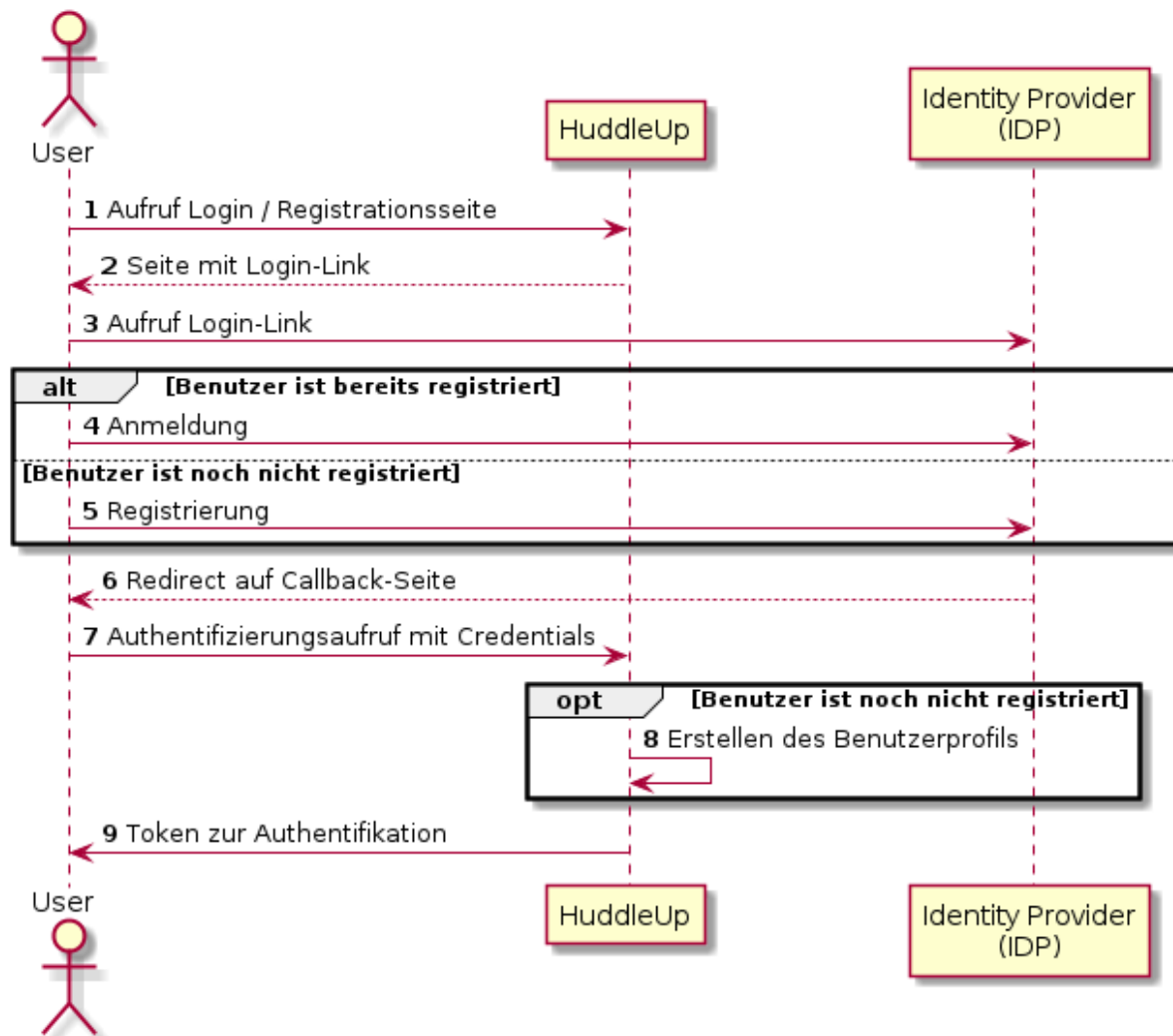


Abbildung 4.2: Registration

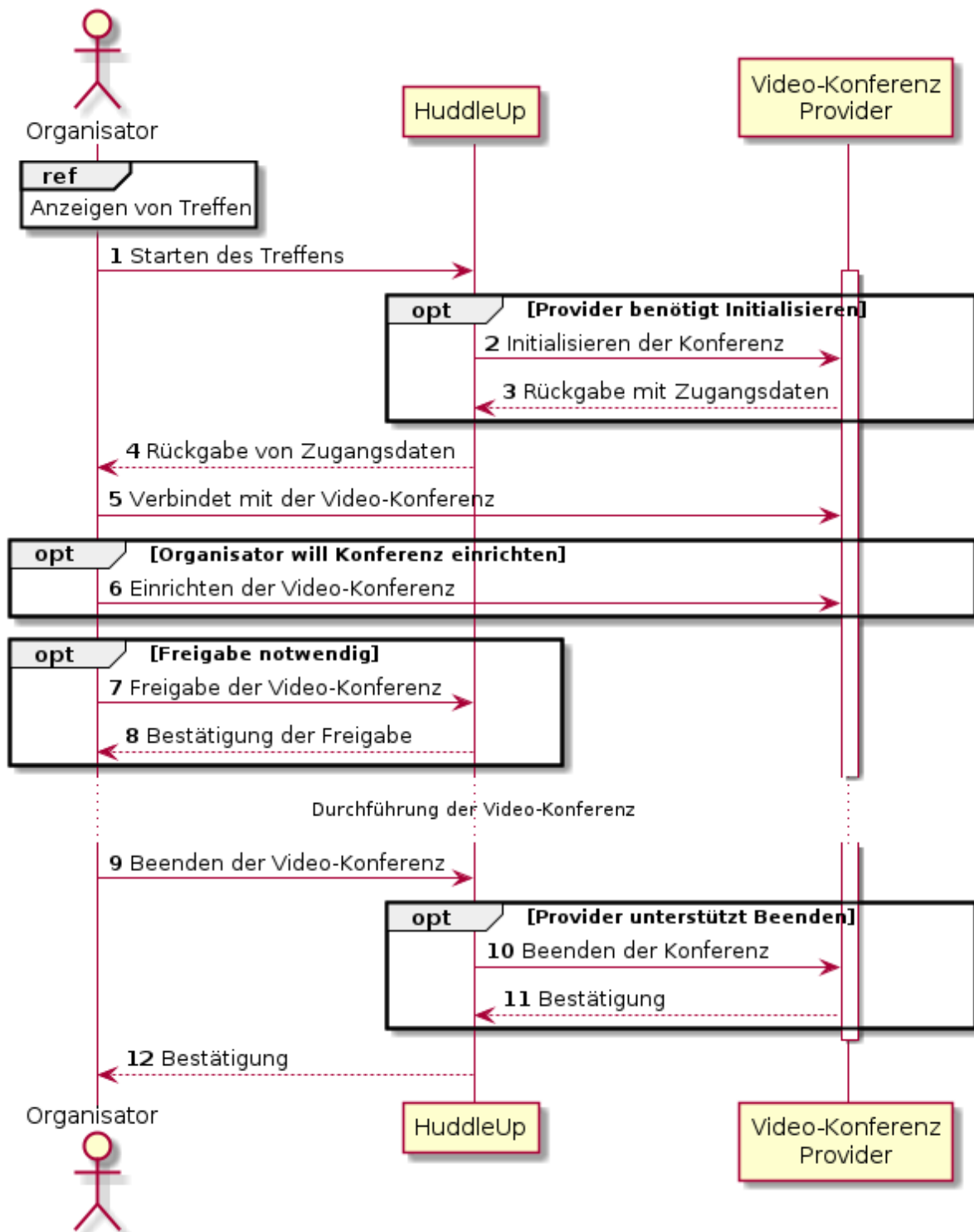


Abbildung 4.3: Durchführung

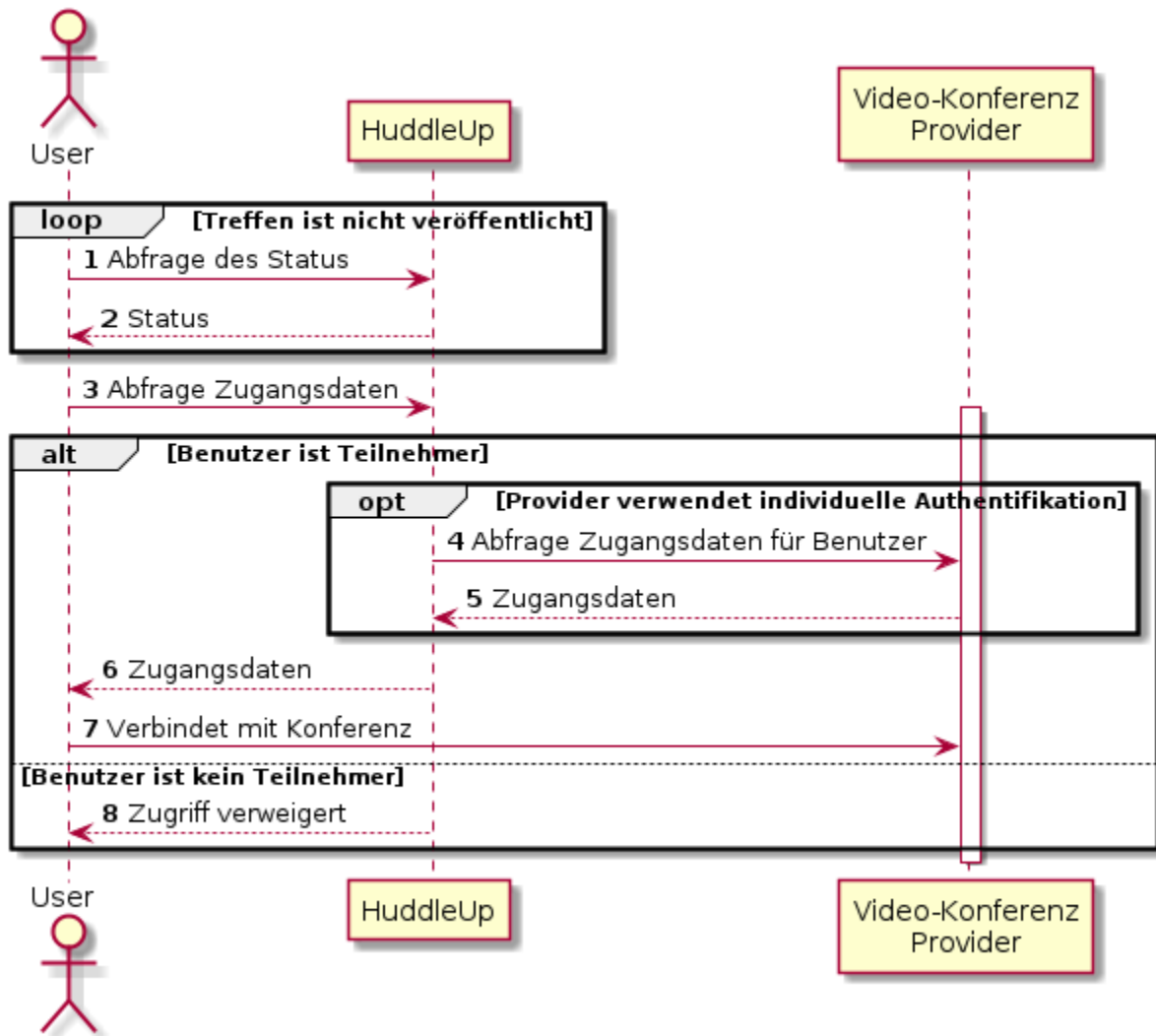


Abbildung 4.4: Teilnahme

Kapitel 5

Domänenanalyse

Dieses Kapitel analysiert die Hauptdomäne des Systems und erklärt die darin enthaltenen Konzepte.

5.1 Domänenmodell

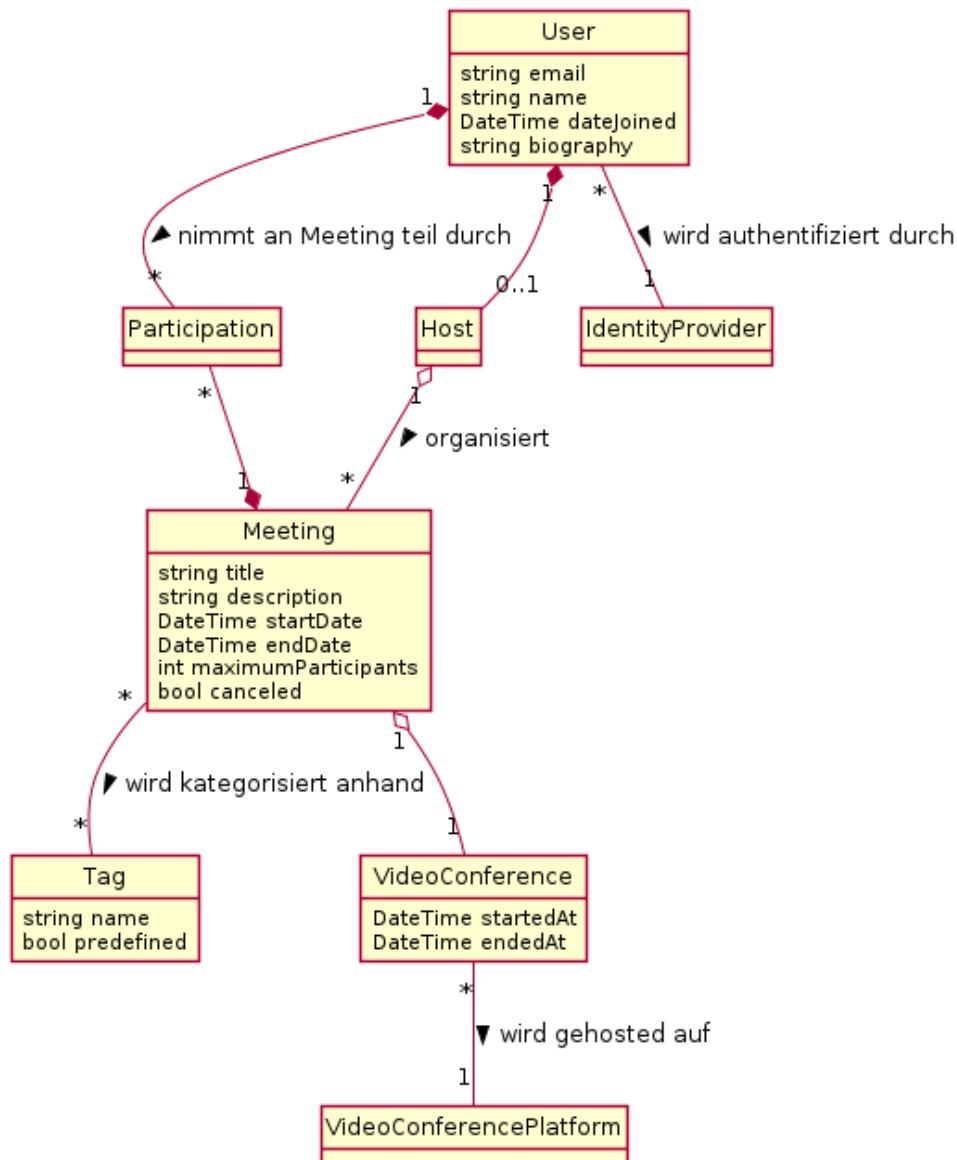


Abbildung 5.1: Domänenmodell

5.2 Konzepte

User Benutzer sind die Endanwender des Systems. Sie organisieren Meetings, oder nehmen an ihnen teil. Benutzer nehmen somit abhängig vom Anwendungskontext sowohl die Rolle des Teilnehmers, wie auch des Organisers an. Die Authentisierung eines Benutzers, bzw. die Bestätigung dessen Identität wird durch einen Identity Provider durchgeführt.

Identity Provider Ein Identity Provider ist eine Instanz, die für die Authentisierung von Benutzern zuständig ist. Dabei kann es sich zum Beispiel um einen [OpenID Connect](#)-Anbieter wie Google handeln.

Meeting Ein Meeting ist ein Treffen zu einem spezifischen Thema. Es wird zu einem

spezifischen Thema und zu einem spezifischen Zeitpunkt durchgeführt.

Host Treffen werden von einem Organisator erstellt und durchgeführt. Dabei handelt es sich zu Beginn nur um einen Benutzer, in Zukunft kann es sich dabei zum Beispiel auch um Gruppen handeln.

Participation Eine Benutzer kann einem Treffen über eine Teilnahme beitreten. Dies kann vor Meeting-Beginn erstellt werden und vom Organisator an- oder abgelehnt werden.

Tag Treffen können durch Tags versehen werden. Tags sind Stichwörter die Benutzern das Auffinden von Treffen erleichtern. Tags können sowohl durch den Betreiber vordefiniert wie auch durch Benutzer erstellt werden.

Video Conference Eine Videokonferenz ist die effektive Durchführung eines Treffens. Dabei handelt es sich um eine Session in welcher die Teilnehmer des Treffens über Sprach- und Videokanäle miteinander kommunizieren.

Video Conference Provider Videokonferenzen werden durch einen Provider wie zum Beispiel Zoom, oder Jitsi zur Verfügung gestellt.

Kapitel 6

Architektur und Design Spezifikation

6.1 Vorlagen

6.1.1 arc42 Template

Wir haben uns bei der Erstellung der Architektur-Dokumentation an arc42¹ orientiert. Es wurden jedoch nicht alle Sektionen übernommen.

6.1.2 C4 Model

Für die Visualisierung der Architektur wurde das C4 Model² angewendet. Wir verwenden die drei Modelle *Context*, *Containers* und *Components* um geeignete Diagramme pro Abstraktionsstufe darzustellen.

6.1.3 Y-Statements

Zur Unterstützung der Architekturentscheidungen verwenden wir Y-Statements[14][15]. Dabei verwenden wir folgende Vorlage:

Architekturentscheidung: X (Vorlage)

Im Kontext von «Use Case / Story / Component», konfrontiert mit «nichtfunktionaler Anforderung», entschieden wir uns für «Option o» und nicht für «Option o2» um «Qualität q» zu erreichen und akzeptieren die «Konsequenz k».

6.2 Technologien

Komponente	Technologie(n)
------------	----------------

¹<https://arc42.org/overview>

²<https://c4model.com>

Frontend	<ul style="list-style-type: none"> • React³ • Material UI⁴ (Material Design-Bibliothek für React-Komponenten) • React Testing Library⁵ (Bibliothek für das Testen von React-Komponenten) • Apollo Client⁶ (GraphQL-Client) • i18next⁷ (Mehrsprachigkeit) • Formik⁸ (Bibliothek für das Handling von Formularen in React)
Backend / API	<ul style="list-style-type: none"> • Nest.js⁹ (Node.js-Framework) • Apollo Server¹⁰ (GraphQL-Server) • TypeORM¹¹ (Object-Relational Mapper) • PostgreSQL¹² (Object-Relational Database System) • Jest¹³ (Test Framework) • Istanbul mit nyc¹⁴ (Code Coverage)
Front- und Backend	<ul style="list-style-type: none"> • Typescript¹⁵ • GraphQL¹⁶ • Lerna¹⁷ (zur Verwaltung des Monorepositories)
Tooling	<ul style="list-style-type: none"> • GitHub¹⁸ • GitHub Actions¹⁹ (Ausführung von CI)

Tabelle 6.1: Technologien

³<https://reactjs.org>⁴<https://material-ui.com>⁵<https://testing-library.com/docs/react-testing-library/intro>⁶<https://www.apollographql.com/docs/react>⁷<https://react.i18next.com>⁸<https://formik.org>⁹<https://docs.nestjs.com>¹⁰<https://www.apollographql.com/docs/apollo-server>¹¹<https://typeorm.io>¹²<https://www.postgresql.org>¹³<https://jestjs.io/>¹⁴<https://github.com/istanbuljs/nyc>¹⁵<https://www.typescriptlang.org>¹⁶<https://graphql.org>¹⁷<https://lerna.js.org>¹⁸<https://github.com>¹⁹<https://github.com/features/actions>

6.3 Building Block View

6.3.1 Context View

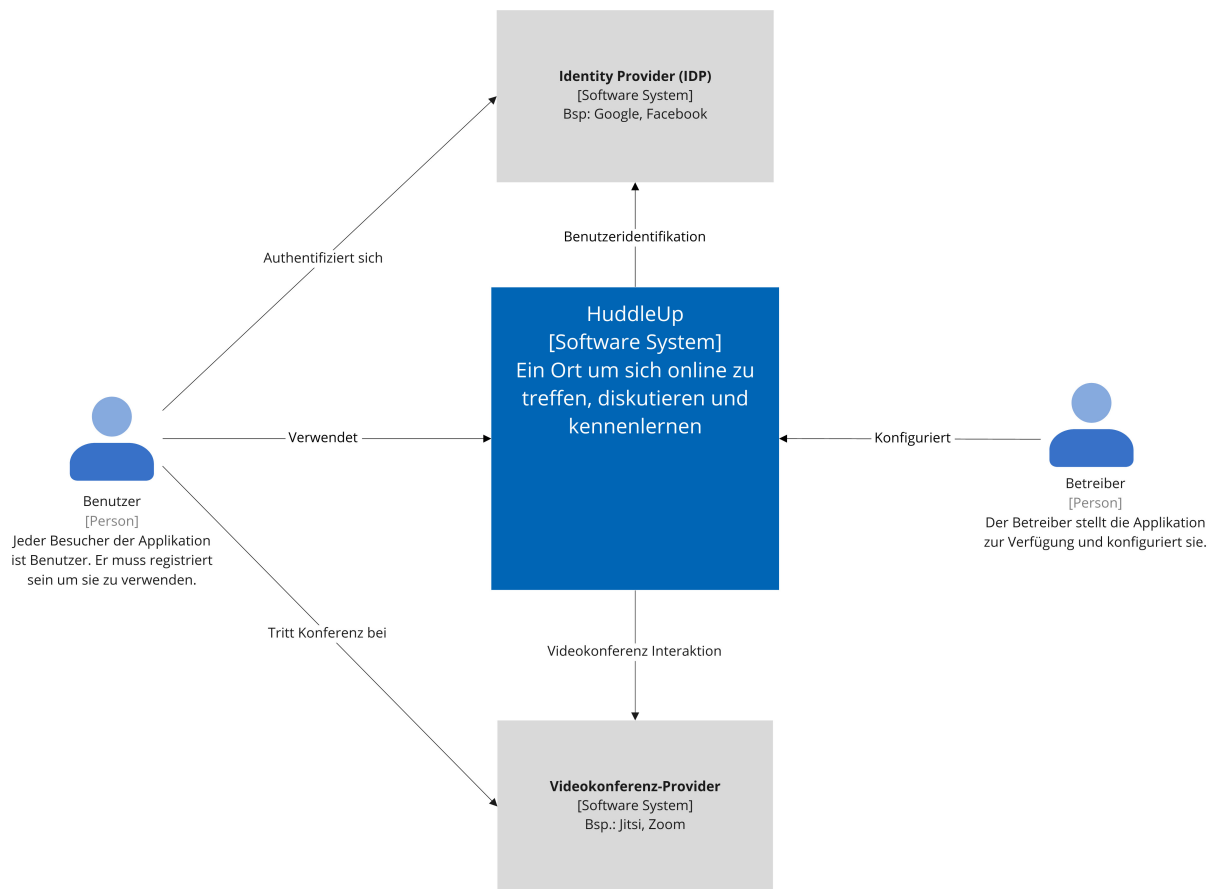


Abbildung 6.1: Context View

System	Beschreibung
Identity Provider (IDP)	Ermöglicht die Authentifizierung eines Benutzers für die Registrierung und das Single-Sign-On (SSO). Die Informationen des Benutzers werden an die Applikation weitergegeben. Die Kommunikation zwischen IDP und der Applikation erfolgt über OpenID Connect.
Videokonferenz-Provider	Es gibt bereits viele etablierte Anbieter auf dem Markt welche Video-Konferenzen mit einem eigenen Produkt wie Zoom Meeting oder Jitsi Meet anbieten. Da die Interaktionsmöglichkeiten je nach Hersteller unterschiedlich sind, stellen wir eine Interface zur Verfügung, welches produktspezifisch implementiert werden kann. Für einen Videokonferenz-Provider braucht es also jeweils eine eigene Implementation unserer API Schnittstelle um mit der Applikation zu interagieren.

6.3.2 Container View

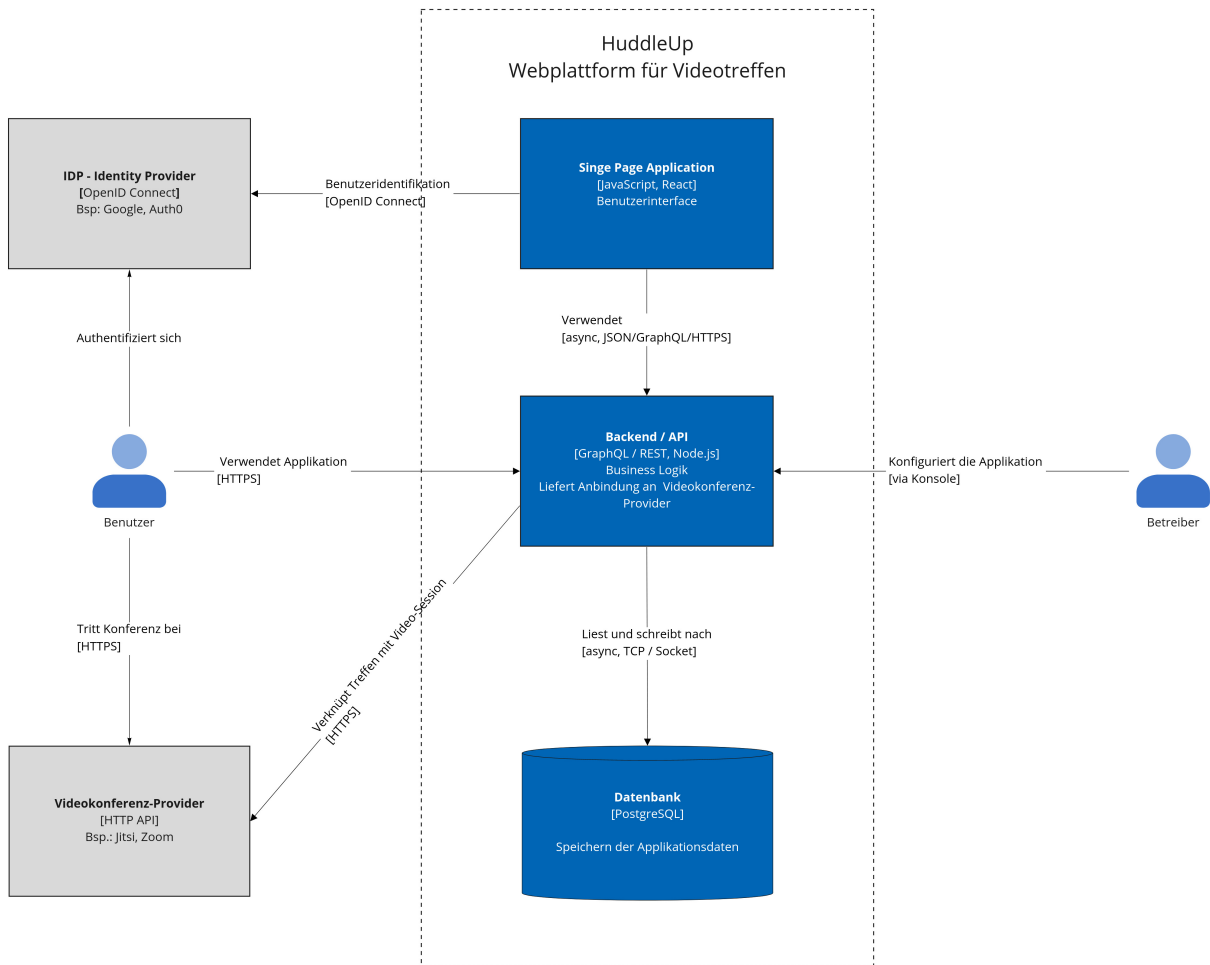


Abbildung 6.2: Container View

Container	Beschreibung
Single Page Applikation	Beinhaltet den <i>Presentation Logic</i> -Layer. Die Single Page Applikation (SPA) basiert auf React und liefert das Benutzerinterface. Die Daten werden asynchron mit Hilfe von Apollo Client über GraphQL vom Backend abgefragt.
Backend / API	Beinhaltet den <i>Business Logic</i> -Layer. Die Applikation basiert auf dem NestJS Framework. Es beinhaltet die Anbindung an den Videokonferenz-Provider. Die Interaktionen mit der Datenbank und dem Videokonferenz-Provider werden asynchron durchgeführt.
Datenbank	Beinhaltet den <i>Data Access</i> -Layer. Als Datenbanksystem wird PostgreSQL genutzt.

6.3.3 Component View

Backend / API

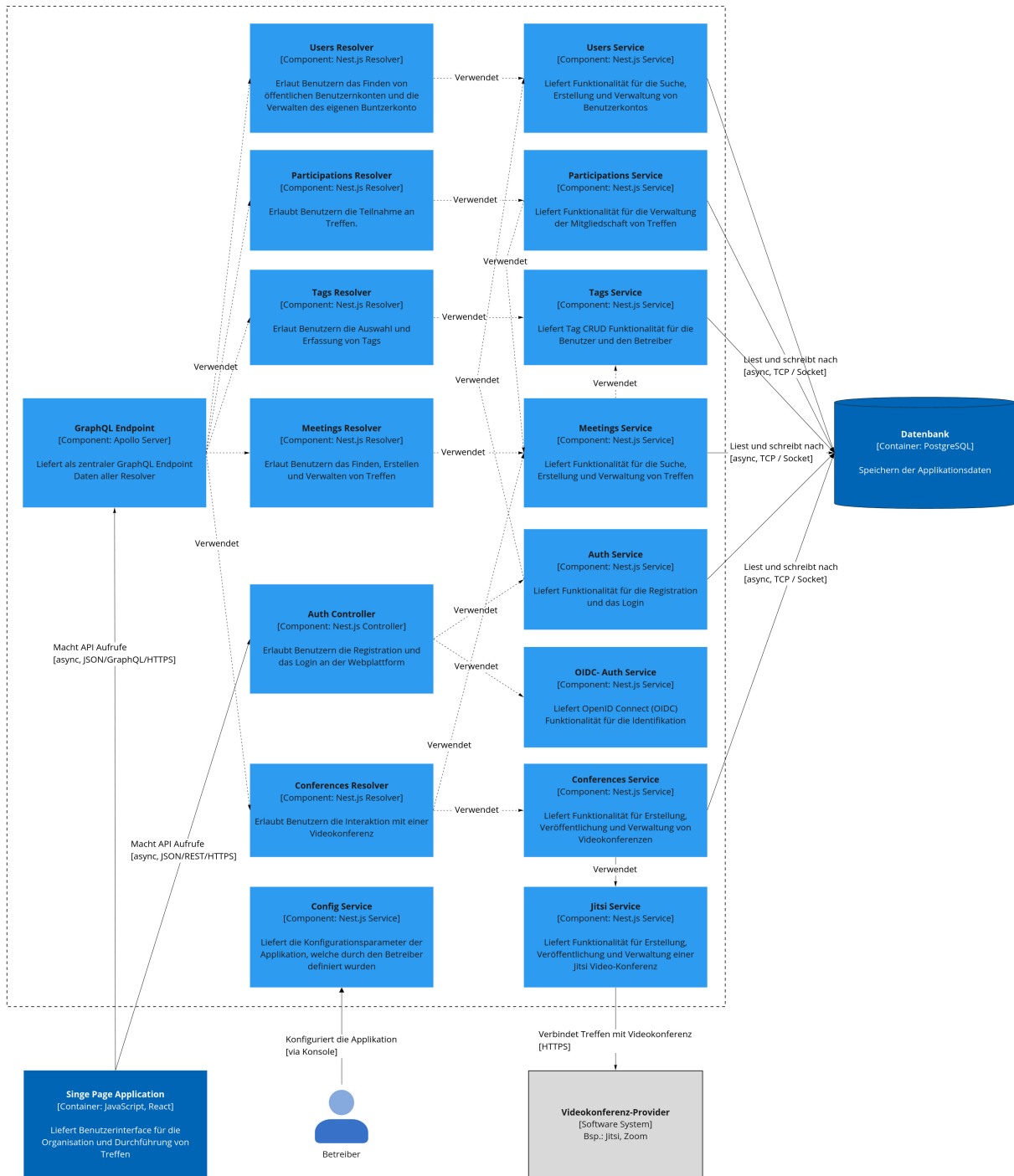


Abbildung 6.3: Backend / API Component View

Komponente	Beschreibung
Single Page Application [Container: JavaScript, React]	Liefert Benutzeroberfläche für die Organisation und Durchführung von Treffen
GraphQL Endpoint [Component: Apollo Server]	Liefert als zentraler GraphQL Endpoint Daten aller Resolver
Users Resolver [Component: Nest.js Resolver]	Erlaubt Benutzern das Finden von öffentlichen Benutzerkonten und die Verwalten des eigenen Benutzerkonto
Participations Resolver [Component: Nest.js Resolver]	Erlaubt Benutzern die Teilnahme an Treffen
Tags Resolver [Component: Nest.js Resolver]	Erlaubt Benutzern die Auswahl und Erfassung von Tags
Meetings Resolver [Component: Nest.js Resolver]	Erlaubt Benutzern das Finden, Erstellen und Verwalten von Treffen
Auth Controller [Component: Nest.js Controller]	Erlaubt Benutzern die Registrierung und das Login an der Webplattform
Conferences Resolver [Component: Nest.js Resolver]	Erlaubt Benutzern die Interaktion mit einer Videokonferenz
Config Service [Component: Nest.js Service]	Liefert die Konfigurationsparameter der Applikation, welche durch den Betreiber definiert wurden
Users Service [Component: Nest.js Service]	Liefert Funktionalität für die Suche, Erstellung und Verwaltung von Benutzerkonten
Participations Service [Component: Nest.js Service]	Liefert Funktionalität für die Verwaltung der Mitgliedschaft von Treffen
Tags Service [Component: Nest.js Service]	Liefert Tag CRUD Funktionalität für die Benutzer und den Betreiber
Meetings Service [Component: Nest.js Service]	Liefert Funktionalität für die Suche, Erstellung und Verwaltung von Treffen
Auth Service [Component: Nest.js Service]	Liefert Funktionalität für die Registrierung und das Login
OIDC-Auth Service [Component: Nest.js Service]	Liefert OpenID Connect (OIDC) Funktionalität für die Identifikation
Conferences Service [Component: Nest.js Service]	Liefert Funktionalität für Erstellung, Veröffentlichung und Verwaltung von Videokonferenzen
Jitsi Service [Component: Nest.js Service]	Liefert Funktionalität für Erstellung, Veröffentlichung und Verwaltung einer Jitsi Video-Konferenz
Datenbank [Container: PostgreSQL]	Speichern der Applikationsdaten
Videokonferenz-Provider [Software System]	Bsp.: Jitsi, Zoom

GraphQL Endpoint	Die GraphQL-Integration in NestJS bietet die Möglichkeit über einen zentralen Endpoint alle Anfragen des Clients entgegen zu nehmen und automatisch an die zuständigen Module weiterzuleiten.
Users Modul	Das <i>Users Modul</i> beinhaltet einen GraphQL-Resolver für Anfragen und Antworten. Der Resolver verwendet einen Service, welcher für die Verwaltung der Benutzer zuständig ist.
Participations Modul	Das <i>Participations Modul</i> beinhalten einen GraphQL-Resolver für Anfragen und Antworten. Der Resolver verwendet einen Service für das Finden, Liefern von Teilnehmer, Einschreiben und Austreten aus Treffen.
Tags Modul	Das <i>Tags Modul</i> beinhaltet einen GraphQL-Resolver für Anfragen und Antworten. Der Resolver verwendet einen Service, welcher für Tag-CRUD-Operationen zuständig ist.
Meetings Modul	Das <i>Meetings Modul</i> beinhaltet einen GraphQL-Resolver für Anfragen und Antworten. Der Resolver verwendet einen Service, welcher Funktionalität für die Suche, Erstellung und Verwaltung von Treffen bereitstellt.
Auth Modul	Das <i>Auth Modul</i> beinhaltet einen REST-Controller für Anfragen und Antworten. Der Controller verwendet den Auth-Service für die Registration und das Login und die interne Authentifikation. Der OIDC-Auth Service wird für die Überprüfung der Identifikation gegenüber dem Identity Provider verwendet.
Conferences Modul	Das <i>Conferences Modul</i> beinhaltet einen GraphQL-Resolver für Anfragen und Antworten. Der Resolver verwendet einen Service für die Erstellung, Veröffentlichung und Verwaltung der Videokonferenzen. Die Videokonferenz-Provider-spezifische Funktionalität wird durch einen dazugehörigen Service übernommen. Da nur eine Implementation existiert, ist dies derzeit der <i>Jitsi-Service</i> .
Config Module	Es existieren mehrere <i>Config Module</i> . Diese sind gleich aufgebaut und unterteilt nach Zuständigkeitsbereich. Es gibt die 6 Konfigurations-Module (<i>app, auth, database, graphql, jitsi und meetings</i>) mit einem dazugehörigen Service. Diese Module werden in den jeweiligen Services wo eine Konfiguration notwendig ist über <i>Dependency Injection</i> eingebunden.

SPA

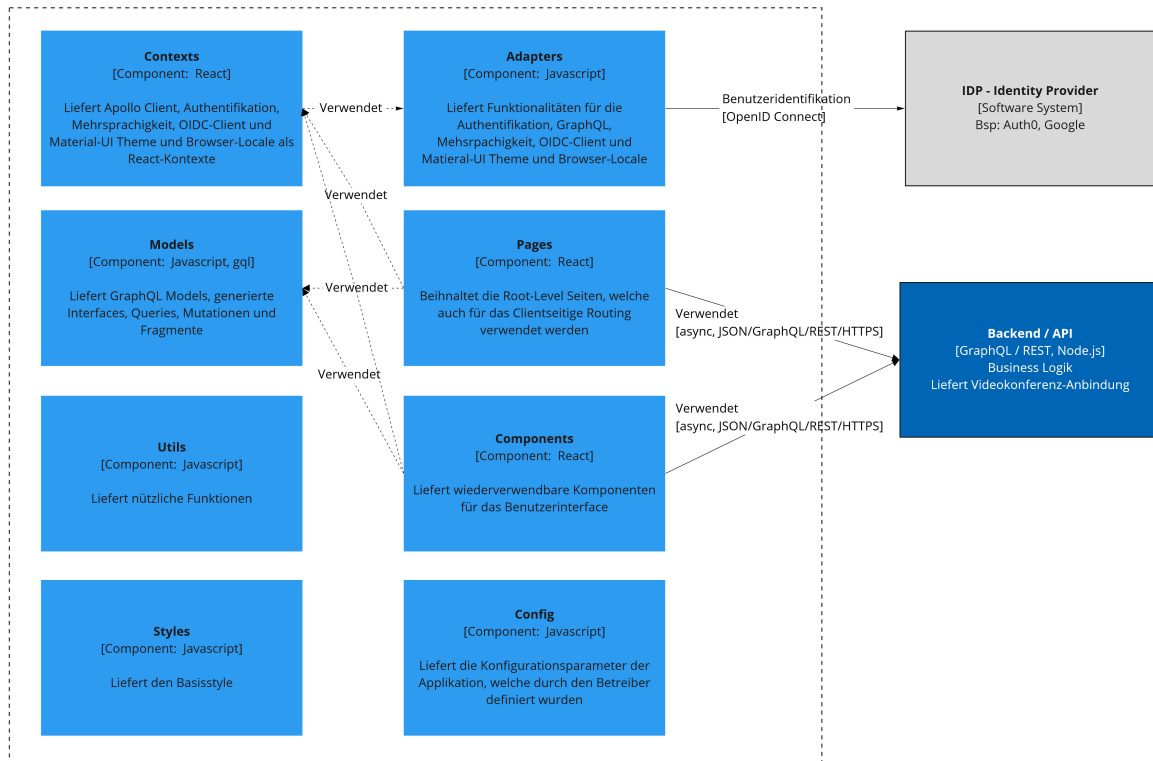


Abbildung 6.4: SPA Component View

Komponente	Beschreibung
Adapters	Adapter stellen die Verbindung von der Applikation zur Aussenwelt her. Dies beinhaltet das Aufrufen von externen Services wie dem Identity Provider, dem GraphQL-Server der API, oder das Lesen von Übersetzungs-Ressourcen.
Contexts	Contexts stellen die Adapter dem Rest der Applikation zur Verfügung. Es handelt sich dabei um React-Contexts ²⁰ und können als eine Form von Dependency Injection angesehen werden.
Models	Models umfassen Funktionalitäten und Definitionen für einzelne Modelle. Dies umfasst GraphQL-Queries, -Fragmente und -Mutationen, aus dem GraphQL-Schema generierte Typen und Interfaces, sowie Hilfsfunktionen.
Pages	Pages sind Route-Level Komponenten, welche die oberste Stufe der Komponentenhierarchie darstellen.
Components	Komplexe Probleme und wiederverwendbare Teile werden in Komponenten unterteilt, welche sich für eine spezifische Aufgabe spezialisieren.

²⁰<https://reactjs.org/docs/context.html>

Utils	Utils beinhalten geteilte Hilfs-Funktionen, welche nicht direkt einem Model, oder Adapter zugewiesen werden können und von mehreren Komponenten verwendet werden.
Styles	Styles beinhaltet die globale Definition des Themes für <i>Material-UI</i> .
Config	Unter Config befindet sich die Schnittstelle zwischen den Einstellungen des Betreibers und der Applikation. Der Aufbau ist vergleichbar mit der Config im Backend. Im Vergleich zum Backend wird jedoch nur ein Config-Service erstellt.

6.4 Runtime View

Der folgende Abschnitt beschreibt wichtige interne Abläufe und Interaktionen zwischen Komponenten der Applikation.

6.4.1 Videokonferenz-Provider

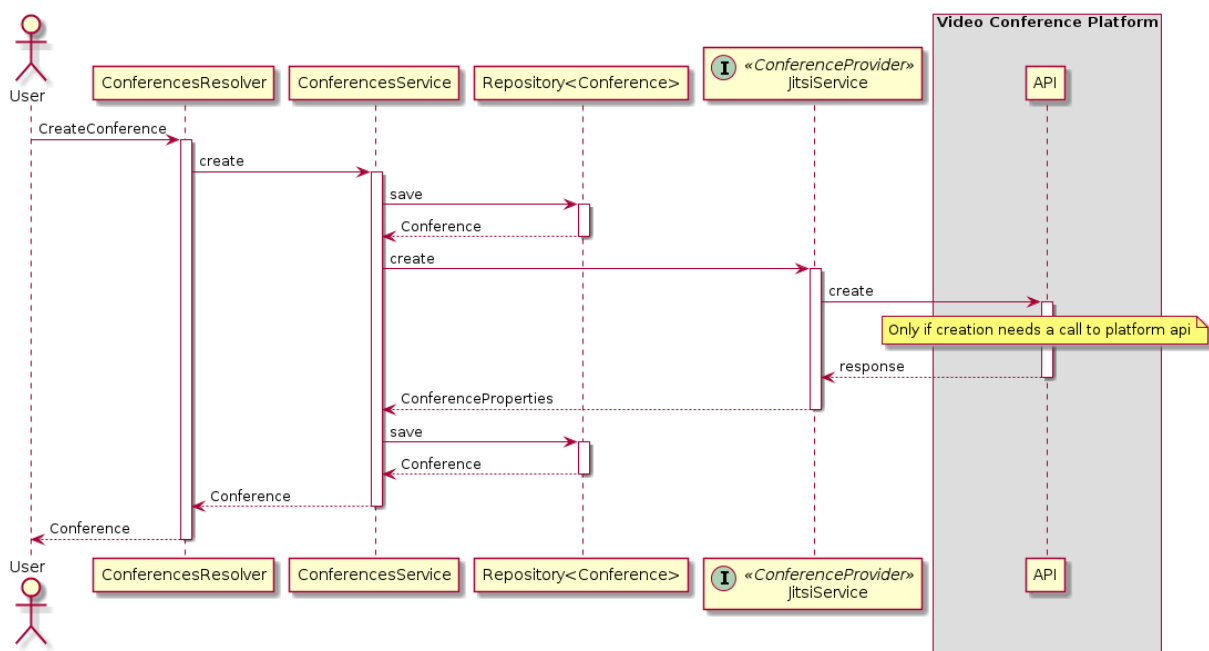


Abbildung 6.5: Interaktion von Komponenten bei der Erstellung einer neuen Konferenz

Das oben stehende Diagramm zeigt die Interaktion einzelner Komponenten des Backends bei der Erstellung einer Videokonferenz. Es bildet nur den fehlerfreien Ablauf ab.

1. Die GraphQL Mutation des Benutzer wird durch den dafür zuständigen `ConferencesResolver` entgegengenommen.
2. Die Operation wird an den `ConferencesService` weitergegeben.
3. Der `ConferencesService` erstellt die Entität und speichert sie über das `Repository` in der Datenbank.

4. Die erstellte Konferenz wird der aktiven Implementation des `ConferenceProvider` Interfaces übergeben (in diesem Falle `JitsiService`)
5. Der `ConferenceProvider` führt die benötigten Aktionen zur Erstellung der Konferenz aus. Dies *kann* den Aufruf an die API der entsprechenden Videokonferenz-Plattform beinhalten.
6. Der `ConferencesService` speichert die vom `ConferenceProvider` erhaltenen Eigenschaften als Attribut der Konferenz ab. Der `ConferencesProvider` hat *keinen* Zugriff auf die Datenbank.
7. Die erstellte Konferenz wird über den Resolver an den Benutzer zurückgegeben.

Weiter Abläufe zum Starten und Beenden der Konferenz sind diesem Ablauf ähnlich.

6.4.2 Authentifikation

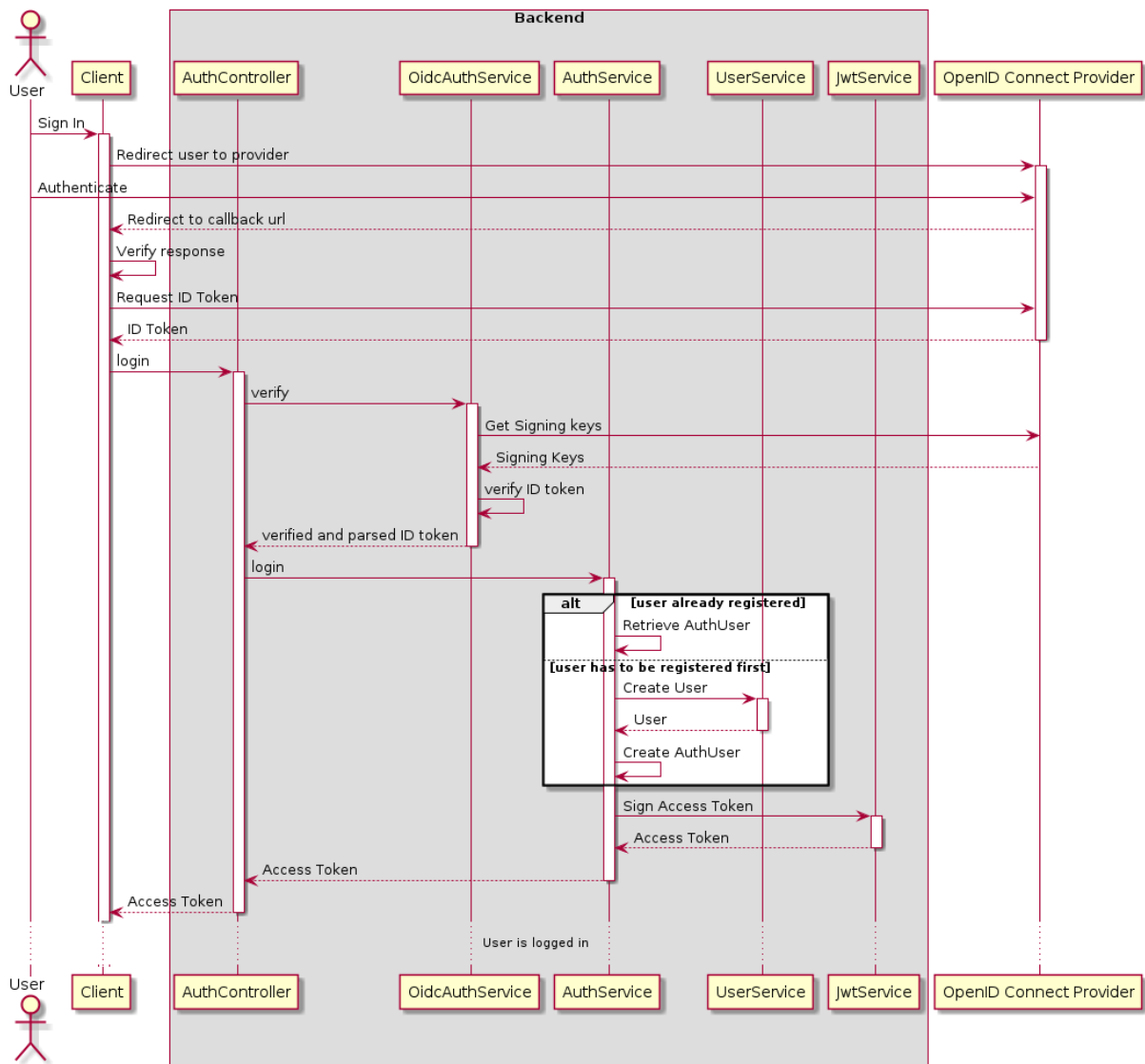


Abbildung 6.6: Interaktion von Komponenten bei der Authentifikation eines Benutzers

Das obenstehende Diagramm zeigt die Interaktion des Clients, des **OpenID Connect-Providers** und Komponenten des Backends bei der Authentifikation eines Benutzers. Es bildet nur den fehlerfreien Ablauf ab. Dieser Ablauf teilt sich in zwei Phasen auf:

1. Authentifikation gegenüber Provider
 - (a) Der Benutzer will sich im Client anmelden
 - (b) Der Client leitet den Benutzer an den vorkonfigurierten OIDC Provider weiter
 - (c) Der Benutzer authentifiziert sich beim Provider und wird zum Client zurückgeleitet
 - (d) Der Client verifiziert die Antwort und fordert den ID Token beim Provider an
2. Authentifikation gegenüber Backend
 - (a) Der Client sendet den ID Token zur Authentifikation an den **AuthController**
 - (b) Der **AuthController** ruft den **OidcAuthService** zur Überprüfung des ID Tokens auf. Diese Überprüfung wird aufgrund der Konfiguration und dem Abruf der Provider-Schlüssel durchgeführt. Der ID Token wird entschlüsselt zurückgegeben.
 - (c) Bei einer erfolgreichen Verifikation wird der Benutzer über den **AuthService** angemeldet. Dieser erstellt einen neuen Benutzer über den **UserService**, falls noch kein dazugehöriger existiert.
 - (d) Der **AuthService** erstellt dann einen Access-Token und lässt diesen durch den **JwtService** signieren.
 - (e) Der signierte Access Token wird an den Client zurückgesendet, welcher diesen für weitere authentifizierte Aufrufe an das Backend verwenden kann.

6.5 Deployment View

Der folgende Abschnitt beschreibt wie die Applikation in einer produktiven Umgebung betrieben werden kann. Es wird dabei hauptsächlich beschrieben, wie das Deployment während der Durchführung aufgebaut war. Die Struktur, oder die Verteilung der Komponenten kann je nach Betreiber von dieser abweichen.

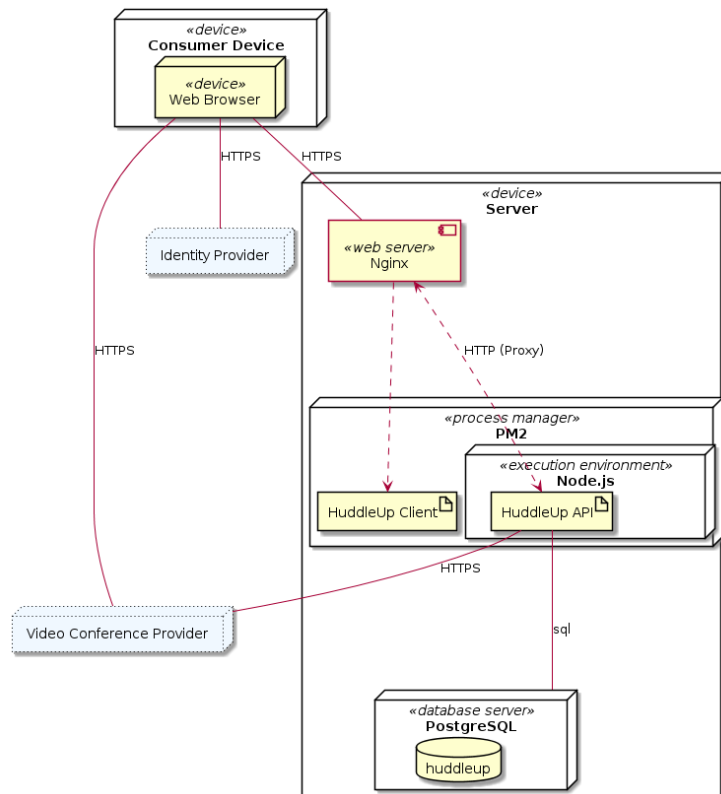


Abbildung 6.7: Deployment-Diagramm

In der beschriebenen Konfiguration befinden sich alle Komponenten auf dem selben Server. Das Front- und das Backend werden dabei beide durch *Nginx*²¹ ausgeliefert. Die Verschlüsselung über HTTPS wird dabei durch *Nginx* übernommen. *PM2*²² kommt für beide Teilapplikationen als Prozessmanager zum Einsatz und stellt sicher, dass beide bei einem Neustart des Systems gestartet werden. Das Frontend kann auch ohne *PM2* ausgeliefert werden, da es sich um statische Dateien handelt. *PM2* wird hier jedoch zur Auslösung des Kompilierungsvorgangs eingesetzt, was den Update-Prozess vereinfacht.

Der *Video Conference Provider* war während der Entwicklung eine eigene Jitsi Instanz und wurde von uns auf einem separaten Server selber gehostet. Da es sich jedoch nicht um einen integralen Bestandteil der Applikation handelt, wird er hier nicht weiter ausgeführt.

6.6 Crosscutting Concepts

Dieses Kapitel basiert auf dem *arc42 Template* und beschreibt generelle Lösungsansätze, welche für verschiedene Teile der Webplattform relevant sind.

6.6.1 User Experience Concepts (UX)

Responsives Webdesign Das einheitliche Design und ein Mobile-First Ansatz wird durch die Verwendung von *Material-UI* erreicht. Dessen *React*-Komponenten implementieren ein *responsives Webdesign* mit anpassbarem Theming. Auf dieser Basis haben wir

²¹<https://nginx.org/>

²²<https://pm2.keymetrics.io>

verschiedene Layouts für die jeweiligen Bildschirmgrößen (Desktop, Mobile) erstellt. Die Icons stammen alle aus dem Material-UI.

Mehrsprachigkeit Für die Mehrsprachigkeit wird die React-Version von **I18next** verwendet. Die Browsersprache des Benutzers wird ausgelesen und wenn die Sprache unterstützt und in der Konfiguration aktiviert ist, wird diese verwendet. Falls die verwendete Sprache nicht vorhanden ist, kommt die vom Betreiber vordefinierte Standardsprache zum Einsatz. Die Übersetzungsdateien haben eine **JSON**-Struktur und werden mit dem Client (**SPA**) ausgeliefert. Es sind bereits Übersetzungen auf Deutsch und Englisch erstellt worden. Die Unterstützung einer neuen Sprache ist durch das Erfassen einer neuen Übersetzungsdatei in der Form `public/locals/[locale]/translation.json` möglich.

Formulare Alle Formulare basieren auf **Formik**. Die Bibliothek vereinfacht und vereinheitlicht die Validierung und Verwendung von Formularen.

6.6.2 Operation Concepts

Logging **NestJS** bietet einen integrierten, konsolenbasierten Logger an, welcher Fehler im System auf der Konsole ausgibt. Der System Logger kann erweitert werden durch eine eigene Implementation. Im produktiven Betrieb ist ein Datei-basiertes Logging hilfreich. Zur Unterstützung der Konfiguration und Formatierung der Logausgaben in eine Datei wird **Winston**²³ verwendet.

Administration Die Administration ist durch das Setzen und Ändern von Environment-Variablen einfach durch den Betreiber machbar. Es gibt eine Environment-Datei für Anpassungen im Frontend und eine für das Backend. Grundlegende, sowie ergänzende Optionen können durch den Betreiber gesetzt werden. Für Änderungen während dem Betrieb wäre eine Erweiterung um ein Administrations-Interface notwendig.

Skalierung Aufgrund der tokenbasierten Authentifikation, die ohne Serversession auskommt, dem modularen Aufbau, den externen Services wie Videokonferenz-Provider und Identity Provider, ist eine horizontale Skalierung mit weiteren Node.js Instanzen möglich.

Caching im Client Mit dem Einsatz von **GraphQL** und **Apollo Client** ist ein Caching standardmässig aktiviert. Das heisst, eine Abfrage über bestehende Daten wird nur erneut vom Backend angefordert, wenn diese nicht im Cache vorhanden ist, oder explizit gewünscht wird. Eine Schwierigkeit ist dabei zu erkennen, wann existierende Daten nicht mehr aktuell sind. Um z.B. den Status eines laufenden Treffens aktuell zu halten, gibt es die Möglichkeit zu Pollen. Dabei wird in einem vordefinierten Intervall das Backend erneut abgefragt.

6.6.3 Abstraktion

Data Mapper Pattern Bei der Verwendung des Data Mapper Patterns werden alle Datenabfragen in einer separaten Klasse definiert und **CRUD**-Operationen werden über

²³<https://github.com/winstonjs/winston>

Repositories durchgeführt.[2]. Die Entitäten werden bewusst schlank gehalten und beinhalten im besten Fall nur die Attribute. Das Data Mapper Pattern wird in der Applikation durch **TypeORM** eingesetzt, welches die Rolle des Data Mappers übernimmt. Repositories werden nicht selbst erstellt, sondern durch TypeORM zur Verfügung gestellt.

Code-First Bei der Verwendung von TypeORM gibt es verschiedene Möglichkeiten für die Schema-Definition, wovon wir den Code-First-Ansatz anwenden. Dabei sind Datenbank-Entitäten als **TypeScript** Klassen erstellt und mit entsprechender Annotation von TypeORM definiert worden. Beim Code-First Ansatz wird aus diesen annotierten Klassen das Datenbankschema erstellt. TypeORM übernimmt bei der Ausführung die objektorientierten Abbildungen um das Datenbankschema und anschliessend die Datenbank zu erstellen.

6.6.4 Datenbank Migrationen

Durch eine Option kann zwischen Entwicklungsbetrieb, bei der automatisch die Datenbank synchronisiert wird und dem produktiven Betrieb, bei der Änderungen über Migrationen durchgeführt werden, gewechselt werden. Die Migrationen bieten Up- und Downgrades an, um das Schema anhand gewünschten Codeänderungen zu mutieren.

6.6.5 Kommunikation und Integration

API-Schnittstelle Mit **GraphQL** lässt sich die **API-Schnittstelle** über ein Schema beschreiben. Gegen ein solches Schema können client-seitig komplexe Datenabfragen und Datenmanipulationen durchführen werden. Der Client kann die Daten entsprechend ihrer Relation verschachtelt zueinander anfordern. Im Backend läuft ein GraphQL-Server, über welchen Resolver die Abfragen und Mutationen auflösen. Ein Resolver interpretiert die Anfrage, übersetzt die Felder in die Geschäftslogik des darunterliegenden Services und liefert die gewünschten Daten an den Client zurück. Das GraphQL-Schema wird durch Annotationen in den TypeScript Klassen definiert. Neben den Resolvern werden für die Authentifikation reguläre **HTTP-Controller** verwendet.

Exception- und Fehlerhandling Fehler werden generell im Backend und Frontend durch Wrapper aufgefangen und verhindern so ein Abstürzen der Applikation. Im Frontend werden **React-Error Boundaries** eingesetzt, welche Fehler der Kind-Komponenten abfangen und somit eine Propagation verhindern. Im Backend kommt der in NestJS eingebaute **Exception-Layer** zum Einsatz, welcher alle nicht behandelten Fehler auffängt und eine benutzerfreundliche Antwort liefert. Um spezifisches Fehlverhalten auszulösen gibt es vordefinierte **HTTP-Exception Klassen**.

Typescript-Definitionen Zur Synchronisation des Frontends mit dem GraphQL-Schema des Backends wird im Frontend das Codegen-Tool von *Apollo* verwendet. Es liest das automatisch generierte Schema des Backends aus und gleicht es mit den im Frontend definierten Queries, Mutationen und Fragments ab. Anschliessend generiert es TypeScript-Interfaces für die Abfragen, welches die Verwendung von Abfragen und Antworten im Frontend stark vereinfacht.

6.6.6 Sicherheit

Authentifikation Die Authentifikation gegenüber der API wird über **JWT**-Token gelöst. Die Token werden durch den *AuthService* generiert und für die Verwendung der GraphQL-API benötigt. Die Überprüfung innerhalb der Resolver und Controller wird über die Guard-Funktion von Nest gelöst.²⁴ Durch die Verwendung eines Guards wird der in der Anfrage enthaltene Token auf dessen Existenz und Korrektheit überprüft.

Autorisation Ob der Benutzer autorisiert ist eine Operation durchzuführen, wird in den Resolver und Services kontrolliert und durch entsprechende Exceptions abgefangen.

6.6.7 Schichtenmodell

Das Backend ist in mehrere Schichten unterteilt, welche spezifische Aufgaben übernehmen und Abhängigkeiten zueinander haben.

- Die **CONTROLLING-SCHICHT** ist für das Entgegennehmen von Daten aus Anfragen zuständig. Es beinhaltet die Resolver und Controller, welche die Anfragen an die zuständigen Services weiterleiten. Die Authentifikation und Autorisation findet in dieser Schicht statt.
- Die **SERVICE-SCHICHT** beinhaltet die Geschäftslogik der Problem domain.
- Die **DATA-ACCESS-SCHICHT** ermöglicht das Lesen, Manipulieren und Speichern der Daten und wird mit dem *Repository Pattern* von der Service-Schicht entkoppelt.
- Die **PERSISTENCE-SCHICHT** beinhaltet die Datenbank-Entitäten.

Die Abhängigkeit ist immer nur von oben nach unten, also von **CONTROLLING-SCHICHT** zu **SERVICE-SCHICHT**, zu **DATA-ACCESS-SCHICHT** zu **PERSISTENCE-SCHICHT**.

Mapping zwischen Schichten Um die Datenobjekte zwischen den Schichten unabhängig voneinander zu halten, werden unterschiedliche Interfaces und Klassen verwendet. Für die Kommunikation mit dem Client wird über Data Transfer Objects (**DTO**) die Struktur der Daten definiert und überprüft. Die Services verwenden eigene Interface oder direkt die Datenbank-Entitäten aus der Persistenz-Schicht.

Module und Schichten Die Architektur von NestJS basiert auf der Verwendung von Modulen²⁵. Ein Modul enkapsuliert die Funktionalität zu einer spezifischen Subdomäne. Dies bedeutet, dass nicht Module einer Schicht zugeordnet werden, sondern jedes Modul in eine Auswahl dieser Schichten aufgeteilt ist.

6.7 Architekturentscheidungen

6.7.1 NestJS

NestJS ist ein in TypeScript geschriebenes Node.js-Framework. Es verwendet eine modulare, stark von Angular inspirierte Architektur. Durch diese Modularität existiert eine Viel-

²⁴<https://docs.nestjs.com/guards>

²⁵<https://docs.nestjs.com/modules>

zahl von Integrationen mit anderen Bibliotheken. Die Flexibilität und breite Unterstützung würde uns deshalb in der Entwicklung eine gute Grundlage bieten.

Architekturentscheidung: NestJS

Im Kontext der Wahl des Backend-Frameworks, konfrontiert mit der Anforderung von Modularität und Flexibilität, entschieden wir uns für NestJS und nicht für eine Einzel-Implementationen auf Basis eines Express-Webservers, um auf einer bereits etablierten Architektur aufzubauen und akzeptieren, dass wir vorgegebene Strukturen einhalten müssen und an das Framework gebunden sind.

6.7.2 GraphQL

GraphQL steht für *Graph Query Language* und ist eine Sprache zur Abfrage und Veränderung von Graphen. Ursprünglich intern von Facebook entwickelt, ist GraphQL seit 2015 als Open Source verfügbar. GraphQL wurde als Alternative zu **REST** entwickelt um die Abfrage von komplexen Datenstrukturen zu vereinfachen. Durch die Möglichkeit, dass der Client nur die Daten abfragt, welche er braucht, können die Anzahl Abfragen reduziert werden.

Architekturentscheidung: GraphQL

Im Kontext der Kommunikation zwischen Front- und Backend, konfrontiert mit der Komplexität der erwarteten Abfragen, entschieden wir uns für GraphQL und nicht für REST, um die Anzahl der Abfragen und Aufarbeitung der Daten zu reduzieren und akzeptieren den zusätzlichen Aufwand für die Einarbeitung in eine uns wenig bekannte Technologie.

6.7.3 Apollo Plattform (GraphQL)

Bei der Implementation von modernen Frontends entsteht schnell der Bedarf für eine Verwaltung des Applikationszustands sowie die Vermeidung unnötiger Abfragen. Ein populäre Kombination für React-basierte Applikationen ist die Kombination aus REST-Abfragen und der Verwaltung des Zustands mit Redux²⁶. Dieser Ansatz benötigt jedoch einen initialen Aufwand für das Erstellen des Redux-Stores und löst das Problem des Caching nur begrenzt. Bei der Verwendung von GraphQL besteht mit dem **Apollo Client** eine Alternative, welches diese beiden Anforderungen abdeckt. Der Apollo Client hat ein integriertes Caching, welches Anfragen nur erneut ausführt wenn benötigt und reduziert somit die Anzahl unnötiger Abfragen. Zusätzlich bietet es durch diesen Cache und die Möglichkeit lokalen Zustand zu verwalten eine Lösung für das Verwalten des Applikationszustands. Voraussetzung dafür ist jedoch, dass der Client richtig konfiguriert wird.

²⁶<https://redux.js.org/>

Architekturentscheidung: Apollo Plattform

Im Kontext der Wahl für die Verwaltung des Applikationszustands, konfrontiert mit dem Implementationsaufwand und Caching-Anforderungen, entschieden wir uns für die Verwendung der Apollo Plattform und gegen den Einsatz eines Redux-Stores um den Implementations-Aufwand zu reduzieren und von den Vorteilen der Plattform zu profitieren und akzeptieren, dass unsere fehlende Erfahrung zu zusätzlichem Aufwand führen kann.

6.7.4 Tokenbasierte Authentifizierung

Architekturentscheidung: Tokenbasierte Authentifizierung

Im Kontext der Authentifikation von Benutzern, konfrontiert mit der Anforderung, dass mehrere Instanzen der Server-Komponente laufen können, entschieden wir uns für eine Token-basierte Authentifizierung und nicht für eine Session-basierte Authentifizierung um keinen *Benutzer-State* auf dem Server zu speichern und akzeptieren die damit verbundene Komplexität.

Kapitel 7

Risikomanagement

7.1 Einführung

Zweck

Dieses Dokument dient der Festhaltung von Risiken, welche identifiziert wurden und während der Projektdauer auftreten können. Die Risiken werden bei Erreichen jedes Meilensteins neu bewertet und neu aufgetretene Risiken erfasst.

Gültigkeitsbereich

Das Dokument ist gültig für die ganze Projektlaufzeit und wird kontinuierlich ergänzt. Es widerspiegelt den aktuellsten Stand, eine Übersicht der Revisionen im Verlauf des Projekts ist im Anhang (Abschnitt [A.9](#)) zu finden.

Referenzen

Dieses Dokument baut unter anderem auf der Risikoanalyse der Studienarbeit der Teammitglieder auf.

7.2 Umgang mit Risiken

Die Ergebnisse aus der Risikoanalyse fließen in die Planung und die Priorisierung von Funktionen mit ein. In der letzten Iteration der Construction-Phase wird ein Puffer eingeplant, um Verzögerungen durch das Eintreten solcher Risiken zu kompensieren. Tritt ein schwerwiegendes Risiko ein, dessen Auswirkungen nicht durch den Puffer kompensiert werden können, wird der geplante Funktionsumfang verringert, um die Fertigstellung sicherzustellen.

Eintrittswahrscheinlichkeit Die Eintrittswahrscheinlichkeiten unterteilen sich in die Quartile *Sehr gering* (0–25 %), *Gering* (25–50 %), *Mittel* (50–75 %) und *Hoch* (75–100 %).

Schaden Das Ausmass der potenziellen Schäden unterteilt sich, aufsteigend nach Schwere, in die vier Bereiche *Sehr tief*, *Tief*, *Mittel* und *Hoch*.

7.3 Identifizierte Risiken

Die erkannten Risiken sind in der nachstehenden Tabelle festgehalten und werden anschliessend genauer erläutert.

Nr.	Risiko	Eintritts-Ws.	Schaden
R1	Fehlendes Know-How von Frameworks	gering	tief
R2	Ausfall von Teammitgliedern	gering	hoch
R3	Mangelnde Qualität der Ergebnisse	sehr gering	mittel
R4	Nicht erfüllbare Funktionalität	sehr gering	tief
R5	Datenverlust	sehr gering	mittel
R6	Mängel der Architektur	gering	tief
R7	Aufwandsunterschätzung	mittel	tief
R8	Abhängigkeiten zu Drittparteien	sehr gering	tief
R9	Nichtfertige Kernfunktionalität blockiert restliche Funktionen	gering	mittel

R1: Fehlendes Know-How von Frameworks

Frameworks erfordern zusätzliche Kenntnisse welche für die Umsetzung bekannt sein müssen

Vorbeugung

- Verwendung häufig eingesetzter und gut dokumentierter Frameworks
- Vorgängig Zeit einplanen bzw. Zeitreserven vorsehen

Verhalten beim Eintreten

- Zeitreserven nutzen um fehlendes Know-How aufzubauen
- Gegenseitige Unterstützung im Team

R2: Ausfall von Teammitgliedern

Langfristiger Ausfall von Mitgliedern aufgrund von Krankheiten (bspw. Covid-19) oder Verletzungen.

Vorbeugung

- Einhalten von Hygiene-Vorschriften

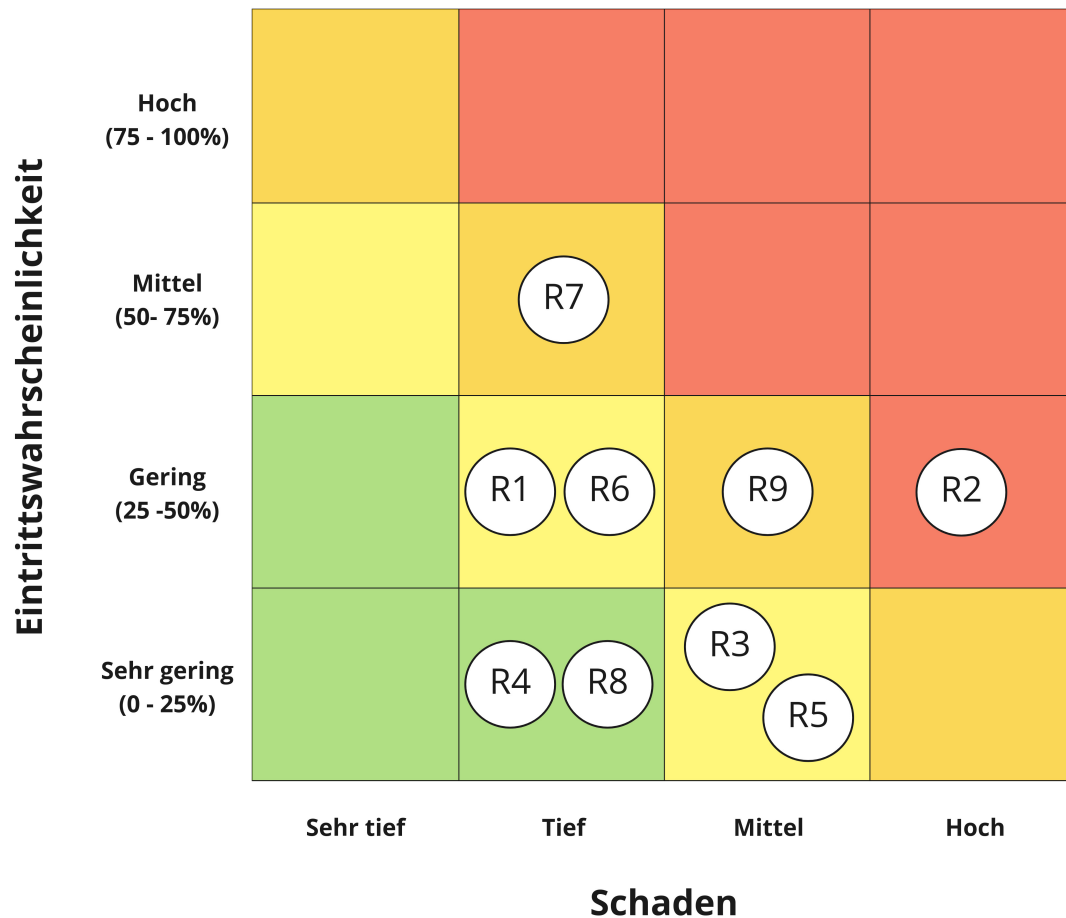


Abbildung 7.1: Risikomatrix

- Regelmässiges *pushen* von Code auf GitHub
- Regelmässige Dokumentation und Wissensaustausch

Verhalten beim Eintreten Reduktion von nicht-essentiellen Funktionen und Neuzuweisung von Aufgaben an die restliche Teammitglieder.

R3: Mangelnde Qualität der Ergebnisse

Lieferartefakte erfüllen nicht die gewünschte Qualität.

Vorbeugung

- Klaren Review-Prozess definieren
- Codeänderungen sind nur durch *Pull Requests* möglich
- *Definition of Done* für jedes Arbeitspaket definieren

R4: Nicht erfüllbare Funktionalität

Die geplante Funktionalität ist dem gewählten Technologie-Stack nicht im gewünschten Rahmen umsetzbar.

Vorbeugung Frühzeitige Entwicklung von Prototypen, um die Kapazitäten der Umgebungen zu erproben.

Verhalten beim Eintreten Reevaluation der betroffenen Funktion, ggf. Ersatz mit einer vergleichbarer Funktionalität.

R5: Datenverlust

Das Repository wird gelöscht, lokale Änderungen gehen verloren oder werden überschrieben.

Vorbeugung

- Regelmässiges *pushen* von Quellcode auf GitHub
- Verwendung von Cloud Services
- Regelmässige Backups

R6: Mängel der Architektur

Die gewählte Architektur kann die geforderten Anforderungen nicht wie geplant erfüllen.

Vorbeugung

- Genügend Zeit einplanen in der *Elaboration*-Phase
- Modulare Architektur verwenden um einfacher Anpassungen und Erweiterungen vorzunehmen

Verhalten beim Eintreten

- Anpassen der Architektur
- Reevaluation der Anforderung falls Architektur-Korrektur nur schwer umsetzbar

R7: Aufwandsunterschätzung

Der Aufwand für die Umsetzung ist höher als geplant.

Vorbeugung

- Reserven einplanen
- Funktionalitäten nach Relevanz klassifizieren und priorisieren

Verhalten beim Eintreten Reserven ausnutzen und Scope anpassen.

R8: Abhängigkeiten zu Drittparteien

Die verwendeten Drittparteien können *Rate Limits* einführen, die Nutzungsbedingungen und **API** ändern oder ihr Angebot einstellen. Dies kann dazu führen, dass gewisse Teile der Applikation nur noch beschränkt oder gar nicht mehr funktionieren.

Vorbeugung

- Evaluation der Drittpartei vor deren Einbindung
- Gewährleisten der Stabilität durch Auslegen der Architektur, sodass der Ausfall einer Drittpartei keinen Einfluss auf das Laufzeitverhalten der Applikation hat.

Verhalten beim Eintreten Reevaluierung der betroffenen Drittpartei, ggf. Ersetzen durch vergleichbare Drittpartei.

R9: Nichtfertige Kernfunktionalität blockiert restliche Funktionen

Ist eine zentrale Funktion noch nicht fertig während ein davon abhängige Funktion bereits implementiert werden sollte, kann dies zu Verzögerungen im Zeitplan kommen und Team-Mitglieder in ihrer Arbeit blockieren. Verzögerungen können auch dadurch entstehen, dass Wissen zuerst aufgebaut werden muss. Dies führt so zu einem grösseren Arbeitsaufwand als geplant und verursacht Verspätungen.

Vorbeugung

- Priorisierung der Implementation von Kernfunktionalitäten und frühzeitige Fixierung deren Interfaces
- Pair Programming bei Unsicherheiten

Glossar

Apollo Client Ein GraphQL-Client der eine eingebaute Integration zur **React** Bibliothek bietet, siehe <https://www.apollographql.com/docs/react>. 59, 67, 70

Application Programming Interface Eine Programmierschnittstelle (von englisch *Application Programming Interface*) ist eine digitale Schnittstelle zwischen zwei Softwaresystemen und erlaubt deren Anbindung zueinander. 80

arc42 Template Das arc42 Template enthält eine Reihe von Tipps und Vorgehensweisen für die Dokumentation und Kommunikation der Architektur. Siehe <https://arc42.org/overview>. 66

C4 Model Das C4 Model beschreibt Modelle zur Visualisierung von Software Architektur. 56

Figma Design-Applikation zur Gestaltung und Prototyping von Benutzer-Interfaces. Siehe <https://www.figma.com>. 13, 115

Formik Eine Open Source Bibliothek für **React** um Formulare zu überwachen, validieren und die Übermittlung zu orchestrieren. Siehe <https://formik.org>. 67

FURPS FURPS ist ein Akronym zur Kategorisierung von Qualitätsattributen von Software. Die einzelnen Buchstaben stehen für folgende Begriffe:

- **F**unctionality (Kapazität, Wiederverwendbarkeit, Sicherheit)
- **U**sability (Erlernbarkeit/Bedienbarkeit, Verständlichkeit)
- **R**eliability (Verfügbarkeit, Fehlertoleranz, Wiederherstellbarkeit)
- **P**erformance (Geschwindigkeit, Ressourcenverwendung, Skalierbarkeit)
- **S**upportability (Wartbarkeit, Erweiterbarkeit, Testbarkeit, Installierbarkeit)

. 13, 49

GraphQL GraphQL ist eine Open Source Datenabfrage- und Manipulationssprache und ein Laufzeitsystem zum Beantworten von Abfragen mit vorhandenen Daten. Siehe <https://graphql.org>. 20, 59, 67, 68, 70

i18next Das auf i18next basierende **React** Framework wird für die Mehrsprachigkeit verwendet. Es bietet viele Funktionalitäten an um die Übersetzungen optimal in die Webplattform zu integrieren. Siehe <https://react.i18next.com>. 67

Jitsi Eine Sammlung mehrerer eigenständiger Open Source Komponenten für Videokonferenzen, siehe <https://jitsi.github.io/handbook/docs/architecture>. 78

Jitsi Meet Ein Browser-Client für **Jitsi**. 21, 58

JSON Web Token Das JWT ermöglicht den Austausch von verifizierbaren Claims. Es ist ein auf **JSON** basiertes Access-Token, siehe <https://jwt.io>. 80

Material Design Von Google entwickelte Design-Sprache. Siehe <https://material.io>. 14, 20, 57

Material-UI *Material Design* ist eine von Google entwickelte Design-Sprache. Die *Material Designs* sind als React Komponenten verfügbar. Siehe <https://material-ui.com>. 66

Monorepository Repository, welches den Code für mehrere Projekte (oder Teilprojekte) enthält, statt diese in separaten Repositories zu verwalten . 15

MVP Abkürzung für *Minimum Viable Product* (dt. *minimal brauchbares Produkt*), die erste minimal funktionsfähige Version eines Produkts. 30, 34

NestJS Ein progressives Node.js-Framework für die Erstellung effizienter, zuverlässiger und skalierbarer serverseitiger Anwendungen. Siehe <https://docs.nestjs.com>. 59, 67, 69

OAuth (Open Authorization) Ein offenes Protokoll zur Authorisierung von Benutzern. Siehe <https://oauth.net>. 21, 78

OpenID Connect Eine Authentifizierungsschicht basierend auf **OAuth**, siehe <https://openid.net/connect>. 21, 54, 58, 65

PostgreSQL Kurz *Postgres*, ist ein freies, objektrelationales Datenbankmanagementsystem, siehe <https://www.postgresql.org>. 59

Pull Request Ein Arbeitsablauf um Quellcode-Änderungen von einem Branch (Quellcode-Zweig) in die eigentliche Quellcode-Basis zu übernehmen. 15

React Eine JavaScript Bibliothek zum Erstellen von Benutzeroberflächen, siehe <https://reactjs.org>. 59, 66, 77

SAML XML-Framework zum Austausch von Authentifizierungs- und Autorisierungsinformationen, siehe https://www.oasis-open.org/committees/tc_home.php. 21

Scrum Scrum ist ein agiles Vorgehensmodell zur Softwareentwicklung. Die Entwicklung wird in gleichmässige Iterationen durchgeführt, die in Scrum **Sprint** genannt werden. Es wird damit pro Sprint eine Lieferungen erreicht.. 1, 12

Single Page Application Eine Webapplikation, bei welcher das Generieren des User Interface und Routing rein clientseitig ausgeführt wird. 1

Sprint Beschreibt eine Zeiteinheit, in der das Team einen zuvor besprochenen Plan zur Erreichung eines Zwischenziels umsetzt. 29, 78

TypeORM TypeORM ist einer der besten **Object Relational Mapper** in der Node.js Welt, siehe <https://typeorm.io>. 68

TypeScript Von Microsoft entwickelte, stark typisierte Programmiersprache. Es handelt sich dabei um ein Superset von JavaScript und kann zu JavaScript kompiliert werden <https://www.typescriptlang.org>. 1, 68

Unified Process Wiederholendes und schrittweises Vorgehensmodell zur Softwareentwicklung. 1, 12, 28, 80

Use Case Ein Anwendungsfall der mögliche Szenarien mit den beteiligten Akteuren des betrachteten Systems auf ein fachliches Ziel aufzeigt. 13

User Story Im Rahmen der agilen Softwareentwicklung formulierte Software-Anforderung. 13

Y-Statements Ein von Olaf Zimmermann beschriebener Ansatz um Entscheidungen anhand eines Skeleton von Aussagen festzuhalten. 56

YouTrack Projektmanagement-Tool, siehe <https://www.jetbrains.com/youtrack>. 13, 31

Zoom Meeting Eine amerikanische Videokonferenz-Plattform, siehe <https://explore.zoom.us/de-de/meetings.html>. 58

Akronyme

- API** Application Programming Interface. 1, 58, 68, 76, 95, 97, *siehe Application Programming Interface*
- CI** Continius Integration. 15, 30, 31
- CLI** Command-Line Interface. 96
- CRUD** Create Read Update Delete. 61, 67
- DTO** Data Transfer Object. 69
- HTTP** Hypertext Transfer Protocol. 68
- IDP** Identity Provider. 58
- JSON** JavaScript Object Notation. 22, 67, 78
- JWT** JSON Web Token. 21, 69, 96, *siehe JSON Web Token*
- OIDC** OpenId Connect. 61
- ORM** Object Relational Mapper. 79
- REST** Representational State Transfer. 20, 21, 61, 70
- SPA** Single Page Application. 59, 67
- SSO** Single Sign-on. 58
- UP** Unified Process. 12, 28, *siehe Unified Process*

Literaturverzeichnis

- [1] GitHub Docs. *GitHub flow*. URL: <https://docs.github.com/en/free-pro-team@latest/github/collaborating-with-issues-and-pull-requests/github-flow>. (besucht am 07.03.2021).
- [2] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2003.
- [3] Claudia Furrer. *Guidelines für Studien- und Bachelorarbeiten im Studiengang Informatik*.
- [4] *Huddle* — Definition on Huddle by Merriam-Webster. URL: <https://www.merriam-webster.com/dictionary/huddle>. (besucht am 06.06.2021).
- [5] *Human Computer Interaction Design (M-HCD / I)*. URL: https://studien.hsr.ch/allModules/32684_M_HCD.html. (besucht am 02.06.2021).
- [6] *Jitsi API* — How to Add Video Meetings to Your Site and App. URL: <https://jitsi.org/api/>. (besucht am 12.06.2021).
- [7] Daniel Keller. *Software Engineering Experience*. 2021. URL: <http://www.danielkeller.info/book/se-buch-keller.pdf>.
- [8] Craig Larman. *Applying Evolutionary Requirements*. URL: https://www.craiglarman.com/wiki/downloads/applying_uml/larman-ch5-applying-evolutionary-requirements.pdf. (besucht am 10.06.2021).
- [9] Craig Larman. *Applying Evolutionary Use Cases*. URL: https://www.craiglarman.com/wiki/downloads/applying_uml/larman-ch6-applying-evolutionary-use-cases.pdf. (besucht am 10.06.2021).
- [10] *Media - Meetup*. URL: <https://www.meetup.com/media>. (besucht am 01.06.2021).
- [11] *OpenID Connect* — Google Identity — Google Developers. URL: <https://developers.google.com/identity/protocols/oauth2/openid-connect>. (besucht am 12.06.2021).
- [12] *SAML Configuration Options - Auth0*. URL: <https://auth0.com/docs/protocols/saml-protocol>. (besucht am 12.06.2021).
- [13] *Speculative Generality*. URL: <https://sourcemaking.com/refactoring/smells/speculative-generality>. (besucht am 12.06.2021).
- [14] Olaf Zimmermann. *Making Architectural Knowledge Sustainable—Industrial Practice Report and Outlook*. URL: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=31345>. (besucht am 11.03.2021).
- [15] Olaf Zimmermann. *Y-Statements*. URL: <https://medium.com/olzzio/y-statements-10eb07b5a177>. (besucht am 11.03.2021).

Abbildungsverzeichnis

1.1	Übersicht über Phasen, Sprints und Meilensteine	12
1.2	Wireframe für die Übersicht aller Treffen	14
2.2	Startseite	16
2.1	Vereinfachter Ablauf für Benutzer	17
2.3	Übersicht von laufenden und anstehenden Treffen	18
2.4	Detailansicht eines Treffens als Teilnehmer, mit Teilnahmeselektion	18
2.5	Detailansicht eines gestarteten Treffens als Organisator	19
3.1	Übersicht über Phasen, Sprints und Meilensteine	28
4.1	Use-Case-Diagramm	36
4.2	Registration	50
4.3	Durchführung	51
4.4	Teilnahme	52
5.1	Domänenmodell	54
6.1	Context View	58
6.2	Container View	59
6.3	Backend / API Component View	60
6.4	SPA Component View	62
6.5	Interaktion von Komponenten bei der Erstellung einer neuen Konferenz	63
6.6	Interaktion von Komponenten bei der Authentifikation eines Benutzers	64
6.7	Deployment-Diagramm	66
7.1	Risikomatrix	74
A.1	Risikomatrix, Stand MS1	113
A.2	Risikomatrix, Stand MS2	114
A.3	Risikomatrix, Stand MS3	115
A.4	Startseite	116
A.5	Anmeldung	116
A.6	Benutzerprofil	117
A.7	Alle Treffen	118
A.8	Meine Treffen	118
A.9	Treffen erstellen	119
A.10	Teilnahme an Treffen	120
A.11	Durchführung von Treffen	120

Tabellenverzeichnis

3.1	Zeitliche Rahmenbedingungen	28
3.2	Projektphasen	29
3.3	Umsetzung von Sprints	29
4.3	UC 1: Registrierung	37
4.4	UC 2: Benutzerprofil CRUD	38
4.5	UC 3: Anzeigen und Filtern von Treffen	39
4.6	UC 4: Einschreiben von Treffen	40
4.7	UC 5: Anzeigen eigener Treffen	41
4.8	UC 6: Erstellen von Treffen	41
4.9	UC 7: Durchführung von Treffen	43
4.10	UC 8: Kategorisierung von Treffen	44
4.11	UC 9: Regeln für Treffen anpassen	44
4.12	UC 10: Absagen von Treffen	45
4.13	UC 11: Teilnahme an Treffen	46
4.14	UC 12: Tag CRUD	47
4.15	Nicht-funktionale Anforderungen	49
6.1	Technologien	57
A.1	Beurteilte Anforderungen bei der Recherche von Videokonferenz-Plattformen	91
A.2	Beurteilung Big Blue Button	91
A.3	Beurteilung Jitsi	92
A.4	Beurteilung Zoom	93
A.5	Beurteilung Google Meet	93

Anhang A

Anhang

A.1 Aufgabenstellung



Aufgabenstellung Bachelor Thesis „Video-Konferenzen für einander Unbekannte“

Aufgabenstellung Bachelor Thesis „Video-Konferenzen für einander Unbekannte“

**Autor: Frank Koch
Version: 2.0**

**Erstellt am: 22. Februar 2021
Letzte Änderung am: 19. Mai 2021**

1. Personen

Diplomanden: Flavio Böni und Severin Hauser

Betreuer: Prof. Frank Koch, Dozent für Wirtschaftsinformatik

2. Ausgangslage, Problembeschreibung

Im Kontext der aktuellen Pandemie werden persönliche Kontakte aufgrund von Home-Office und Fernstudium reduziert. Anstatt vor Ort verlagern sich Treffen deshalb auf Video-Konferenz-Plattformen wie Zoom.

Während diese Plattformen den Austausch mit bereits bestehenden Kontakten erleichtern, ist das Kennenlernen neuer Menschen nur noch begrenzt möglich. Die Problematik zeigt sich am Beispiel von Studierenden im ersten Semester: Sie kennen Ihre Mitsstudenten noch nicht und die von der Hochschule eingesetzten Kommunikationsmittel richten sich nur auf den Unterricht aus. Veranstaltungen der Studentenvereine fallen aus, oder werden nur über E-Mails angekündigt. Es entfallen also viele Möglichkeiten, Mitsstudenten mit gemeinsamen Interessen in einem informellen Rahmen kennen zu lernen.

3. Aufgabenstellung

Ausgehend von der Problemstellung soll eine Web-Plattform für Video-Treffen erstellt werden. Die Plattform soll sich einander unbekanntenen Personen eine Möglichkeit zum Kennenlernen und Austausch über gemeinsame Interessen bieten. Angemeldete Benutzer können Treffen zu einem Thema an einem spezifischen Zeitpunkt ausschreiben. Durch eine Kategorisierung der Treffen wird die Suche nach Gleichgesinnten vereinfacht. Benutzer können sich für geplante Treffen einschreiben, oder bereits laufenden Treffen beitreten. Ein Online-Konferenzraum wird auf den geplanten Zeitpunkt eines Treffens erstellt und der Zugang über die Plattform ermöglicht. Die Video-Session wird durch einen Organisator geführt, welcher die Regeln für das Treffen bestimmt und die Teilnehmer diesbezüglich kontrollieren kann.

Die Plattform soll frei verfügbar und selbst betreibbar sein. Sie soll deshalb möglichst modular implementiert werden, sodass sie auf die Bedürfnisse des jeweiligen Betreibers angepasst werden kann. Beispiele für diese Modularität sind etwa die Wahl der Video-Konferenz-Plattform, oder des Anmelde-Mechanismus.

Funktionale Anforderungen

Erstellen von Treffen

Benutzer der Applikation können geplante Treffen erstellen. Der Ersteller des Treffens wird automatisch dessen Organisator. Organisatoren können Treffen bis zu deren Durchführung bearbeiten, oder absagen. Bei der Erstellung werden Attribute erfasst:

- Titel und Beschreibung des Treffens
- Datum und Uhrzeit des Treffens
- Maximale Anzahl Teilnehmer
- Kategorisierung des Treffens

Kategorisierung von Treffen

Die Kategorisierung von Treffen wird durch Auswahl von Tags gemacht. Die verfügbaren Tags können durch die Applikation vordefiniert und durch Benutzer ergänzt werden (vergleichbar mit Hashtags auf Twitter).

Suchen von Treffen

Alle laufenden und geplanten Treffen werden den Benutzern in einer Feed-Ansicht präsentiert. Benutzer können diese nach Datum und Tags filtern und durchsuchen.

Teilnahme an Treffen

Benutzer können sich für ein geplantes Treffen einschreiben.

Durchführen von Treffen

Zum Zeitpunkt des Treffens kann der Organisator das Treffen starten. Dabei wird von der Applikation eine Konferenz auf dem Video-Konferenz-Provider erstellt. Die Teilnehmer können dann über den generierten Zugang der Konferenz beitreten. Ist das Treffen abgeschlossen wird es durch den Organisator in der Applikation beendet.

Benutzerprofil

Benutzer der Applikation können ihr Benutzerprofil mit ihrem Namen und Biografie versehen. Diese Daten sind von anderen Benutzern aufrufbar. Benutzer sehen in ihren eigenen Profilen ihre geplanten und vergangenen Treffen.

Optional

Die Applikation kann um eine Vielzahl an Funktionen erweitert werden. Soweit genügend Zeit vorhanden ist, können zum Beispiel eine Auswahl der folgenden Features umgesetzt werden:

- *Bewertung von Treffen*
 - *Teilnehmer eines Treffens können den Organisator anhand eines vom Betreiber konfigurierbaren Bewertungskriterienkataloges bewerten.*
 - *Themengebiete*
Vordefinierte Kategorien als Ergänzung zu Tags
 - *Erstellung von persistenten Treffen*
Treffen die immer aktiv und ohne Moderation sind, z.B. im Sinne einer Kaffee-Ecke
 - *Freunde*
Benutzer können anderen Benutzern folgen, oder diese zu ihren Freunden hinzufügen.
 - *Matching von Benutzern mit ähnlichen Interessen*
 - *Benachrichtigungen*
Implementation von E-Mail-Benachrichtigungen über demnächst startende Treffen, neue Treffen mit gewissen Tags, oder anderer Benutzern welchen der Benutzer folgt.
 - *Einladungs-Funktion zu Treffen*
 - *Gastzugänge*
 - *Erweiterte Konfiguration von Treffen*
Organisatoren die Möglichkeit geben, Teilnehmer zu bestätigen oder abzulehnen, Treffen auf privat zu setzen und nur über Einladungen zugänglich machen.
 - *Kommentarfunktion*
Teilnehmern die Möglichkeit geben in einer Kommentarspalte Fragen zum Treffen zu stellen
 - *Gruppen*
Die Möglichkeit Gruppen zu erstellen, unter welchen Treffen organisiert werden (z.B. ein Studentenverein hat einen Gruppen-Account). Treffen können nur Gruppenmitgliedern zugänglich gemacht werden.
 - *Private Treffen*
Benutzer können direkt ein Treffen mit einem anderen Benutzer durchführen.
 - *Moderation der Applikation*
Moderations-Mittel zur Durchsetzung der Nutzungsregeln einführen. Dies kann z.B. Funktionen für Warnungen sowie temporäre und permanente Bans beinhalten.
 - *Administrations-Panel*
Interface für Administratoren zur Verwaltung, Moderation und Konfiguration der Applikation.
-

Nicht-Funktionale Anforderungen

Functionality

- Video-Konferenzen werden durch Anbindung an eine externe Konferenz-Plattform wie Zoom oder Jitsi durchgeführt

Usability

- Die Web-Applikation soll auf Desktop- und Mobilgeräten verwendbar sein
- Die Applikation soll übersetzbar sein

Supportability

- Der Anmeldemechanismus und die verwendete Video-Konferenz-Lösung sollen austauschbar sein
- Die Applikation soll als Open Source entwickelt werden
- Eine generische Herangehensweise soll die Applikation für den Betreiber individuell konfigurierbar bzw. erweiterbar machen (Customizing)

4. Zur Durchführung

Mit dem Betreuer finden Besprechungen gemäss Absprache statt. Die Besprechungen sind von den Diplomanden mit einer Traktandenliste vorzubereiten und die Ergebnisse in einem Protokoll zu dokumentieren, das dem Betreuer per E-Mail zugestellt wird.

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsergebnisse in vorläufigen Versionen abzugeben.

5. Dokumentation und Abgabe

Die Dokumentation zur Projektplanung und –Verfolgung ist gemäss den Richtlinien der Abteilung Informatik anzufertigen. Die Detailanforderungen an die Dokumentation der Recherche- und Entwicklungsergebnisse werden entsprechend dem konkreten Arbeitsplan festgelegt.

Die Dokumentation ist vollständig digital sowie ausgedruckt in drei Exemplaren abzugeben.

Neben der Dokumentation sind abzugeben:

- ein Poster zur Präsentation der Arbeit
- alle zum Nachvollziehen der Arbeit notwendigen Ergebnisse und Daten (Quellcode, Buildskripte, Testcode, Testdaten usw.)
- Material für eine Abschlusspräsentation (ca. 30')

6. Termine

Es gelten folgende Termine:

- **22.02.21** Beginn der Bachelorarbeit
 - Bis **09.06.21** Erfassung des Abstracts im Online-Tool <https://abstract.hsr.ch/> Die Studierenden geben den Abstract für die Diplomarbeitsbroschüre zur Kontrolle an ihren Betreuer/Examinator frei.
 - Bis **15.06.21** Der Betreuer/Examinator gibt das Dokument mit dem korrekten und vollständigen Abstract der Broschüre zur Weiterverarbeitung an das Studiengangsekretariat frei.
 - **18.06.21 12:00** Abgabe des Berichts an den Betreuer
 - **18.06.21** Hochladen aller Dokumente auf archiv-i.hsr
 - Bis **16.08. – 10.09.2021** Mündliche Prüfung
-

7. Beurteilung

Eine erfolgreiche Bachelorarbeit erhält 12 ECTS-Punkte (1 ECTS Punkt entspricht einer Arbeitsleistung von ca. 25 bis 30 Stunden). Die Bewertung erfolgt gemäss nachstehender Kriterien.

Gesichtspunkt	Gewicht
1. Organisation, Durchführung (Projektplanung u. Nachführung Arbeit gemäss Projektplan, Selbstständigkeit, Einsatz, Zusammenarbeit mit Auftraggeber, Betreuer)	1/6
2. Bericht (Inhalt des Projektschlussberichts, Gliederung, Darstellung, Sprache der gesamten Dokumentation)	1/6
3. Inhalt	
3.1 Problemanalyse (Vorstudie, Literaturstudium, Anforderungsspezifikation, Anforderungsanalyse, Domainanalyse)	1/6
3.2 Lösungsentwurf (Lösungsvarianten und deren Beurteilung, Variantenentscheid, Konzept, Entwurf)	1/6
3.3 Realisierung und Test	1/6
4. Präsentation und Mündliche Prüfung zur Bachelorarbeit	1/6

Im Übrigen gelten die Abläufe und Regelungen Studien□ und Bachelorarbeiten im Studiengang Informatik.

Rapperswil, den 22.02.21

Frank Koch

A.2 Installationsanleitung und Benutzerhandbuch

Da die Applikation öffentlich verfügbar ist, wurden die Installationsanleitung und das Benutzerhandbuch als Dokumentations-Website erstellt und über GitHub Pages¹ gehostet.

Handbuch

<https://huddle-up.github.io/handbook>

Benutzerhandbuch (Deutsch)

<https://huddle-up.github.io/handbook/de/docs/use/intro>

Benutzerhandbuch (Englisch)

<https://huddle-up.github.io/handbook/docs/use/intro>

Installationsanleitung

<https://huddle-up.github.io/handbook/docs/host/intro/>

Quellcode

<https://github.com/huddle-up/handbook>

A.3 Evaluation Videokonferenz-Provider

A.3.1 Einführung

Die Durchführung von Videokonferenzen ist eine Kernfunktionalität von *HuddleUp*. Gemäss der Aufgabenstellung (Abschnitt A.1) und den daraus abgeleiteten nicht-funktionalen Anforderungen (Abschnitt 4.2) wird diese Funktionalität nicht durch die Applikation, sondern durch einen externen Videokonferenz-Provider umgesetzt.

Um die Interaktionsmöglichkeiten mit solchen Providern zu erfassen wurden in einem ersten Schritt vorhandene Plattformen recherchiert. In einem zweiten Schritt wurde eine der Plattformen zur Umsetzung gewählt. In einem dritten Schritt wurde die gewählte Plattform im Detail evaluiert.

A.3.2 Vorhandene Videokonferenz-Plattformen

Beurteilung

Bei der Recherche wurde auf mehrere Anforderungen geachtet. Dabei wurden die Anforderungen in zwingend benötigt (*must*), wünschenswert (*should*) und optional (*could*) unterteilt. Weitere Anforderungen wurden erfasst, aber nicht bewertet (*n.b.*).

Name	Beschreibung	Bewertung
API vorhanden	Die Möglichkeit über eine Schnittstelle mit der Plattform zu interagieren. Zu bevorzugen wäre eine HTTP-Schnittstelle.	must
Konferenzen erstellen	Konferenzen sollen über die Schnittstelle erstellbar sein.	must

¹<https://pages.github.com>

Konferenzen konfigurieren	Die Einstellungen einer Konferenz sollen konfigurierbar sein. Beispiele dafür: Anzahl Teilnehmer, Moderatoren-Rechte, Titel der Konferenz.	should
Konferenzen schützen	Es soll möglich sein, den Zugang zu einer Konferenz einzuschränken.	should
Metainformationen	Die Möglichkeit Informationen zu einer laufenden Konferenz abfragen zu können. Beispiele dafür: Anzahl Teilnehmer, Dauer der Konferenz, Teilnehmer in der Konferenz.	could
Open-Source	Verfügbarkeit des Quellcodes.	could
Eigenes Hosting	Die Plattform kann auf eigener Infrastruktur installiert werden.	could
Erweiterbarkeit	Die Plattform kann durch den Betreiber / Anwender durch zusätzliche Funktionen erweitert werden.	could
Kosten	Kosten welche für die Verwendung der Plattform entstehen	n.b.

Tabelle A.1: Beurteilte Anforderungen bei der Recherche von Videokonferenz-Plattformen

A.3.3 Resultate

Big Blue Button

Homepage <https://bigbluebutton.org/>

Dokumentation <https://docs.bigbluebutton.org/>

Bemerkungen Starker Fokus auf Einsatz an Schulen.

Anforderung	Vorhanden
API vorhanden	✓ Ausführliche HTTP-API vorhanden.
Konferenzen erstellen	✓
Konferenzen konfigurieren	✓
Konferenzen schützen	✓
Metainformationen	✓
Open-Source	✓ https://github.com/bigbluebutton/bigbluebutton
Erweiterbarkeit	✓
Eigenes Hosting	✓
Kosten	Self-Hosted: Abhängig von Hosting und Nutzlast; Commercial-Hosting: 50 - 150\$+ pro für 50 Benutzer

Tabelle A.2: Beurteilung Big Blue Button

Jitsi

Homepage <https://jitsi.org/>

Dokumentation <https://jitsi.github.io/handbook/docs/intro>

Bemerkungen *Jitsi* stellt eine öffentliche Instanz² mit Einschränkungen zur Verfügung. Wenn selber gehosted sind alle Features verfügbar. Zusätzlich gibt es das kommerzielle *Jitsi as a Service*³ von den Jitsi-Entwicklern, welches zusätzliche Erweiterungen offeriert.

Anforderung	Vorhanden (kommerziell / self-hosted / öffentliche Instanz)
API vorhanden	✓/ ✓/ ? Eine self-hosted Instanz bietet nicht eine API an, kann aber durch Module ergänzt werden.
Konferenzen erstellen	✓/ ✓/ ✓
Konferenzen konfigurieren	✓/ ✓/ ✗
Konferenzen schützen	✓/ ✓/ ✗
Metainformationen	✓/ ✓/ ✗
Open-Source	✗/ ✓ https://github.com/jitsi
Erweiterbarkeit	✓✓/ ✗
Eigenes Hosting	✓
Kosten	Öffentliche Instanz: gratis; Self-Hosted: Abhängig von Hosting und Nutzlast; Commercial-Hosting: bis 25 Teilnehmer gratis, bis 300 Teilnehmer 99\$ pro Monat

Tabelle A.3: Beurteilung Jitsi

Zoom

Homepage <https://zoom.us/>

Dokumentation <https://marketplace.zoom.us/docs/api-reference/zoom-api/>

Bemerkungen Interaktion mit API benötigt Berechtigung von Benutzer. Bei Lizenzen wird zwischen Organisationen und Benutzern unterschieden. Pro Konferenz braucht es nur eine Lizenz um erweiterte Funktionen zu ermöglichen. Eine Lizenz muss nicht direkt an einen Benutzer gebunden sein.

Anforderung	Vorhanden
API vorhanden	✓ Ausführliche HTTP-API vorhanden.
Konferenzen erstellen	✓
Konferenzen konfigurieren	✓
Konferenzen schützen	✓
Metainformationen	✓
Open-Source	✗
Erweiterbarkeit	✓
Eigenes Hosting	✗

²<https://meet.jit.si>

³<https://jaas.8x8.vc/>

Kosten	Gratis (Gruppen-Meetings auf 40 Minuten beschränkt), sonst mind. 149\$ pro Lizenz pro Jahr
--------	---

Tabelle A.4: Beurteilung Zoom

Google Meet

Homepage <https://meet.google.com/>

Dokumentation -

Bemerkungen Eine Lizenz ist direkt an den Benutzer gebunden.

Anforderung	Vorhanden
API vorhanden	✘ Keine offizielle API, gewisse Möglichkeiten durch die Google Calendar API.
Konferenzen erstellen	?
Konferenzen konfigurieren	?
Konferenzen schützen	?
Metainformationen	✘
Open-Source	✘
Erweiterbarkeit	✘
Eigenes Hosting	✘
Kosten	Gratis (Gruppen-Meetings auf 1 Stunde beschränkt), sonst mind. 8\$ pro aktiver User pro Monat

Tabelle A.5: Beurteilung Google Meet

Microsoft Teams

Microsoft Teams wurde nicht bewertet, da dessen Fokus auf die Einbindung in eine Organisation liegt.

A.3.4 Wahl des Providers

Aufgrund der Recherche und der Bewertung wurde beschlossen die Erstimplementation mit *Jitsi* umzusetzen. Für die Wahl sprechen mehrere Gründe:

Flexibilität bei der Entwicklung Für die Erstellung einer Konferenz auf *Jitsi* benötigt es im einfachsten Fall nur eine generierte URL. Dies bedeutet, dass wir die Anbindung inkremental machen können und nicht bereits zu Beginn die gesamte Schnittstelle umsetzen müssen.

Flexibilität für Betreiber Durch die öffentliche Instanz, das eigene Hosting sowie das *Jitsi as a Service*-Angebot bietet *Jitsi* dem Betreiber mehrere Möglichkeiten dieses seinen Bedürfnissen entsprechend einzusetzen. Gerade die öffentliche Instanz ist attraktiv, da der Betreiber zwar auf viele Funktionen verzichten, dafür nur die Kosten für die *HuddleUp* Instanz übernehmen muss.

Kosten Die Kosten für die Videokonferenz-Plattform sind unserer Meinung nach ei-

ner der Hauptkriterien, ob ein Betreiber sich für *HuddleUp* entscheidet. Die frei verfügbare öffentliche Instanz, sowie die Kosten für eigenes Hosting, oder *Jitsi as a Service* sind im Vergleich zu den anderen Anbietern am attraktivsten.

Erweiterbarkeit Ein Grossteil der Funktionalität von Jitsi kann konfiguriert, oder durch Erweiterungen angepasst werden.

A.3.5 Evaluation Jitsi

In diesem Abschnitt werden die Möglichkeiten ausgeführt, wie mit *Jitsi* die in der Bewertung verwendeten Anforderungen umgesetzt werden können. Es wird dabei zwischen der öffentlichen *Jitsi* Instanz⁴ sowie einer selbst gehosteten Instanz unterschieden. Das *Jitsi as a Service* wird nicht im Detail behandelt, da dieses sich in vielen Bereichen unterscheidet.

Jitsi Übersicht

Um die anschliessenden Ausführungen zur Umsetzung zu ergänzen, wird hier zuerst ein kurzer Überblick über die Architektur und wichtige Abläufe von *Jitsi* gegeben.

Architektur *Jitsi* ist eine Sammlung mehrerer eigenständiger Komponenten. Während wir für eine genaue Beschreibung auf die Dokumentation⁵ verweisen, wollen wir die wichtigsten hier kurz erwähnen:

- **Jitsi Meet** Obwohl der Name *Jitsi Meet* oft für die gesamte Applikation verwendet wird, handelt es sich dabei nur um den Browser-Client. Die JavaScript-Applikation dient als Frontend für die Serverseitige *Jitsi Videobridge*.
- **Jitsi Videobridge** Die *Jitsi Videobridge* ist das Herzstück von *Jitsi*. Alle Audio- und Video-Streams einer Konferenz laufen über die *Videobridge*.
- **Jicofo** *Jicofo* steht für *Jitsi Conference Focus* und ist für die Erstellung und Verwaltung von Konferenzen zuständig.
- **Prosody** *Prosody*⁶ wird von Jitsi zur Kommunikation zwischen den einzelnen Komponenten eingesetzt. Die Funktionalität von *Prosody* kann durch Module / Plugins erweitert werden, wobei *Jitsi* bereits mehrere zur Verfügung stellt⁷. *Prosody* wird nicht von *Jitsi* selbst entwickelt.

Als Konsequenz aus dieser Aufteilung findet die Konfiguration einer *Jitsi* Instanz über die verschiedenen Komponenten verteilt statt.

Konzepte *Jitsi* versucht die Erstellung und Durchführungen von Konferenzen so einfach wie möglich zu halten. Dies zeigt sich an den folgenden Beispielen:

- **Registration** *Jitsi* hat keine Benutzer. Es benötigt folglich keinen Account um eine Konferenz zu erstellen, oder dieser beizutreten. Es bietet jedoch die Möglichkeit, Benutzer zu authentifizieren und so den Zugriff auf die Instanz einzuschränken.

⁴<https://meet.jit.si>

⁵<https://jitsi.github.io/handbook/docs/architecture>

⁶<https://prosody.im/>

⁷<https://github.com/jitsi/jitsi-meet/tree/master/resources/prosody-plugins>

- **Erstellen von Konferenzen** Eine Konferenz wird automatisch erstellt, wenn deren URL aufgerufen wird. Wird zum Beispiel `https://meet.domain.tld/HuddleUp` aufgerufen, wird eine Konferenz mit dem Titel «HuddleUp» erstellt. Besteht bereits eine Konferenz unter der URL, so tritt man dieser bei.

A.3.6 Umsetzung von Anforderungen

API

Jitsi bietet keine öffentliche HTTP-API an. Dies kann jedoch ergänzt werden, indem *Prosody* durch entsprechende Plugins erweitert wird. Während dies zwar nicht einer ausgefeilten API gleichkommt, kann so trotzdem eine Vielzahl an Funktionen zur Verfügung gestellt werden. Da die öffentliche *Jitsi*-Instanz diese Komponenten nicht öffentlich zugänglich macht, ist dies dort nicht möglich.

Öffentliche Instanz	Self-Hosted
Nicht möglich	Verwenden von Prosody-Plugins

Konferenzen erstellen

Konferenzen können direkt über die URL generiert werden. Dabei besteht die Gefahr, dass unter einer URL bereits eine bestehende Konferenz vorhanden ist. Dies kann reduziert werden indem ein zusätzlicher Präfix verwendet wird (`https://meet.domain.tld/PREFIX/HuddleUp`), oder die URL einen zufällig generierten Teil beinhaltet.

Öffentliche Instanz	Self-Hosted
Möglich	Möglich

Konferenzen konfigurieren

Da *Jitsi* keine dedizierte API zur Verfügung stellt, gibt es keine Möglichkeiten, Konferenzen explizit bei der Erstellung zu konfigurieren. Für gewisse Aspekte gibt es jedoch Lösungen:

User Interface Das dem Benutzer angezeigte User-Interface kann durch Parameter in der Konferenz-URL⁸, oder global in der *Jitsi*-Konfiguration angepasst werden.

Öffentliche Instanz	Self-Hosted
Konfiguration nur über Config-String in der Konferenz-URL möglich.	Möglich

Moderatoren-Rechte *Jitsi* kann so konfiguriert werden, dass der erste Benutzer der die Konferenz betritt zum Moderator wird. Dass der Organisations eines Treffens auch dessen Moderator wird, kann erreicht werden indem sicher gestellt wird, dass dieser die Konferenz als erstes betritt.

⁸<https://jitsi.github.io/handbook/docs/user-guide/user-guide-advanced>

Öffentliche Instanz	Self-Hosted
Nicht möglich, da Moderatoren-Funktion nicht aktiviert	Möglich

Konferenzen schützen

Um den Zugang zu einer Konferenz zu schützen gibt es drei Möglichkeiten:

1. Prosody-interne Authentifizierung
2. Verwendung von **JWT**
3. Warteraum aktiviert

Prosody-interne Authentifizierung Der Zugriff auf *Jitsi* kann für Benutzer eingeschränkt werden⁹. Dafür gibt es zwei Varianten. Die erste Variante ist, die interne Authentifizierung von *Prosody* zu verwenden. Dazu wird ein Benutzer mit Benutzername und Passwort bei Prosody registriert. Anschliessend können Konferenzen nur noch erstellt werden, wenn der erste Benutzer (und somit der Moderator) sich mit diesen Angaben authentifiziert. Weitere Teilnehmer können ohne Anmeldung beitreten. Diese Lösung hat jedoch einige Nachteile:

- Die Erstellung von Benutzern ist nur über eine **CLI** möglich. Läuft die *Jitsi*-Instanz also nicht auf dem selben Server wie *HuddleUp* ist dies erschwert möglich.
- Die Zugänge sind nicht auf Konferenzen einschränkbar. Kennt ein Benutzer bereits einen Zugang, kann er jeder Konferenz beitreten, oder neue erstellen.
- Es entsteht ein Zusatzaufwand bei der Synchronisierung der Benutzer von *HuddleUp* und *Prosody*.

Öffentliche Instanz	Self-Hosted
Nicht möglich	Möglich

Verwendung von JWT Eine weitere Variante um den Zugriff auf Konferenzen zu schützen ist durch die Verwendung von **JWT**.¹⁰ Diese können in der URL zur Konferenz mitgesendet werden und bieten einige Vorteile:

- Die Token können auf einzelne Konferenzen eingeschränkt werden.
- In den Token können Benutzerinformationen mitgesendet werden, so haben Benutzer in der Konferenz automatisch den selben Anzeigenamen wie in *HuddleUp*
- Die Token können zeitlich limitiert werden, es muss sich also nicht um deren Invalidierung gekümmert werden.

Ein Nachteil dieser Lösung ist, dass alle Benutzer einen Token benötigen. Dies bedeutet, dass zum Beispiel Gastzugänge über *HuddleUp* erstellt werden müssten.

⁹<https://jitsi.github.io/handbook/docs/devops-guide/secure-domain>

¹⁰<https://jitsi.github.io/handbook/docs/devops-guide/secure-domain>

Öffentliche Instanz	Self-Hosted
Nicht möglich	Möglich

Warterraum aktiviert Der Moderator einer Konferenz kann eine Warterraum-Funktion aktivieren. Jeder neue Benutzer muss dann von einem Moderator bestätigt werden um der Konferenz beitreten zu können. Auf der öffentlichen Instanz ist diese Option zwar vorhanden, jeder Teilnehmer in der Konferenz kann jedoch Benutzer bestätigen, oder die Funktion wieder deaktivieren.

Öffentliche Instanz	Self-Hosted
Möglich (mit Einschränkungen)	Möglich

Metainformationen

Metainformationen können über die Private HTTP-API der *Jitsi Videobridge*¹¹ sowie über die Einbindung von *Prosody*-Plugins abgerufen werden.

Öffentliche Instanz	Self-Hosted
Nicht möglich	Möglich

A.4 Systemtests

A.4.1 Einleitung

Die Systemtests sind sogenannte *Black Box* Tests. Sie werden auf der Grundlage der Produktspezifikation (Leistungsanforderungen, Erfolgsgarantie der Use Cases usw.) ohne Zugriff auf den Quellcode definiert. Wir haben die Systemtests am Ende der *Construction*-Phase als Benutzer manuell durchgeführt. Im Gegensatz zu den *White Box* Tests, welche auf der internen Struktur (Architektur, Code) basieren und automatisch durchgeführt werden.^[7]

A.4.2 Testfall 1: Registrierung

Testbeschreibung Es soll möglich sein sich über die Webplattform zu registrieren. Für die Authentifikation wird der Benutzer an den vorkonfigurierten Identity Provider weitergeleitet. Nach der Registrierung kann eine erneute Anmeldung automatisch über ein *Single-sign on* des Identity Provider geschehen.

Testziele

1. Registrierung eines neuen Benutzers mit E-Mail und Passwort über Weiterleitung an den Identity Provider
2. Anmelden mit einem bestehenden Login über den Identity Provider
3. *Single sign-on* während der Gültigkeitsdauer des Tokens

¹¹<https://github.com/jitsi/jitsi-videobridge/blob/master/doc/rest.md>

4. Für die Registrierung wird an den vorkonfigurierten Identity Provider weitergeleitet

A.4.3 Testfall 2: Benutzerprofil bearbeiten

Testbeschreibung Es soll möglich sein nach der Anmeldung das Benutzerprofil zu bearbeiten und das öffentliche Benutzerprofil des eigenen und anderer Benutzer zu sehen. Das Benutzerprofil soll mit allen damit bezogenen Daten gelöscht werden.

Testziele

1. Das eigene Benutzerprofil kann geladen und bearbeitet werden
2. Das öffentliche Benutzerprofil ist von einem anderen Benutzer aus als Teilnehmer oder Organisator eines Treffens ersichtlich
3. Das Benutzerprofil löschen

A.4.4 Testfall 3: Anzeigen und Filtern von Treffens

Testbeschreibung Es sollen alle laufenden und geplanten Treffens über die Suche gefunden werden können. Dazu wird in der Suche nach einem Titel, Beschreibung, Tag, Von- und Bisdatum gesucht. Die gleiche Suche nach allen eigenen Treffens soll über die Ansicht Meine Treffens durchgeführt werden.

Testziele

1. Geplante, laufende und eigene Treffens werden in vorsortierten und nachladbaren (paged) Ansichten unterteilt dargestellt
2. Es werden nur Treffer angezeigt, welche den Filterkriterien (alle Kombinationen) entsprechen angezeigt
3. Pro Treffens ist ersichtlich ob man Teilnehmer oder Organisator ist

A.4.5 Testfall 4: Erstellen von Treffens

Testbeschreibung Es soll ein eigenes Treffens mit allen Attributen erstellt werden. Das Treffens soll bearbeitet und abgesagt werden.

Testziele

1. Als angemeldeter Benutzer ein neues Treffens erstellen
2. Das Treffens mittels Titel, Beschreibung, Tags, Von-, Bisdatum und maximale Teilnehmer erfassen
3. Das Treffens speichern und erneut bearbeiten
4. Das Treffens absagen

A.4.6 Testfall 5: Tags erfassen und auswählen

Testbeschreibung Es sollen vom Betreiber vordefinierte Tags zur Verfügung stehen. Es sollen zudem eigene Tags erfasst werden können. Die eigenen und die vom Betreiber erfassten Tags sind für alle Benutzer verwendbar.

Testziele

1. Tags als Betreiber vorkonfigurieren und die Webplattform starten
2. Als Benutzer eigene Tags erfassen und die vorkonfigurierten Tags auswählen
3. Nach dem Speichern des Treffens nach diesen Tags Suchen

A.4.7 Testfall 6: Teilnahme an Treffen

Testbeschreibung Es soll an einem geplanten Treffen teilgenommen werden können. Das Treffen hat bereits andere Teilnehmer und es wird gleich durch den Organisator gestartet.

Testziele

1. Nach dem geplanten Treffen suchen
2. Die bereits registrierten Teilnehmer sind im Treffen ersichtlich
3. An dem Treffen teilnehmen
4. Das Treffen mit einem anderen Benutzer (Organisator) freigeben
5. Der Videokonferenz nach der Freigabe des Organisators beitreten

A.4.8 Testfall 7: Durchführung von Treffen

Testbeschreibung Es soll ein geplantes Treffen durchgeführt werden. Das Treffen ist kurz vor dem Startzeitpunkt.

Testziele

1. Als Organisator die Videokonferenz über das Treffen starten
2. Der Videokonferenz beitreten und minimal einrichten
3. Die Videokonferenz für alle Teilnehmer freigeben
4. Mit einem anderem Benutzer der Videokonferenz beitreten
5. Die Videokonferenz über die Applikation beenden

A.5 Systemtest Protokolle

Test Case	Tester	Datum	Erfolg	Kommentar
Testfall 1	Flavio Böni	12.6.20201	Ja	Weiterleitung und Anmeldung funktioniert
Testfall 2	Flavio Böni	12.6.20201	Ja	Erstellung, Bearbeitung und Löschung ok
Testfall 3	Flavio Böni	12.6.20201	Ja	Ansichten und Filter werden berücksichtigt
Testfall 4	Flavio Böni	12.6.20201	Ja	Funktioniert einwandfrei
Testfall 5	Flavio Böni	12.6.20201	Ja	Wird berücksichtigt
Testfall 6	Flavio Böni	12.6.20201	Ja	Teilnahme und beitreten funktioniert
Testfall 7	Flavio Böni	12.6.20201	Ja	Durchführung und Prozess funktioniert

A.6 Kommentare zu den Tests

- Nach dem Abmelden erfolgt eine erneute Anmeldung automatisch. Für den Benutzer kann das ohne Hinweis etwas verwirrend sein.
- Das Abmelden nach der Tokengültigkeit geschieht nicht automatisch. Der Zustand wird abgefangen und durch einen Fehler dargestellt.
- Durch das Löschen des Benutzerprofils werden alle enthalten Daten des Benutzers gelöscht.
- Die Freiheiten bei der Erstellung von Tags kann schnell zu einer grossen Sammlung von Tags führen.
- Das Treffen muss über die Webplattform durch den Organisator beendet werden. Dieser Prozessschritt ist für den Benutzer umständlich.
- Der Zugang zur Videokonferenz ist nach dem geplanten Ende nicht mehr möglich und das Treffen wird nicht mehr als Live dargestellt.

A.7 Usability Tests

A.7.1 Einleitung

Die Grundlagen und das Vorgehen zur Durchführung eines Usability-Tests, sind uns aus der Vorlesung des gleichzeitig besuchten Moduls Human Computer Interaction Design (HCD) der OST, unterrichtet von Dr. Sibylle Peuker, bekannt.[5]

Ein Usability-Test ist eine qualitative Methode zur Überprüfung der Gebrauchstauglichkeit eines Produktes. Potenzielle User führen typische Aufgaben aus und werden dabei beobachtet. Vorher und nachher gibt es eine strukturierte Befragung.

Das Ziel ist sicherzustellen, dass zukünftige User die Plattform benutzen können und ihre Ziele effektiv, effizient und zufriedenstellend erreichen können. Für uns Entwickler ist es eine Unterstützung bei Designentscheidungen insbesondere in Steitfällen. Auch Experten können sich täuschen und es hilft offene Fragen zu klären.

A.7.2 Vorgehen bei Usability-Tests

1. Testkonzept erstellen
2. Testpersonen rekrutieren

3. Testvorbereitungen
4. Prototyp testfähig machen
5. Durchführung der Tests
6. Auswertung der Testergebnisse
7. Erarbeiten von Verbesserungsvorschlägen / Anpassung des Konzepts

A.7.3 Testkonzept

Wissensziele/Fragestellungen

- Sind die einzelnen Funktionen auffindbar?
 - Registration, Login und Logout
 - Benutzerprofil bearbeiten
 - Treffen suchen nach Suchbegriffen, Tags oder Datum
 - Anzeigen weiterer Treffen
 - Registrierung eines Treffens
 - Teilnahme eines Treffens
 - Erstellung oder Bearbeitung eines Treffens
 - Durchführung eines Treffens
 - Unterscheidung Alle Treffen, Meine Treffen und ihren einzelnen Unterteilungen
- Ist die Tag-Selektion verständlich, insbesondere eigene die Erfassung von eigenen?
- Ist die von/bis Datum-Selektion verständlich?
- Sind die Prozesse zur Durchführung eines Treffens verständlich und funktionieren diese?
- Sind die Funktionen/Buttons verständlich beschriftet und gekennzeichnet?
- Kann die Plattform auch mit geringen Vorkenntnissen bedient werden?
- Sind Texte und Begriffe verständlich?
- Sind Fehlermeldungen verständlich?
- Findet man auch zu den Sonderfällen?

A.7.4 Testform und Inhalt

Remotetest

Idealer wäre ein Feldtest, aber durch die eingeschränkten Möglichkeiten zurzeit ist ein Remote Test mit der Testperson vorzuziehen. Der Testleiter und die Testperson sind dabei am eigenen PC zu Hause.

Inhalt

Die Testperson muss für die Durchführung den PC-Bildschirm in Microsoft Teams teilen und das Mikrofon aktiviert haben. Bei der Durchführung der Szenarien wird eine Aufzeichnung des Bildschirms mit Ton gemacht. Die Aufzeichnung wird vom Testleiter gestartet und hilft als Ergänzung bei der Protokollierung. Wichtig ist, dass die Testperson die Gedanken laut ausspricht. Die Aufzeichnung ist einfach mit der in Windows 10 integrierten Xbox Game Bar möglich.¹²

Vorinterview

- Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?
- Welche Konferenz-Provider kennst du, oder würdest du einsetzen?
- Hast du schon einmal an einem Video-Treffen teilgenommen?
- Über welche Plattform und Geräte würdest du aktuell ein Video-Treffen durchführen?
- Was wäre für dich wichtig um an einem spezifischen Video-Treffen teilzunehmen? (Bsp. Thema, Organisator, Mitglieder, Dauer, Regeln, etc.)

Nachinterview

- Wie ist dein allgemeiner Eindruck?
- Was hat dir gut gefallen?
- Was war nicht so gut?
- Wie nützlich beurteilst du die Plattform?
- Würdest du diese Plattform einsetzen oder verwenden?
- Was fehlt?

A.7.5 Szenarienaufbau

Die einzelnen Aufgaben für die Testperson werden als glaubwürdige Geschichten mit passender Formulierung beschrieben. Das Verhalten der Testperson soll antizipiert werden. Es wird eine Aufgabe pro Szenario gestellt und es sollte darauf geachtet werden, dass keine Wörter aus dem Interface im Szenario vorkommen.

A.7.6 Szenario 1

Vorbereitung

Da es sich hier um eine Test-Umgebung handelt, kannst du bei der ersten Aufgabe eine fiktive E-Mail-Adresse eingeben. Verwende dabei die Domain '@test.test'. Beispiel: 'hans.muster@test.test'.

¹²<https://www.heise.de/tipps-tricks/Bildschirm-aufnehmen-mit-Windows-10-4226449.html>

Beschreibung/Aufgabe

Du hast kürzlich von der Open Source Plattform *Huddle Up* gehört, welche von einem Freund gehostet wird und die du sogleich auf deinem PC über die Adresse <https://stage.huddle-up.org> im Browser (Firefox, Chrome, Edge) öffnest. Die Plattform hast du noch nie verwendet, weißt aber, dass mit unbekanntenen Personen in Video-Treffen über interessante Themen gesprochen werden kann. Als erstes möchtest du nun sehen ob aktuell ein Treffen geplant ist. Nachdem du eine Übersicht der Treffen gesehen hast, musst du leider den PC für jemand anders freigeben und willst nicht, dass diese Person die Treffen sieht.

Wissensziele

- Ist der Registration-, Login- und Logout-Prozess klar?

A.7.7 Szenario 2

Beschreibung/Aufgabe

Der PC ist wieder frei und du möchtest das statt deinem richtigen Namen ein Fantasienamen auf der Plattform verwendet wird. Zusätzlich willst du, dass andere etwas über dich erfahren können.

Wissensziele

- Ist der Prozess zum Benutzerprofil bearbeiten klar?

A.7.8 Szenario 3

Beschreibung/Aufgabe

Du siehst einige Treffen aber keines interessiert dich auf den ersten Blick. Demnächst sind Abstimmungen und dich interessiert das Thema Politik. Du willst deshalb schauen ob es ein Treffen dazu gibt.

Wissensziele

- Treffen suchen nach Suchbegriffen, Tag oder Datum intuitiv?
- Anzeigen weiterer Treffen intuitiv?
- Unterscheidung Entdecken und Meine Treffen sowie der einzelnen Unterteilungen?

A.7.9 Szenario 4

Beschreibung/Aufgabe

Du willst am Treffen mit dem Titel *I'm watching you* teilnehmen. Das Treffen läuft aktuell und du möchtest diesem beitreten. Begrüße den Moderator und frag ihn ob er dich hören und sehen kann. Du merkst, dass er dich nur testet und verlässt deshalb die Videokonferenz wieder.

Wissensziele

- Prozess Teilnahme eines Treffen klar?

A.7.10 Szenario 5

Beschreibung/Aufgabe

Du bist genervt über den schlechten Organisator und möchtest dein eigenes Treffen organisieren. Das Thema ist dir überlassen, es soll aber möglichst gut auffindbar sein. Verwende alle Optionen die dir zur Verfügung stehen. Die Dauer soll entsprechend dem Thema angepasst werden. Der Testleiter will deinem Treffen in 5 Minuten beitreten können.

Wissensziele

- Sind die Prozesse zur Durchführung eines Treffens verständlich und funktionieren diese?
- Ist die Tag-Selektion verständlich, insbesondere eigene Erfassen?
- Ist die von/bis Datum-Selektion verständlich, insbesondere bis Datum?
- Findet man auch zu den Sonderfällen?

A.7.11 Recruiting screener

Kriterien

- Grundsätzlich interessiert am Thema

Eine Testperson, die

- Erfahrung hat im Organisieren von Treffen

Eine Testperson, die

- Noch nie an einer Videokonferenz teilgenommen hat

3 Testperson, die

- Schon an Videokonferenzen teilgenommen haben

Demographie

- Alter 18+
- Geschlecht und Bildungsniveau möglichst ausgeglichen

A.8 Usability Test Protokolle

A.8.1 Testprotokoll Person 1

Max Freudiger mit Erfahrung im Organisieren von Treffen

- Alter: Vor Pensionierung
- Geschlecht: Männlich

- Beruf oder Kenntnisse: Viel Erfahrung als Organisator

Vorinterview

Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?

- Ja

Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?

- Ja

Welche Konferenz-Provider kennst du, oder würdest du einsetzen?

- Teams Jitsti, BBB ,Zoom ,Webex

Über welche Plattform und Geräte würdest du aktuell ein Video-Treffen durchführen?

- Notebook, Teams Jitsi von Handy

Was wäre für dich wichtig um an einem spezifischen Video-Treffen teilzunehmen?

- Reminder (Benachrichtigung z.B. per E-Mail), Thema, Organisator, Teilnehmer

Nachinterview

Wie ist dein allgemeiner Eindruck?

- Noch etwas unausgereift

Was hat dir gut gefallen?

- Wert Leute zu Treffen über Themen, Marktplatz

Was war nicht so gut?

- Noch einige unfertigen Funktionen

Wie nützlich beurteilst du die Plattform?

- Braucht Leute die es gut führen

Würdest du diese Plattform einsetzen oder verwenden?

- Je nach Situation ja

Was fehlt?

- Verwaltung von Tags

Protokoll Szenario 1

- Registration gefunden, ok
- Keine Registration gemacht da schon ein Login vorhanden
- Sortierung negativ aufgefallen, hat stark verwirrt
- Zuerst gedacht dass er das Fenster schliessen muss
- Danach Profil aufrufen und Logout klar, aber etwas verwirrt durch Aufgabenstellung

Protokoll Szenario 2

- Profil in der Navigation erneut angeklickt
- Aktualisierung des Profils kein Problem
- Überprüft Änderung oben rechts über den Namen

Protokoll Szenario 3

- Sieht nur die 5 Einträge und denkt es gibt den Eintrag gar nicht da die *Mehr anzeigen* Schaltfläche nicht auffällt

Protokoll Szenario 4

- Zusätzliches Problem mit Sortierung und Anzahl angezeigter Einträge
- Versucht nun aber Suche und findet das Meeting
- Zurück über den Browser verwendet
- Irritiert weil das aktuelle Treffen nicht sichtbar ist
- Sieht nun den Pager und merkt das es nur ein Auszug ist
- Wählt alle laden und sieht das laufende Treffen
- Verwendet den Zurückbutton aber durch die Suche wird keine History angelegt und er landet wieder im Profil
- Klickt auf Live und es geht nicht aber merkt das er auf den Titel klicken muss
- Merkt das es 2 Teilnehmer hat
- Ist verwirrt über den Teilnehmer Button, ist davon ausgegangen das es schon startet
- Sieht seine Registration
- Ist verwirrt wieso das zweimal *S* steht bei Teilnehmer, merkt nicht das er klicken kann
- Erwartet im Dropdown *teilnehmen* und nicht das *Teilnehmer* steht
- Probiert *nicht teilnehmen* aus
- Tritt bei, verwirrt durch gemeinsame *allow Micro und Video* Meldung in Jitsi
- Verwirrt das Fenster sich nicht schliesst, weiss aber das er es schliessen muss um zurück in die Webplattform zu kommen

Protokoll Szenario 5

- Sieht er ist Teilnehmer im vorherigen Treffen
- Neues Treffen Button sei etwas weit weg, aber ok
- Überrascht über die vielen vorhandenen Tags
- Tag Bearbeitung ok

- Datum, Zeit ok
- Prozess für die Durchführung gefällt ihm
- Schaut sich den Raum an, ok
- Wundert sich wieso er schon starten kann, gibt aber frei
- Das Meeting beendet geht etwas unter und wird nicht getestet

A.8.2 Testprotokoll Person 2

Lena Muster hat schon an Treffen teilgenommen

- Alter: 40
- Geschlecht: Weiblich
- Beruf oder Kenntnisse: Primarschullehrerin mit wenig technischen Kenntnissen

Vorinterview

Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?

- Ja

Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?

- Ja

Welche Konferenz-Provider kennst du, oder würdest du einsetzen?

- Jitsi Meet (nutze ich privat)
- Microsoft Teams (geschäftlich für Nutzung vorgegeben)
- Skype: kenne ich von früher
- Zoom: würde ich nicht einsetzen (Datenschutz)
- Webex: davon weiss ich, weil mein Partner dies geschäftlich nutzt

Über welche Plattform und Geräte würdest du aktuell ein Video-Treffen durchführen?

- Jitsi Meet (privat)
- Microsoft Teams (geschäftlich)

Was wäre für dich wichtig um an einem spezifischen Video-Treffen teilzunehmen?

- Thema muss für mich relevant sein
- Ich muss Aussicht auf einen Mehrwert haben, wenn ich daran teilnehme
- Der/die OrganisatorIn muss vertrauenswürdig sein
- Das Treffen darf nicht zu lange dauern (max. 30-45 min)
- Die Regeln sollten so ausgestaltet sein, dass alle zu Wort kommen / wer nicht spricht, schaltet sein Mikrofon auf stumm / bei kleinen Meetings (bis ca. 10 Personen) sollen alle mit Kamera sichtbar sein

Nachinterview

Wie ist dein allgemeiner Eindruck?

- Die Plattform ist im Inhalt, der Handhabung und dem Layout benutzerInnenfreundlich

Was hat dir gut gefallen?

- Dass ich über die Suche Meetings zu für mich interessanten Themen entdecken konnte
- Die sehr einfache Aufmachung/Gestaltung der Plattform

Was war nicht so gut?

- Dass die Meetings, nach welchen in gesucht habe, in der Trefferliste nicht zuoberst erschienen, sondern zusammen mit den bereits angezeigten Meetings

Wie nützlich beurteilst du die Plattform?

- Ich kann mir z. B. vorstellen, dass die Plattform nützlich sein könnte, wenn in einem grösseren Unternehmen die Diskussionskultur gefördert oder unterstützt werden soll. So können alle, die sich für ein Thema interessieren, in den entsprechenden Meetings mitdiskutieren, ohne am selben Ort sein zu müssen.

Würdest du diese Plattform einsetzen oder verwenden?

- Ich kann mir vorstellen, die Plattform zu verwenden, wenn ich mich zu einem Thema mit anderen austauschen oder darüber etwas lernen möchte. Dann kann ich auf der Plattform nach geeigneten Konferenzen suchen.

Was fehlt?

- Wenn ich ein Meeting entdecken möchte, fände ich es zum Stöbern hilfreich, wenn ich die Themen mittels Kategorien durchsuchen könnte.

Protokoll Szenario 1

- Externe Registration brauchte Hilfe
- Etwas verwirrt wie ich abmelde, Fenster schliessen oder PC ausschalten
- Abmelden gefunden
- Verwirrt wegen *D* als Icon des Profils

Protokoll Szenario 2

- Kannte das Single-Sign-On nicht und war überrascht das es automatisch wieder einloggte
- Klickt auf Mein Profil in der Navigation obwohl schon in Mein Profil
- Speichern geprüft über den Namen oben rechts im Profile

Protokoll Szenario 3

- Klar das dafür gesucht werden kann
- Abstimmung als Suchwort gewählt
- Politik als Tag ausgewählt
- Datum von und bis eingegeben aber nur das Datum ohne Uhrzeit
- Danach ohne von/bis Datum gesucht
- Genervt 'mühsam' das es nicht gleich geht und über die Auswahl das Datum und die Uhrzeit gewählt werden muss
- Suche hatte ein Fehler geworfen, musste eingreifen damit das Resultat ersichtlich wird

Protokoll Szenario 4

- Von sich aus die Seite aktualisiert und sah das Treffen danach
- Link ins Treffen nicht gefunden, zuerst auf das Öffentliche Profil geklickt und nicht realisiert das man über den Titel ins Treffen kommt. Als erstes auf Live geklickt um ins Treffen zu gelangen, da es rot ist.
- Teilnehmen und beitreten in die Videokonferenz unklar, dachte es sei schon gemacht mit teilnehmen. Erklärung notwendig das Jitsi extern läuft

Protokoll Szenario 5

- Es braucht ein Verständnis für den Benutzer/in das die Videokonferenz extern läuft und deshalb auch in der Webapplikation wieder geschlossen werden muss
- Unklar bei Tags Eingabe, dass nicht kommaseparierte Liste akzeptiert wird. Nicht gemerkt das Enter verwendet werden kann für einzelne Tags
- Verwirrt, da die 0 immer stehenbleibt aber akzeptiert das auch 09 funktioniert
- *Als erster* bemängelt, müsste zusätzlich auch *Als erste* stehen
- Dem Treffen beigetreten und versucht direkt in der Videokonferenz Personen einzuladen
- Verwirrt, dass sie wieder zurück in die Applikation gehen muss um die Konferenz für andere Freizugeben nachdem sie der Konferenz beigetreten war
- Auch an das Stoppen des Treffens hätte sie nicht gedacht. Es braucht ein Verständnis dafür, dass die Videokonferenz separat läuft. Eine Erwähnung vor dem Tests über diese Umstände hätte das Prozessverständnis laut ihrer Aussage einfacher gemacht.

A.8.3 Testprotokoll Person 3

Hans Meier hat schon an Treffen teilgenommen

- Alter: 35

- Geschlecht: Männlich
- Beruf oder Kenntnisse: Leiterfunktion in einem Grosskonzern

Vorinterview

Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?

- Ja

Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?

- Ja

Welche Konferenz-Provider kennst du, oder würdest du einsetzen?

- WebEx, Microsoft Teams

Über welche Plattform und Geräte würdest du aktuell ein Video-Treffen durchführen?

- WebEx

Was wäre für dich wichtig um an einem spezifischen Video-Treffen teilzunehmen?

- Thema, Organisator
- Interesse am Thema
- Einfaches Teilnehmen
- Erinnerung ein paar Minuten vor dem Termin

Nachinterview

Wie ist dein allgemeiner Eindruck?

- Nette Website. Optisch ansprechend

Was hat dir gut gefallen?

- Recht intuitiv

Was war nicht so gut?

- Funktionen waren z.T. nicht fertig bzw. funktionsfähig

Wie nützlich beurteilst du die Plattform?

- Es gibt sicherlich Möglichkeiten. Ich selbst benutze aber ausser im Geschäft keine Video-Calls

Würdest du diese Plattform einsetzen oder verwenden?

- Aktuell kein Bedarf

Protokoll Szenario 1

- Etwas verwirrt das SSO funktioniert und gemerkt das unser Logout nicht wirklich ausloggt

- Brauchte Hilfe bei der externen Registration da es etwas unklar beschrieben war, das registriert werden muss und nicht ein bestehendes Login erstellt wurde.
- Lange gesucht bis er das Logout gefunden hatte, da er die Navigation oben rechts erwartete

Protokoll Szenario 2

- Bemängelt das kein Feedback nach dem Speichern kommt
- Zurück funktioniert nicht wenn vorher auf mein Profil geklickt wurde

Protokoll Szenario 3

- Mit dem Tag Politik gesucht und das Treffen gefunden
- Unterscheidung der beiden Listen Meine Treffen und Entdecken bemängelt
- Er hat die Links oben Meine Treffen und Entdecken erst bei meiner Mitteilung gesehen
- Mehr Anzeigen und Suche intuitiv genutzt

Protokoll Szenario 4

- Treffen gefunden und Teilnahme sowie öffnen der Konferenz klar nach vorheriger Instruktion das Jitsi extern läuft

Protokoll Szenario 5

- Erstellung von Treffen mit allen Optionen einwandfrei durchgeführt
- Prozess zur Durchführung klappte gut, da er immer sofort das Videokonferenz Fenster geschlossen hatte und wieder zurück in die Webapplikation kehrte um den nächsten Prozesspunkt durchzuführen.

A.8.4 Testprotokoll Person 4

Mike Müller hat schon an Treffen teilgenommen

- Alter: 28
- Geschlecht: Männlich
- Beruf oder Kenntnisse: Gamer, technisch versiert

Vorinterview

Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?

- Ja

Hast du schon einmal ein Video-Treffen organisiert oder warst du ein Moderator?

- Ja

Welche Konferenz-Provider kennst du, oder würdest du einsetzen?

- GoToMeeting, Teams, Jitsi, Discord

Über welche Plattform und Geräte würdest du aktuell ein Video-Treffen durchführen?

- Desktop

Was wäre für dich wichtig um an einem spezifischen Video-Treffen teilzunehmen?

- Keine Übertragungsverzögerung

Nachinterview

Wie ist dein allgemeiner Eindruck?

- Clean Interface

Was hat dir gut gefallen?

- Nicht überladen

Wie nützlich beurteilst du die Plattform?

- Nützlich für Themen wie Politik wo nicht nur Kollegen in Frage kommen

Würdest du diese Plattform einsetzen oder verwenden?

- Persönlich eher weniger

Protokoll Szenario 1

- Registrieren klar, Logout klar

Protokoll Szenario 2

- Feedback beim Speichern des Profils fehlt

Protokoll Szenario 3

- Entdecken Link intuitiv gefunden
- Enter sollte Suche auslösen
- Gross- und Kleinschreibung soll nicht geprüft werden bei der Suche
- Suche soll auch Tags finden nicht nur Beschreibung und Titel
- Auf Tag klicken soll Suche danach auslösen

Protokoll Szenario 4

- Einwandfrei, alles gefunden

Protokoll Szenario 5

- Problemlose Navigation und Prozessverständnis

A.9 Revisionen der Risikoanalyse

Nachfolgend sind die Anpassungen an der Risikoanalyse (Kapitel 7) im Verlaufe des Projekts ersichtlich.

MS1: Projektplan

Erste Version nach Erreichen des Meilensteins *MS1: Projektplan* (10. März).

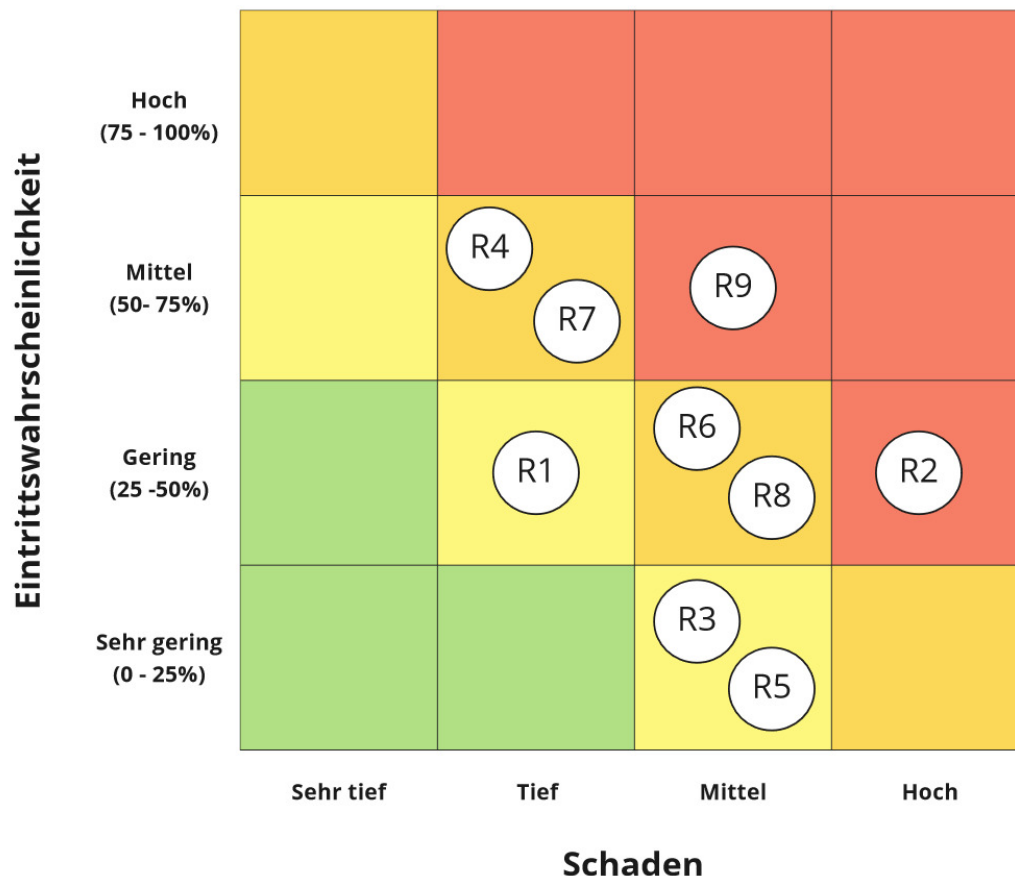


Abbildung A.1: Risikomatrix, Stand MS1

MS2: End of Elaboration

Zweite Version nach Erreichen des Meilensteins *MS2: End of Elaboration* (05. April)

R4: Nicht erfüllbare Funktionalität

Die Anforderungsanalyse und die Evaluation von Technologien haben Unsicherheiten beseitigt und mögliche Probleme hervorgehoben. Somit ist dieses Risiko besser abschätzbar worden.

Eintrittswahrscheinlichkeit Senkung von *Mittel* auf *Gering*

R6: Mängel der Architektur

Durch die genauere Definition und die vorgegebenen Strukturen von *Nest.js* ist die grobe Architektur gefestigt.

Schaden Senkung von *Mittel* auf *Tief*

R8: Abhängigkeiten zu Drittparteien

Die eingesetzten Drittparteien verfügen entweder über keine Rate Limits, oder sind Open Source und austauschbar.

Schaden Senkung von *Mittel* zu *Tief*

Eintrittswahrscheinlichkeit Senkung von *Gering* zu *Sehr gering*

R9: Nichtfertige Kernfunktionalität blockiert restliche Funktionen

Die Erfassung der Anforderungen und die Evaluation der Technologien haben geholfen wichtige Kernfunktionen zu bestimmen. Diese wurden entsprechend für die Umsetzung priorisiert.

Eintretenswahrscheinlichkeit Senkung von *Mittel* zu *Gering*.

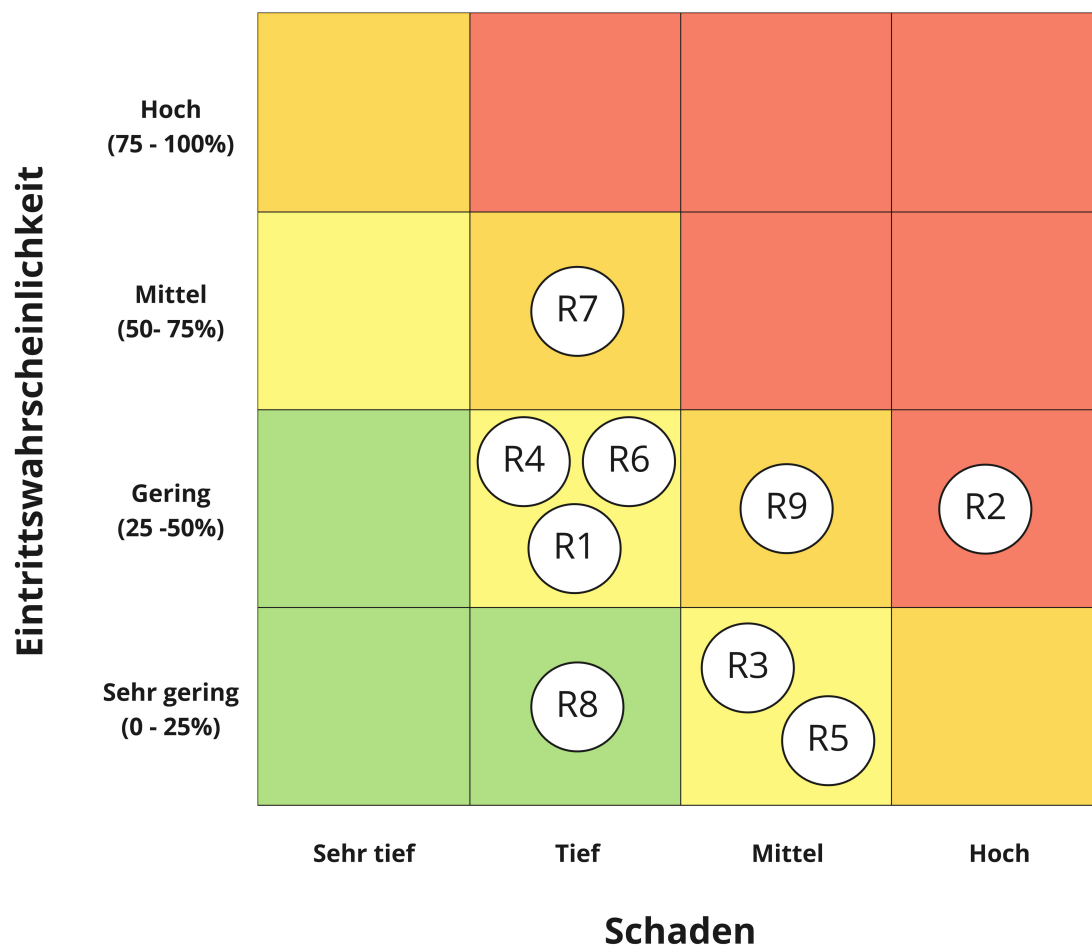


Abbildung A.2: Risikomatrix, Stand MS2

MS3: Alpha (MVP)

Dritte Version noch Erreichen des Meilensteins *MS3: Alpha (MVP)* (03. Mai)

R4: Nicht erfüllbare Funktionalität

Die Umsetzung der Grundfunktionen und die Arbeit mit den gewählten Technologien haben gezeigt, dass die geplante Funktionalität technisch erreichbar sein sollte. Die Grundfunktionen sollten alle im vorhandenen Zeitrahmen umsetzbar sein.

Eintrittswahrscheinlichkeit Senkung von *Gering* auf *Sehr gering*

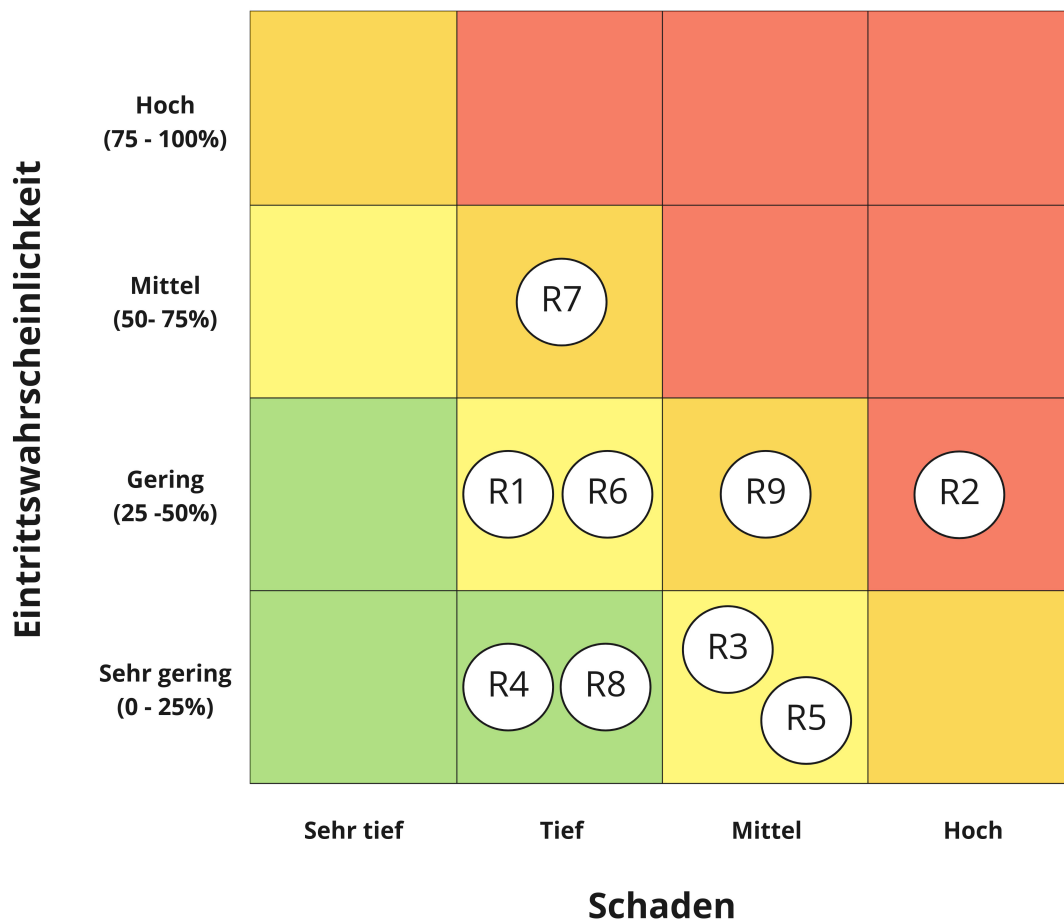


Abbildung A.3: Risikomatrix, Stand MS3

A.10 Wireframes

In der *Elaboration*-Phase wurden Wireframes mit **Figma** für die spätere Umsetzung erstellt.

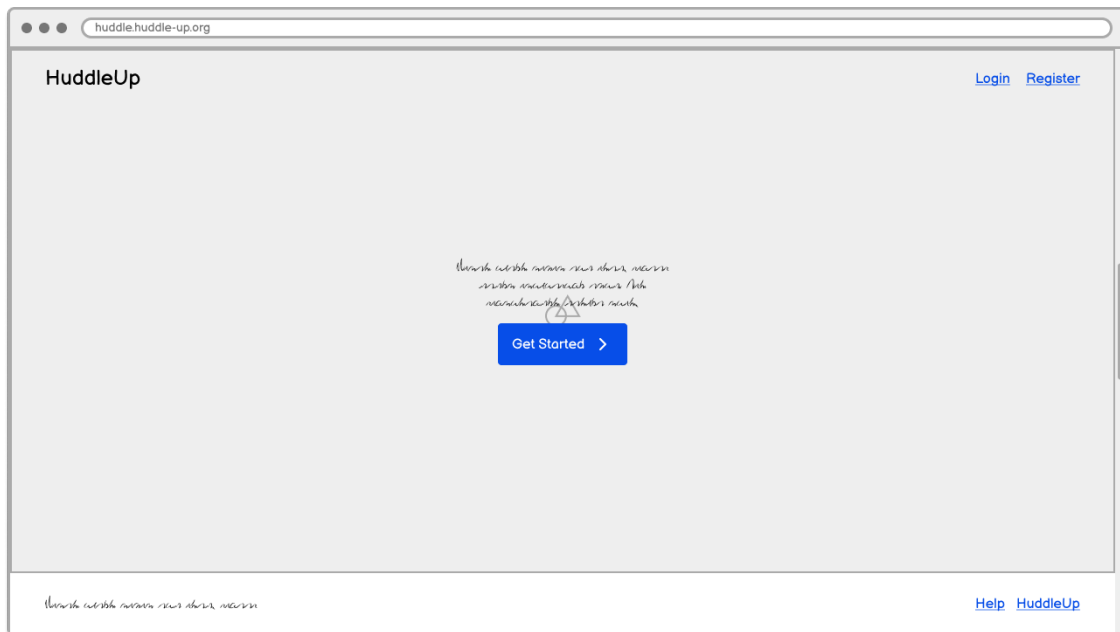


Abbildung A.4: Startseite

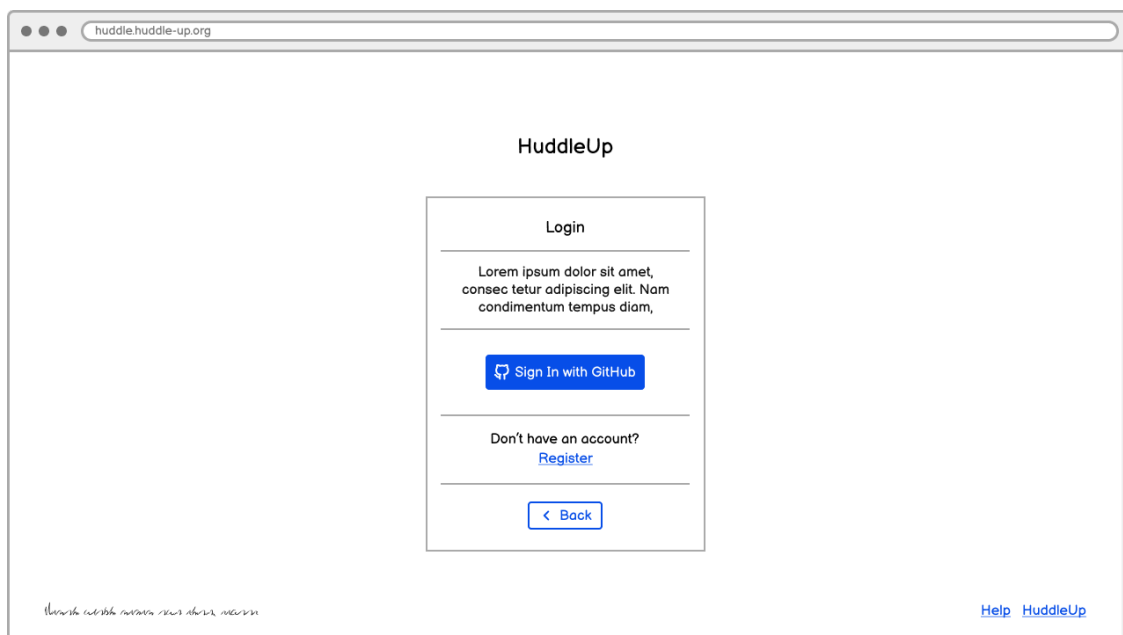


Abbildung A.5: Anmeldung

huddle.huddle-up.org

HuddleUp Discover My Meetings Create Meeting John Doe

Profile

Profile

Your Information

Full Name: John Doe

Email: john.doe@mail.test

Save

Public Profile

This information will be visible to all users.

Biography

Lorem ipsum dolor sit amet, consectetur ui i iadipiscing elit.

Danger Zone

Delete Account

Delete your account and remove all data associated with it. Beware, this is irreversible!

Delete Account

Help HuddleUp

Abbildung A.6: Benutzerprofil

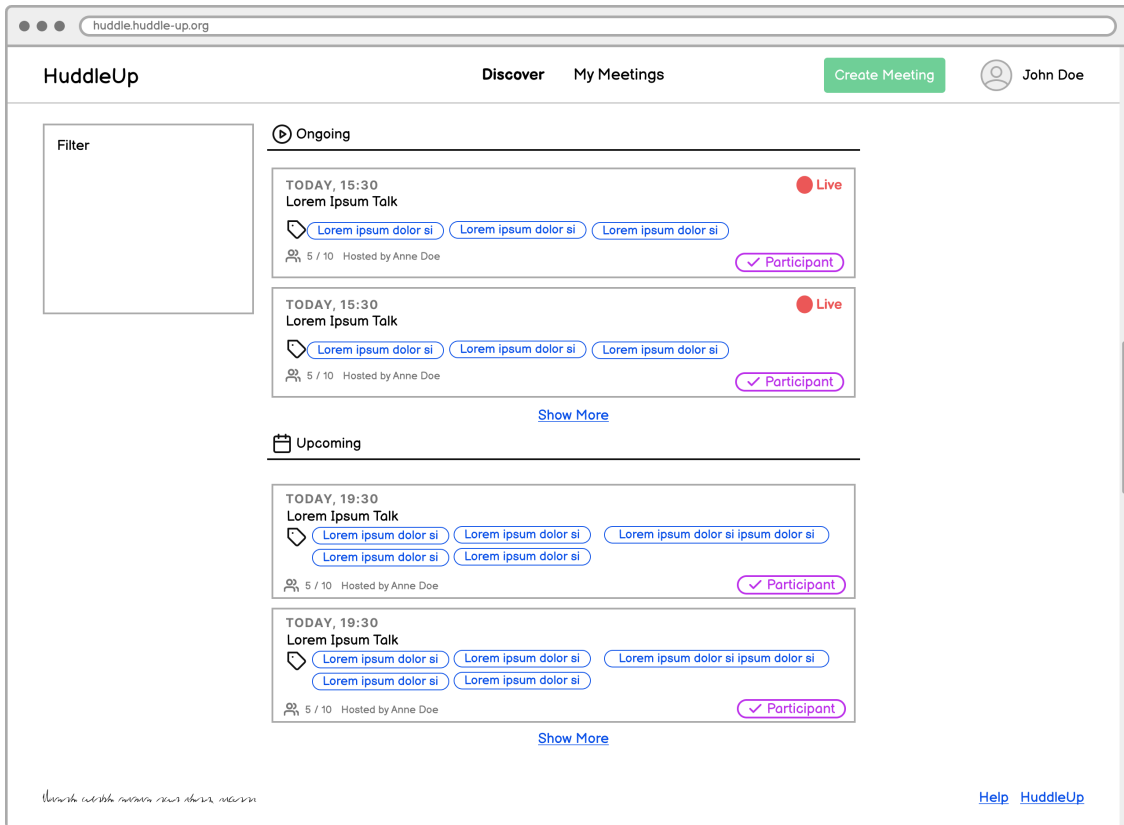


Abbildung A.7: Alle Treffen

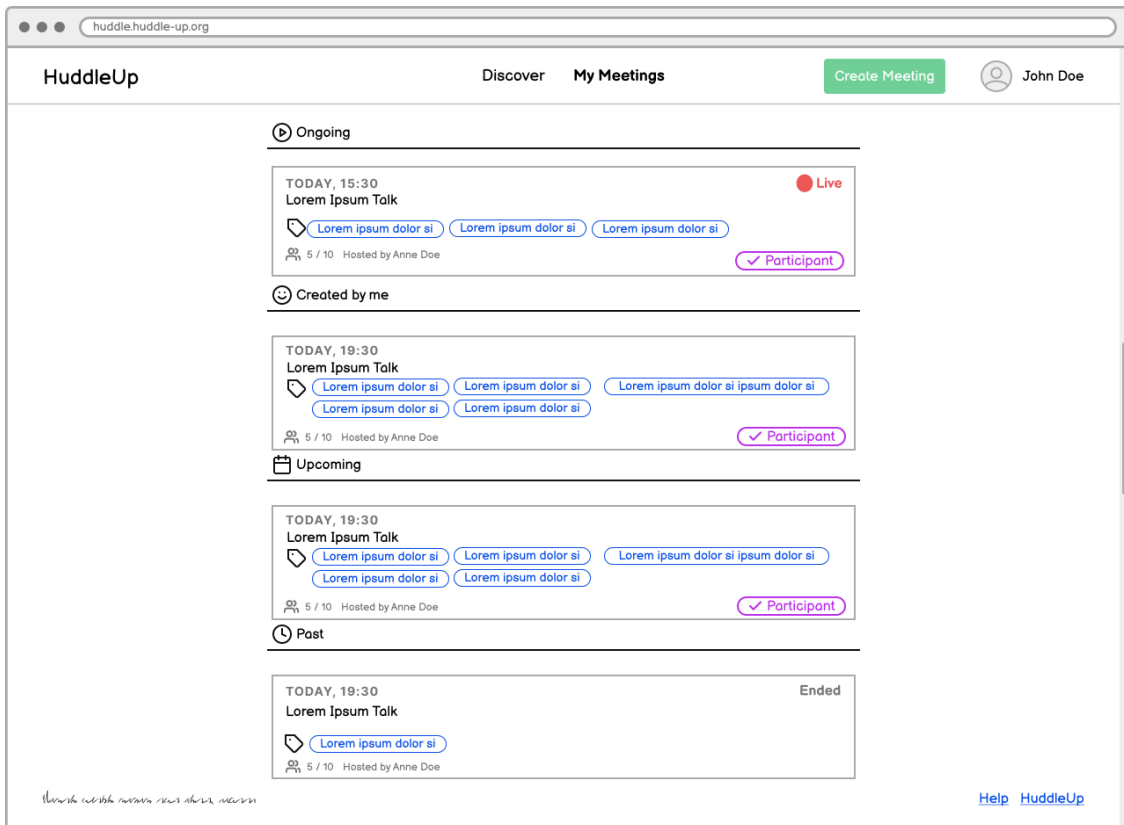


Abbildung A.8: Meine Treffen

The screenshot shows a web browser window with the URL `huddle.huddle-up.org`. The page title is "HuddleUp". In the top navigation bar, there are links for "Discover" and "My Meetings", a green "Create Meeting" button, and a user profile icon for "John Doe".

The main content area is titled "New Event" and contains three sections:

- About the Event:** A "Title" field with the text "Cat Up!" and a "Full Description" text area containing the placeholder text "Lorem ipsum dolor sit amet, consectetur ui i iadipiscing elit."
- Date and Time:** Two date and time pickers. The "Start" field is set to "29.06.2021 15:30" and the "Estimated End" field is set to "29.06.2021 19:00".
- Participation:** A "Maximum Participants" slider set to 5. Below the slider are two checked checkboxes: "Limit number of participants" and "Allow participation without confirmation".

At the bottom of the form, there are "Cancel" and "Create Event" buttons. In the bottom left corner, there is a small, faint logo, and in the bottom right corner, there are links for "Help" and "HuddleUp".

Abbildung A.9: Treffen erstellen

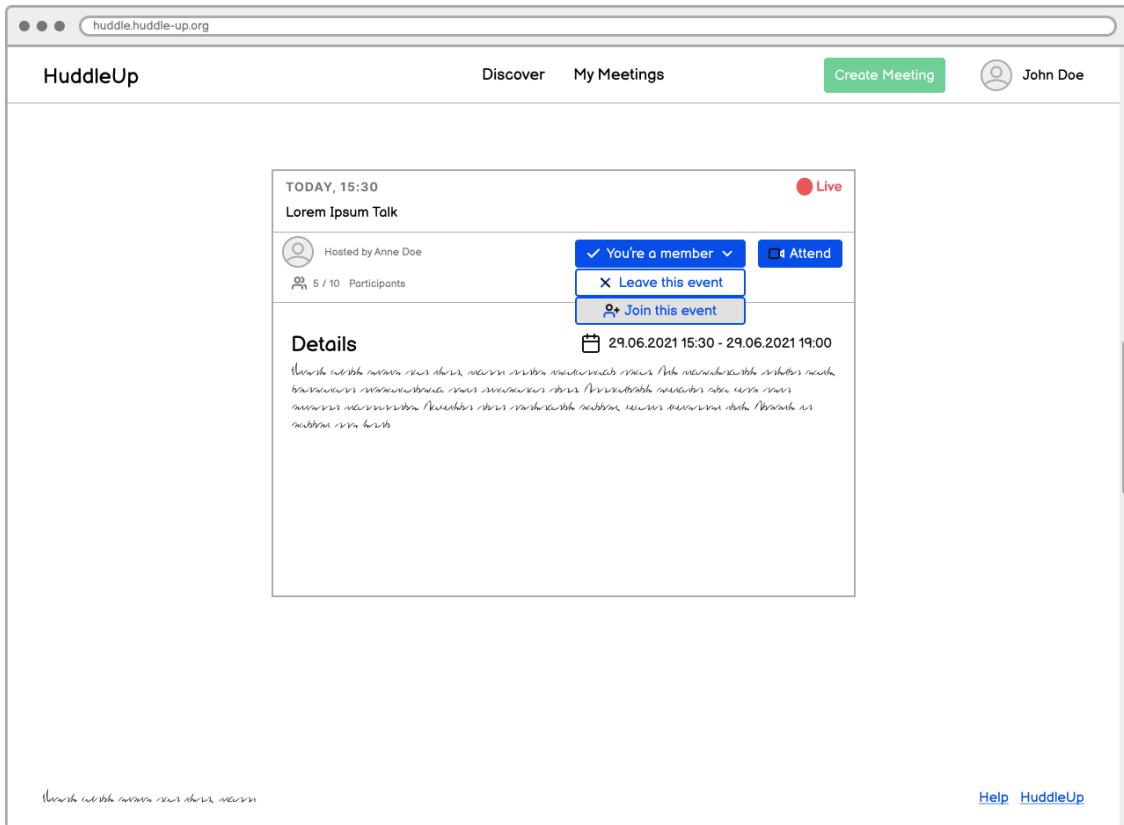


Abbildung A.10: Teilnahme an Treffen

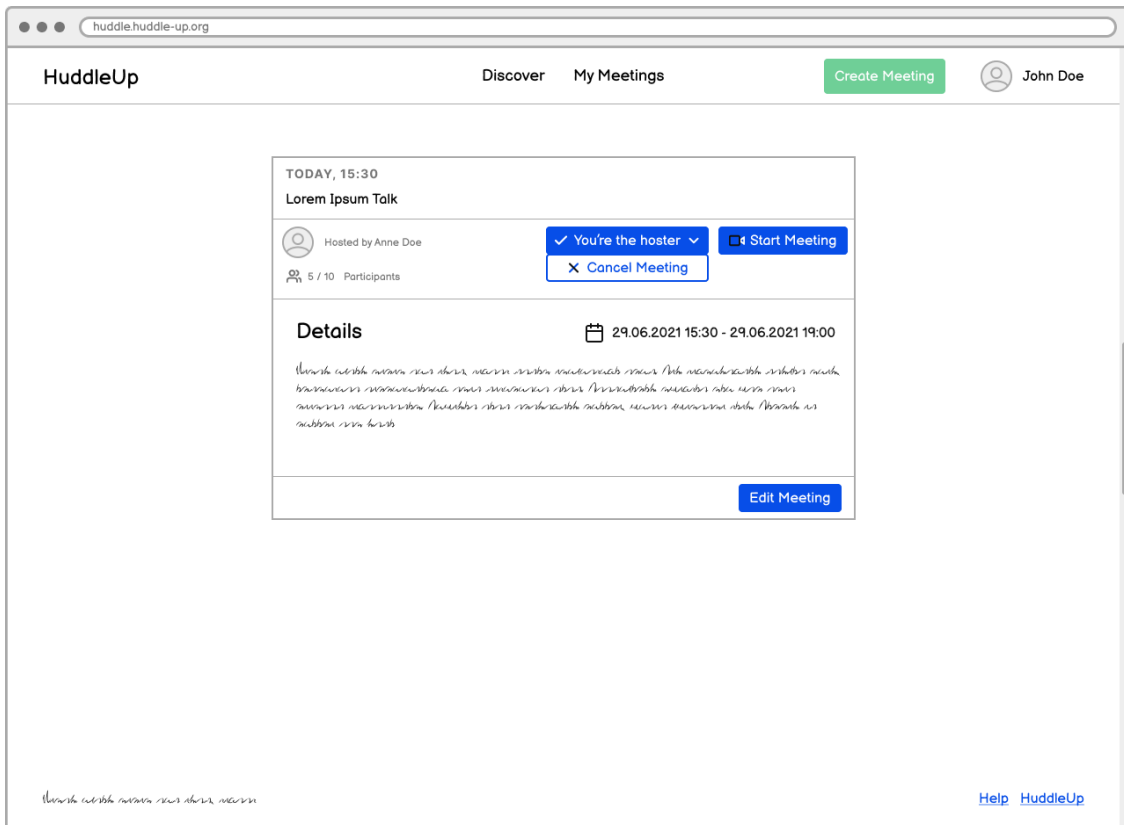


Abbildung A.11: Durchführung von Treffen