

# Konzept und Vision

## Motion Capture für E-Learning

ausgearbeitet von

Lali Nurtaev und Tristan Schmele

vorgelegt an der

TECHNISCHEN HOCHSCHULE KÖLN  
CAMPUS GUMMERSBACH  
FAKULTÄT FÜR INFORMATIK UND  
INGENIEURWISSENSCHAFTEN

im Studiengang

MEDIENINFORMATIK (M.A.)

Projektleiter: Prof. Dr. Horst Stenzel  
Technische Hochschule Köln

Betreuer: Jannis Möller  
Technische Hochschule Köln

Gummersbach, im Februar 2022

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Problemstellung . . . . .	3
1.2	Ziel . . . . .	3
<b>2</b>	<b>Anforderungsmanagement</b>	<b>4</b>
2.1	Stakeholderanalyse . . . . .	4
2.2	Nutzergruppen . . . . .	4
2.3	Anforderungen . . . . .	5
<b>3</b>	<b>Marktrecherche</b>	<b>6</b>
3.1	Bestehende Produkte . . . . .	6
3.1.1	3D Animation . . . . .	6
3.1.2	E-Learning . . . . .	7
3.2	Bestehende Frameworks . . . . .	7
3.3	State of the Art . . . . .	8
<b>4</b>	<b>Avatar Animation</b>	<b>10</b>
4.1	Modell . . . . .	10
4.2	Autorigging . . . . .	10
<b>5</b>	<b>Systemarchitektur</b>	<b>11</b>
5.1	Motion Capture . . . . .	11
5.2	Geeignete Frameworks . . . . .	11
5.3	Architektur Empfehlungen . . . . .	12
<b>6</b>	<b>Prototyp</b>	<b>13</b>
6.1	Erstellung in Figma und Anima . . . . .	13
6.2	Nutzerinterviews . . . . .	13
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>15</b>
<b>8</b>	<b>Fazit</b>	<b>16</b>
	<b>Abbildungsverzeichnis</b>	<b>17</b>
	<b>Tabellenverzeichnis</b>	<b>18</b>
	<b>Literaturverzeichnis</b>	<b>19</b>

# **1 Einleitung**

## **1.1 Problemstellung**

## **1.2 Ziel**

## 2 Anforderungsmanagement

In diesem Kapitel werden Stakeholder und Anforderungen des Systems gesammelt und präsentiert.

### 2.1 Stakeholderanalyse

Als Stakeholder des Systems sind primär Lehrkräfte und Lernende zu identifizieren, welche die Lehrinhalte generieren und konsumieren sollen. Darüber hinaus Bildungseinrichtungen an sich und Anbieter von Motion Capture Technologie.

Stakeholder	Kategorie	Bezug	Beschreibung
Lehrkräfte	Individuum	Interesse	Erstellen von Videos
		Interesse	Veröffentlichen von Lehrvideos
		Interesse	Austausch mit anderen Benutzern
		Anteil	Selbsterstellte Inhalte
		Anteil	Selbst aufgezeichnete MoCap Daten
		Anrecht	Datenschutz/-sicherheit
Lernende	Individuum	Interesse	Konsumieren von geteilten Inhalten
		Interesse	Austausch mit anderen Benutzern
		Anrecht	Datenschutz/-sicherheit
Bildungseinrichtungen	Organisation	Interesse	Verwalten der Kurse von mehreren Lehrkräften
		Anteil	Inhalte der Kurse
		Anteil	Daten der Kurse
		Anrecht	Datenschutz/-sicherheit
MoCap Unternehmen	Organisation	Interesse	MoCap Systeme verkaufen
		Interesse	MoCap Studio vermieten

Tabelle 2.1: Stakeholdertabelle

### 2.2 Nutzergruppen

Benutzer könne innerhalb des Systems Inhalte erstellen und Inhalte konsumieren. Dabei sind Kursleiter für die Erstellung von Inhalten und Kursteilnehmer für das Konsumieren von Inhalten zuständig. Ein Benutzer ist dabei nicht limitiert zu einer Benutzergruppe und kann in verschiedenen Kursen verschiedene Gruppen vertreten.

## 2 Anforderungsmanagement

Gruppe	Beschreibung
Administrator	Verwaltet das System. Kann Benutzeraccounts deaktivieren und löschen.
Kursverwalter	Verwaltet einen oder mehrere Kurse. Kann Kursteilnehmer zu Kursleitern oder Moderatoren ernennen. Kann beim erstellen eines neuen Kurses direkt einen anderen Benutzer als Kursleiter festlegen.
Kursleiter	Leiter eines Kurses. Ist für die Veröffentlichung von Kursinhalten und die Verwaltung dieses Kurses zuständig. Kann Kursteilnehmer zu Moderatoren ernennen. Kann Kursteilnehmer aus dem Kurs kicken oder bannen.
Moderator	Kann Kursteilnehmer aus dem Kurs kicken oder bannen. Kann Kommentare von Kursteilnehmern verwalten.
Kursteilnehmer	Kann veröffentlichte Inhalte konsumieren. Kann veröffentlichte Inhalte kommentieren.

Tabelle 2.2: Benutzergruppentabelle

### 2.3 Anforderungen

Anforderungen werden in der Regel in Funktional und Nicht-Funktional sortiert, wobei die Definition für Nicht-Funktionale Anforderungen nicht einheitlich festgelegt ist. Für dieses Projekt werden die Anforderungen daher in Funktional (F) und Qualitativ (Q) sortiert, wobei Qualitativ beispielsweise Performance des Systems beinhaltet. Da das primäre Ziel des Projekts einem Proof of Concept ähnelt, werden die Funktionalen Anforderungen priorisiert und Qualitative Anforderungen können zu gunsten der Umsetzung einer Funktionalen Anforderung zunächst vernachlässigt werden.

Bezeichnung	Beschreibung
F10	Das System muss dem Lehrer die Möglichkeit bieten MoCap Daten einzusehen.
F20	Das System muss dem Nutzer die Möglichkeit bieten Videos interaktiv zu betrachten.
F30	Das System muss dem Lehrer die Möglichkeit bieten Inhalte in seinen Kursen zu teilen
F40	Das System muss dem Lehrer die Möglichkeit bieten Inhalte in seinen Kursen zu teilen
F50	Das System muss dem Schüler die Möglichkeit bieten geteilte Inhalte in seinen Kursen zu konsumieren.
F60	Das System muss dem Lehrer die Möglichkeit bieten Inhalte zu speichern.
F70	Das System muss dem Lehrer die Möglichkeit bieten eigene Inhalte zu verwalten.
F80	Das System muss fähig sein Informationen wie Titel, Beschreibung und Länge zu erstellten Videos anzugeben.
F90	Das System sollte Avatare für die Präsentation der MoCap Daten abbilden.
F100	Das System sollte dem Lehrer die Möglichkeit bieten eigene Kurse zu erstellen.
F110	Das System sollte dem Schüler die Möglichkeit bieten Kursen beizutreten.
F120	Das System sollte fähig sein eine interaktive Vorschau von Clips zu ermöglichen.
F130	Das System sollte dem Lehrer die Möglichkeit bieten Clips zu einem neuen Inhalt hinzuzufügen
F140	Das System sollte dem Lehrer die Möglichkeit bieten die Cliplänge anzupassen.
F150	Das System sollte dem Lehrer die Möglichkeit bieten Audiodateien zu erstellen.
F160	Das System sollte dem Lehrer die Möglichkeit bieten Textinhalte zu erstellen.
F170	Das System sollte dem Lehrer die Möglichkeit bieten erstellte Textelemente zu bearbeiten.
Q10	Das System soll fähig sein die Videoinhalte stabil auf durchschnittlichen Computern darzustellen.
Q20	Das System soll fähig sein die Videoinhalte stabil bei durchschnittlichen Internetverbindungen darzustellen

Tabelle 2.3: Funktionale und Qualitative Anforderungen

## 3 Marktrecherche

In diesem Kapitel werden auf dem Markt bestehende Produkte und Frameworks analysiert. Dabei werden sowohl die Stärken, als auch die Schwächen der einzelnen Produkte und Frameworks erfasst, um abzuwägen welche davon für die Umsetzung des Systems vorteilhaft sind. Zudem werden State of the Art Produkte für die Darstellung von 3D Bewegungen vorgestellt. So können entweder Teilaspekte übernommen oder optimiert werden.

### 3.1 Bestehende Produkte

Die Marktrecherche zu den bereits bestehenden Produkten wird in 3D Animation und E-Learning aufgeteilt. Unter dem Unterkapitel 3D Animation werden Konkurrenzprodukte vorgestellt, die 3D Avatare im Lehrkontext verwenden. Im Unterkapitel E-Learning werden E-Learning Plattformen vorgestellt, auf denen das zu erstellende System veröffentlicht werden kann.

#### 3.1.1 3D Animation

##### Youtube Videos - Yoga mit Modi

Auf dem Youtube-Kanal „Yoga with Modi“ werden Yoga-Übungen über einen 3D Avatar ausgeführt und vorgestellt. Der Avatar repräsentiert Narendra Modiden, den Premierminister Indiens. In den Übungsvideos werden unterschiedliche Yoga-Stellungen vorgeführt. Dabei werden sie aus verschiedenen Betrachtungswinkeln (z.B. seitlich, vorne, von oben) und über mehrere Bildausschnitte mit unterschiedlichen Zoom-Stufen gezeigt. Über Bildausschnitte auf bestimmte Körperregionen können Ausführungen detailliert vorgestellt werden. Während der Vorführung der Übungen durch den Avatar wird die Ausführung über ein Voice-Over erläutert. Erläuterungen werden über Animationen und Texte eingeblendet. Jedoch wird bei den Videos nicht ersichtlich, ob die Yoga Übungen mittels Motion Capture erstellt werden.

**Stärken:** Die Youtube Videos ermöglichen eine einfache Bedienung für Lernende und sind auf unterschiedlichen Sprachen zugänglich. Über Playlisten können die Videos vom Lehrenden oder Lernenden beliebig aneinander gereiht werden. Die Texte, Animationen und das Voice Over hilft bei dem Nachmachen der Übungen und dem Verständnis für Yoga.

**Schwächen:** Eine Interaktivität durch den Lernenden ist nicht gegeben. Die Videos können nur angehalten, zurück oder weiter gespult werden. Des Weiteren können die Ausführungen nur aus dem vorgegebenen Winkel und Bildausschnitt betrachtet werden.

#### Charamel

Das Unternehmen Charamel bietet eine E-Learning Plattform mit virtuellen Trainern in Form von 3D Avataren an. Die Anwendungsszenarien der Trainer besteht aus Unterweisungen, Schulungen und Trainings beispielsweise im Brandschutz, Sicherheit oder im Fahrbereich. Die Avatare dienen als interaktive Gesprächspartner für Lernende. Auf der Plattform wird eine Übersicht des Lernfortschritts und eine Statistik bereitgestellt. Die Plattform ist über den Browser zugänglich und die Lerninhalte werden auf 30 Sprachen zur Verfügung gestellt. Die Lerninhalte können von Unternehmen individuell über Fragenelemente oder erstellte Avatare angepasst werden. Es wird nicht ersichtlich, ob für die Bewegungen der Avatare Motion Capture verwendet wird.

**Stärken:** Die E-Learning Plattform bietet durch die interaktiven Avatare als Trainer eine auf den Lernenden angepasste Lernsituation. Kunden können ihre Lerninhalte über bestimmte Fragenelemente oder die Erstellung und Nutzung eigener Avatare mit VuppetMaster anpassen. Das Produkt ist auf dem Handy, Tablet und Computer abrufbar, da es im Browser zur Verfügung gestellt wird.

**Schwächen:** Die Verwendung der E-Learning Plattform für virtuelle Trainer ist kostenpflichtig. Jedoch wird mit einem Kauf zusätzlich ein Support von Charamel angeboten.

#### TriCat

TriCat bietet virtuelle Lern- und Arbeitswelten und E-Learning Lösungen. Für ein virtuelles Zusammentreffen werden Räume angeboten, durch die mittels einen selbst angepassten 3D Avatar navigiert werden kann. Der Lernende hat die Möglichkeit Situationen aus unterschiedlichen Perspektiven (Ego, Verfolger, frei steuerbar) zu betrachten. Es werden Konferenzen, Trainings und Ersthelfer-Kurse angeboten, an denen mit einem Headset teilgenommen werden kann. Zudem können Veranstaltungen individuell gestaltet werden und beinhalten u.a. Chats, Umfragen und Whiteboards. Darüber hinaus hat TriCat einen 3D Viewer, in welchem Teilnehmer mit 3D Modellen interagieren können. Obwohl Teilnehmer mit einem 3D Avatar durch die virtuelle Welt navigieren können, kommt Motion Capture nicht zum Einsatz.

**Stärken:** Das System bietet Lernenden ein Präsenzgefühl und Interaktionsmöglichkeiten mit anderen Teilnehmern. Es werden unterschiedliche Übungsszenarien und Elemente zum gemeinsamen Arbeiten zur Verfügung gestellt.

**Schwächen:** Für die Verwendung der virtuellen Welten muss eine eigene Software installiert werden. Die Verwendung des Systems ist ebenfalls wie bei Charamel kostenpflichtig, enthält aber einen Support.

#### 3.1.2 E-Learning

### 3.2 Bestehende Frameworks

#### Unity und Unreal

Unity ist eine Entwicklungsumgebung, die für die Erstellung von Spielen und weiteren interaktiven 3D Anwendungen verwendet wird. In Unity können 3D Avatare erstellt,



bearbeitet und mit Motion Capture Daten verbunden werden. Zudem ist es möglich in Unity ein User Interface mit sämtlichen Funktionen zu entwickeln. Über ein WebPlayer Plugin kann eine in Unity erstellte 3D Anwendung im Webbrowser implementiert werden. Eine Alternative zu Unity ist die Unreal Engine, die ebenfalls die Option bietet eine 3D Anwendung im Webbrowser zu implementieren.

#### **Web Graphics Library (WebGL)**

WebGL ist eine Programmierschnittstelle, die es ermöglicht interaktive 3D Grafiken im Webbrowser anzuschauen. Über Unity, Unreal oder three.js erstellt Inhalte können nach WebGL exportiert werden. WebGL wird verwendet, um komplizierte 3D Objekte in verschiedenen Browsern anzeigen zu können.

#### **Three.js**

Die Programmiersprache JavaScript bietet eine Bibliothek und Programmierschnittstelle namens three.js, mit welcher 3D Modelle und Animationen erstellt und im Webbrowser angezeigt werden können. Über three.js kann eine Szene gerendert und aus einer bestimmten Kameraperspektive animiert und gerendert werden. Three.js eignet sich sehr gut für 3D Grafiken und Animationen, die eine Webanwendung verschönern.

#### **Viewer**

Es existieren unterschiedliche Model Viewer, die für die interaktive Darstellung von 3D Animationen und Motion Capture Daten im Web verwendet werden können. Durch eine einfache Einbettung des Viewers in den HTML Code wird der Modell Viewer auf in einer Webanwendung angezeigt. Ein Beispiel dafür ist der Model Viewer, in welchen 3D Modelle im Datenformat gltf/glb eingelesen werden können. Im Viewer ist es möglich ein Modell zu drehen und ranzuzoomen. Der Model Viewer bietet zwar eine unkomplizierte Einbettung eines Fensters für das Anschauen von 3D Animationen, hat jedoch Einschränkungen, welche das Datenformat, den Webbrowser und die Funktionen für das Betrachten betreffen.

### **3.3 State of the Art**

Da wir als Team keine Vorkenntnisse in der 3D Animation haben, haben wir uns dazu entschieden State of the Art Produkte für die Darstellung von 3D Bewegungen zu betrachten. So können Erkenntnisse zu Ansätzen für die spätere Implementierungsmöglichkeiten gewonnen werden.

#### **Cologne Motion Capture Database (CMCD)**

Die CMCD ist durch eine Bachelorarbeit am Institut für Medien- und Phototechnik der TH Köln entstanden. Die Datenbank enthält aufgenommene Motion Capture Daten, die in einer Liste angeordnet, nach Kategorie ausgewählt und über einen Model Viewer angeschaut werden können. Für die Darstellung der 3D Animationen wird ein Model

Viewer und WebGL verwendet. Im Viewer werden die Bewegungen über ein Skelettmodell ohne Avatar angezeigt. Dem Nutzer ist es möglich das Skelettmodell beliebig zu drehen, heranzuzoomen und zu pausieren. Zudem gibt es die Möglichkeit die Hüfte zu fixieren, sodass Bewegungen auf einem Punkt ausgeführt werden. Das Bewegungsaufnahme wird über eine Schleife abgespielt und kann über einen Pause-Button gestoppt werden. Der Viewer für die Anzeige der Motion Capture Daten ermöglicht eine interaktive Betrachtung, um die dargestellten Bewegungen aus bestimmter Perspektive und Kameraeinstellung anschauen zu können.

#### **Avatar SDK**

Avatar SDK ist ein im Web zugängliches Toolkit für die Erstellung und Anapssung eines eigenen 3D Avatars. Es bietet die Möglichkeit ein Avatar Modell in Ganzkörper-, Kopf- und Kopf-Schulter-Darstellung über ein Foto zu generieren und für eine Betrachtung zu drehen. Das erstellte Modell kann bestimmte Bewegungen wie Zwinkern, Blinzeln und Lächeln ausführen. Dem Modell können verschiedene Frisuren, Haar- und Hautfarbe zugewiesen werden. Die Anwendung mit allen Funktionen wurde in Unity erstellt und über WebGL dargestellt.

#### **Mixamo**

Mixamo ist eine Webanwendung, auf der 3D Animationen zur Verfügung gestellt werden. Die 3D Animationen werden über Gifs aufgelistet und angezeigt. Beim Auswählen eines Gifs erscheint die Animation in einem Model Viewer. Im Viewer wird dem Nutzer die Möglichkeit gegeben das 3D Modell zu drehen, ranzoomen und die Animation zu pausieren. So kann die Bewegung wie bei den zwei vorherigen Produkten interaktiv betrachtet werden. Für die Darstellung der Animationen im Viewer verwendet Mixamo Adobe.

## 4 Avatar Animation

Für das zu erstellende System werden 3D Avatare verwendet, um aufgenommene Motion Capture Daten darzustellen. Für die Darstellung von Tanzbewegungen werden jeweils ein weiblicher und ein männlicher Avatar angeboten, um verschiedene Rollen zu symbolisieren. Diese sollten für eine optimale Bewegungserkennung realitätsnah sein. Im ersten Schritt werden Optionen aufgezählt, welcher Avatar in der Implementierung verwendet werden kann. Im zweiten Schritt wird erläutert, wie Motion Capture Daten einfach mittels Avatare ausgeführt werden können.

### 4.1 Modell

Eine Möglichkeit der Avatar Erstellung ist das im Kapitel 3.3 vorgestellte Toolkit Avatar SDK. Durch das Auswählen eines Ganzkörper Bildes wird ein 3D Avatar Modell erstellt. Das Modell kann sowohl im Toolkit, als auch in Blender angepasst werden. Der Vorteil von Avatar SDK ist die schnelle und einfache Erstellung von Avataren durch Fotos. Hierfür ist keine Erfahrung in der 3D Erstellung notwendig. Der Nachteil besteht darin, dass das Exportieren eines Avatars in der Ganzkörperdarstellung zuzeit noch nicht angeboten wird. Es kann jedoch ein 3D Kopf heruntergeladen werden. Dieser müsste in Blender auf einen Körper plaziert werden.

Eine weitere Möglichkeit ist die Verwendung bereits erstellter Avatare, welche auf unterschiedlichen Plattformen wie Mixamo, free3d und renderpeople angeboten werden. Die Avatare unterscheiden sich im Dateiformat, Aussehen und Realitätsgrad. Der Vorteil dieser Option ist die große Auswahl an realitätsnahen Avataren, dessen Modell in Ganzkörperdarstellung kostenfrei heruntergeladen werden kann. Die Webseite renderpeople.com bietet viele Avatare, die durch das Einscannen von Fotos entstanden sind. Aus diesem Grund werden ein weiblicher und männlicher Avatar dieser Webseite für das Projekt verwendet.

### 4.2 Autorigging

Das Verbinden der Motion Capture Daten mit den 3D Avataren kann in unterschiedlichen Anwendungen und auf verschiedene Arten erfolgen. Da in dem Projekt eine größeren Anzahl an Motion Capture Daten entstehen kann, sollte das Rigging automatisch erfolgen. In Blender wurde das Rigging zunächst sowohl manuell, als auch über das Plugin Rigify getestet. Da das Projekt in anderen Entwicklungsumgebungen erstellt wird, wird es empfohlen das Autorigging in Unity, Unreal oder anderen Programmen auszuführen. Hierzu gibt es in den jeweiligen Programmen Funktionen oder Plugins, die verwendet werden können.

## 5 Systemarchitektur

Im folgenden werden einzelne Kernpunkte der Architektur des konzipierten Systems genannt und Empfehlungen für die Implementierung festgehalten.

### 5.1 Motion Capture

Um die Bewegungsdaten für die Inhalte des Systems aufzuzeichnen gibt es mehrere Möglichkeiten. Neben mehreren Versionen für optisches Motion Capture, können die Aufzeichnungen auch mit geringem zusätzlichem Aufwand mit VR (Virtual Reality) Systemen erstellt werden. In der Forschung zu E-Learning mit Motion Capture wird unter anderem mit dem VIVE VR System von HTC gearbeitet (Kasapakis u. Dzardanova, 2021), welches mit zusätzlichen Trackern an Armen, Beinen und Körper in der Lage ist Körperbewegungen vollständig aufzuzeichnen.

Der Vorteil solcher VR Systeme besteht vor allem in ihrem niedrigen Einstiegspreis. Jedoch sind optische Motion Capture Systeme durch eine höhere Menge an Daten in der Lage präzisere Aufzeichnungen zu erstellen. Da dem Projekt Team ein CapturyLive Motion Capture System zur Verfügung steht, ist zu empfehlen weiterhin mit diesem zu arbeiten. Sollte ein anderes Team jedoch zu Testzwecken ein neues Motion Capture System benötigen, wird empfohlen zunächst mit der kostengünstigeren Variante zu arbeiten.

Der geplante Einsatz für Motion Capture Technologie in diesem konzipierten System beschränkt sich in Praxis auf das Erstellen von Bewegungsdaten, welche vom eigentlichen System genutzt werden. Daher ist im finalen Einsatz die Quelle dieser Bewegungsdaten nicht relevant, solange diese in vom System nutzbaren Formaten vorliegen.

### 5.2 Geeignete Frameworks

Ein weiterer essentieller Bestandteil des Systems ist die Darstellung der animierten 3D-Avatare. Hierzu wurden in Kapitel 3 bereits mehrere potentielle Frameworks genannt:

- Unity
- Unreal Engine
- WebGL

Sowohl Unity als auch Unreal Engine sind Tools zur Erstellung von Video Spielen. Beide können in eine Website eingebunden werden und verfügen bereits über eingebaute Funktionen, um Kameras in einem 3D Raum zu steuern. Die wichtigsten Unterschiede sind hierbei die Bedienung der Software selbst, sowie ihre kommerziellen Lizenzen.

Unity kann kostenfrei genutzt werden, bis zu jährlichen Einnahmen von 100.000 USD. Anschließend kostet die Lizenz bis 200.000 USD Einnahmen 399 USD pro Jahr und darüber 1.800 USD im Jahr (Unity Technologies, 2022). Unreal Engine kann kostenfrei genutzt werden, bis zu Gesamteinnahmen von 1 Millionen USD und anschließend fallen 5% der Quartaleinnahmen als Lizenzgebühren an, für jedes Quartal mit Einnahmen von mehr als 10.000 USD (Epic Games, Inc., 2022).

WebGL ist eine JavaScript API zur Darstellung von 2D und 3D Grafiken im Web Browser (Mozilla Foundation, 2022). Bei einem kommerziellen Einsatz dieser API fallen keine Lizenzgebühren an. Allerdings muss die Implementierung vollständig selbst übernommen werden, da WebGL keine Entwicklungsumgebung wie Unity oder Unreal Engine bietet.

Da die Anforderungen an die 3D Darstellung innerhalb des Systems nicht über grundlegende Kamerabewegungen hinausgehen, wird empfohlen mit WebGL zu arbeiten, da hierbei im Falle eines späteren kommerziellen Nutzen des Systems keine extra Kosten anfallen. Ist keine kommerzielle Nutzung des Systems geplant oder sollten mehr Entwickler mit Unity oder Unreal Engine vertraut sein, als mit WebGL, ist jedoch empfehlenswert mit diesen zu arbeiten.

### 5.3 Architektur Empfehlungen

Bezüglich der Systemarchitektur ist vor allem wichtig zu Entscheiden, ob das System als Monolith oder in Form von Microservices implementiert werden soll. Monolithen bieten hierbei vor allem den Vorteil, dass die Implementierung selbst keine weiteren Herausforderungen stellt, während beim implementieren von Microservices ein von Martin Fowler benanntes “MicroservicePremium” anfällt (Fowler, 2015a). Ob ein neues System zunächst immer als Monolith erstellt werden sollte (Fowler, 2015b) oder ob Systeme die mit Microservices laufen sollen niemals als Monolith starten sollten (Tilkov, 2015), ist keine eindeutig geklärte Frage.

In einem Artikel von 2019 beschäftigt sich Herbert Dowalil mit dieser Frage und bietet einen Kompromiss. Die Architekturform, welche er “Modulith” nennt, soll als Vorbereitung auf einen Umstieg auf eine Microservice Architektur dienen und dabei alle Vorteile eines Monolithen in der Implementierung behalten (Dowalil, 2019). Hierbei werden bereits vorab die einzelnen Microservices grob eingeteilt und im System so unabhängig wie möglich implementiert, wobei die steigende Komplexität der vollständigen Trennung vermieden wird, da diese Services immernoch als Teil des Monolithen erstellt werden. Diese können anschließend in einzelnen Schritten vollständig vom Monolith gelöst werden, um am Ende eine Microservice Architektur zu erhalten.

Im Rahmen dieses Projekts wird empfohlen dieser Idee eines “Modulithen” zu folgen. Hierfür ist es essentiell als ersten Schritt die Domäne des Systems sinnvoll einzuteilen, um die interne Unterteilung vorzubereiten. Dabei bietet es sich an mit einem so genannten Bounded Context Modell zu arbeiten (Fowler, 2014). In diesem wird der Zusammenhang einzelner Aufgaben und Elemente einer Domäne abgebildet und abhängig von diesen Zusammenhängen unterteilt. Ohne diese Unterteilung besteht das Risiko, dass die erstellten Subsysteme zu eng miteinander gekoppelt werden und somit der Aufwand bei einem Übergang zu Microservices steigt.

## 6 Prototyp

Im Rahmen eines viertägigen Design Sprints wurde ein Prototyp erstellt, welcher die Screens des zu erstellenden Systems enthält. Der Prototyp ist anklickbar und in Figma entwickelt worden. Dieser wurde durch Nutzer in einem Interview getestet und die Ergebnisse dokumentiert.

### 6.1 Erstellung in Figma und Anima

Für die Erstellung des Prototypen wurde das Tool Figma verwendet, mit welchem ein interaktives User Interface gebaut werden kann. Der Prototyp ist aus der Sicht des Lehrenden dargestellt. Der Fokus des Prototypen liegt auf einer interaktiven Betrachtung eines 3D Modells. Das 3D Modell wurde auf der Webseite free3d im fbx-Format heruntergeladen. Für den interaktiven Viewer wurde das Plugin Anima und das Designtool Spline verwendet. Spline ermöglicht das Arbeiten mit 3D Objekten. In einer Szene wird das 3D Modell importiert. Die Szene wird als Frame in HTML exportiert. Dieses Frame kann über Anima im Prototypen in Figma eingebettet werden. So kann ein Nutzer das 3D Modell im Fenster interaktiv betrachten, indem er es drehen und heranzoomen kann.

Der Prototyp enthält ein Homescreen, in welchem die Kursübersicht eines Lehrenden angezeigt wird. Dem Nutzer ist es möglich ein neues Video zum Thema Yoga in einem Kurs zu erstellen. Der erste Schritt zur Erstellung eines Videos ist das Auswählen eines 3D Clips. Aus einer Auflistung von Bildern zu Yogastellungen wählt der Nutzer eine Stellung aus, die ihm im interaktiven Viewer angezeigt wird. Im nächsten Schritt kann der Nutzer eine Audio erstellen, die über dem Clip in einer Timeline eingefügt wird. Im weiteren Schritt hat der Nutzer die Möglichkeit einen Text zu erstellen, welcher im Viewer eingefügt wird. Am Schluss kann ein Titel, eine Beschreibung und ein Titelbild für das erstellte Video hinzugefügt werden.

### 6.2 Nutzerinterviews

Für das Testen des Prototypen wurden Sportlehrer von verschiedenen Schulen angeschrieben. Da wir keine Antwort erhalten haben, wurden Bekannte für Nutzerinterviews genommen. Das Nutzerinterview bestand aus mehreren kleinen Aufgaben, bei der der Nutzer durch den Prototypen durchnavigieren sollte und anschließenden Fragen. Die Aufgabe bestand darin ein neues Video zu den Kursen hinzuzufügen, indem alle dazugehörigen Schritte und Screens durchlaufen werden mussten. Die Ergebnisse wurden wie folgt dokumentiert:

**Positives:** Die Nutzer haben die interaktive Ansicht des 3D Modells angemerkt. Der Aufbau und Ablauf des Systems erinnert an ein Videoschnittprogramm und ist aus

diesem Grund verständlich.

**Negatives:** Den Nutzern hat das Bearbeiten und Entfernen von hinzugefügten Audio- und Textdateien und weitere Werkzeuge gefehlt. Des Weiteren war die Video- und Dateiübersicht nicht übersichtlich.

**Ideen:**

## **7 Zusammenfassung und Ausblick**

Zusammenfassung der Handlungsempfehlungen



## **8 Fazit**

(Evaluation der Ergebnisse)

# **Abbildungsverzeichnis**

# Tabellenverzeichnis

2.1	Stakeholdertabelle . . . . .	4
2.2	Benutzrgruppentabelle . . . . .	5
2.3	Funktionale und Qualitative Anforderungen . . . . .	5

# Literaturverzeichnis

- [Dowalil 2019] DOWALIL, Herbert: *Modulith First! Der angemessene Weg zu Microservices*. <https://www.informatik-aktuell.de/entwicklung/methoden/modulith-first-der-angemessene-weg-zu-microservices.html>. Version: 2019. – Last accessed 24 April 2021
- [Epic Games, Inc. 2022] EPIC GAMES, INC.: *Download Unreal Engine*. <https://www.unrealengine.com/en-US/download>. Version: 2022. – Last accessed 11 March 2022
- [Fowler 2014] FOWLER, Martin: *BoundedContext*. <https://martinfowler.com/bliki/BoundedContext.html>. Version: 2014. – Last accessed 8 March 2022
- [Fowler 2015a] FOWLER, Martin: *MicroservicePremium*. <https://martinfowler.com/bliki/MicroservicePremium.html>. Version: 2015. – Last accessed 8 March 2022
- [Fowler 2015b] FOWLER, Martin: *MonolithFirst*. <https://martinfowler.com/bliki/MonolithFirst.html>. Version: 2015. – Last accessed 8 March 2022
- [Kasapakis u. Dzardanova 2021] KASAPAKIS, Vlasios ; DZARDANOVA, Elena: Using High Fidelity Avatars to Enhance Learning Experience in Virtual Learning Environments. In: *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2021, S. 645–646
- [Mozilla Foundation 2022] MOZILLA FOUNDATION: *WebGL*. [https://developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API). Version: 2022. – Last accessed 11 March 2022
- [Tilkov 2015] TILKOV, Stefan: *Don't start with a monolith*. <https://martinfowler.com/articles/dont-start-monolith.html>. Version: 2015. – Last accessed 8 March 2022
- [Unity Technologies 2022] UNITY TECHNOLOGIES: *Compare Unity Plans*. <https://store.unity.com/compare-plans>. Version: 2022. – Last accessed 11 March 2022