



HTL - Perg

Höhere Abteilung für Informatik

Diplomarbeit

GetFitAgain

Projektteam: Gregor Arbeithuber
Michael Köck
Franz-Filip Schörghuber

Projektbetreuer: Prof. Dipl.-Ing. Michael Stumpfl

In Zusammenarbeit mit Fa. Therapiezentrum Aist
Betreuer Herr Ronald Voglhofer

Bearbeitungszeitraum: 01.10.2017 – 05.04.2018

Eidesstattliche Erklärung

Hiermit versichern wir, die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der von uns angegebenen Quellen angefertigt zu haben. Alle Stellen, die wörtlich oder sinngemäß aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Perg, _____ Unterschrift _____

Gregor Arbeithuber

Perg, _____ Unterschrift _____

Michael Köck

Perg, _____ Unterschrift _____

Franz-Filip Schörghuber

Gender-Erklärung

Für die bessere Lesbarkeit werden in dieser Diplomarbeit personenbezogene Bezeichnungen, die sich zugleich auf Frauen und Männer beziehen, generell nur in der im Deutschen üblichen maskulinen Form angeführt.

Dies soll jedoch keinesfalls eine Geschlechterdiskriminierung oder eine Verletzung des Gleichheitsgrundsatzes zum Ausdruck bringen.

Danksagung

An dieser Stelle möchten wir uns bei allen Personen bedanken, die uns bei der Entstehung der Diplomarbeit unterstützten und uns zur Seite standen.

Besonderer Dank gilt unserem Diplomarbeitsbetreuer Prof. Dipl.-Ing. Michael Stumpfl, der uns die gesamte Projektdauer für technische und organisatorische Fragen zur Verfügung stand. Weiters bedanken wir uns bei unserem Auftraggeber Herrn Ronald Voglhofer, der uns diese Diplomarbeit ermöglicht hat.

Kurzfassung

Aufgabenstellung

Derzeit ist es so, dass bei vielen Physiotherapeuten sehr viel Aufwand benötigt ist, um Trainingspläne für ihre Kunden zu erstellen. Die Physiotherapeuten, unter anderem auch unser Auftraggeber, erstellen die Trainingspläne oft in einfachen Textverarbeitungsprogrammen. In diese werden immer die benötigten Übungen hineinkopiert und es muss darauf geachtet werden, dass die Position der Bilder und des Textes stimmt. Diese Trainingspläne werden dann in einem nicht einheitlichen Format ausgedruckt und den Kunden mit nach Hause gegeben. Um dieses Problem zu lösen, wurde im Rahmen der Projektarbeit des Vorjahres vom Projektteam bereits ein einfaches Verwaltungsprogramm erstellt, in dem der Physiotherapeut seine Trainingspläne per Mausklick erstellen kann, welche schließlich als PDF abgespeichert werden. Aufbauend auf dieses Programm soll als Diplomarbeit eine mobile Anwendung für seine Kunden entwickelt werden, die es ihnen ermöglicht, auf ihren Smartphones die Trainingspläne abzurufen und ganz einfach von zu Hause abzuarbeiten.

Realisierung

Die mobile Anwendung ist als plattformunabhängige HTML5-App auf Basis des Angular Frameworks in Kombination mit dem Ionic Framework realisiert, dadurch ist es möglich, die App auf jedem beliebigen Smartphone abzurufen. Zusätzlich wird dem Benutzer zu jeder Übung eine Computeranimation angezeigt, die mit der freien Software Blender erstellt wurde. Dadurch sieht der Kunde genau, wie eine Übung durchzuführen ist. Außerdem soll der Physiotherapeut die Möglichkeit haben, die Fortschritte seiner Kunden anzusehen, Feedback zu bekommen und dieses ebenfalls von der App aus zu betrachten.

Ergebnis

Die App wird nach Abschluss des Projektes an das Therapiezentrum Aist geliefert. Eine kurze Einschulung anhand der Bedienungsunterlagen wird von der Entwicklergruppe durchgeführt werden. Den benötigten Server, eine NAS-Festplatte der Firma Synology, hat unser Auftraggeber, Physiotherapeut Ronald Voglhofer, gratis angeboten. Herr Voglhofer wird seinen Kunden die App zur Verfügung stellen.

Abstract

Task

Nowadays, many physiotherapists need a lot of effort to create a training plan for their clients. The physiotherapists, including our client, often create the training plans in simple word processing programs. They usually just copy the required exercises into a document and must ensure that the position of the images and the text is correct. These training plans are then printed out in a non-uniform format and taken home by the clients. In order to solve this problem, the project team has already created a simple desktop-client as a project work of the previous year, in which the physiotherapist can create his training plans with simple mouse clicks and they are finally saved as a PDF. Based on this program, during this diploma thesis a mobile application was developed to enable the therapist's clients to call up their training plans on their smartphones and process the exercises at home.

Realization

The mobile application is implemented as a platform-independent HTML5 app based on the Angular framework in combination with the Ionic Framework, which makes it possible to use the app on any smartphone. Moreover, for each exercise the user will be presented with a computer animation created with the free software Blender. This allows the customer to see exactly how he has to perform an exercise. In addition, the physiotherapist should have the opportunity to use the app to look at the progress of his clients and to get feedback from them.

Result

The app will be delivered to the Therapy Center Aist after the completion of the project. A short training based on the user manuals for our software will be carried out by the developer group. Our client, physiotherapist Ronald Voglhofer, offered the required server, a Synology NAS hard disk, free of charge. Mr. Voglhofer will provide the app for his clients.

Inhalt

Inhalt	7
1 Impressum	14
1.1 Schule	14
1.2 Schuljahr	14
1.3 Klasse	14
1.4 Projektname	14
1.5 Projektleiter	14
1.6 Projektteam	15
1.7 Betreuungslehrer	15
2 Einleitung	16
2.1 Ist-Zustand	16
2.2 Motivation	16
2.3 Ziel der Arbeit	17
2.4 Geschäftsziele	17
3 Projektdefinition	18
3.1 Projektumfang	18
3.2 Meilensteine	18
3.3 Projektstrukturplan	19
3.4 Organisation	20
4 Entstehung und Planung	21
4.1 Ideenfindung	21
4.2 IVM – Matrix	21
5 Realisierung	22
5.1 Anwendungsfälle	22
5.2 Architektur	22

5.3	Backend	23
6	Datenbank	24
6.1	Datenbankmodell	24
6.2	Beschreibung Entitäten	25
7	Darstellung	28
7.1	Willkommensfenster	28
7.2	Loginfenster	29
7.3	Home als Benutzer	30
7.4	Home als Therapeut	30
7.5	Mein Training	31
7.6	Übungen	31
7.7	Detailansicht einer Übung	32
8	Technologien	34
8.1	Verwendete Technologien	34
8.1.1	Angular	34
8.1.2	Ionic	34
8.1.3	Warum wurde Ionic gewählt?	34
8.1.4	HTML	35
8.1.5	TypeScript	35
8.1.6	JavaScript	36
8.1.7	Node.js	36
8.1.8	MySQL	36
8.1.9	MySQL auf der Synology	37
8.1.10	Synology Server	40
8.2	Verwendete Entwicklungssysteme	42
8.2.1	Blender	42

8.2.2	3DS Max	42
8.2.3	Autodesk Maya	42
8.2.4	Technologievergleich der drei Systeme	43
8.2.4.1	Allgemeines	43
8.2.4.2	Vorgehensweise	43
8.2.4.3	Renderer	44
8.2.4.4	Warum wurde Blender gewählt?	44
8.2.5	Visual Studio	45
8.2.6	Visual Studio Code	45
8.2.7	Atom Editor	46
8.2.8	Ionic DevApp	46
8.2.9	Notepad++	46
8.2.10	Postman	47
8.3	XAMPP als Testsystem	47
8.3.1	Konfiguration	47
8.3.2	Virtueller Host	47
8.3.3	Konfiguration eines virtuellen Hosts	48
8.4	REST	48
8.4.1	Allgemein	48
8.4.2	Anfrage und Antwort	49
8.4.3	GET Anfrage	49
8.4.4	POST-Anfrage	50
8.4.5	PUT-Anfrage	51
8.4.6	DELETE-Anfrage	52
8.4.7	Requirements	53
8.4.8	Implementierung	53

8.4.8.1 Definition URIs	53
8.4.8.2 Verwendete HTTP-Methoden	53
8.4.8.3 Implementierung mit SlimFramework	54
8.4.9 Assoziationstabellen	55
8.4.10 Benötigte Ordnerstruktur	56
8.5 Verwendete Bibliotheken	57
8.5.1 OpenGL	57
8.6 Sonstige verwendete Software	57
8.6.1 VMware-Player	57
8.6.2 GitHub	57
9 Implementierung	58
9.1 Computeranimationen der Übungen	58
9.1.1 Erstellen von 3D-Modellen	58
9.1.1.1 Blender	58
9.1.1.2 Beispiel Hantelbank	59
9.1.1.2.1 Vorgehensweise	59
9.1.1.3 3DS-Max	60
9.1.1.4 Maya	61
9.1.2 Erstellen eines Skeletts	61
9.1.2.1 Blender	62
9.1.2.1.1 Vorgehensweise	62
9.1.2.2 3DS-Max	65
9.1.2.2.1 Vorgehensweise	65
9.1.2.3 Maya	66
9.1.3 Animation von 3D-Objekten	66
9.1.3.1 Blender	66

9.1.3.2 Beispiel Abduktoren in Seitenlage	66
9.1.3.3 3DS-Max	67
9.1.3.4 Maya	67
9.1.4 Korrektur von Verformungen	68
9.1.4.1 Blender	68
9.1.4.2 3DS-Max	68
9.1.4.3 Maya	69
9.1.5 Kamera- und Lichtsetup	69
9.1.5.1 Allgemein	69
9.1.5.2 Blender	70
9.1.6 Render Engines in Blender	71
9.1.6.1 Blender Render	71
9.1.6.2 Cycles Render	71
9.1.7 Render-Setup	72
9.1.7.1 Allgemein	72
9.1.7.2 Blender	72
9.1.8 Integrieren der Animationen in die App	74
9.1.9 Animationen mit Motion Capture	75
9.2 App	77
9.2.1 Installation von Angular und Ionic	77
9.2.2 Anbindung des REST-Webservice in Angular	78
9.2.2.1 CORS Anywhere	78
9.2.2.2 Zugriff in Angular in der Methode ngOnInit	78
9.2.3 Login	80
9.2.3.1 SHA512	81
9.2.4 Bilderübertragung mittels REST-Webservice in Angular	81

9.2.5	Erstellung eines Timers in Angular	82
9.2.5.1	Observable-Timer	82
9.2.6	Verwendete Ionic-Komponenten	83
9.2.6.1	Ion-List	83
9.2.6.2	Ion-Toolbar	84
9.2.6.3	Ion-Thumbnail	84
9.2.6.4	Ion-Label	84
9.2.6.5	Ion-Card	85
9.2.6.6	Ion-Alert	85
9.2.6.7	Ion-Button	87
9.2.6.8	Ion-Icon	87
9.2.7	Ionic-Menu Component	87
9.3	Feedbackverwaltung in der App	90
9.3.1	Allgemein	90
9.3.2	Planung Feedback	90
9.3.3	Datenbank	91
9.3.4	Visualisierung Menü	92
9.3.5	Visualisierung Feedbackverwaltung Kunde	93
9.3.6	Visualisierung Feedback Therapeut	96
9.3.6.1	Allgemein	96
9.3.6.2	Fragenverwaltung	96
9.3.6.3	Feedbackverwaltung	98
9.3.7	Feedbackstatistiken	99
9.3.7.1	Diagramme	100
9.3.7.2	Visualisierung der Diagramme	100
9.3.7.3	Visualisierung Fragenart 3	101

10	Zusammenfassung	102
10.1	Testen	102
10.1.1	Testen der App	102
10.1.2	Testen der Computeranimationen	102
10.2	Probleme und Lösungen	103
10.3	Resümee	103
10.4	Zukunft	104
11	Literatur- und Quellenverzeichnis	105
11.1	Abbildungsverzeichnis	105
11.2	Literaturverzeichnis	108
12	Im Anhang	110
12.1	Lebenslauf Franz-Filip Schörghuber	110
12.2	Lebenslauf Michael Köck	111
12.3	Lebenslauf Gregor Arbeithuber	112

1 Impressum

1.1 Schule

Das Projekt wurde im Rahmen der Abschlussarbeit (Diplomarbeit) an der HTBLA Perg erstellt.

HTBLA Perg

Machlandstraße 48

4320 Perg

Tel. 0 72 62 / 539 26

Fax. 0 72 62 / 539 26 - 6

Schulkennzahl: 411457



Abbildung 1: HTL-Perg (fcp.at, 2017)

1.2 Schuljahr

2017/2018

1.3 Klasse

5AHIF

1.4 Projektname

GetFitAgain

1.5 Projektleiter

Franz-Filip Schörghuber

1.6 Projektteam

Gregor Arbeithuber

Michael Köck

Franz-Filip Schörghuber

1.7 Betreuungslehrer

Als Mobile-Computing-Professor und Professor im Fach Technische Informatik steht uns Herr Professor Dipl.-Ing. Michael Stumpfl als Betreuungslehrer mit Rat und Tat zur Seite.

2 Einleitung

2.1 Ist-Zustand

Im Rahmen der Projektarbeit der 4. Klasse wurde bereits eine Desktop-Anwendung zur interaktiven Erstellung von Trainingsplänen erstellt. Außerdem hat der Physiotherapeut damit die Möglichkeit, seine Kunden und Übungen zu verwalten, also neue hinzuzufügen und bestehende zu ändern oder zu löschen.

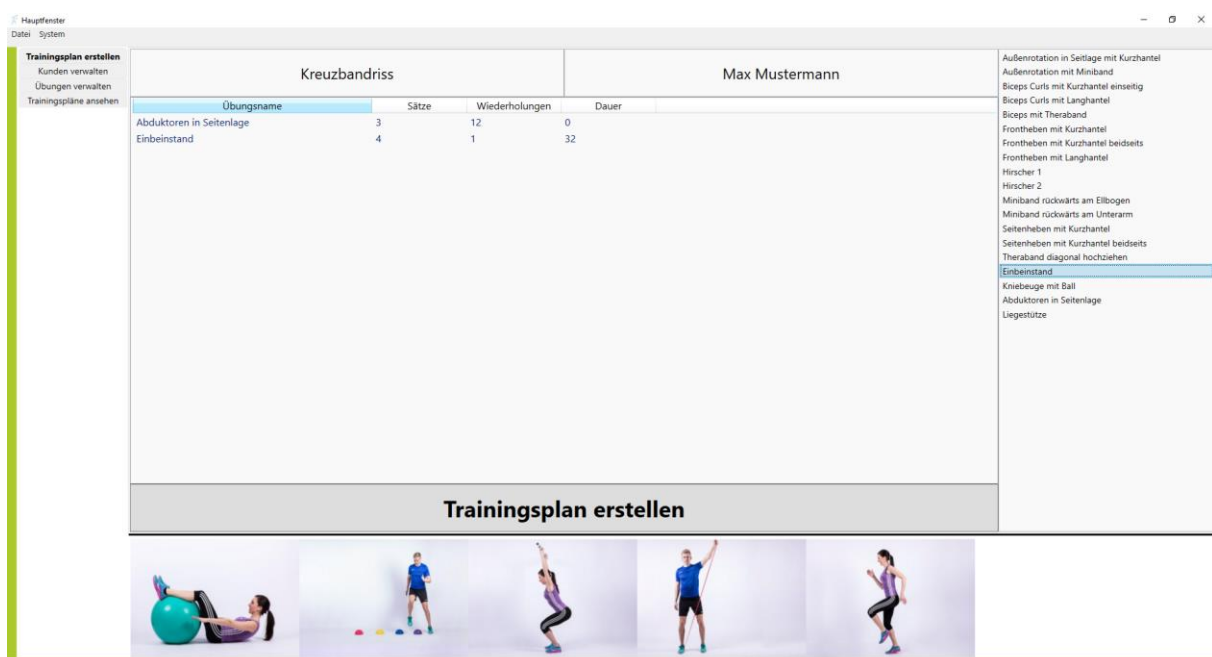


Abbildung 2: Desktop-Anwendung (Ist-Zustand)

Durch unsere Anwendung hat der Physiotherapeut die Möglichkeit, seine erstellten Trainingspläne als PDF zu exportieren und auszudrucken. Diese Trainingspläne gibt er daraufhin seinen Kunden mit, damit diese zu Hause weiter trainieren können.

2.2 Motivation

Da wir im Rahmen des Regelunterrichts leider keine Erfahrungen mit dem HTML5-Framework und Computeranimationen machen konnten, sahen wir diese Arbeit als Möglichkeit, uns damit intensiver auseinander zu setzen. Da diese Technologien in der Zukunft im-

mer wichtiger werden und wir später im Berufsleben mit hoher Wahrscheinlichkeit damit zu tun haben werden, haben wir darin eine große Chance gesehen.

Die Zusammenarbeit mit einem Auftraggeber und die daraus entstehenden Herausforderungen bringen außerdem zusätzliche Erfahrung für unsere Zukunft im Berufsleben.

2.3 Ziel der Arbeit

Das Ziel der Diplomarbeit ist die Erstellung einer plattformunabhängigen mobilen App auf Basis aktueller Technologien. Durch die App wird es für die Kunden des Physiotherapeuten leichter, ihre Übungen durchzuführen, weil sie zu jeder Übung – zusätzlich zur Beschreibung einer Übung – eine Computeranimation angezeigt bekommen, in der genau zu sehen ist, wie die derzeitige Übung durchzuführen ist. Ebenso soll eine engere Bindung zwischen dem Physiotherapeuten und seinen Kunden ermöglicht werden, da sie in der App die Möglichkeit bekommen, dem Physiotherapeuten ein Feedback darüber zu geben, ob sich ihre Beschwerden durch das Training verbessert haben.

2.4 Geschäftsziele

- ✓ Werbewirkung für den Physiotherapeuten
- ✓ Kontaktierungsmöglichkeit des Physiotherapeuten durch Feedback
- ✓ Einfache Erklärung der Übungen durch Computeranimationen
- ✓ Zeitersparnis für den Therapeuten

3 Projektdefinition

3.1 Projektumfang

Der Projektumfang beinhaltet die Erstellung einer plattformunabhängigen HTML5-Anwendung, in der der Kunde seine Trainingspläne abrufen kann. Dabei soll ihm zu jeder Übung die Beschreibung, die Anzahl an durchzuführenden Sätzen, Wiederholungen und eventuell einer Dauer und eine Computeranimation angezeigt werden. Dabei kann der Kunde mit der App durch einen wiederholten Klick auf einen Button genau mitzählen, wie oft er die Übung noch durchführen muss. In einem weiteren Menüpunkt kann der Kunde dem Therapeuten ein Feedback über eine Besserung seiner Beschwerden geben, dieses Feedback kann der Physiotherapeut in seiner Administrationsoberfläche der App ansehen.

3.2 Meilensteine

Meilenstein	Soll-Termin	Ist-Termin	Projektmitglieder
Projektstart	20.06.2017	20.06.2017	alle
Projektinitialisierung abgeschlossen	10.09.2017	07.09.2017	alle
Oberfläche der App	18.02.2018	25.02.2018	Michael Köck
Computeranimationen erstellen	18.02.2018	20.02.2018	Gregor Arbeithuber
Auswertung von Nutzerdaten	18.02.2018	19.02.2018	Franz-Filip Schörghuber
Funktionalität für Kunden bereitstellen	27.02.2018	25.02.2018	Michael Köck
Integrierung der Animationen in die App	27.02.2018	24.02.2018	Gregor Arbeithuber
Integrierung der Nutzerdaten in die App (Statistiken)	27.02.2018	24.02.2018	Franz-Filip Schörghuber
Rendern der Animationen	15.03.2018	15.03.2018	Gregor Arbeithuber
Zusammenführen der einzelnen Teile der App	05.03.2018	02.03.2018	Michael Köck
Diplomarbeitsschrift fertig	01.04.2018	03.04.2018	alle
Diplomarbeit abgeschlossen	05.04.2018	05.04.2018	alle

3.3 Projektstrukturplan



Abbildung 3: Projektstrukturplan

3.4 Organisation

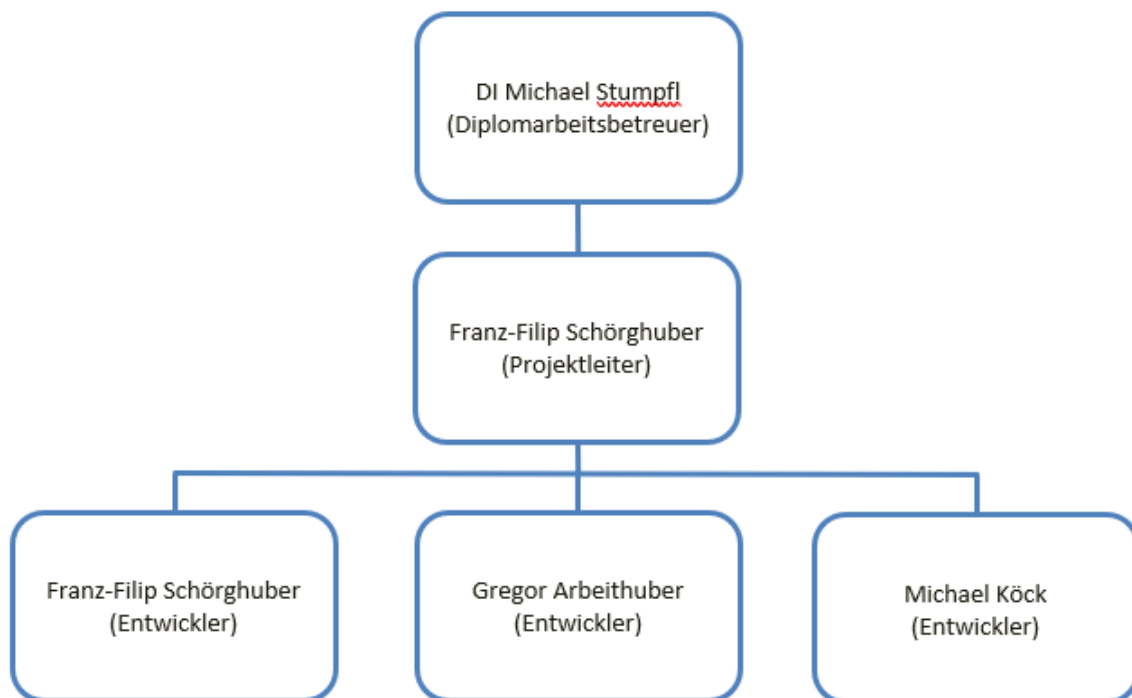


Abbildung 4: Projekthierarchie

4 Entstehung und Planung

4.1 Ideenfindung

Nachdem wir als Projektarbeit des Vorjahres für unseren Auftraggeber eine Software zur Kundenverwaltung und zum Ausdrucken seiner Trainingspläne erstellt haben, sind wir gemeinsam auf die Idee gekommen, dass im 21. Jahrhundert bedruckte Zettel etwas aus der Mode gekommen sind. Somit stand fest: Wir programmieren eine App mit Computeranimationen, die Cross-Plattform fähig ist und den Kunden die Möglichkeit gibt, dem Physiotherapeuten Feedback darüber zu geben, ob sich ihre Beschwerden durch das Training verbessern.

4.2 IVM – Matrix

Die IVM Matrix ermöglicht einen Überblick über verschiedene Teile des Projektes. Dabei wird zwischen Verantwortlichen (V), Mitwirkenden (M) und den Informierten (I) der jeweiligen Teilbereiche unterschieden.

	Franz-Filip Schörghuber	Michael Köck	Gregor Arbeithuber	DI Michael Stumpfl	Ronald Voglhofer
Computeranimationen	I	I	V	I	I
HTML5-App	I	V	I	I	I
Nutzerdatenauswertung/Statistiken	V	I	I	I	I
Diplomschrift	V	V	V	I	I

I ... informiert

M ... mitwirkend

V ... verantwortlich

5 Realisierung

5.1 Anwendungsfälle

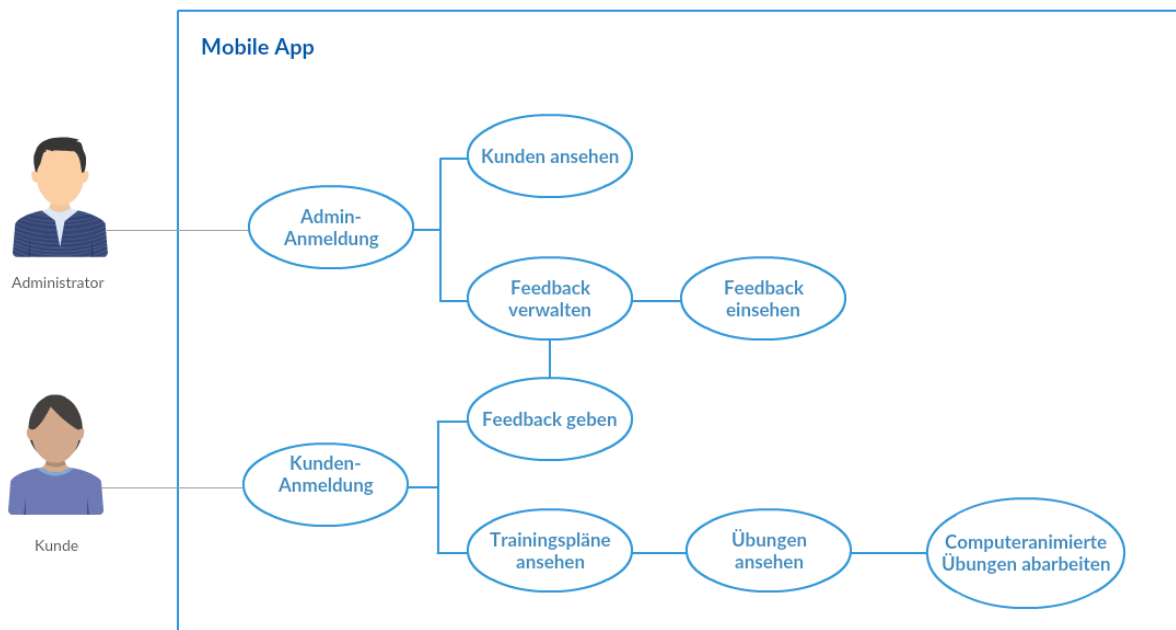


Abbildung 5: Use-Case

5.2 Architektur

Das Zusammenspiel unseres Projekts der 4. Klasse und der Diplomarbeit ist in folgender Grafik ersichtlich:

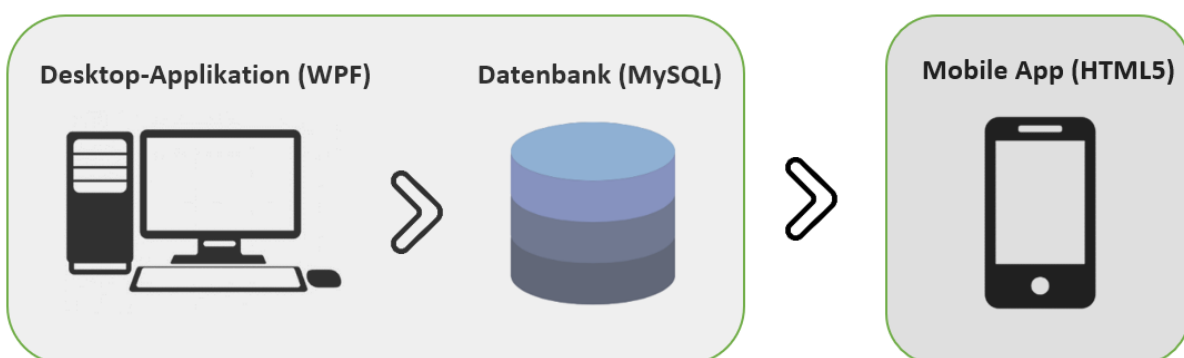


Abbildung 6: Architektur

Mit der Desktop-Anwendung des Therapeuten werden somit die Trainingspläne erstellt und mithilfe des REST-Webservice in die Datenbank gespeichert. Die Datenbank liegt auf dem

NAS des Physiotherapeuten, darauf wird mit der App über das Internet zugegriffen. Nachdem man sich also in die App eingeloggt hat, werden alle zugeteilten Trainingspläne von der Datenbank geladen und angezeigt.

5.3 Backend

Die Anforderungen des Physiotherapeuten waren eindeutig: Es sollten möglichst wenig Kosten anfallen. Um dies umzusetzen, sollte der bereits vorhandene Synology NAS in das Projekt eingebunden werden. Somit wurde bereits einiges an Planungszeit eingespart, da der Server eine Datenbank und Web-Inhalte hosten kann. Weiters wurde als Datenbank die MySQL-Datenbank gewählt, da sie direkt auf der Synology durch ein Plugin zu installieren ist und noch dazu völlig kostenlos ist. Durch die Performance des Servers ist es nicht möglich, einen WCF-Service oder Ähnliches auf der Synology zu hosten. Deswegen wurde für die Schnittstelle auf das PHP Microframework „SlimFramework“ zurückgegriffen, welches sich durch seinen schmalen Footprint auszeichnet. Denn das Framework benötigt weder viel Speicherplatz auf der Festplatte, noch viel RAM oder CPU Leistung.

6 Datenbank

6.1 Datenbankmodell

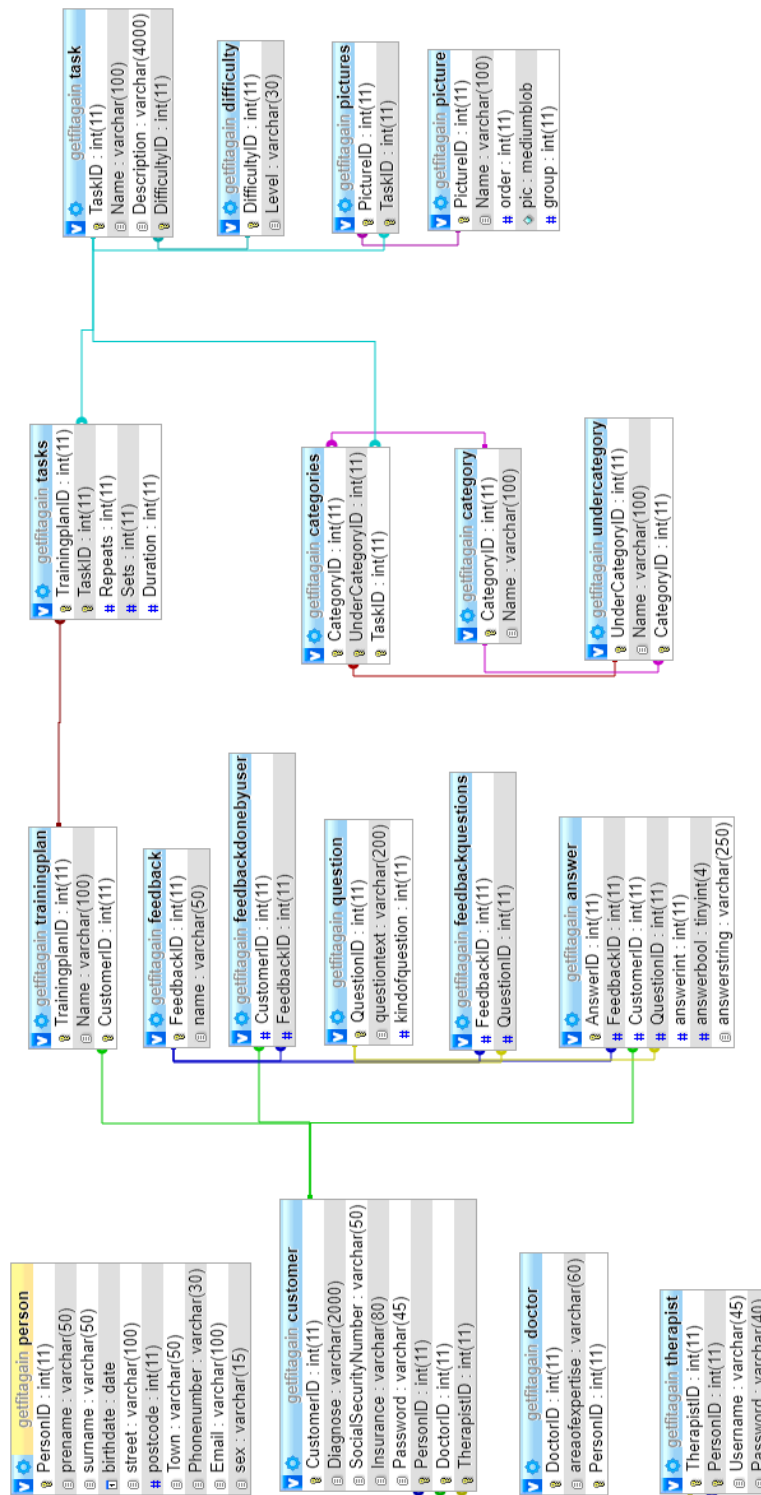


Abbildung 7: Datenbankmodell

6.2 Beschreibung Entitäten

Tabelle	PK	FK	Attribute	Typ	Bedeutung
Person	PersonID	-	firstname	Varchar	Vorname einer Person
			Surname	Varchar	Nachname einer Person
			Birthdate	Date	Geburtsdatum einer Person
			Street	Varchar	Straße in der die Person lebt
			Postcode	Int	PLZ der Stadt in der die Person lebt
			Town	Varchar	Stadt in der die Person lebt
			Phonenumber	Varchar	Telefonnummer der Person
			Email	Varchar	Emailadresse der Person
			Sex	Varchar	Geschlecht
Doctor	DoctorID	PersonID	Areaofexpertise	Varchar	Fachgebiet
Therapist	TherapistID	PersonID	Username	Varchar	Username für den Therapeuten zum Anmelden
			Password	Varchar	Passwort für den Therapeuten zum anmelden
Customer	CustomerID	PersonID DoctorID TherapistID	Diagnose	Varchar	Diagnose des Customers
			Socialsecurity-Number	Varchar	SVN des Customers
			Insurance	Varchar	Versicherung des Customers
			Password	Varchar	Passwort für die Anmeldung
Trainingplan	TrainingplanID	CustomerID	Name	Varchar	Name des Trainingsplans
Difficulty	DifficultyID	-	Level	Varchar	Schwierigkeits-Grad

Tabelle	PK	FK	Attribute	Typ	Bedeutung
Task	TaskID	DifficultyID	Description	Varchar	Beschreibung der Übung
Tasks	Assoziations-Tabelle	TrainingplanID TaskID	Repeats	Int	Anzahl Wiederholungen pro Übung
			Sets	Int	Anzahl der Sets pro Übung
			Duration	int	Wielange man die Übung durchführen soll
Picture	PictureID	-	Name	Varchar	Name des Bildes
			Order	Int	Reihenfolge des Bildes innerhalb der Bildgruppe
			Pic	BLOB	Das Bild zu einem Base64 String konvertiert
			Group	int	Zu welcher Bildgruppe das Bild gehört
Pictures	Assoziations-tabelle	PictureID TaskID	-	-	-
			-	-	-
Category	CategoryID	-	Name	Varchar	Name der Kategorie
Undercategory	UnderCategoryID	CategoryID	Name	Varchar	Name der Unterkategorie
Categories	Assoziations-Tabelle	CategoryID UndercategoryID TaskID	-	-	-
			-	-	-
			-	-	-
Feedback	FeedbackID	-	Name	Varchar	Name des Feedbacks
Feedbackdonebyuser	Assoziations-tabelle	CustomerID FeedbackID	-	-	-
			-	-	-
Question	QuestionID	-	Questiontext	Varchar	Text der Frage
			kindfrage	Int	Art der Frage
Feedbackquestions	Assoziations-tabelle	FeedbackID QuestionID	-	-	-
			-	-	-

Tabelle	PK	FK	Attribute	Typ	Bedeutung
Answer	AnswerID	FeedbackID CustomerID QuestionID	Answerint	Int	Antwort für Antwortmöglichkeit 1
			Answerbool	Tinyint	Antwort für Antwortmöglichkeit 2
			answerstring	varchar	Antwort für Antwortmöglichkeit 1

7 Darstellung

7.1 Willkommensfenster

Diese Seite erscheint sobald man die App startet und noch nicht eingeloggt ist. Damit man unsere Therapie-App überhaupt aktiv richtig benutzen kann, muss man sich zu allererst einloggen. Dies erfolgt mit einem Klick auf den Login-Button, nachdem man die App gestartet hat. Mit einem Klick auf den Signup-Button gelangt man auf die Signup-Seite, diese wird im nächsten Kapitel erläutert.

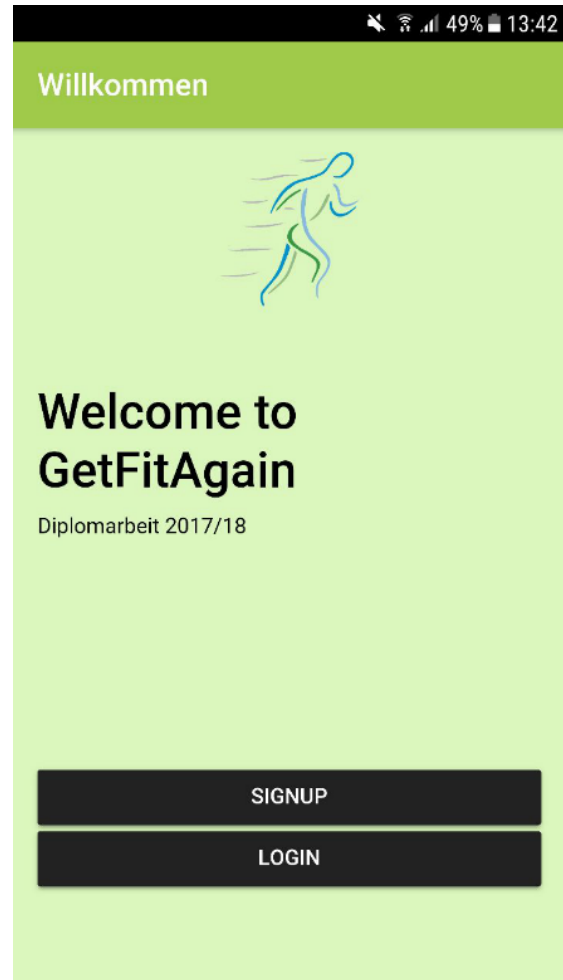


Abbildung 8: Willkommensfenster der App

7.2 Loginfenster

In diesem Fenster wird der Benutzer aufgefordert, dass er seine Emailadresse und sein Passwort eingibt. Jeder Benutzer kann bei seinem Besuch beim Physiotherapeuten ein Passwort festlegen, welches es zu benutzen gilt. Als Emailadresse wird jene verwendet, die der Physiotherapeut in seiner Datenbank gespeichert hat. Als Therapeut verwendet man die gleichen Logindaten wie beim Desktopprogramm. Ist der Login nicht erfolgreich, erscheint ein Pop-up, welches den Benutzer darauf hinweist, seine Daten erneut einzugeben. Hat der Benutzer seine Emailadresse oder sein Passwort vergessen, kann er auf den Button „Logindaten vergessen?“ klicken und er wird zur oben genannten Signup-Seite weitergeleitet. In diesem Fenster (siehe Abbildung 9) wird dem Benutzer erklärt, wie er Zugangsdaten erhalten kann bzw. wann und wo er das Therapiezentrum-Aist erreichen kann um zum Beispiel das Passwort zu ändern.

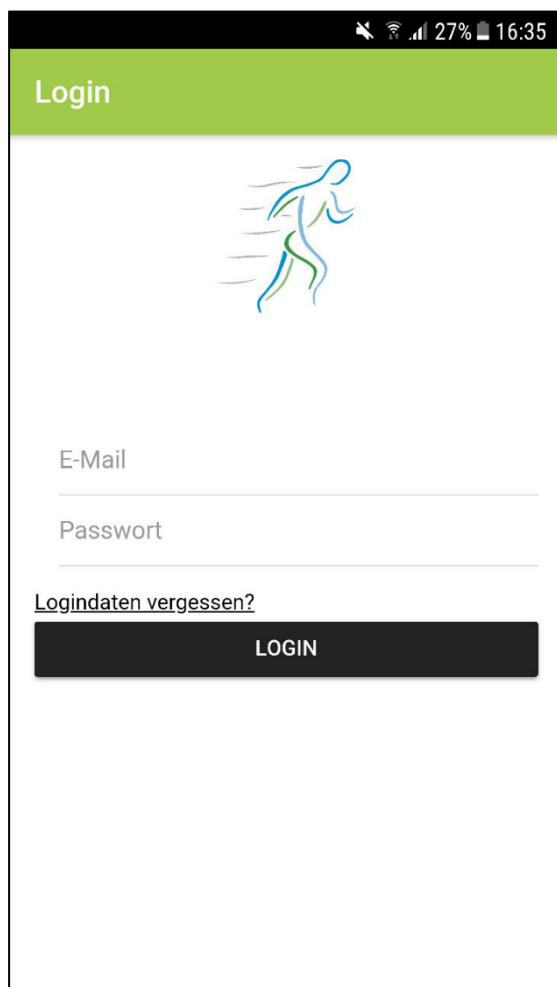


Abbildung 10: Loginfenster App



Abbildung 9: Signup-Seite App

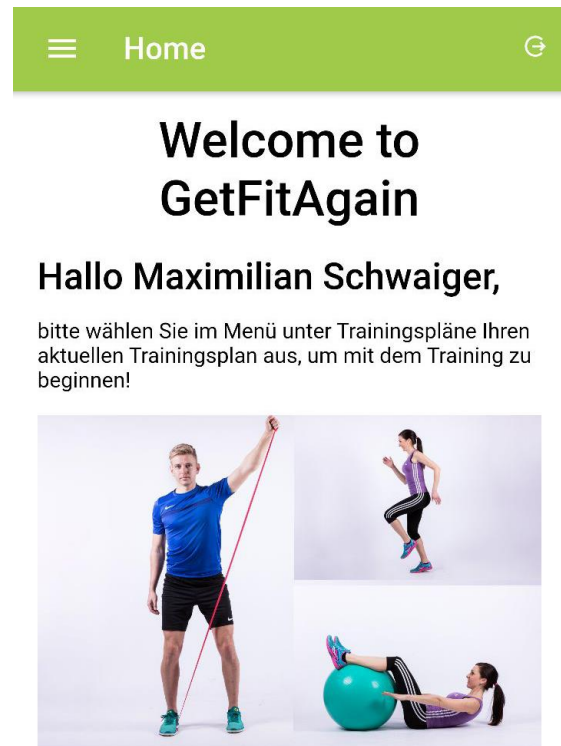
7.3 Home als Benutzer

Ist der Login erfolgreich, gelangt man zum Homescreen der App, hier wird jeder Benutzer mit seinem Namen begrüßt, weiters findet er unterhalb des Bildes die Telefonnummer, die Emailadresse und weitere Kontaktdaten zu seinem persönlichen Therapeuten. Ist der Benutzer ein Therapeut, wird er nur begrüßt und erhält eine kleine Anleitung, wie er die App bedienen kann. Links oben findet jeder eingeloggte Benutzer einen Menüknopf, dieser öffnet ein Seitenleistenmenü mit den jeweiligen Unterpunkten.

Rechts oben kann man sich wieder ausloggen und man gelangt wieder zum Willkommensfenster.

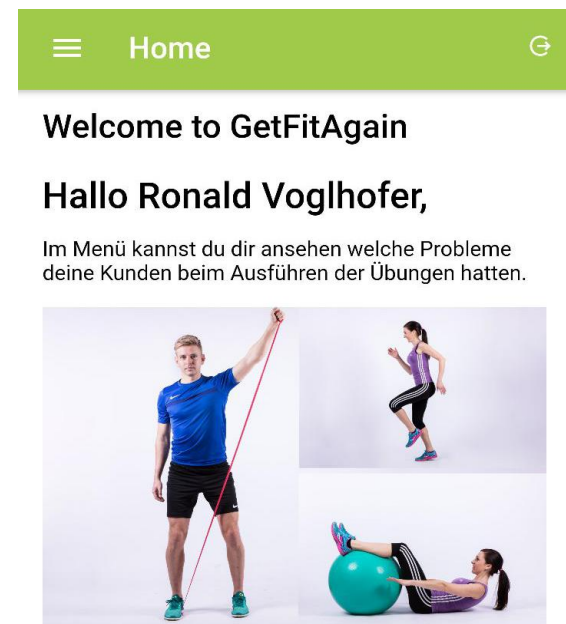
7.4 Home als Therapeut

Neben dem Endanwender kann sich natürlich auch der Therapeut bei der App anmelden. Dieser meldet sich jedoch mit denselben Login-Daten an, die er auch verwendet, wenn er sich bei der Desktop-Anwendung anmeldet. Der Therapeut hat natürlich ein komplett anderes Menü bzw. einen anderen Home-Screen. Zum Beispiel wird der Therapeut darauf hingewiesen, was er im Menü mit der App machen kann, unter anderem kann er Feedback auswerten und verwalten.



Ihr zuständiger Therapeut ist Ronald Voglhofer, bei Fragen ist er unter 0723626701 bzw. voglhofer@therapiezentrum-aist.at erreichbar.

Abbildung 11: Homescreen-Kunde App



Klicke im Menü auf Fragen verwalten um neue Fragen zu entwickeln

Abbildung 12: Homescreen-Therapeut

7.5 Mein Training

Klickt man nun im Menü auf den Button „Mein Training“, gelangt man zu einer Seite, in der alle bisherigen Trainingspläne aufgelistet sind, die man je beim Physiotherapeuten erhalten hat. Der letzte Trainingsplan der Liste ist immer der Aktuellste. Unter Abbildung 13 wäre dies „Wadenzerrung“, durch Anklicken des gewünschten Trainingsplans gelangt man in eine detailliertere Ansicht (siehe Kapitel 7.6).

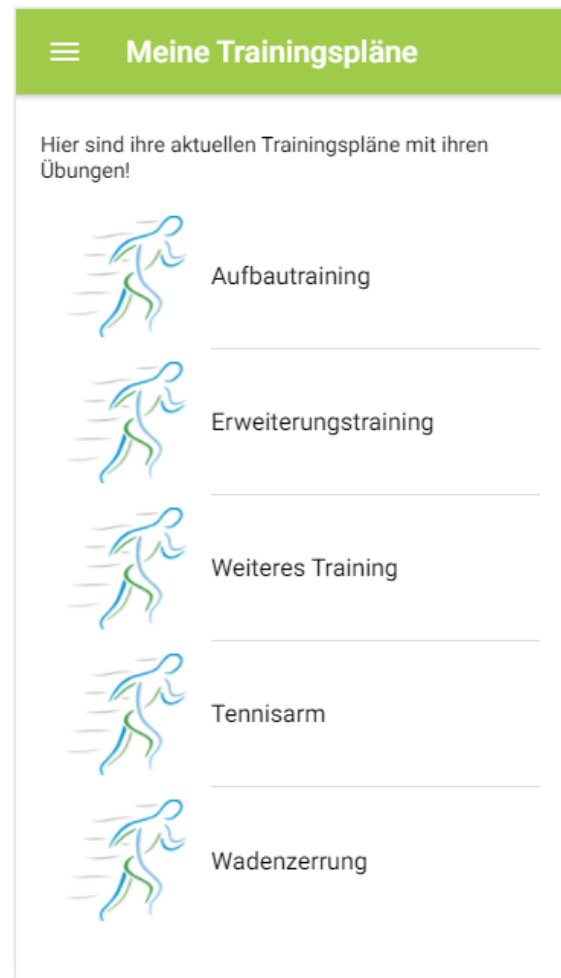


Abbildung 13: Meine Trainingspläne App

7.6 Übungen

Die nächste Seite ist eine weitere Auflistung, diesmal der einzelnen Übungen des vorher ausgewählten Trainingsplans, mit einer kleinen Beschreibung. Diese Auflistung dient hauptsächlich dazu, die App übersichtlich zu halten, da es ansonsten viel zu viele Informationen auf einer Seite geben würde und man sich erdrückt fühlen würde. Klickt man nun auf eine einzelne Übung, gelangt man in eine Detailansicht (siehe Kapitel 7.7).

7.7 Detailansicht einer Übung

In der Detailansicht angelangt, findet man alle Parameter der Übung vor. Wie lange muss der Benutzer die Übung durchführen? Wie oft muss er sie wiederholen? Wie viele Sätze an Wiederholungen muss er machen. Diese Fragen werden in diesem Fenster beantwortet. Weiters gibt es in dieser Detailansicht einen Button „Animationen ein-/ausblenden“. Mit einem Klick auf diesen Button öffnet sich eine im MPEG-4-Kompressionsformat hinterlegte Computera-
nimation.

Diese Animation soll dem Benutzer dabei helfen, die Übung zu verstehen. Durch einen weiteren Klick auf den Button schließt sich die Animation wieder. Der Button „Starte Übung“ startet, wie der Name schon sagt, die Übung. Im Konkreten hat eine Übung eine Zeitangabe, fängt ein Zähler an im Sekundentakt runter zu zählen, erreicht er schließlich den Wert 0, erscheint ein Pop-Up, welches den Benutzer darauf hinweist, dass er eine Wiederholung abgeschlossen hat und sich der Zähler bei verbleibende Wiederholungen um 1 verringert hat. Er kann nun mit einem erneuten Klick die nächste Wiederholung abarbeiten. Erreicht der Benutzer 0 verbleibende Wiederholungen, wird der Zähler der Sätze um 1 verringert. Bei 0 Sätzen wird dem Benutzer mitgeteilt, er habe nun die gesamte Übung abgeschlossen und er soll die nächste Übung ein Fenster weiter zurück beginnen.

Hat man jedoch keine Dauer angegeben, wird sofort eine Wiederholung runter gezählt, dies ist zum Beispiel der Fall, wenn man Kniebeugen oder ähnliche Übungen macht, die nicht auf Dauer ausgerichtet sind. Hierbei hat man den Vorteil, dass man nicht mitzählen muss.

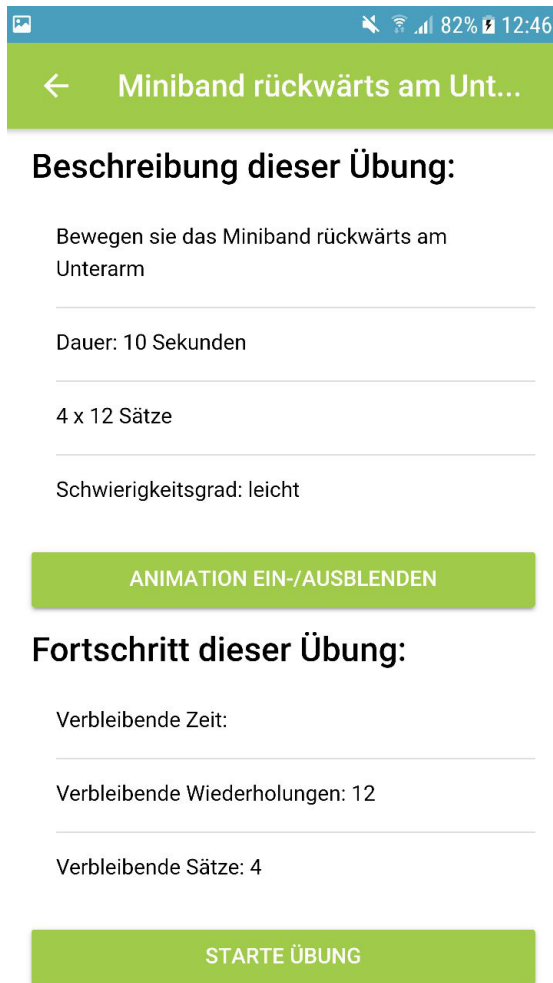


Abbildung 14: Detailansicht einer Übung



Abbildung 15: Detailansicht einer Übung mit Animation

8 Technologien

8.1 Verwendete Technologien

8.1.1 Angular

Angular ist ein von Google entwickeltes, sehr umfangreiches, clientseitiges JavaScript-Framework. Das Framework basiert auf der von Microsoft entwickelten Programmiersprache TypeScript und wird meist zur Entwicklung von Front-End-Webapplikationen verwendet. Angular 2 ist der Nachfolger von AngularJS (Version 1) und unterscheidet sich maßgeblich von diesem, unter anderem durch dynamisches Laden. Weiters haben die Anforderungen von mobilen Plattformen eine erhöhte Priorität (Mobile-First-Ansatz). (vgl. [wikipedia.org](https://de.wikipedia.org), 2018)



Abbildung 16: Angular-Logo
(de.wikipedia.org, 2018)

8.1.2 Ionic

Das Open-Source-Framework Ionic wird zur Erstellung von Hybrid-Apps verwendet. Es setzt auf das JavaScript-Framework Angular auf. Da der Fokus von Ionic auf Front-Ends liegt, stellt das Framework zahlreiche Komponenten einer Programmbibliothek zur Verfügung. Ionic-Hybrid-Apps kann man außerdem über den Google Play Store bzw. über den Apple App Store vertreiben. Das Framework wird aktuell von Drifty Co weiterentwickelt. (vgl. [wikipedia.org](https://de.wikipedia.org), 2018)



Abbildung 17: Ionic-Logo
(de.wikipedia.org, 2018)

8.1.3 Warum wurde Ionic gewählt?

Das Framework Ionic wurde gewählt, weil es der App ein natives Gefühl verleiht. Die Darstellung der einzelnen Komponenten passen sich immer an das gerade verwendete Betriebssystem an. Durch die Kombination mit Angular werden auch dessen Vorteile eingebunden, unter anderem hat man mit Angular die Möglichkeit, HTML-Tags um Direktiven zu erweitern, welche die Lesbarkeit des Quellcodes um ein Vielfaches erhöht. (vgl. [workline](https://workline.com), 2017)

8.1.4 HTML

Die textbasierte Ausgangssprache, Hypertext Markup Language (HTML), dient zur semantischen Formatierung eines Textes. Die visuelle Darstellung wird vom Webbrowser übernommen und nicht wie angenommen durch die Sprache selbst. Aktuell befindet sich HTML in der Version 5.2 und wird von W3C (World Wide Web Consortium) weiterentwickelt. Fast alle HTML-Elemente werden von einem Start- und einem End-Tag markiert. (vgl. Wikipedia, 2018)

`<p>Ein Tag zur Erklärung</p>`



Abbildung 18: HTML-Logo
(wikipedia.org, 2018)

8.1.5 TypeScript

Die von Microsoft entwickelte Programmiersprache TypeScript ist eine Erweiterung von JavaScript, dies bedeutet auch, dass jeder JavaScript-Code gleichzeitig auch TypeScript-Code ist und im Endeffekt immer zu JavaScript kompiliert wird. TypeScript ergänzt JavaScript, wie der Name schon sagt, um ein Typensystem, somit werden viele Fehler bereits zur Kompilierungszeit auffällig, nicht erst zur Laufzeit. Die Programmiersprache TypeScript erschien im Jahr 2012 und befindet sich aktuell in der Version 2.7.1 (vgl. wikipedia.org, 2018)



Abbildung 19: TypeScript-Logo
(wikipedia.org, 2018)

Die App wurde mithilfe von TypeScript in Kombination mit Ionic und Angular entwickelt.

8.1.6 JavaScript

Die 1995 entwickelte dynamisch typisierte, objektorientierte, jedoch auch klassenlose Skriptsprache Javascript, ist vor allem für die dynamische HTML-Generierung bekannt. Es lassen sich unter anderem Benutzerinteraktionen auswerten oder Inhalte dynamisch verändern.

(vgl. wikipedia.org, 2018)



Abbildung 20: JS-Logo (wikipedia.org, 2018)

Diese Skriptsprache bietet die Basis für das von uns eingesetzte Framework Angular und der Programmiersprache TypeScript.

8.1.7 Node.js

Node.js bietet eine ressourcensparende Architektur und wird in der JavaScript-Laufzeitumgebung ausgeführt. Es ermöglicht eine große Anzahl an gleichzeitig bestehenden Netzwerkverbindungen, da es sich um den serverseitigen Teil der Softwareentwicklung kümmert. Node.js wird vor allem zur Entwicklung und Betrieb von Netzwerkanwendungen verwendet. (vgl. wikipedia.org, 2018)



Abbildung 21: Logo-Node (wikipedia.org, 2018)

8.1.8 MySQL

MySQL ist eine sehr weit verbreitetes relationales Datenbanksystem und ist sowohl als Open-Source-System als auch als kommerzielle Enterprise-Version erhältlich. In unserer Diplomarbeit verwenden wir die Open-Source-Version, da es für diese Zwecke ausreichend ist. MySQL arbeitet als klassisches Client-Server-System, in unserem Fall reprä-



Abbildung 22: MySQL-Logo (de.wikipedia.org, 2018)

sentiert es die Datenbank, auf die die REST-API bzw. die Webanwendung zugreift. Die REST-API übersetzt diese Anfragen schließlich in ausführbaren Code. (vgl. wikipedia.org, 2018)

8.1.9 MySQL auf der Synology

PhpMyAdmin wird benötigt, um die Datenbank auf der Synology zu verwalten. Über das Paketzentrum der Synology kann man das Verwaltungstool schnell und einfach installieren.

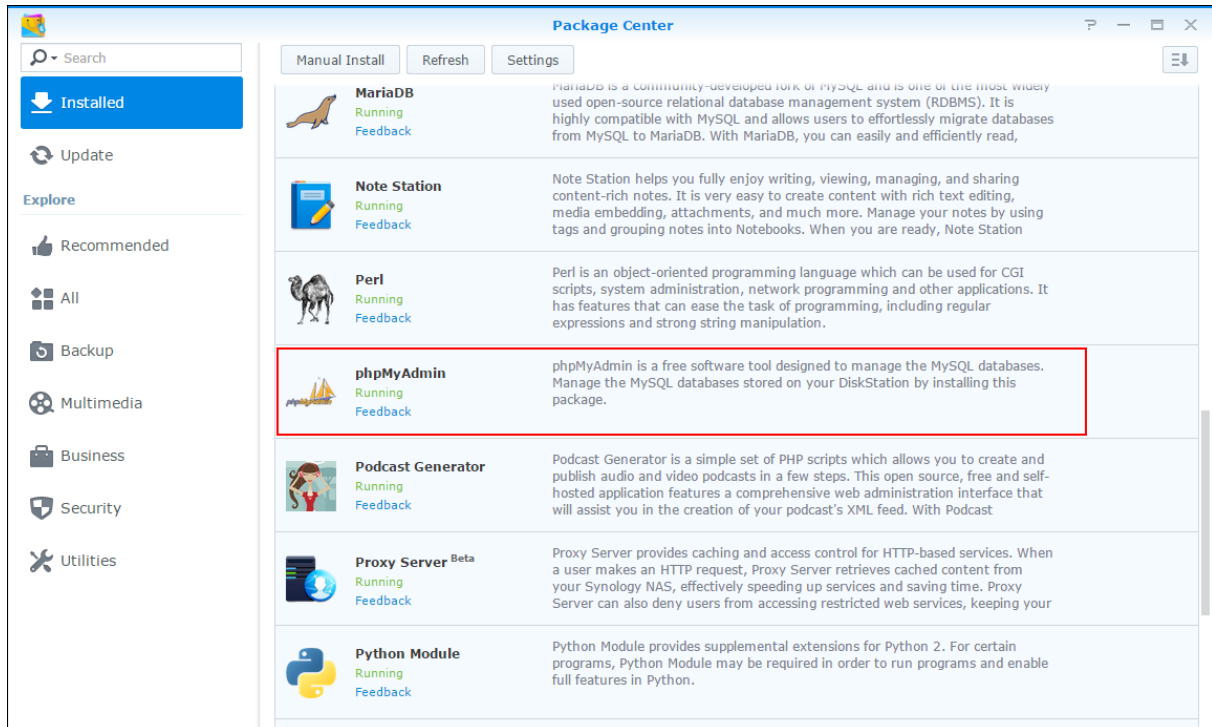


Abbildung 23 Installation phpMyAdmin

Danach können alle Funktionen der Datenbank verwendet werden:

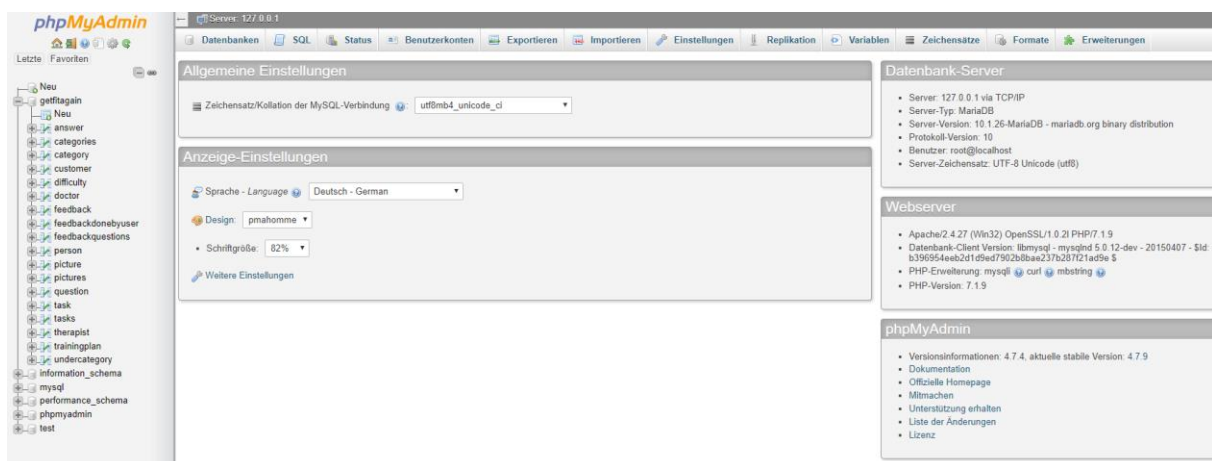


Abbildung 24 phpMyAdmin Menü

Im Rahmen der Entwicklung der Datenbank und der REST-API war es oft nötig, die Datenbank umzugestalten. Dabei war es aber auch nötig vorher Sicherungen zu erstellen und diese, falls nötig, später wieder einzuspielen. Mit der Option „Exportieren“ kann man sich die

gesamte Datenbank, inklusive Daten und Fremdschlüssel, in ein SQL-Skript exportieren lassen.

Abbildung 25 Exportieren der Datenbank

Als Beispiel wurde hier die Tabelle Doctor verwendet:

```
--
-- Tabellenstruktur für Tabelle `doctor`
--

CREATE TABLE `doctor` (
  `DoctorID` int(11) NOT NULL,
  `areaofexpertise` varchar(60) DEFAULT NULL,
  `PersonID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Daten für Tabelle `doctor`
--

INSERT INTO `doctor` (`DoctorID`, `areaofexpertise`, `PersonID`) VALUES
(1, 'Allgemeinmedizin', 11),
(2, 'Allgemeinmedizin', 12),
(3, 'Anästhesiologie', 13),
(4, 'Anästhesiologie', 14),
(5, 'Anatomie', 15),
(6, 'Anatomie', 16),
```

```
(7, 'Arbeitsmedizin', 18),  
(8, 'Arbeitsmedizin', 19),  
(9, 'Arbeitsmedizin', 20),  
(10, 'Arbeitsmedizin', 21),  
(11, 'Allistair J. Moss', 22),  
(12, 'Allgemeinmedizin', 23),  
(13, 'Allgemeinmedizin', 24),  
(14, 'Allgemeinmedizin', 25),  
(15, 'Allgemeinmedizin', 26);
```

Falls beim Entwickeln nun ein Fehler passiert, ist es möglich, durch dieses Skript die Datenbank wiederherzustellen. Um die Datenbank wiederherzustellen, muss das Skript wieder importiert werden:

The screenshot shows a database management interface with a menu bar containing 'Struktur', 'SQL', 'Suche', 'Abfrage', 'Exportieren', 'Importieren', and a help icon. Below the menu is a title bar 'Importiere in die Datenbank "getfitagain"'. The main content area is divided into sections:

- Zu importierende Datei:**
 - Text: 'Datei kann komprimiert (gzip, bzip2, zip) oder unkomprimiert sein. Der Dateiname einer komprimierten Datei muss mit .[Format].[Komprimierung] enden. Beispiel: .sql.zip'.
 - Text: 'Durchsuchen Sie Ihren Computer: Keine ausgewählt (Maximal: 2.048KiB)'.
 - Text: 'Sie können auch per Drag & Drop eine Datei auf einer beliebigen Seite legen.'
 - Text: 'Zeichencodierung der Datei:
- Teilweiser Import:**
 - ☒ 'Import abbrechen, wenn die maximale PHP-Scriptlaufzeit erreicht wird. (Damit ist es möglich, große I'.
 - Text: 'Diese Anzahl Abfragen (für SQL) überspringen, beginnend von der ersten:
- Andere Optionen:**
 - ☒ 'Fremdschlüsselüberprüfung aktivieren'
- Format:**
 -
- Formatspezifische Optionen:**
 - Text: 'SQL-Kompatibilitätsmodus:
 - ☒ 'AUTO_INCREMENT nicht für Nullwerte verwenden'

At the bottom left is an 'OK' button.

Abbildung 26 Importieren des Datenbankscripts

8.1.10 Synology Server

Um den Zugriff auf die Datenbank mit der API zu ermöglichen, ist es notwendig die zuvor lokal entwickelte API auf der Synology zu hosten. Dafür muss in der „Webstation“ der Synology in dem Reiter „Allgemeine Einstellungen“ ein geeigneter HTTP-Backend Server und die PHP-Version ausgewählt werden.

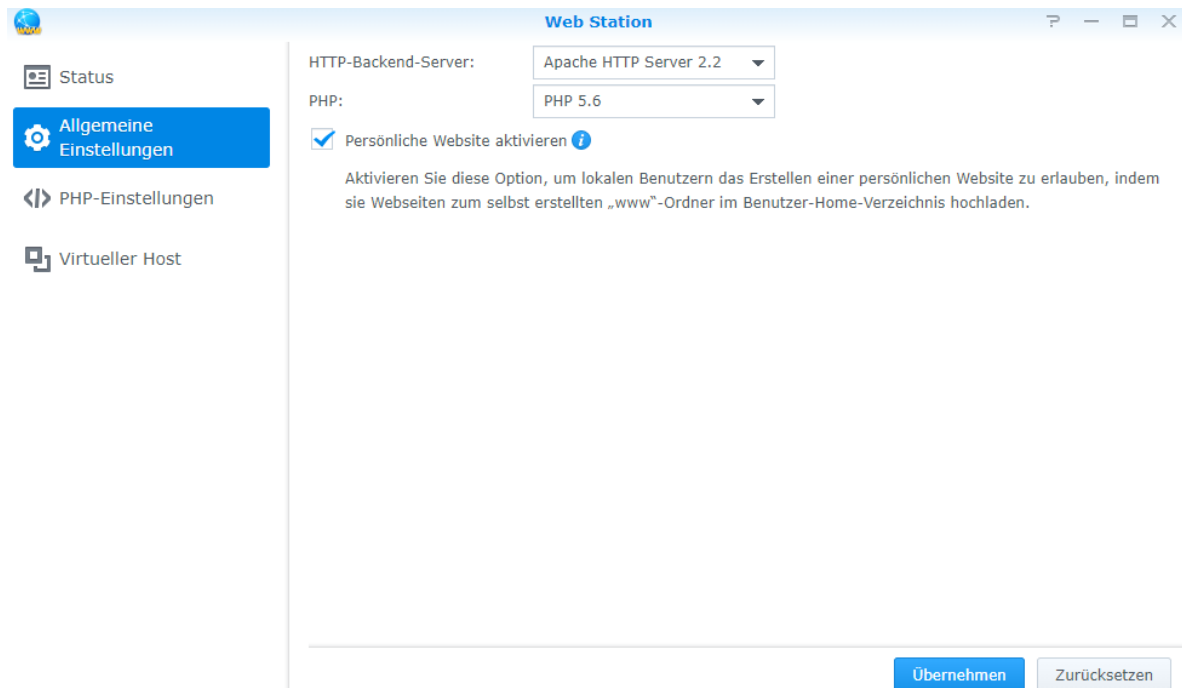


Abbildung 27 Synology Serverkonfiguration

Somit können jetzt im Ordner „web“ die benötigten Ressourcen gehostet werden. Die REST-API von GetFitAgain wird hier im Ordner „api“ gehostet.

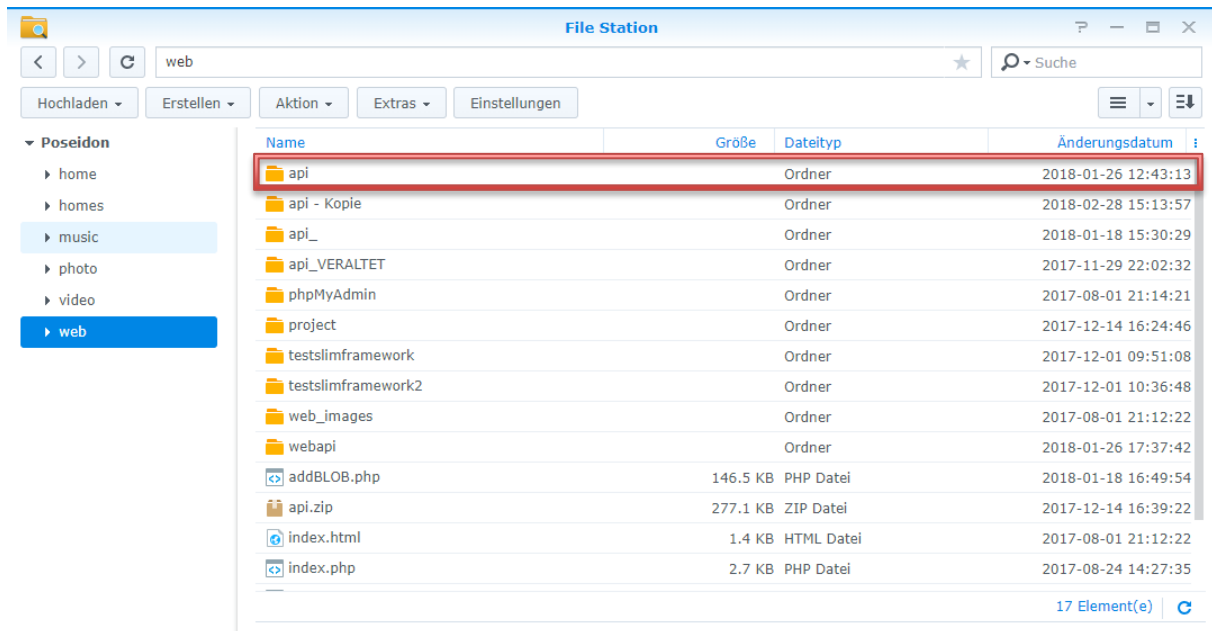


Abbildung 28 API im Filesystem

Um die API nun auch über das Internet zu erreichen ist es notwendig, der Synology einen DDNS zuzuweisen.

Synology bietet dafür aber bereits eine Möglichkeit. Der NAS muss lediglich mit einem Synologykonto verknüpft werden und danach kann man dem NAS eine „QuickConnectID“ zuweisen. Aus dieser wird dann der Hostname generiert:

<input type="checkbox"/>	Modellname	Seriennummer	QuickConnect ID ⓘ	DDNS-Hostname	IP ⓘ	Weitere Hilfsquellen
<input type="checkbox"/>	DS216+II	1740O6N949510	HOGWARTS99 🔗	● hogwarts99.synology.me 📧 🔗	178.190.101.72	💬 📄

Abbildung 29 DDNS Hostname

Somit ist die Synology nun auch über das Internet erreichbar, wenn man den Hostnamen kennt.

8.2 Verwendete Entwicklungssysteme

Die nachfolgenden drei Entwicklungssysteme (8.2.1 – 8.2.3) werden zum Modellieren, Texturieren und Animieren von 3D-Objekten verwendet. Damit können unter anderem 3D-Bilder, Videos oder Computerspiele entwickelt werden.

8.2.1 Blender

Blender ist eine freie Software, welche als Cross-Plattform-Anwendung zur Verfügung steht. Somit ist sie sowohl auf Windows, Linux, als auch macOS lauffähig. Die Oberfläche verwendet OpenGL (siehe 8.5.1), welches eine einheitliche Darstellung – unabhängig von Hardware und Plattform – ermöglicht.



Abbildung 30: Blender-Logo
(blender.org, 2018)

Blender wird zum Erstellen der Computeranimationen zu den jeweiligen Übungen verwendet. Damit wird eine Auswahl der benötigten Übungen modelliert, texturiert und animiert.

8.2.2 3DS Max

3DS-Max ist eine kostenpflichtige Software, die ebenfalls auf den im vorigen Punkt genannten drei Betriebssystemen lauffähig ist. Das Programm wird von der Firma Autodesk als Educational Edition für Schüler und Studenten zur Verfügung gestellt.



Abbildung 31: 3DS-Max-Logo (autodesk.com, 2018)

3DS-Max wird als Teil der Diplomarbeit verwendet, um einen Technologievergleich der 3D-Entwicklungssysteme anzufertigen.

8.2.3 Autodesk Maya

Autodesk Maya ist eine kostenpflichtige Software. Das Programm wird ebenfalls von der Firma Autodesk als Educational Edition zur Verfügung gestellt.



Abbildung 32: Maya-Logo (autodesk.com, 2018)

Autodesk Maya wird als Teil der Diplomarbeit verwendet, um einen Technologievergleich der 3D-Entwicklungssysteme anzufertigen.

8.2.4 Technologievergleich der drei Systeme

8.2.4.1 Allgemeines

Sowohl 3DS-Max, als auch Maya werden mit einem etwas größeren Funktionsumfang als Blender geliefert. 3DS-Max enthält beispielsweise Material zum Erstellen von Architekturplänen wie Stiegen oder Rohre. Der Grund liegt darin, dass beide Softwares kostenpflichtig sind. Blender ist dagegen frei erhältlich, es sind dafür weniger Basismaterialien enthalten, die Oberfläche ist nicht so einfach aufgebaut und man benötigt daher auch mehr Einarbeitungszeit. Grundsätzlich kann man aber mit allen drei Entwicklungssystemen nahezu identische Ergebnisse erzielen, bei Blender wird lediglich mehr Aufwand benötigt. Ein großer Vorteil von Blender ist außerdem, dass man sehr viele Online-Tutorials zu nahezu allen möglichen Gegebenheiten findet, da es kostenlos ist und damit von vielen Menschen verwendet wird.

Die Oberfläche der drei Entwicklungssysteme kann unter folgenden URIs verglichen werden:

Blender:

Vgl. docs.blender.org, 2018

3DS Max:

Vgl. help.autodesk.com, 2018

Maya:

Vgl. knowledge.autodesk.com, 2018

8.2.4.2 Vorgehensweise

Ein großer Unterschied zwischen Blender und den beiden Autodesk-Anwendungen ist aber die Vorgehensweise bei der Erstellung und Animation von 3D-Objekten. Während man in 3DS-Max und Maya überwiegend mit Mausklicks auf bestimmte Symbole - welche in verschiedenen Leisten angezeigt werden - arbeitet, ist man mit Blender sehr viel auf Tastenkürzel und Tastenkombinationen angewiesen, mit denen man beispielsweise Objekte vergrößern, verschieben oder rotieren kann. Dadurch wird es möglich, dass man mit Blender eine viel schnellere Arbeitsgeschwindigkeit erreicht.

8.2.4.3 Renderer

3DS-Max bietet sehr viele verschiedene Renderer an, die je nach Situation ausgewählt werden. Möchte man beispielsweise ein fotorealistisches Ergebnis, wählt man den Mental Ray Renderer. Der deutlich schnellere Scanline Renderer hingegen wird verwendet, wenn die Genauigkeit nicht sehr wichtig ist. Weitere Renderer wären der von Autodesk entwickelte ART-Renderer, der NVIDIA IRAY Renderer oder der Quicksilver Hardware Renderer, welcher Grafik-Hardware für den Render-Prozess verwendet. (vgl. autodesk.com, 2018)



Abbildung 33: NVIDIA Mental Ray - Logo (nvidia.com, 2018)

Maya hingegen bietet einen Software-Renderer und einen Hardware- bzw. den neueren Hardware 2.0 Renderer an, mit Letzterem sind fotorealistische Szenen zu rendern. Weiters gibt es noch einen Vektor-Renderer, welcher für stilisierte Renderings wie beispielsweise Cartoon, Tonalismus (Malerei) oder Drahtgittermodelle verwendet wird. (vgl. autodesk.com, 2018)

Außerdem existiert sowohl für 3DS-Max, als auch für Maya ein Plugin für den von Solidangle und Sony Pictures Imageworks entwickelten Ray Tracing Renderer Arnold, welcher in über 300 Film- und Animationsstudios der Welt verwendet wird. Aufgrund der hohen Render-Geschwindigkeit von Arnold wurde er beispielsweise für den Oscar-prämierten Film Gravity verwendet. (vgl. solidangle.com, 2018)

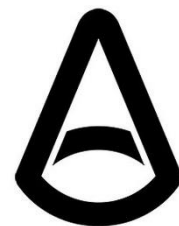


Abbildung 34: Solidangle-Logo (vimeo.com, 2018)

Blender verfügt grundsätzlich über drei Renderer, wobei zum fotorealistischen Rendern nur der auf Raytracing basierende Cycles Render verwendet wird. Als Raytracing bezeichnet man einen auf der Aussendung von Strahlen basierenden Algorithmus zur Ermittlung der Sichtbarkeit von dreidimensionalen Objekten im Raum. Das kann noch durch das Radiosity-Verfahren erweitert werden, welches den Weg und das Auftreffen von Lichtstrahlen auf ähnliche Weise berechnet. Beim Cycles-Render kann man auswählen, ob man mithilfe der CPU oder der GPU rendern möchte. (vgl. blender.org, 2018)

8.2.4.4 Warum wurde Blender gewählt?

Der Hauptgrund, warum Blender als Entwicklungssystem gewählt wurde ist, weil die Compu-teranimationen in der App verwendet werden, die der Physiotherapeut an seine Kunden weitergibt. Somit ist es nicht möglich, die Animationen mithilfe einer Educational-License für

Maya oder 3DS-Max zu erstellen, weil diese nur zur nicht kommerziellen Nutzung verwendet werden darf. Da man mit Blender die Kosten einer Lizenz gänzlich vermeiden und grundsätzlich sehr ähnliche Ergebnisse erzielen kann, wurde schlussendlich Blender gewählt. Ein weiterer Grund ist die schnellere Arbeitsgeschwindigkeit in Blender, da man mithilfe von Tastenkürzeln den Arbeitsablauf stark beschleunigen kann.

8.2.5 Visual Studio

Visual Studio ist eine von Microsoft angebotene Entwicklungsumgebung für verschiedene Hochsprachen (darunter C, C++, C#, Python).



Abbildung 35: Visual Studio-Logo (wikimedia.org, 2018)

Sie wurde für die Implementierung der einfachen Datei-Kopierer-Anwendung (vgl. Kapitel 9.1.7) zur Beschleunigung des Renderprozesses verwendet.

8.2.6 Visual Studio Code

Visual Studio Code ist eine weitere, von Microsoft angebotene, Entwicklungsumgebung. Anders als sein großer Bruder Visual Studio, spezialisiert sich Visual Studio Code auf die Entwicklung von Webanwendungen. Dabei wird die Entwicklung mit Javascript, Typescript, HTML, CSS, PHP, SQL ermöglicht.



Abbildung 36: Visual Studio Code-Logo

Außerdem ermöglicht Visual Studio Code auch das Installieren und Benutzen von anderen Packages und die Quellcodeverwaltung mittels GitHub.

Visual Studio Code wurde wegen der Übersichtlichkeit, Code Vervollständigung und anderer nützlicher Features gewählt.

8.2.7 Atom Editor

Atom ist ein Open-Source Editor, der auf dem Electron Framework basiert. Atom integriert einen Paketmanager „apm“ und ein Versionierungsplugin für Git. Desweiteren bietet Atom Syntaxvorhebung und Code Completion. Atom ist mit vielen Plugins und Themes leicht und schnell zu erweitern.



Abbildung 37 Atom Editor

Im Projekt wurde Atom hauptsächlich zur Entwicklung der REST-API verwendet.

8.2.8 Ionic DevApp

Ionic DevApp ist eine mobile App zum Testen der erstellten mobilen Applikation. Diese App verbindet sich automatisch mit der erstellten App, solange sich das Mobilgerät im gleichen Netzwerk wie das Entwicklungsgerät befindet. Ionic DevApp ist für Android und iOS erhältlich.



Abbildung 38: Ionic DevApp-Logo (play.google.com, 2018)

8.2.9 Notepad++

Notepad++ ist ein kostenfreier, leichtgewichtiger Texteditor, der ASCII bzw. Unicode-Kodierungen unterstützt. Im Gegensatz zum Standard Windows-Editor wird für die meisten Programmiersprachen Syntax-Hervorhebung und Auto-Vervollständigung unterstützt, deshalb entschieden wir uns, diesen Editor für die schnelle Einsicht in unterschiedliche Daten zu verwenden.



Abbildung 39: Notepad++ Logo (de.wikipedia.org, 2018)

8.2.10 Postman

Postman ist eine Open-Source Software, basierend auf dem Electron-Framework, zum Testen von Schnittstellen. Außerdem kann man nicht nur die Art der Anfrage bestimmen (GET, POST, PUT, DELETE), sondern auch die Header und HTML Fragmente.

Im Projekt wurde der Service benutzt um den Client zu simulieren und den Service sowohl lokal als auch auf der Synology zu testen.



Abbildung 40: Postman (get-postman.com, 2018)

8.3 XAMPP als Testsystem

Da auf der Synology die MySQL Datenbank und der Apache Webserver verfügbar sind, wurde ein Testsystem benötigt, das beides beinhaltet. Da XAMPP bereits aus dem Unterricht bekannt war, fiel die Entscheidung relativ schnell und leicht.

8.3.1 Konfiguration

Das Konfigurieren von XAMPP gestaltet sich recht einfach, da nur für den Webserver und die Datenbank freie Ports benutzt werden müssen.

Der Webserver ist mit Port 80 und die Datenbank mit Port 3306 erreichbar. Somit sind beide mit ihrem Standard Port zu erreichen.

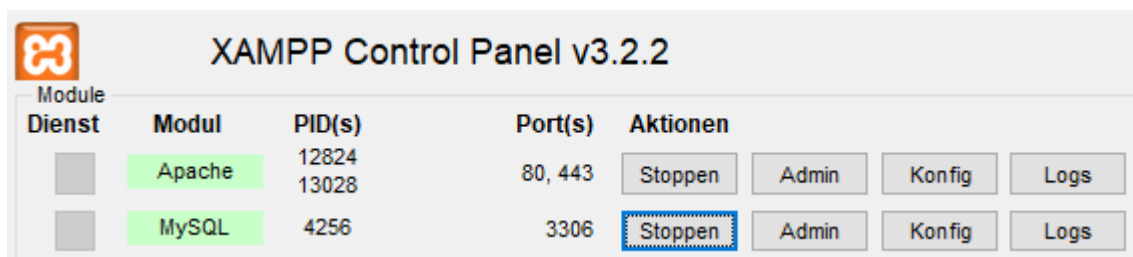


Abbildung 41: XAMPP Control Panel

8.3.2 Virtueller Host

Um dem REST-API URL-Standard zu entsprechen ist es nötig, einen virtuellen Host einzurichten. Ein virtueller Host ist eine zusätzliche Domain, die auf demselben Server betrieben wird.

Ursprünglicher API-Aufruf: http://localhost/api/src/index.php/<table_name>/<parameters>

API Aufruf mit virtuellen Host: http://api/<table_name>/<parameters>

8.3.3 Konfiguration eines virtuellen Hosts

In der „hosts“ Datei im Verzeichnis (C:\Windows\System32\drivers\etc) muss die Adresse 127.0.0.1 und der Name hinzugefügt werden, hier wird der Name „api“ verwendet.

```
# localhost name resolution is handled within DNS itself.  
127.0.0.1      localhost  
127.0.0.1      api  
::1           localhost
```

Danach müssen der Datei „httpd-vhost.conf“ im Verzeichnis (xampp\apache\conf\extra) diese Einträge hinzugefügt werden:

```
#Slimframework API  
<VirtualHost *:80>  
  DocumentRoot "C:\xampp\htdocs\api\src"  
  ServerName api  
</VirtualHost>
```

8.4 REST

8.4.1 Allgemein

REST steht für Representational State Transfer und ist ein sogenanntes „Programmierparadigma“. Ein Programmierparadigma soll den Entwickler dabei unterstützen, „guten“ Code zu erstellen. REST hat das Ziel eine einheitliche Schnittstelle zu schaffen. Denn das Internet oder World Wide Web stellt eigentlich eine enorm große REST Anwendung dar.

Vgl. (Wikipedia.org, 2018)

8.4.2 Anfrage und Antwort

GetFitAgain verwendet REST, um Datensätze über den Webservice aus der Datenbank lesen, erstellen, bearbeiten oder löschen zu können. Als Dateiübertragungsformat wird dabei JSON benutzt. Um überhaupt mit den Daten zu arbeiten, muss der Client eine Anfrage an den Webservice schicken. Dieser überprüft dann, ob die Anfrage korrekt ist und schickt eine Antwort zurück.

Der Webservice unterstützt folgende Anfragen:

- GET
- POST
- PUT
- DELETE

8.4.3 GET Anfrage

Die API verwendet GET Anfragen nur zum Auslesen von Daten. Wenn nun die API unter der URI <http://api/person/1> mit einer GET-Anfrage angesprochen wird, erfolgt nach korrekter Anfrage diese Antwort:

```
{
  "status": 200,
  "status_message": "data found",
  "data": [
    {
      "PersonID": "1",
      "prename": "Beatrice",
      "surname": "Alford",
      "birthdate": "1990-11-14",
      "street": "4564 Urna. Road",
      "postcode": "9261",
      "Town": "Inverness",
      "Phonenumber": "06642974446",
    }
  ]
}
```

```
        "Email": "sollicitudin.adipiscing.ligula@aliquam.ca",  
        "sex": "w"  
    }  
]  
}
```

Die ersten 2 Teile der Nachricht, „status“ und „status_message“, geben Auskunft über den Erfolg der Anfrage. Der dritte Teil („data“) beinhaltet die Daten der angeforderten Ressourcen.

8.4.4 POST-Anfrage

Anders als bei einer GET-Anfrage, kann man mit einer POST-Anfrage und einer PUT-Anfrage Daten im Body der Anfrage übergeben. Die POST-Anfrage wird hierbei zur Erstellung von Daten benutzt.

Beispielaufruf für eine POST-Anfrage im Postman:

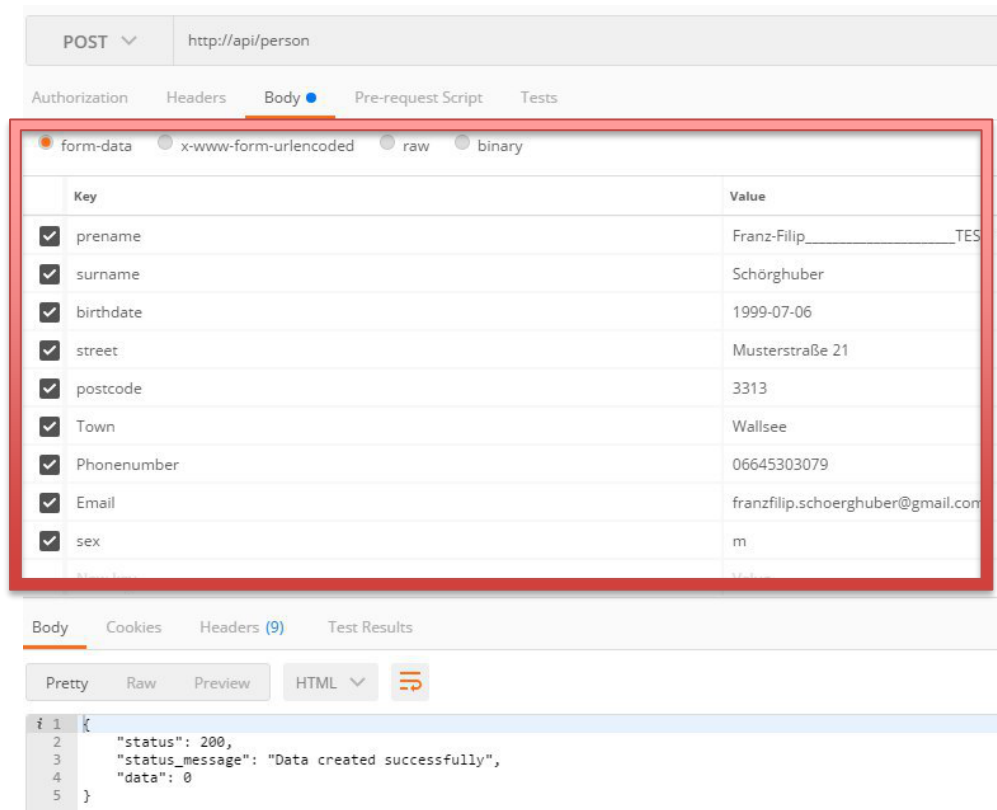


Abbildung 42 POST Anfrage in Postman

Wenn nun die API unter der URI <http://api/person> mit einer korrekten POST-Anfrage angesprochen wird, dann wird als Antwort als Status „200“ und als Status Message „Data created successfully“ zur Bestätigung zurückgesandt. In Abbildung 38 ist im rot markierten Bereich auch der Body Teil der Anfrage sichtbar. Dieser Teil besteht immer aus allen Attributen der angesprochenen Ressource.

8.4.5 PUT-Anfrage

Die PUT-Anfrage wird zum Bearbeiten von Daten verwendet.

Beispielaufruf für eine POST-Anfrage im Postman:

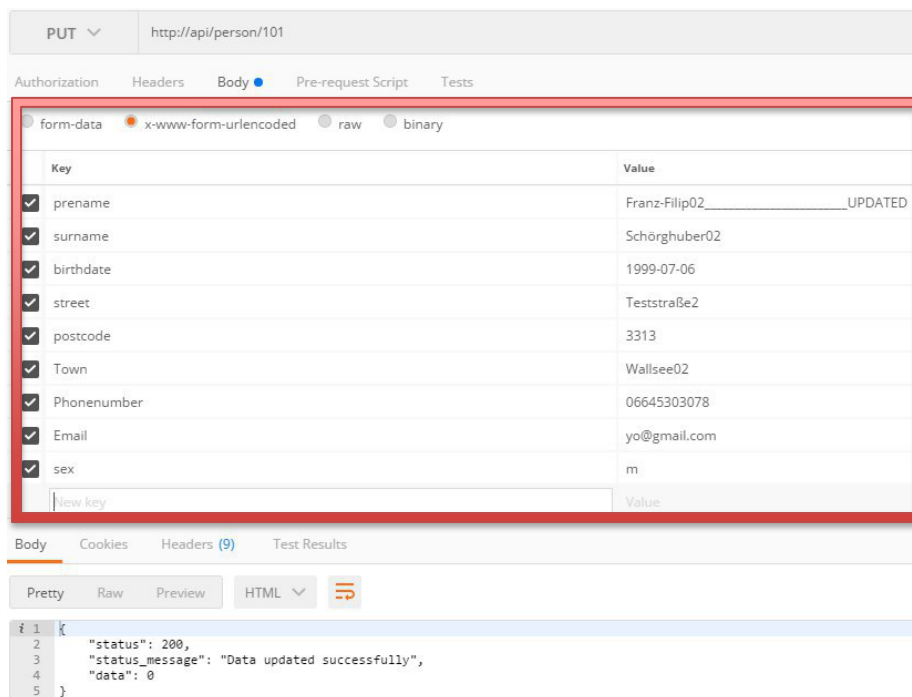


Abbildung 43 PUT Anfrage im Postman

Wenn nun die API unter der URI <http://api/person/101> mit einer korrekten PUT-Anfrage angesprochen wird, dann wird als Antwort als Status „200“ und als Status Message „Data updated successfully“ zur Bestätigung zurückgesandt. In Abbildung 39 ist im rot markierten Bereich auch der Body Teil der Anfrage sichtbar. Dieser Teil besteht immer aus allen Attributen der angesprochenen Ressource. Die Zahl in der Anfrage wird als eindeutige Identifikation benötigt.

8.4.6 DELETE-Anfrage

Die API verwendet DELETE-Anfragen nur zum Löschen von Daten. Wenn nun die API unter der URI <http://api/person/1> mit einer DELETE-Anfrage angesprochen wird, erfolgt nach korrekter Anfrage diese Antwort:

```
{  
  "status": 200,  
  "status_message": "Data deleted successfully",  
  "data": 0  
}
```

Die Zahl in der URI wird für die eindeutige Identifikation benötigt.

8.4.7 Requirements

Das Framework benötigt einen Apache oder Nginx Server als Backend. Der Server sollte mit einer `.htaccess` File so konfiguriert werden, dass alle Anfragen zu einem, als Frontcontroller definiertem, PHP File geschickt werden. Als Frontcontroller ist das „index.php“ File definiert.

8.4.8 Implementierung

8.4.8.1 Definition URIs

Die URIs müssen dieses Schema haben:

- Grundbaustein
 - `<Hostname>/api`
- Route
 - `/<Tabellenname>`
- Zusätzliche Attribute
 - Als zusätzliche Attribute werden nur IDs akzeptiert
 - Beispiel: `/1`
 - Es können aber auch mehrere IDs benötigt werden, wie zum Beispiel bei den Assoziationstabellen: `/1/3/4`

URIs die ein anderes Schema aufweisen bekommen einen Error 404.

8.4.8.2 Verwendete HTTP-Methoden

Die API akzeptiert GET, POST, PUT und DELETE Anfragen. Dabei hat jede Methode eine einzelne Aufgabe.

1. GET
 - a. Liest Daten aus der Datenbank aus.
2. POST
 - a. Speichert einen neuen Datensatz in die Datenbank.
3. PUT
 - a. Verändert einen bereits bestehenden Datensatz in der Datenbank.
4. DELETE
 - a. Löscht einen bestehenden Datensatz aus der Datenbank.

8.4.8.3 Implementierung mit SlimFramework

Wie bereits erwähnt, funktioniert das SlimFramework nach einem gewissen Schema. Dieses fordert, dass die Anfrage in einem Frontcontroller zerplückt wird und dann die benötigte Route aufgerufen wird. In dieser Route befindet sich dann die eigentliche Funktion, die die Aufgabe ausführt und die Antwort liefert. Als Beispiel URI wird hier <http://api/person> verwendet, eventuell kommt hier noch eine ID hinzu, dann würde die URI so aussehen: <http://api/person/<ID>>.

Zunächst muss eine htaccess File erstellt werden, die alle Anfragen auf das index.php File weiterleitet:

```
1. RewriteEngine on
2. RewriteCond % {REQUEST_FILENAME} ! -d
3. RewriteCond % {REQUEST_FILENAME} ! -f
4. RewriteRule.index.php[L]
```

In der „index.php“ wird der Frontcontroller definiert. Dieser fragt von der eingehenden Anfrage die Art und die URI ab. Das bedeutet, dass für jede Tabelle eigene Routen angelegt werden müssen.

Diese Routen werden wie folgt angelegt:

```
1. //Get all persons
2. $app - > get('/person', function(Request $request, Response $response) {
3.     getAllPersons();
4. });
```

Der Controller trennt nun die Anfrage in die benötigten Informationen. Aus dieser Anfrage bekommt man somit die Art der Anfrage und auch die angeforderte Ressource. Ausgehend von unserer Beispiel-URI wird hier die Ressource „person“ angefordert.

Laut dieser Route wird nun die Methode „getAllPersons“ aufgerufen. Diese befindet sich im Unterordner „Table“ in dem File „person.php“. Sie wurde am Anfang des Dokuments eingebunden:

```
1. include_once('Table/person.php');
```

In jeder Route wird eine spezifische Methode aufgerufen. Diese erfüllt dann die angeforderte Aufgabe. In diesem Fall wäre die Aufgabe alle Datensätze der Tabelle Person abzurufen.

8.4.9 Assoziationstabellen

Assoziationstabellen verknüpfen 2 oder mehrere Tabellen mit ihren Primärschlüssel.

In der Assoziationstabelle „FeedbackDoneByUser“ wird gespeichert ob ein User ein Feedback abgeschlossen hat. Assoziationstabellen in der REST-API sind komplexer zu erstellen, da sie nicht über einen Primärschlüssel verfügen, sondern über 2 oder mehr Fremdschlüssel.

Im Fall von „FeedbackDoneByUser“ sieht man das am deutlichsten bei der Delete-Methode. Normalerweise würde man nur einen Schlüsselwert benötigen, um einen Datensatz eindeutig zu löschen. Hier werden aber alle Schlüsselwerte die in der Tabelle gespeichert sind benötigt, um eine eindeutige Löschung zu ermöglichen.

Was würde passieren, wenn nur mit einer ID, angenommen der CustomerID, in der „FeedbackDoneByUser“ Tabelle gelöscht werden würde?

- Es würden alle Datensätze mit dem Schlüsselwert gelöscht werden, nicht nur einer.

Um eben das zu umgehen müssen beide Schlüsselwerte mitgeschickt werden.

- Somit wird in der Tabelle der Datensatz mit der CustomerID und der FeedbackID gelöscht

8.4.10 Benötigte Ordnerstruktur

In dem unten ersichtlichen Bildschirmfoto kann man die Grundverzeichnisstruktur erkennen.

Im Verzeichnis `src` befindet sich der gesamte benötigte Source Code:

- `index.php`
 - Empfängt die Anfrage, zerpflückt sie und leitet sie an das jeweilige PHP File im Ordner `Table` weiter
- `.htaccess`
 - Kümmt sich um die Weiterleitung der Anfragen an die `index.php` File
- `othermethods.php`
 - Implementiert die benötigten Methoden um eine Antwort zu schicken und mehrere Datensätze in ein Array zu speichern
- Unterordner `Table`
 - Alle Entitäten aus der Datenbank in PHP Files realisiert

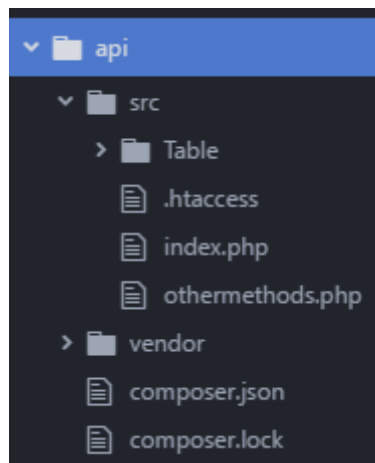


Abbildung 44 Ordnerstruktur SlimFramework

8.5 Verwendete Bibliotheken

8.5.1 OpenGL

OpenGL ist eine Grafikbibliothek, welche eine plattform- und programmiersprachenübergreifende Programmierschnittstelle zur Entwicklung von 2D- und 3D-Computergrafikanwendungen zur Verfügung stellt. Damit können komplexe 3D-Szenen in Echtzeit dargestellt werden.



Abbildung 45: OpenGL-Logo (khronos.org, 2018)

Die OpenGL-API ist in der Regel durch Systembibliotheken oder als Teil der Grafikkartentreiber implementiert. (vgl. wikipedia.org, 2018)

3D-Szenen werden sowohl in Blender, 3DS-Max und Autodesk Maya mithilfe von OpenGL dargestellt. Somit ist es möglich, seine 3D-Modelle in Echtzeit zu bearbeiten und Veränderungen sofort darzustellen.

8.6 Sonstige verwendete Software

8.6.1 VMware-Player

Die kostenlose Virtualisierungssoftware VMware-Player von der Firma Dell wurde während der Diplomarbeit speziell dazu verwendet, Tests durchzuführen. Außerdem wurde mit Hilfe dieser Software eine virtuelle Maschine mit dem Betriebssystem Windows 10 für unseren Diplomarbeitsbetreuer bereitgestellt, damit er am Ende der Diplomarbeit alle Programme, die er zum Testen benötigt, fertig installiert hat. (vgl. wikipedia.org, 2018)

8.6.2 GitHub

Als Versionsverwaltung für die Diplomarbeit wurde GitHub gewählt.

GitHub ist eine Code-Hosting-Plattform der Firma GitHub Inc. Damit ist es möglich, dass mehrere Personen mit dem gleichen Projekt arbeiten. Hier wurde immer der aktuelle Stand der Diplomarbeit hochgeladen.



Abbildung 46: Github-Logo (github.com, 2018)

9 Implementierung

9.1 Computeranimationen der Übungen

Da in Blender sehr viel mit Tastenkombinationen gearbeitet wird, werden in diesem Kapitel zu den jeweiligen Schritten in Klammer die dazugehörigen Tastenkürzel angegeben.

9.1.1 Erstellen von 3D-Modellen

In jedem 3D-Entwicklungsprogramm hat man die Möglichkeit, verschiedene Objekte zu erstellen, vergrößern, verkleinern, rotieren, verschieben oder generell die Form mithilfe von verschiedenen Tools zu verändern. Mit Objekten sind beispielsweise Formen wie Würfel, Zylinder, Flächen oder Dreiecke gemeint. Somit ist es möglich, jedes beliebige reale Objekt nachzubilden, indem man einfach verschiedene 3D-Objekte erstellt und diese entsprechend anpasst.

9.1.1.1 Blender

Um in Blender eine Form hinzufügen, öffnet man auf der linken Seite den *Create*-Reiter (*Shift+A*), in dem alle verfügbaren Formen angezeigt werden. Man hat somit alle wichtigen Basisformen abgedeckt, die man dann entsprechend verbinden kann, um gewünschte Objekte zu erstellen. Möchte man beispielsweise einen Tisch erstellen, klickt man auf *Cube*, diesen kann man daraufhin entlang den jeweiligen Achsen vergrößern, verschieben und rotieren. Erstellt man nun mehrere Würfel, kann man diese, nachdem man sie angepasst hat, zu einem Tisch verbinden. Zuletzt färbt man sie noch ein und fügt bestimmte Effekte hinzu, beispielsweise Glas oder Holz.



Abbildung 47: Blender-Formen

9.1.1.2 Beispiel Hantelbank

Um eine Hantelbank zu modellieren, benötigt man grundsätzlich nur Würfel und Zylinder. Diese muss man dann entsprechend platzieren, bis man das gewünschte Ergebnis erreicht.

9.1.1.2.1 Vorgehensweise

Zuerst fängt man am Boden an und fügt zwei Würfel und einen Zylinder hinzu. Diese verkleinert bzw. vergrößert (*S*) man entlang den jeweiligen Achsen, bis sie die gewünschte Form annehmen. Eventuell müssen sie folglich noch rotiert bzw.



Abbildung 48: Hantelbank (Blender)

verschoben (*G*) werden. So macht man es jetzt für alle Teile der Hantelbank, bis sie so aussieht wie bei Abbildung 48. Um den metallischen Effekt hinzuzufügen, muss man zuerst rechts oben in der Leiste ein Material erstellen, welches man daraufhin den Formen zuweist, die aus Metall bestehen. Um das Material zu bearbeiten, verwendet man den Node-Editor, welchen man im Dropdown-Menü links unten auswählt.

Einstellungen für den metallischen Effekt:

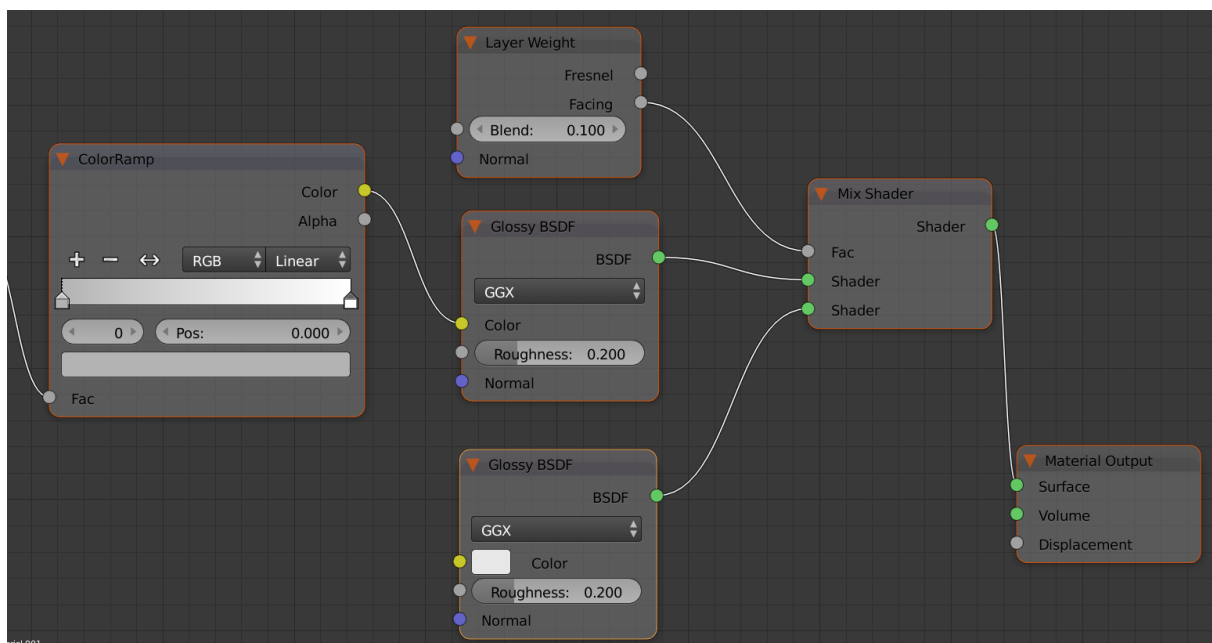


Abbildung 49: Node-Editor (Blender)

Mit der Color Ramp stellt man die Farbe ein, die von einem eher dunklen silber zu einem helleren übergeht, somit erreicht man die typische stahlartige Farbe. Die beiden Elemente *Glossy BSDF* sorgen für eine Reflektion, die bei metallischen oder gläsernen Objekten benötigt wird. Gemeinsam mit dem Layer Weight Node kann man mithilfe vom *Mix Shader* einstellen, wie stark die beiden Farben vermischt werden sollen. Am Ende wird es dann noch zum *Material Output* verbunden, somit hat man den metallischen Effekt nun auf der Oberfläche des Materials. Dieses Material kann man nun auf die jeweiligen Objekte anwenden, indem man es im Material-Reiter auswählt.



Abbildung 50: Material zuweisen (Blender)

Als letzten Schritt kann man noch alle Teile der Hantelbank auswählen und mit *Strg+J* verbinden, somit hat man sie in einem Objekt und kann sie leichter in andere Szenen kopieren. Außerdem ist es notwendig, für jeden Teil auszuwählen, dass man das Material auf das Objekt selbst anwenden möchte. Dazu wählt man im Dropdown-Menü neben dem Materialnamen *Object* aus. Wählt man hingegen *Data* aus, wird das Material auf die Objekt-Daten angewendet, wenn man die Objekte also verbindet, können keine verschiedenen Materialien gespeichert werden, da sie nicht auf die einzelnen Objekte selbst angewendet wurden.

9.1.1.3 3DS-Max

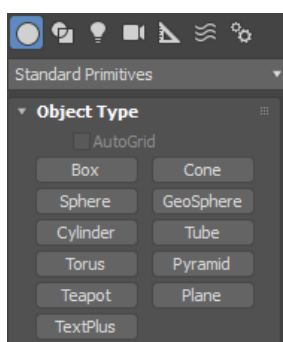


Abbildung 51: Materialien (3DS Max)

In 3DS Max gibt es ebenfalls einen Reiter, in dem Formen hinzugefügt werden können. Dabei unterscheidet man wieder in Standard-Primitiven, die die Basisformen abdecken, und andere Formen, beispielsweise Knochen, Lichter, eine Kamera oder auch verschiedene Kräfte, die auf die umgebenden Materialien wirken sollen. Die Vorgehensweise ist hier wieder sehr ähnlich zu Blender, die Formen werden so lange angepasst und verbunden, bis man sein gewünschtes Ergebnis erzielt.

Um einem Material einen Effekt hinzuzufügen, arbeitet man in 3DS-Max mit dem sogenannten Material-Editor, welchen man oben in der Leiste (bzw. mit der Taste *M*) findet. Es öffnet sich nun ein Fenster, in dem man das Material beliebig bearbeiten kann. Ähnlich wie in Blender gibt es verschiedene *Nodes*, die miteinander verbunden werden, somit kann man nahezu jeden beliebigen Effekt nachbilden.

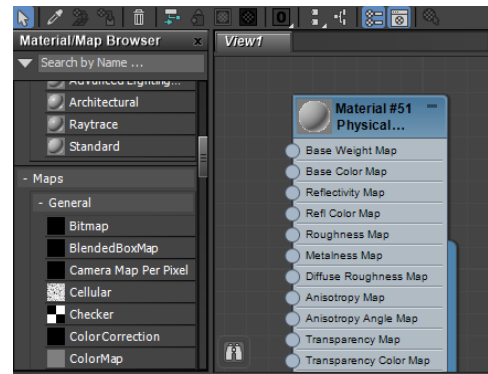


Abbildung 52: Material Editor (3DS Max)

9.1.1.4 Maya

In Maya findet man alle vorhandenen Formen in der Leiste oben. Dabei unterscheidet man zwischen Polygons, welche üblicherweise aus Eckpunkten, Kanten und Oberflächen bestehen, und NURBS (Non uniform rational B-Spline) Formen, welche auf mathematischen Funktionen und Kurven basieren. NURBS-Objekte brauchen zwar weniger Speicher, können aber das Animieren von Objekten sehr kompliziert machen, daher ist es üblich, dass man NURBS-Objekte vor dem Animieren in Polygon-Objekte konvertiert.



Abbildung 53: Materialien (Maya)

Ein Material kann in Maya erstellt werden, indem man den Hypershade-Editor öffnet, dieser befindet sich in der sogenannten Status Line, welche sich direkt unter der Menüleiste befindet. Darin kann man wieder ähnlich wie bei den anderen beiden Entwicklungssystemen ein Material erstellen und diesem beliebige Effekte hinzufügen. Dafür gibt es ebenfalls wieder einen Node-Editor, in dem die verschiedenen Nodes verbunden werden, bis gewünschte Effekte entstehen.

9.1.2 Erstellen eines Skeletts

Um einen 3D-Charakter bewegen zu können, benötigt man mehrere sogenannte Knochen, die dann an den entsprechenden Punkten mit dem Modell verknüpft werden. Somit kann man mehrere Körperteile mit den jeweiligen Knochen verbinden und kann diese anschließend animieren.

9.1.2.1 Blender

In Blender gibt es zwar Rigging-Tools, die Modelle automatisiert mit vorgefertigten Skeletten verbinden, man erzielt jedoch bessere Ergebnisse, wenn man das Skelett selbst aufbaut, indem man die einzelnen Knochen erstellt, entsprechend vergrößert, rotiert und an die richtige Position bewegt. Am Schluss verlinkt man die Knochen mit dem Modell und lässt sich die Gewichtungen, also wie stark sich bestimmte Körperteile bei den Bewegungen verformen sollen, generieren.

9.1.2.1.1 Vorgehensweise

Als ersten Schritt öffnet man sein Charakter-Modell in Blender. Jetzt fängt man damit an, den ersten Knochen zu platzieren, dazu fügt man einen neuen Knochen hinzu (*Shift+A -> Armature -> Single Bone*) und bewegt diesen an die Position des untersten Wirbelknochens, vereinfacht etwa im Bereich des Bauchnabels. Als nächstes vergibt man rechts im Bone-Tab einen Namen für diesen Knochen, beispielsweise *Wirbel1.m*. Das *.m* steht für Mitte, ist in diesem Fall nicht unbedingt notwendig, aber ermöglicht einen besseren Überblick über das Skelett. Wichtig wird es erst später, wenn man die Knochen für Arme, Beine, usw. erstellt, da man diese nur auf einer Seite erstellen muss und daraufhin duplizieren und spiegeln kann. Sehr hilfreich ist die

Einstellung *X-Ray*, dadurch sieht man die Knochen durch das Modell durch und kann sie punktgenau platzieren. Der nächste Schritt ist, den nächsten Wirbelknochen zu erstellen. Mit der Tastenkombination *Shift+E* kann man vom untersten Wirbelknochen weg einen neuen Knochen erstellen, diesen platziert man dann wieder entsprechend. So geht es jetzt weiter bis zum Kopf hoch, es sollten zumindest vier Wirbelknochen erstellt werden und für Hals und Kopf jeweils einer. Nach jedem platzierten Knochen sollte man außerdem überprüfen, ob er sich an der richtigen X-, Y- und Z-Position befindet.

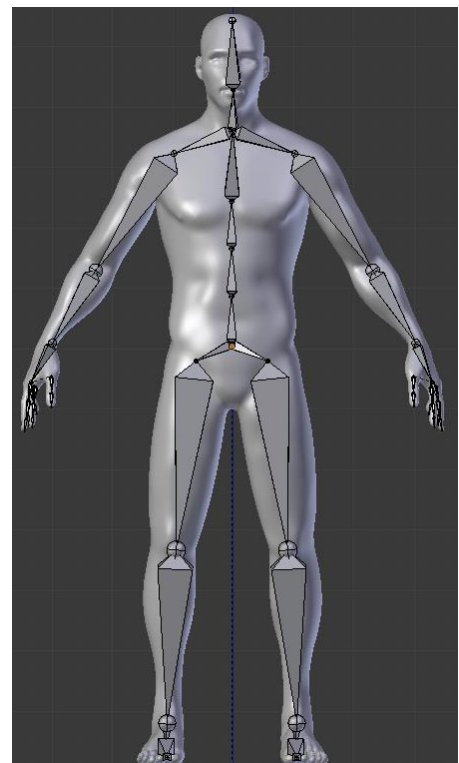


Abbildung 54: Skelett (Blender)

Dafür wechselt man mit Klicks auf Zahlen auf dem Ziffernblock in die verschiedenen Ansichten:

- 1 – Frontansicht
- 3 – Seitenansicht
- 7 - Vogelperspektive

Die Knochen sollten so verschoben werden, dass sie sich immer etwa in der Mitte eines Körperteils befinden.

Wie in Abbildung 54 ersichtlich, müssen jetzt noch die Arm- und Beinknochen erstellt werden. Das muss vorerst nur auf einer Körperhälfte geschehen, die Knochen für die andere Körperhälfte kann man sich im Anschluss automatisch erzeugen lassen.

Dafür ist es wichtig, jedem erstellten Knochen ein *.l* für Knochen auf der linken Seite beziehungsweise *.r* für Knochen auf der rechten Seite anzuhängen. Für die Armknochen erstellt man dazu vom obersten Wirbelknochen weg einen Knochen, der Richtung Schulter geht. Von diesem Schulterknochen weg werden jetzt bis in die Fingerspitzen weitere Knochen erstellt, wobei



Abbildung 55: Hand-Knochen (Blender)

bei den Fingerknochen im Knochen-Tab die Option *Keep Offset* ausgewählt, bzw. das Häkchen bei *Connected* entfernt werden muss, was bedeutet, dass sie nicht direkt mit dem Mittelhandknochen verbunden sind, sondern sich mit gleichmäßigem Abstand mit ihm bewegen und eine fixe Distanz einhalten. Für die Beine macht man genau das gleiche, erst erstellt man einen Knochen für die Hüfte und verbindet diesen dann bis unten in die Zehenspitzen. Im Normalfall ist es nicht notwendig, Knochen für jede einzelne Zehe zu machen.

Eine Besonderheit bei den Beinen bilden die Knie. Möchte man beispielsweise das Bein anheben und dabei das Knie abbiegen, ist das ohne zusätzliche Konfiguration nicht so einfach möglich. Dazu muss eine kinematische Kette angelegt werden, die bestimmt, wieviele Knochen mitbewegt werden sollen, wenn ein bestimmter Knochen (in diesem Fall das Knie) bewegt wird. Dazu geht man folgendermaßen vor:

- Erstellen von einem Knochen horizontal vom Knie weg (zwischen Ober- und Unterschenkel) und von der Ferse weg

- Dabei ist es wichtig, die Eigenschaft *Deform* auszuschalten, da diese Knochen das Modell nicht verformen sollten
- Name des Fersenknochens: ik.ferse.l bzw. ik.ferse.r
- Name des Knieknochens: ik.knie.l bzw. ik.knie.r
- Das *ik* – also inverse Kinematik – soll hier einfach widerspiegeln, dass diese Knochen Teil der kinematischen Kette sind.
- Bei beiden Knochen *Strg+P* -> *Clear Parent*
- Der Knie-Knochen kann nun etwas nach vorne geschoben werden (*Keep Offset*)
- In den Pose Mode wechseln, den Unterschenkelknochen auswählen und im Bone Constraints Tab den Bone Constraint *Inverse Kinematics* hinzufügen
 - Hier werden die Einstellungen laut Abbildung 56 bzw. vorgenommen (für die rechte Seite mit der Erweiterung .r)

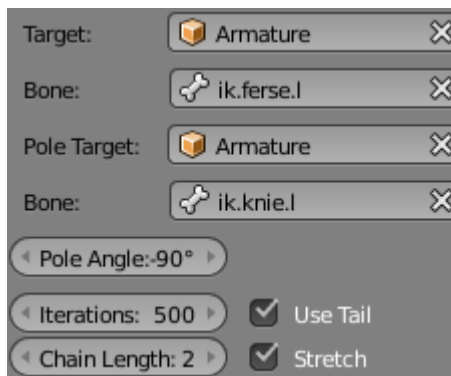


Abbildung 56: IK Settings (Blender)

Die Chain Length sagt aus, wie viele Knochen in der Kette bewegt werden sollen, wenn dieser Knochen bewegt wird. Der Pole Angle muss evtl. angepasst werden, falls sich das Bein beim zuweisen des Pole Target verdreht. Dasselbe macht man auch für die rechte Seite.

Hat man diese Einstellungen vorgenommen, ist es möglich, das Knie anzuheben, indem man den Knochen *ik.ferse.l* bzw. *ik.ferse.r* nach oben zieht.

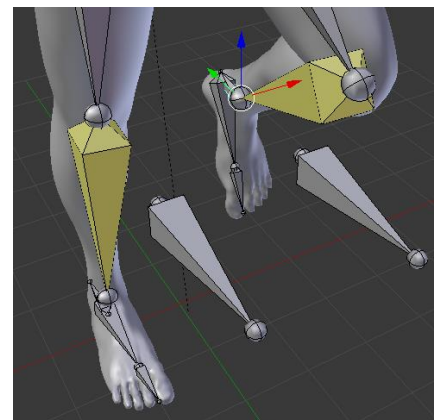


Abbildung 57: IK Knie (Blender)

Hat man das Skelett nun einseitig erstellt, alles richtig positioniert und korrekte Namen vergeben, kann man die eine Körperhälfte duplizieren. Dabei geht man folgendermaßen vor:

- ✓ Im Edit Mode (*Tab*) alle Knochen einer Körperhälfte selektieren (*Strg+B*)
- ✓ Curser zentrieren (*Shift + C*)

- ✓ Pivot auf 3D-Cursor ändern (Dropdown, Leiste unter dem 3D-View Editor)
- ✓ Knochen duplizieren und einmal Linksklick (*Shift+D*)
- ✓ Armature -> Mirror, X local auswählen (Leiste unter dem 3D-View Editor)
- ✓ W -> Flip Names

Jetzt hat man die Knochen der einen Körperhälfte auf die andere dupliziert und die Namen automatisch ersetzt, aus einem .l wurde .r bzw. aus .r wurde .l. Somit hat man ein fertiges Skelett, das jetzt nur mehr mit dem 3D-Modell verbunden werden müssen. Dazu selektiert man mit Shift zuerst das 3D-Objekt und klickt dann auf einen beliebigen Knochen. Jetzt verbindet man diese mit *Strg+P -> With Automatic Weights*. Damit werden automatisch Gewichtungen für die einzelnen Knochen erstellt, die bestimmen, wie stark sich Körperteile bei der Animation verformen. Jetzt hat man einen fertig animierbaren Charakter.

9.1.2.2 3DS-Max

Das Erstellen eines Skeletts kann natürlich auch in 3DS Max wie oben beschrieben erfolgen, 3DS Max stellt dafür aber Tools zur Verfügung, die einem viel Arbeit ersparen. So gibt es bereits ein fertiges Skelett, das man nur noch vergrößern bzw. verkleinern und richtig platzieren muss, danach kann man es schon mit dem Objekt verbinden.

9.1.2.2.1 Vorgehensweise

Als Erstes fügt man das sogenannte *Biped* hinzu, was übersetzt so viel heißt wie Zweibeiner, also ein menschliches Skelett. Dabei kann man einstellen, wieviele Knochen für die jeweiligen Körperteile erstellt werden sollen, also zum Beispiel 4 Wirbelknochen und 5 Fingerknochen pro Hand. Jetzt muss man das Skelett wieder entsprechend platzieren, rotieren, verkleinern bzw. vergrößern, bis alle Knochen an der richtigen Position sind. Dazu gibt es in 3DS Max die Quad-Ansicht, in der man seine Szene von 4 verschiedenen Ansichten betrachten kann, was es sehr leicht möglich macht, das Skelett zu platzieren. Danach wählt man das 3D-Modell aus und wählt rechts im Modifier-Tab aus der Modifier-List den *Skin-Modifier* aus. Den *Skin-Modifier* verwendet man, um dem Biped das 3D-Objekt zuzuweisen. Wird ihm ein Skelett zugewiesen, erhält jeder Knochen automatisch eine Gewichtung. Um das Skelett zuzuweisen, geht man weiter unten im Modifier-Tab auf *Add Bones* und fügt das komplette Biped hinzu. Der Charakter ist jetzt fertig und man kann damit Animationen erstellen.

9.1.2.3 Maya

In Maya gibt es seit der Version Maya 2017 das wohl einfachste und am besten funktionierende Rigging-Tool. Mithilfe des *Quick Rig*, welches im Rigging-Reiter zu finden ist, werden einem 3D-Objekt automatisch Knochen an den jeweiligen Körperteilen zugewiesen. Dabei kann man wählen, ob man die Knochen Schritt für Schritt zuweisen will, was für kompliziertere Modelle notwendig sein wird, oder ob man einfach mit einem Klick ein fertiges Skelett erstellen möchte, das dann dem Modell zugewiesen wird. Das funktioniert sehr gut für Standardobjekte, also menschliche Objekte. Hat man nun sein Skelett erstellt, ist es nicht mehr notwendig, dieses mit dem 3D-Objekt zu verbinden, das ist im vorigen Schritt mithilfe des Quick-Rig-Tools bereits automatisch passiert. Somit ist es in Maya möglich, mit nur wenigen Mausklicks einen fertig animierbaren Charakter zu erstellen.



Abbildung 58: Quick-Rig (Maya)

9.1.3 Animation von 3D-Objekten

Hat man jetzt ein fertiges Modell inklusive Skelett, kann dieses an allen Knochen rotiert, verschoben und skaliert werden. Animationen entstehen, indem man bei verschiedenen Positionen zu dazugehörigen Zeitpunkten Schlüsselbilder einfügt, mithilfe von inverser Kinematik wird dann immer die Bewegung vom letzten Schlüsselbild zum nächsten errechnet und entsprechend dargestellt. Am Ende wird diese Abfolge von Bewegungen gerendert und es entsteht ein Animationsvideo.

9.1.3.1 Blender

In Blender wechselt man dazu einfach in den Pose-Mode und kann jeden Knochen per Mausklick skalieren, rotieren und verschieben.

9.1.3.2 Beispiel Abduktoren in Seitenlage

Das Ziel ist es, diese Übung wiederholt auszuführen, mit einer gewissen Anzahl von Sätzen und Wiederholungen. Dabei soll von folgender Anfangsposition in die Endposition gegangen werden und wieder zurück.

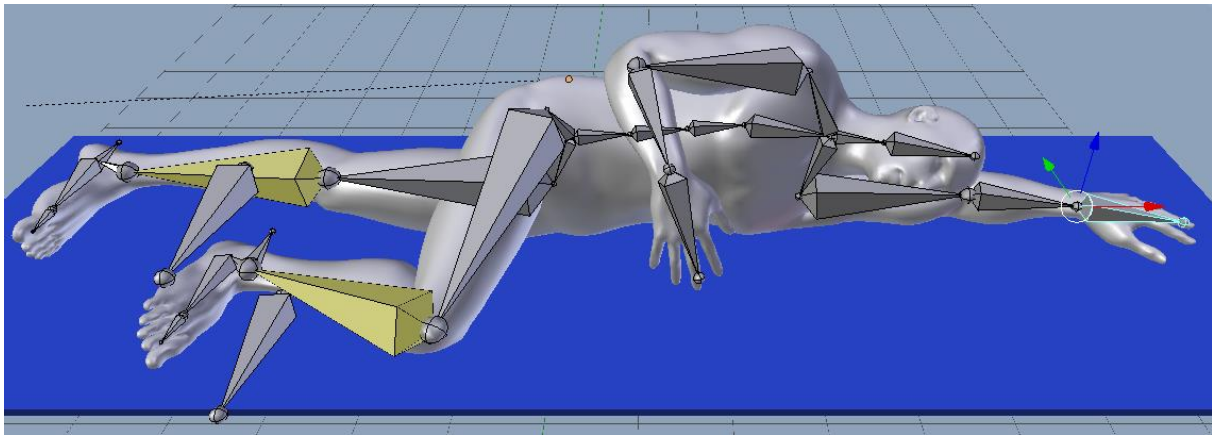


Abbildung 59: Ausgangsposition Abduktoren in Seitenlage

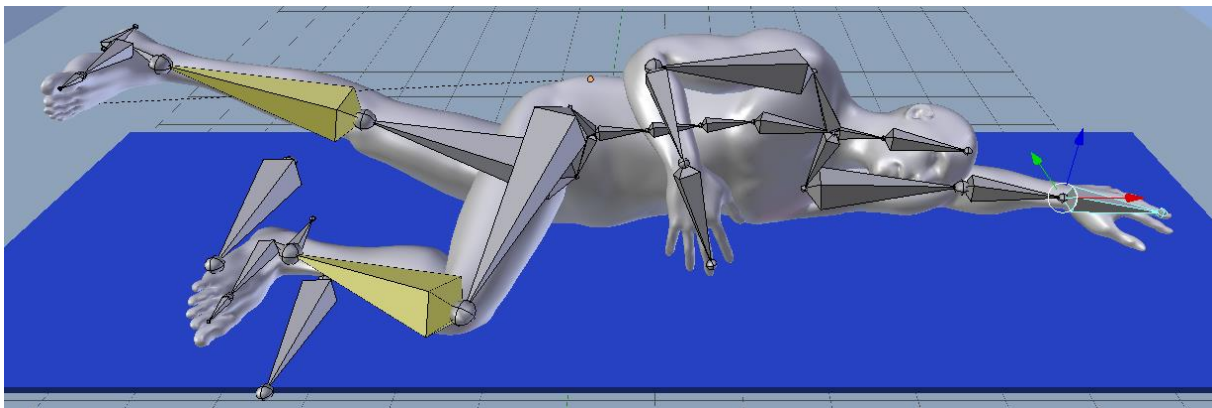


Abbildung 60: Endposition Abduktoren in Seitenlage

Als Erstes ist sicherzustellen, dass sich der Mauszeiger an Position 0 der Timeline befindet und sich die Knochen in der Ausgangsposition befinden. Daraufhin markiert man mit A alle Knochen und erzeugt ein Keyframe mit I, hier wählt man am besten LocRotScale, es werden also zu allen Knochen die Größe, Rotation und die Position gespeichert. Zusätzlich gäbe es noch einen Auto Key Mode, der automatisch Keyframes setzt, wenn man Knochen bewegt. In den Kapiteln 9.1.4, 9.1.5 und 0 werden weitere Schritte beschrieben.

9.1.3.3 3DS-Max

In 3DS Max ist das Vorgehen genau gleich, Keyframes kann man rechts unter der Timeline platzieren bzw. den Auto Key Modus einschalten.

9.1.3.4 Maya

In Maya ist das Vorgehen ebenfalls gleich, Keyframes kann man ebenfalls rechts unter der Timeline platzieren bzw. den Auto Key Modus einschalten. Alternativ kann man über den Tab *Animation* auch Keyframes platzieren.

9.1.4 Korrektur von Verformungen

Bei der Bewegung von Knochen kommt es fast immer zu bestimmten unrealistischen Verformungen des Modells an dieser Stelle. Diese können durch eine Neukonfiguration der Gewichtungen vermieden werden, es muss einfach eingestellt werden, dass sich umliegende Körperteile nicht so stark verformen sollen.

9.1.4.1 Blender

Dazu selektiert man in Blender den Knochen, bei dessen Animation sich umliegende Körperteile unrealistisch verformen. Danach drückt man irgendwo auf das 3D-Objekt und wählt im Dropdown unten den Weight-Paint-Mode aus. Hier bekommt man nun ein Menü, in dem man wählen kann, ob man Gewichtungen addieren, subtrahieren, etc. möchte. Rund um den selektierten Knochen wird der Körper

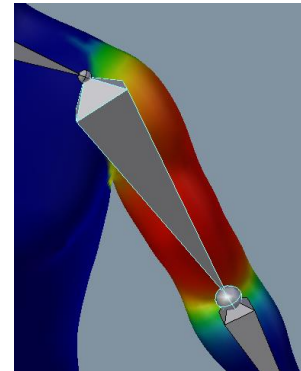


Abbildung 61: Weight-Paint-Mode (Blender)

jetzt farblich dargestellt, rot steht dabei für eine starke und grün für eine schwache Verformung an dieser Stelle, blaue Stellen verformen sich gar nicht bei Bewegung dieses Knochens. Wählt man im Menü *Add* aus, werden Gewichtungen dazu addiert, klickt man also mit der Maus auf eine Stelle, wird diese zunehmend rot eingefärbt. Bei *Subtract* werden Gewichtungen entfernt, es geht also auf grün bzw. blau zu.

Eine weitere Möglichkeit wäre die Verwendung des Sculpt-Mode, in dem man dem Modell Masse hinzufügen, entfernen, Oberflächen glätten oder Löcher füllen kann. Ändert man aber sein Modell in einer bestimmten Pose kann es sein, dass es in vielen anderen Posen stark verformt erscheint.

9.1.4.2 3DS-Max

In 3DS Max gibt es auch einen Weight-Paint-Mode. Diesen erreicht man, indem man sein 3D-Objekt selektiert und im Modify-Tab auf *Edit Envelopes* klickt. Weiter unten in diesem Reiter kann man auswählen, für welchen Knochen man die Gewichtungen bearbeiten will.

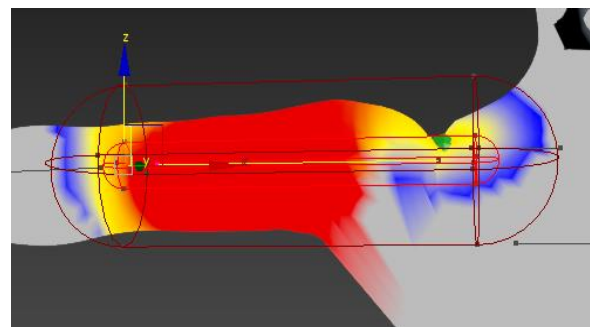


Abbildung 62: Weight-Paint-Mode (3DS Max)

Ebenfalls wie in Blender wird jetzt farblich

dargestellt, wie stark sich der ausgewählte Körperteil verformt. Zusätzlich gibt es in 3DS Max

noch den *Mirror Mode*, der ebenfalls weiter unten in diesem Reiter zu finden ist. Damit werden alle Änderungen der Gewichtungen auf die andere Körperhälfte gespiegelt.

9.1.4.3 Maya

In Maya arbeitet man ebenfalls mit dem Weight-Paint-Mode. Dazu wählt man zuerst sein gesamtes Objekt aus und wählt im Anschluss im Dropdown links oben *Rigging* aus, in der oberen Leiste erscheint dann ein *Skin-Menüpunkt*. Klickt man folglich auf *Paint Skin Weights* erscheint ein neues Fenster, in welchem man die Gewichtungen einstellen kann. Dabei kann man wieder auswählen, für welchen Körperteil man die Gewichtungen anpassen möchte. Setzt man unten bei *Use Color Ramp* ein Häkchen, bekommt man wieder die gewohnte Ansicht, in der Gewichtungen farblich dargestellt sind. Alles andere funktioniert genauso wie in Blender und 3DS Max.

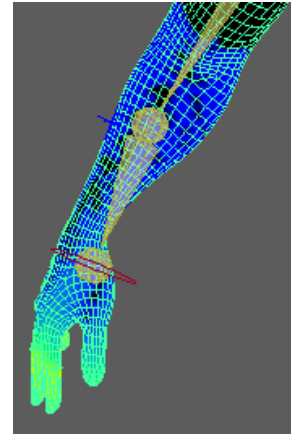


Abbildung 63: Weight Paint - Rechter Arm (Maya)

9.1.5 Kamera- und Lichtsetup

Da alle Übungen in Blender gerendert wurden, ist sowohl in diesem Kapitel als auch in Kapitel 9.1.7 keine spezifische Beschreibung für 3DS Max und Maya vorhanden.

9.1.5.1 Allgemein

Um ein Video rendern zu können, muss man erst eine Kamera platzieren, die festlegt welcher Ausschnitt gerendert werden soll. Das alleine reicht jedoch nicht, da die Render-Engine nicht weiß, wie hell oder dunkel sie das Objekt darstellen soll. Ohne Licht würde man nur eine schwarze 2D-Fläche von allen Objekten erhalten. Das Licht muss man daher so platzieren, dass es den gesamten Körper beleuchtet. Zusätzlich muss noch die Lichtstärke eingestellt werden, also wie hell es den Ausschnitt beleuchtet. Im Idealfall verwendet man nur ein Licht, das von schräg oben auf das Objekt leuchtet. Für aufwendigere Animationen sind aber auf jeden Fall mehrere Lichter notwendig.

9.1.5.2 Blender

In Blender ergibt sich somit folgende Einstellung:

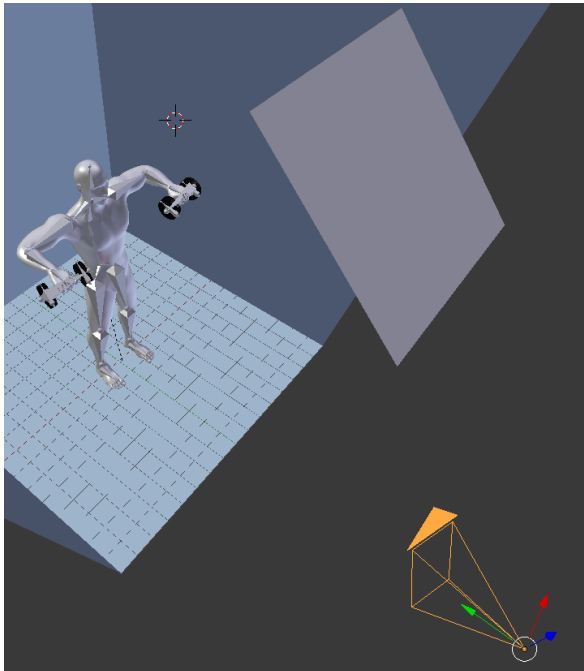


Abbildung 64: Kamera und Licht (Blender)

Die weiße Fläche stellt dabei ein Licht dar. Um so ein Licht zu erstellen, benötigt man nur ein *Plane*-Objekt (*Shift+A -> Mesh -> Plane*), das man so vergrößert, dass es in etwa den ganzen Körper beleuchtet. Danach muss man nur noch im Material-Tab unter *Surface den Emission-Shader* auswählen, es geht also von der Fläche ein Licht aus. Um weißes Licht zu erhalten reicht es, das Attribut Farbe auf dem Standardwert zu belassen. Zuletzt stellt man noch die Lichtstärke ein. Im Normalfall sollte ein Wert im Bereich von 3-5 reichen, um den gesamten Körper zu beleuchten. Es kommt natürlich auch auf die Größe des Lichts und die Entfernung vom Objekt an, im Bild oben beträgt die Lichtstärke zum Beispiel 4.

Jetzt muss nur noch die Kamera platziert werden und man kann die Animation rendern. Dazu fügt man eine neue Kamera hinzu (*Shift+A -> Camera*). Im Render-Tab unter Dimensions stellt man dann ein, ob man die Animation im Hochformat oder Querformat rendern möchte. Für Hochformat wählt man für X 1080 und für Y 1920, Querformat genau umgekehrt. Das obige Bild sollte im Hochformat gerendert werden, da das Modell steht. Als nächstes klickt man auf die Kamera und wählt unten unter View -> Kamera aus (*Numpad 0*). Jetzt kann man die Kamera wie gewohnt platzieren (rotieren, verschieben).

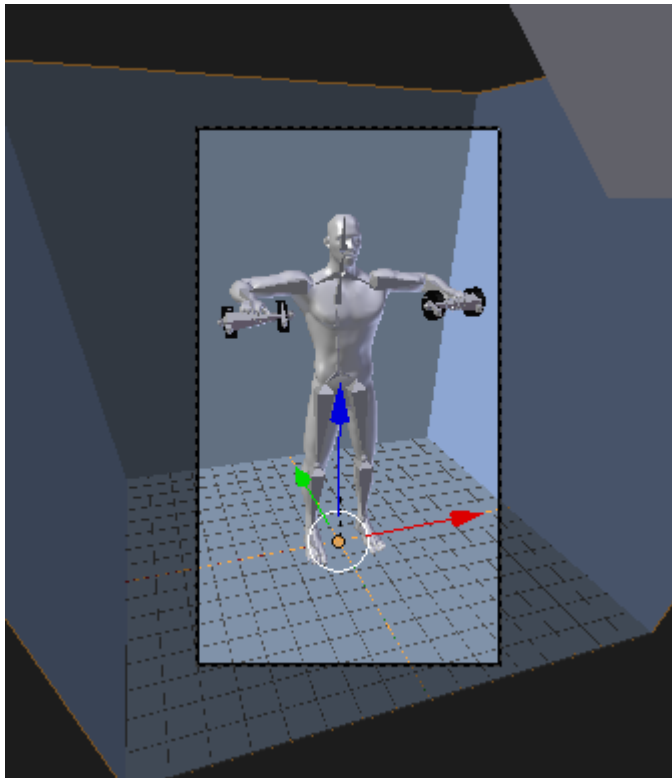


Abbildung 65: Kamera (Blender)

Jetzt ist man fertig und muss die Animation nur noch rendern.

9.1.6 Render Engines in Blender

9.1.6.1 Blender Render

Der Blender Render oder auch Blender Internal ist ein nicht fotorealistischer Renderer, der schon in den 90er-Jahren entwickelt wurde. Obwohl er einige Features wie Raytracing, Reflexionen und Subsurface Scattering - also die Streuung von Licht – beinhaltet, wird er heute nur noch selten verwendet, da man damit keine realistischen Bilder darstellen kann. Das bringt aber den Vorteil mit sich, dass er sehr schnell ist und zu Testzwecken bezüglich der Kamerapositionierung oder Ähnlichem verwendet werden kann. Auch für nicht fotorealistische Animationen wie Cartoons kann er gut verwendet werden.

9.1.6.2 Cycles Render

2011 erschien dann der physikalisch-basierte Cycles Render, welcher fotorealistische Bilder erzeugen kann. Er bietet eine Unmenge an neuen Features, welche unter [1] nachgelesen werden können.

(vgl. Trammell, 2016)

9.1.7 Render-Setup

9.1.7.1 Allgemein

Um eine Animation zu erhalten, muss man die erstellte Szene rendern. Vereinfacht gesagt bedeutet dies, dass aus der erstellten Szene mit den verschiedenen 3D-Objekten eine Animation erstellt wird. Das kann sehr viel Zeit in Anspruch nehmen und benötigt auch viel Rechenleistung der CPU.

Um Datenverlust zu vermeiden, sollte zuerst jedes Frame als Bild (vorzugsweise PNG) gerendert werden, somit bleiben die Bilder erhalten, wenn man zwischendurch abbrechen muss. Am Schluss werden diese dann zu einem Video zusammengefügt (MP4).

9.1.7.2 Blender

Da man realistische Bilder erzeugen möchte, wird als Erstes oben der Cycles Render ausgewählt. In Blender gibt es zum Rendern den *Render*-Reiter rechts. Hier hat man im vorigen Schritt bereits das Kameraformat richtig eingestellt. Jetzt kann man noch einstellen, wieviele Frames gerendert werden sollen und welche Framerate man verwendet. Eine Framerate

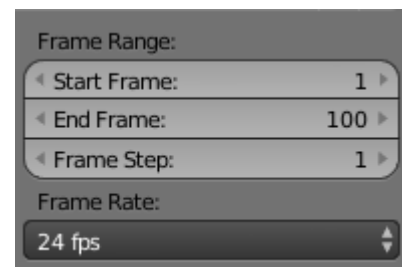


Abbildung 66: Frame-Settings (Blender)

von 24 FPS ist für Animationen ein Standard. Rendert man jetzt also Frame 1 bis 100, würde diese Animation knapp über 4 Sekunden dauern. Ein Spezialfall entsteht bei Übungen, welche eine Auf- und Ab oder Vor- und Zurück-Bewegung beinhalten, wie beispielsweise die Kniebeuge, bei der man zuerst runter und dann wieder rauf geht. Hier ist es nicht notwendig, die gesamte Animation zu rendern, da ja genau die gleichen Bewegungen bei der nächsten Hälfte der Animation benötigt werden. Dazu wurde eine einfache C#-Konsolenanwendung entwickelt, welche die erste Hälfte an Bildern kopiert und die Nummern hinter dem Namen automatisch vertauscht.

```
private static void CopyFiles(string folderPath)
{
    string[] files = Directory.GetFiles(folderPath);
    for(int i=files.Length-2, j=3; i>=0; i--, j = j+2)
    {
        string sourceFile = Path.Combine(folderPath, files[i]);
        int number = i + j;
        string ending;
```



```

if(number < 10)
{
    ending = "0" + number + ".png";
} else
{
    ending = "" + number + ".png";
}
string destFile = Path.Combine(folderPath, sourceFile.Substring(0, sourceFile.Length - 6)
+ ending);

File.Copy(sourceFile, destFile);
}
}

```

Mit der Methode `Directory.GetFiles(folderPath)` werden alle Dateien aus diesem Pfad ausgelesen und in die Liste gespeichert. Diese werden dann in der folgenden Schleife durchlaufen und kopiert. So wird beispielsweise aus `Kniebeuge0001.png` `Kniebeuge0100.png`.

Unter Output kann man jetzt einstellen, welches Format die gerenderte Animation haben soll und wo sie hingespeichert wird. Hier wählt man zuerst PNG aus und gibt einen Speicherort an, hierzu erstellt man vorzugsweise einen Ordner mit dem

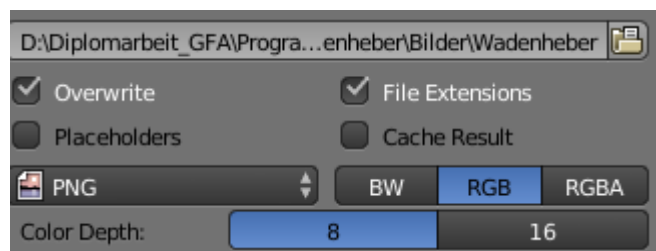


Abbildung 67: Output-Settings (Blender)

Namen *Bilder*, in den alle gerenderten Bilder kommen. Weiters kann man noch die Farbtiefe und den Farbmodus – Schwarzweiß, RGB oder RGB mit Alphakanal – einstellen. Da die Animationen in der mobilen App abgespielt werden, wird hier RGB mit einer Farbtiefe von 8 gewählt, damit die Datei nicht zu groß wird. Jetzt muss man nur noch oben auf *Render Animation* klicken und es werden die eingestellten Frames als PNG gerendert.

Beim Wadenheber musste nur die Hälfte gerendert werden, da die zweite Hälfte wieder den gleichen Ablauf hat. Also verwendet man jetzt das C#-Programm, welches die Dateien kopiert, mit dem folgenden Aufruf:

```
C:\Diplomarbeit\RenderHelper\bin>dotnet FileCopier.dll
```

```
C:\Diplomarbeit\Computeranimationen\Blender\Renders\Wadenheber\Bilder
```

Es werden jetzt die Bilder im angegebenen Pfad kopiert und man hat eine fast fertige Animation.

Jetzt fehlt nur noch, dass man sie in einen MP4-Film umwandelt. Dazu geht man oben in die *Video Editing* Ansicht. Hier stellt man sich auf Frame 0 und klickt auf Add -> Image. Jetzt wechselt



Abbildung 68: H.264-Render (Blender)

man in den Dateipfad, in dem die Bilder gespeichert sind, wählt alle aus (A) und fügt sie hinzu. Es wurde jetzt ein Image-Strip hinzugefügt, der von Frame 0 bis zum vorhin angegebenen Frame geht. Danach geht man wieder in die Default Ansicht und stellt im Render-Reiter unter Output den Pfad um und das Dateiformat auf H.264. H.264 ist eine leistungsstarke Videokompressions-Technologie, welche hochauflösende Videos mit niedriger Bit-Rate verschlüsseln kann. Unten wählt man noch das Dateiformat der Animation aus, also MP4. Klickt man jetzt auf *Render Animation*, wird die Animation als MP4 erstellt und man hat seine fertig verwendbare Animation.

9.1.8 Integrieren der Animationen in die App

Um die Animationen in der HTML5-App verwenden zu können, muss man sie persistent an einem von überall erreichbarem Ort abspeichern. Da man in relationale Datenbanken nicht unbedingt große Binärdaten wie MP4-Videos einfügen sollte, sind die Animationen auf Google Drive abgelegt und freigegeben. Diese kann man in der HTML5-App ganz einfach mit dem *video*-Tag einbinden. Der *video*-Tag wird verwendet, um Videos eines beliebigen Formats, in diesem Fall MP4, in eine Webseite einzubetten.

```
<video style="width: 310px; display: block; margin: auto;" autoplay loop controls>
```

```
  <source src="{{animationUrl}}" type='video/mp4' />
```

```
</video>
```

Im *src*-Attribut ist die Variable eingetragen, in der der Pfad zur jeweiligen Übung spezifiziert ist. Diese wurde dafür in TypeScript von der Datenbank abgerufen und in einer Variable gespeichert. Wie genau man einer Variable die Werte zuweist, ist im Punkt 9.2.2 genauer beschrieben. Dazu hat jede Übung in der Datenbank den Google Drive Link zur jeweiligen MP4-Animation eingetragen. Außerdem ist im *style*-Attribut eingetragen, dass die Animation

zentriert dargestellt werden soll und wie breit die Animation sein soll, die Höhe wird dazu entsprechend automatisch skaliert. Das Attribut *autoplay* sorgt dafür, dass das Video automatisch abgespielt wird, wenn es geladen wurde und loop führt dazu, dass es in einer Schleife immer wieder abgespielt wird. Controls fügt unten eine Leiste hinzu, in der es einen Play-Button und den aktuellen Zeitpunkt im Video anzeigt.

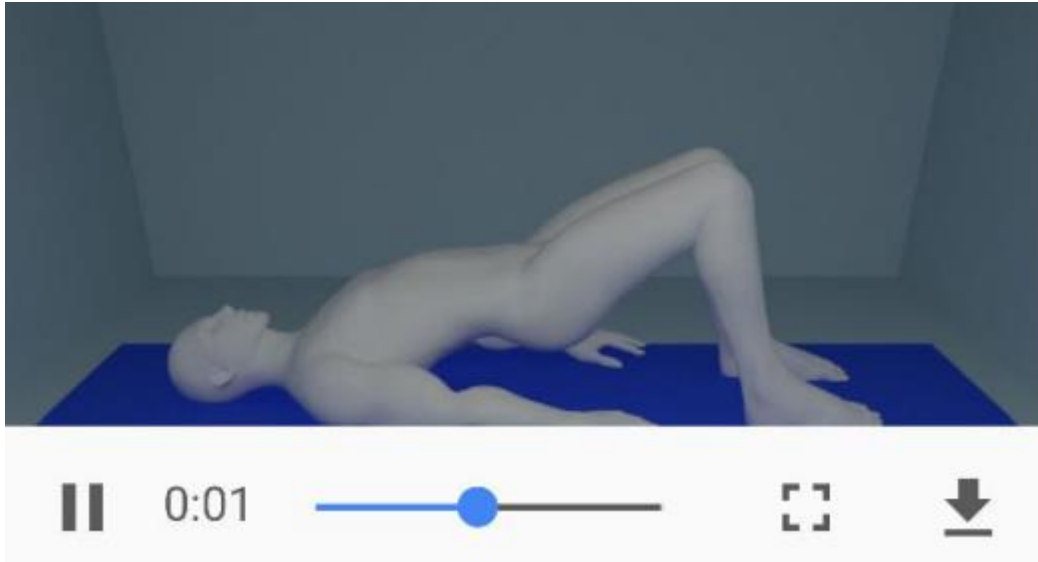


Abbildung 69: Video-Tag (Animation)

Eine genauere Darstellung der Animation ist in Abbildung 15 (Kapitel 7.7) ersichtlich.

9.1.9 Animationen mit Motion Capture

Eine weitere Möglichkeit, Computeranimationen oder auch komplett computeranimierte Filme zu erstellen, ist Motion Capture. Motion Capture ist eine der populärsten CGI-Techniken (computer-generated imagery) und wurde bereits in Filmen wie *Der Hobbit* oder *Avatar – Aufbruch nach Pandora* verwendet, wo menschliche Schauspieler durch Science-Fiction-Figuren ersetzt wurden.

Das Prinzip hinter Motion Capture ist es, menschliche Bewegungen aufzunehmen, zu kopieren und diese zu realistischen Computer-Simulationen zu transformieren. Es gibt mehrere Techniken, Motion Capture zu realisieren. Eine Möglichkeit ist die Verwendung von LED-Lichtern. Dabei zieht der Schauspieler im Normalfall einen Anzug an, auf dem mehrere LED-Lichter befestigt sind. Die Bewegung dieser LED-Lichter wird von der Motion Capture Software – mit Hilfe von mehr als 100 Kameras - aufgenommen und kann so am Computer direkt betrachtet werden.

Nachfolgend sind einige Vorteile gegenüber herkömmlichen 3D-Animationen aufgelistet:

- ✓ Echtzeit-Ergebnisse der Animationen
- ✓ Deutlich weniger Arbeit als Animationen per Hand zu erstellen
- ✓ Schnellere, einfachere und genauere Darstellung von menschlichen Bewegungen

Natürlich bringt das aber auch einige Nachteile mit sich. Es wird spezielle Motion Capture Software und Hardware benötigt, welche sehr leicht in den fünf- bis sechststelligen Eurobereich gehen können. Daher lohnt es sich nicht für kleinere Unternehmen – und schon gar nicht für unsere Diplomarbeit – diese Geräte anzuschaffen. Außerdem benötigt man einen Raum in dem man die Kameras montieren kann, es sollte hier auch genügend Platz vorhanden sein. Im Normalfall ist es noch notwendig, die Animationen nach dem Motion Capture Verfahren trotzdem etwas nachzubearbeiten. (vgl. computerstories.net, 2011)



Abbildung 70: Motion Capture - Planet der Affen (denofgeek.com, 2016)

9.2 App

9.2.1 Installation von Angular und Ionic

Vor der Installation von Angular bzw. Ionic muss man zu allererst Node.js und npm installieren. Hierbei ist es wichtig zu bedenken, dass Angular und Ionic zumindest Node.js in der Version 6.9.x und npm mindestens in der Version 3.x.x benötigen. Hat man bereits aus früheren Anwendungen bzw. Entwicklungen npm oder Node.js installiert kann man die Versionen mit den Kommandozeilenbefehlen *node -v* bzw. *npm -v* prüfen:

Ionic wird nun mit einem einfachen Kommandozeilenbefehl installiert. Um Problemen aus dem Weg zu gehen empfiehlt es sich die Eingabeaufforderung als Administrator auszuführen. Der Kommandozeilenbefehl lautet *npm install -g ionic cordova*. *-g* wird verwendet um den Aufruf global am Gerät durchzuführen, hierbei ist zu bedenken, dass man Administrator-Rechte braucht um einen Aufruf global durchzuführen. Als nächstes gibt man an ob man die Ionic-Pro-SDK dazu installieren möchte.

Sobald alle Befehle reibungslos abgeschlossen sind, kann es auch schon losgehen die App zu entwickeln.

9.2.2 Anbindung des REST-Webservice in Angular

9.2.2.1 CORS Anywhere

Um während der Implementierungsphase dauerhaft testen zu können, hat sich jedes Projektmitglied eine lokale Testdatenbank eingerichtet. Diese Datenbank wurde mittels XAMPP lokal gehostet. Dies hat den Vorteil, dass jeder eigene Testdaten erstellen konnte und somit sein System auf Herz und Nieren prüfen konnte.

Vor dem Zugriff auf die Datenbank, ist es weiters nötig, einen lokalen Proxy einzurichten, da die Datenbank ansonsten mit dem automatisch erstellten Webservice von Angular einen Konflikt auslöst. Für meinen Teil habe ich das Programm „CORS Anywhere“ verwendet.

Die Konfiguration dieses Programms sieht wie folgt aus:

```
var host = process.env.HOST || '0.0.0.0';
// Listen on a specific port via the PORT environment variable
var port = process.env.PORT || 8101;

var cors_proxy = require('cors-anywhere');
cors_proxy.createServer({
  originWhitelist: [] // Allow all origins
}).listen(port, host, function () {
  console.log('Running CORS Anywhere on ' + host + ':' + port);
});
```

Ausgeführt wird dieses Programm mit dem Programm NodeJS, siehe Kapitel 8.1.7.

9.2.2.2 Zugriff in Angular in der Methode `ngOnInit`

`OnInit` ist ein Modul, welches durch den Import des npm Package *@angular/core* eingebunden werden kann. *ngOnInit* ist eine Methode, die sofort aufgerufen wird, speziell noch bevor irgendwelche Kind-Methoden aufgerufen werden. (vgl. angular.io, 2018)

Um nun einen GET Aufruf abzusetzen, muss man den Service *HttpClient* zuerst in der Datei *app.module.ts* hinzufügen. Dies erfolgt durch folgenden Import-Befehl:

```
import { HttpClient } from '@angular/common/http';
```

Als nächstes erstellt man für die Entität, welche man mit dem GET-Aufruf aufrufen möchte, ein Interface mit Properties für alle Attribute. Es ist darauf acht zu geben, dass man exakt dieselben Namen wählt, die von der API verwendet werden.

```
interface IPerson {  
  
    PersonID: number;  
    prename: string;  
    surname: string;  
    birthdate: any;  
    street: string;  
    postcode: string;  
    Town: string;  
    Phonenumber: string;  
    Email: string;  
    sex: string;  
}
```

Durch das Schlüsselwort *export* kann man das Interface öffentlich machen und somit von anderen Klassen darauf zugreifen. Für meinen Teil habe ich beschlossen, die GET-Aufrufe *async* zu machen, da es die Lesbarkeit des Codes erhöht und man den ausgelesenen Wert, mit der Hilfe des Schlüsselwortes *await*, direkt in eine Variable speichern kann, ohne vorher noch weitere Methoden auszuführen. Da man den Konstruktor einer Klasse nicht *async* setzen kann und ich trotzdem direkt beim erstmaligen Aufruf die Daten auslesen wollte, kam die Methode *ngOnInit* zum Einsatz. Ein Aufruf um zum Beispiel alle Customer und eine Person mit der *PersonenID* 1 auszulesen und diese in ein Array von *ICustomer* bzw. in eine Variable *IPerson* zu speichern, würde so aussehen:

```
public customers: ICustomer[];
public person: IPerson;

constructor(public http: HttpClient) {}

public async ngOnInit() {

  const result : any = await
  this.http.get(<a href="http://localhost:8101/http://localhost:8081/api/src/customer">http://localhost:8101/http://localhost:8081/api/src/customer</a>)
  .toPromise();

  const resultPerson : any = await
  this.http.get(<a href="http://localhost:8101/http://localhost:8081/api/src/person/1">http://localhost:8101/http://localhost:8081/api/src/person/1</a>)
  .toPromise();

  this.customers = result.data;
  this.person = resultPerson.data;
}
```

9.2.3 Login

Damit der Benutzer überhaupt die Gelegenheit hat, um die App zu benutzen muss er sich einloggen. Hierzu muss ihm der Physiotherapeut Login-Daten zur Verfügung stellen. Dies erfolgt beim ersten Besuch des Kunden im Therapiezentrum, der Benutzer wird gefragt welche Emailadresse er benutzen möchte und schließlich wird er noch gefragt ob er ein Passwort eingeben kann, mit diesem kann er sich später bei der App anmelden. Dieses Passwort wird mittels SHA512-Algorithmus verschlüsselt in der Datenbank gespeichert.

Der Benutzer der App gibt nun einfach seine Emailadresse und sein Passwort in die jeweiligen Eingabefenster ein und klickt Login, im Hintergrund wird nun sein Passwort wieder mittels SHA512-Algorithmus verschlüsselt und mit der Datenbank verglichen.

9.2.3.1 SHA512

Da TypeScript bzw. Angular keine automatisch eingebundene Bibliothek hat, die einen SHA512-Algorithmus implementiert hat, musste ich auf eine externe Bibliothek zurückgreifen. Für diese Diplomarbeit wurde die Bibliothek *js-sha512* verwendet, da sie bereits Methoden implementiert hat, um sich nur durch einen einzigen Methodenaufruf den Hash generieren zu lassen. (vgl. emn178, 2017)

Zu Beginn der Entwicklung wollte ich die Bibliothek *crypto-js* verwenden, jedoch traten bei der Generierung der Hashwerte Probleme auf, da die Bibliothek keine Methode hat um sich Hashes automatisch generieren zu lassen. Sie gab nur ein Byte-Array zurück, welches man durch selbst implementierten Code, in einen Hash umwandeln müsste.

9.2.4 Bilderübertragung mittels REST-Webservice in Angular

Da wir nicht alle Bilder lokal speichern wollen, weil es einfach zu viel Speicherplatz in Anspruch nehmen würde, speicherten wir die Bilder und einen Link zu den Animationen in der Datenbank, auf diese griffen wir wie üblich mittels REST zu.

Bilder werden in unserer Datenbank als BLOB gespeichert und sind Base64 kodiert. Da man in HTML in einem Image-Tag einfach eine Base64-Zeichenkette, welche automatisch zu einem Bild konvertiert wird, angeben kann, war es einfach, die Bilder abzurufen. Durch einen einfachen GET-Request (wie bereits unter 9.2.2 angesprochen) kann man nun das Bild abfragen. Zur Verdeutlichung nachfolgend ein Beispiel.

```
var picture: string;

const imgResult: any = await
this.http.get("http://localhost:8101/http://localhost:8081/api/src/picture/1").toPromise();

this.picture = "data:image/jpeg;base64," + imgResult.data[0].pic;
```

Nun hat man in *this.picture* einen base64-String, davor steht „data:image/jpeg;base64,“ dies deshalb, damit der HTML-Tag „img“ weiß, dass es sich um einen String, nicht wie normal üblich um einen Pfad zu einem Bild handelt. Zur Verdeutlichung wird im nachfolgenden Beispiel der Aufruf in der HTML-Datei gezeigt.

```

```

9.2.5 Erstellung eines Timers in Angular

Da viele Übungen zeitbasiert sind, macht es natürlich Sinn in unsere App einen Timer einzubauen. Dieser Timer ermöglicht ein effizienteres Training, da der Benutzer somit nicht die App verlassen muss, um einen Countdown oder Ähnliches zu starten, er kann per Knopfdruck einfach entscheiden, wann er mit der Übung beginnen will.

Im nachfolgenden Kapitel werde ich die Schritte zur Entwicklung eines Timers erläutern.

9.2.5.1 Observable-Timer

Um einen Timer zu programmieren, muss man die Klasse *TimerObservable* der ReactiveX-Bibliothek (vgl. reactive.io, 2018) importieren.

```
import { TimerObservable } from 'rxjs/observable/TimerObservable';
```

Als nächstes habe ich eine Variable *timer* erstellt und diese mit dem Ergebnis des Methodenaufruf *create* des Objekts *TimerObservable* belegt.

```
let timer2 = TimerObservable.create(0,1000).subscribe(t => {this.tick -= 1;

    if(this.tick<=0){

        timer2.unsubscribe();
        alert.present();
        this.wiederholungenleft -=1;
        this.isenabled=true;
        if(this.wiederholungenleft<=0){
            alert.dismiss();
            alert2.present();
            this.setsleft -= 1;
            this.wiederholungenleft += this.taskrepeats;
        }
        if(this.setsleft <= 0 ){
            alert.dismiss();
            alert2.dismiss();
            alert3.present();
            this.setsleft += this.tasksets;
        }
    }
});
```

Der Timer zieht von *this.tick* alle 1000ms 1 ab, bis der Wert 0 erreicht. *this.tick* ist zu Beginn der Wert der Dauer der Übung in Sekunden. Sobald man 0 erreicht, wird der Timer mit der Methode „unsubscribe()“ beendet und ein Alert-Fenster öffnet sich, welches den Benutzer darauf hinweist, dass die Übung fertig ist.

9.2.6 Verwendete Ionic-Komponenten

Ionic verwendet sogenannte Komponenten, um schnell ein geeignetes User-Interface zu erstellen. Jedes dieser Ionic-Komponenten ist individuell konfigurierbar und kann mit den Angular-Direktiven angepasst werden. Außerdem passen sie sich immer an das verwendete Betriebssystem des mobilen Geräts an. Das bedeutet, dass eine Komponente zum Beispiel unter Android anders aussehen kann als unter iOS. Nachfolgend erläutere ich, welche Komponenten ich verwendet habe.

Alle nachfolgenden Komponenten sind unter folgendem Link zu finden:

<https://ionicframework.com/docs/components/>

9.2.6.1 Ion-List

Eins der am häufigsten verwendeten Komponenten ist die sogenannte Ion-List, bei dieser Arbeit wurde sie unter anderem bei der Anzeige der Übungen oder bei der Anzeige der Details zu den jeweiligen Übungen. Ion-Lists kann man weiters mit vielen Komponenten, zum Beispiel Ion-Thumbnail, kombinieren. Eine Ion-List besteht meist aus Ion-Items, welche wiederum die einzelnen Teile der Liste repräsentieren.

```
<ion-list>
```

```
  <ion-item text-wrap>{{taskdescription}}</ion-item>
```

```
  <ion-item text-wrap *ngIf="!timerzero"> Dauer: {{taskduration}} Sekunden<br></ion-item>
```

```
  <ion-item text-wrap> {{tasksets}} x {{taskrepeats}} Sätze<br></ion-item>
```

```
  <ion-item text-wrap> Schwierigkeitsgrad: {{level}}<br></ion-item>
```

```
</ion-list>
```

9.2.6.2 Ion-Toolbar

Eine Ion-Toolbar kann man entweder im Header oder im Footer platzieren, je nachdem ob man die Toolbar oberhalb oder unterhalb des eigentlichen Inhalts haben möchte. Eine Toolbar bleibt immer fixiert in ihrer Position und wird meist für Suchfelder verwendet. Man kann direkt im ion-toolbar-Tag angeben, ob ein Button *menuToggle* erstellt werden soll.

```
<ion-header>
  <ion-toolbar color="secondary">
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Meine Trainingspläne</ion-title>
  </ion-toolbar>
</ion-header>
```

9.2.6.3 Ion-Thumbnail

Ion-Thumbnail wird ausschließlich in Kombination mit der Ion-List verwendet, da es im Endeffekt nur eine Erweiterung der bestehenden Ion-List ist. Konkret wird die List dahingehend erweitert, dass man neben dem Text auch noch ein Bild einfügen kann und es so optisch ansprechender gestaltet.

```
<ion-list>
  <ion-item *ngFor="let c of currentTrainingplan" (click)="openTrainingsplan(c.PlanID)">
    {{c.Name}}
    <ion-thumbnail item-start (click)="openTrainingsplan(c.PlanID)">
      
    </ion-thumbnail>
  </ion-item>
</ion-list>
```

9.2.6.4 Ion-Label

Labels werden innerhalb von Ion-Items verwendet um zum Beispiel ein ion-Input zu beschreiben. Ion-Labels können entweder *floating*, *fixed* oder *stacked* sein. Je nach Attribut verhält sich der Text im Ion-Label anders. Im nachfolgenden Beispiel würde der Text zum Beispiel nach oben rücken sobald man mit der Eingabe beginnt.

```
<ion-item>
```

```
  <ion-label floating>E-Mail</ion-label>
```

```
  <ion-input type="text" required [(ngModel)]="logData.username"
    ngControl="username" name="username" value=""></ion-input>
```

```
</ion-item>
```

In diesem Beispiel soll der Benutzer in einem Eingabefenster seine Emailadresse eingeben. Damit er weiß, dass die Emailadresse gemeint ist, gibt man den Text „E-Mail“ mit Hilfe eines Ion-Labels an.

9.2.6.5 Ion-Card

Eine weitere sehr vielfältige Komponente sind die Ion-Cards. Ion-Cards sind ein toller Weg um Informationen von der Masse herauszuheben, sie können sogar animiert werden. Außerdem ist diese Komponente ideal mit Ion-Lists kombinierbar.

```
<ion-card>
```

```
  <ion-card-header class="header">
```

```
    Meine Übungen:
```

```
  </ion-card-header>
```

```
  <ion-list>
```

```
    <button ion-item *ngFor="let r of resultTasks" (click)="openDetailedTask(r.TaskID)">
```

```
      <div>
```

```
        {{r.Name}}<br>
```

```
        <div class="desc">{{r.Description}}</div>
```

```
      </div>
```

```
    </button>
```

```
  </ion-list>
```

```
</ion-card>
```

9.2.6.6 Ion-Alert

Ion-Alerts funktionieren etwas anders als die bisher angesprochenen Komponenten; Alerts unterscheiden sich zum Beispiel schon beim Aufruf, da sie nicht, wie zum Beispiel Ion-Lists, nur durch ein HTML-Tag aufgerufen werden, sondern bereits im Typescript-File konfiguriert

werden müssen. Zu Beginn müssen AlertController importiert werden und dem Konstruktor als Parameter mitgegeben.

```
import { AlertController } from 'ionic-angular';

constructor(public alertCtrl: AlertController) {}
```

Da Alerts meist durch einen Klick auf einen Button ausgelöst werden sollen, wird eine Methode angelegt, welche dann durch den Button-Aufruf ausgeführt wird.

```
start(){
    let alert = this.alertCtrl.create({
title: ‚Satz fertig‘,
    subTitle: ‚Sie haben diesen Satz abgeschlossen.‘,

    buttons: [‘OK’]

    });
}
```

Alerts sind sehr individuell gestaltbar, folgende weitere Optionen kann man einstellen:

Property	Beschreibung
message	Eine Nachricht die erscheinen soll
cssClass	Wenn man den Alert grafisch anpassen möchte, zum Beispiel die Farbe des Hintergrundes ändern will, dann kann man eine CSS-Klasse angeben.
inputs	Bei Inputs kann man, wie auch bei Buttons, mehrere Optionen einstellen: type, name, placeholder, value, label, checked und id
enableBackdropDismiss	Wenn man außerhalb des Alerts klickt, wird der Alert verworfen
Button-Property	Beschreibung
text	Da es mehrere Buttons mit unterschiedlichen Texten geben kann, kann man die text-Property mehrmals einfügen, um so weitere Buttons zu erstellen
handler	Hier gibt man an, was passieren soll, wenn der Benutzer den Button drückt. (Meist wird ein Methodenaufruf gestartet.)
cssClass	Siehe oben: nur dass der Button diesmal verändert wird.
role	Kann die Werte null oder cancel haben.

Hat man den Alert schließlich erstellt, muss man ihn noch aufrufen, dies erfolgt durch die Methode *present()*:

```
alert3.present();
```

In meinem Teil verwendete ich Alerts um den Benutzer darauf aufmerksam zu machen, dass er eine Übung abgeschlossen hat bzw. dass der Timer abgelaufen ist (siehe 9.2.5).

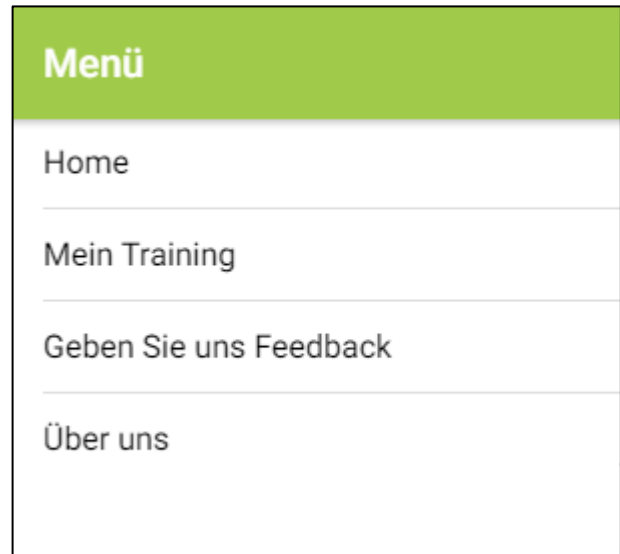


Abbildung 71: Ionic-Menü App

9.2.6.7 Ion-Button

Der Ion-Button unterscheidet sich von einem normalen Button dahingegen, dass er mehr Attribute hat. Ein Button soll per Mausklick etwas ausführen, in diesem Beispiel die Methode *start()*.

```
<button ion-button block (click)="start()" [disabled]="!isEnabled" color="placeholder">Starte Übung</button>
```

9.2.6.8 Ion-Icon

Ionic stellt eine breite Palette von Icons, sogenannten *Ionicons* zur Verfügung. Die meisten Icons sehen unterschiedlich aus, je nachdem auf welchem Betriebssystem man die App verwendet. Manche Ionicons kann man auch durch Attribute spezifizieren. Das nachfolgende Icon ist ein Menü Icon, welches sich an das verwendete Betriebssystem anpasst.

```
<ion-icon name="menu"></ion-icon>
```

9.2.7 Ionic-Menu Component

Die Menü-Komponente funktioniert etwas anders als die anderen Ionic-Komponenten, da sie schließlich für alle Seiten gleich aussehen soll bzw. von allen Seiten aus erreichbar sein soll.

Dies ist der Grund dafür, dass man sich bereits vor Erstellung einer App Gedanken darüber machen sollte, wie man seine App gestalten will. Für diese Arbeit wurde bereits beim Anle-

gen der App ein Seitenmenü gewählt, deshalb generierte Ionic automatisch eine kleine Applikation mit einem Menü, welches man ständig erweitern konnte. In der `app.component.ts` wird ein einfaches Array mit den Inhalten *title* und *component* erstellt. Befüllt wird es schließlich mit den Seiten, die man als Menüauswahl haben möchte und einem Titel, der angezeigt werden soll.

```
pages: Array<{title: string, component: any}>;

constructor(public platform: Platform, public statusBar: StatusBar, public splashScreen:
SplashScreen) {

    this.initializeApp();

    this.pages = [
        { title: 'Home', component: HomePage },
        { title: 'Mein Training', component: MytasksPage },
        { title: 'Geben Sie uns Feedback', component: FeedbackPage},
        { title: 'Über uns', component: AboutPage}
    ];
}
```

In der HTML-Datei `app.html` wird schließlich das äußere Erscheinungsbild festgelegt und mit Klick auf einen Titel die jeweilige Seite aufgerufen. Hier kann man weitere Dinge wie Titel des Menüs, Hintergrundfarbe usw. angeben.

```
<ion-menu [content]="content" color="secondary">

<ion-header>
    <ion-toolbar color="secondary">
        <ion-title>Menü</ion-title>
    </ion-toolbar>
</ion-header>
<ion-content>
<ion-list>
    <button menuClose ion-item *ngFor="let p of pages" (click)="openPage(p)">
```



```
        {{p.title}}  
      </button>  
</ion-list>  
</ion-content>  
</ion-menu>
```

9.3 Feedbackverwaltung in der App

9.3.1 Allgemein

Die Feedbackverwaltung bietet dem Therapeuten die Möglichkeit Feedbacks zu erstellen. Der Kunde wiederum kann diese Feedbacks dann beantworten. Die Feedbacks werden nach der Beantwortung automatisch ausgewertet. Um diesen Teil umzusetzen war es nötig die Datenbank und die App zu erweitern

9.3.2 Planung Feedback

Um das Feedback auch auswerten zu können ist es nötig eindeutige Antworten zu bekommen. Aus diesem Grund gibt es 3 Möglichkeiten eine Frage zu erstellen:

1. Bewertung zwischen 1 – 5
 - a. Beispielfrage: Wie zufrieden sind sie mit der Betreuung? (1 nicht zufrieden, 5 sehr zufrieden)
2. Wahr oder Falsch
 - a. Beispielfrage: Sind Sie Raucher? (Ja oder Nein)
3. Freie Textantwort
 - a. Haben Sie Veränderungsvorschläge für das Therapiezentrum?

Fragemöglichkeiten 1 und 2 sind auswertbar da beide immer über bestimmte Werte verfügen. Die 3 Möglichkeit ist nicht direkt auswertbar, sondern ermöglicht dem Kunden lediglich eine genauere Antwort.

9.3.3 Datenbank

In der Datenbank mussten insgesamt 5 neue Tabellen erzeugt werden:

- Answer
 - Jede Antwort ist einem Kunden, einem Feedback und einer Frage mittels Fremdschlüssel eindeutig zugewiesen
 - Die Antwort beinhaltet außerdem die eigentliche Antwort entweder als String oder Integer
- Feedback
 - Jedes Feedback hat einen Namen, welcher das Thema des Feedbacks beschreibt
- FeedbackDoneByUser
 - Wenn ein Feedback von einem Kunden abgeschlossen worden ist, wird hier eingetragen welcher Kunde welches Feedback erledigt hat
- FeedbackQuestions
 - Ist eine Assoziationstabelle in der für jedes Feedback die Fragen gespeichert werden
- Question
 - Jede Frage hat die eigentliche Frage als String und die Art der Frage (1,2 oder 3) als Integer gespeichert

9.3.4 Visualisierung Menü

In dem unteren Bild ist das Menü für die Feedbackverwaltung zu sehen. Das untere Rechteck ist nur für die Physiotherapeuten zugänglich. Der Kunde hat nur Zugriff auf das obere Rechteck, denn er darf die Fragen nur beantworten.

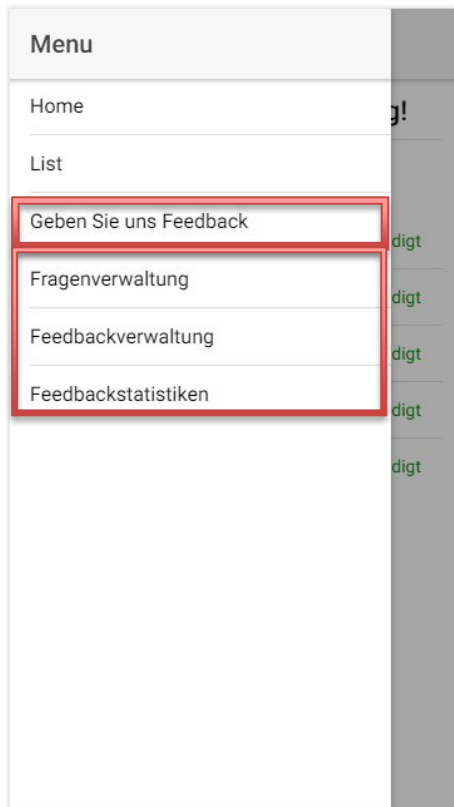


Abbildung 72 Feedbackmenü

9.3.5 Visualisierung Feedbackverwaltung Kunde

Im unteren Bild ist die Ansicht für den Kunden zu sehen. Jeder Kunde hat eine Liste mit Feedbacks vor sich. Wobei jeder Listeintrag den Namen des Feedbacks und den Zustand (Unerledigt oder Erledigt) anzeigt.

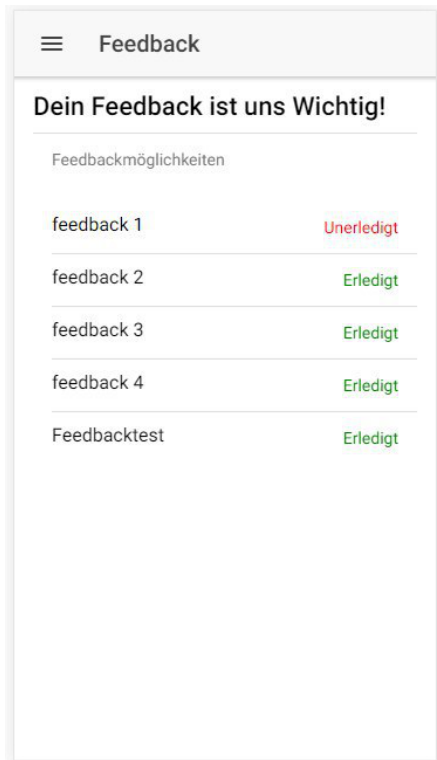


Abbildung 73 Feedbackverwaltung Kunde

Wenn sich der Kunde nun für ein Feedback entschieden hat drückt er auf das jeweilige Listitem. Dadurch wird ein Event ausgelöst durch welches das Menü erscheint:



Abbildung 74 Feedbackmenü Kunde

Nun hat der Kunde 3 Möglichkeiten:

- ✓ Bearbeiten
 - Ein Feedback kann auf zwei Arten bearbeitet werden
 - Neu
 - Das Feedback wird zum Ersten Mal ausgefüllt
 - Erneut
 - Das alte Feedback wird gelöscht und es wird erneut beantwortet
- ✓ Löschen
 - Löscht die Antworten auf das Feedback die der Kunde gegeben hat
- ✓ Abbrechen
 - Bricht die Aktion ab

Wenn der Kunde nun auf den Button „Bearbeiten“ gedrückt hat kommt er auf die nächste Seite:

Abbildung 76 Beantwortung Feedback 1

Abbildung 76 Beantwortung Feedback 2

Alle Fragen die einem Feedback zugewiesen wurden, werden hier dargestellt.

Fragenart 1:

- Schieberegler zwischen 1 und 5

Fragenart 2:

- Checkbox (checked = wahr, unchecked = falsch)

Fragenart 3:

- Eine beliebig lange Antwort (der Kunde kann aber dieses Feld auch leer lassen)

Nachdem der Kunde auf den Button Speichern gedrückt hat, werden alle Eingaben gespeichert und das Feedback als „Erledigt“ markiert. Danach folgt die Rückleitung auf die Menüseite.

9.3.6 Visualisierung Feedback Therapeut

9.3.6.1 Allgemein

Dem Therapeuten muss es möglich sein, Fragen und Feedbacks zu verwalten, sie also zu erstellen, bearbeiten und zu löschen. Außerdem sollen die Feedbacks automatisch ausgewertet werden.

9.3.6.2 Fragenverwaltung

Dieses Bild zeigt die Fragenverwaltung. In dieser Fragenverwaltung können Fragen hinzugefügt, bearbeitet und gelöscht werden. Der Aufbau unterscheidet sich minimal von der Ansicht des normalen Kunden. Beide Pages haben eine Liste als Grundelement. Der Unterschied ist jedoch, dass der Liste in der Fragenverwaltungspage ein statisches Element („Frage hinzufügen“) hinzugefügt wurde. Durch Icons ist am rechten Rand der Liste außerdem die Art der Frage ersichtlich.

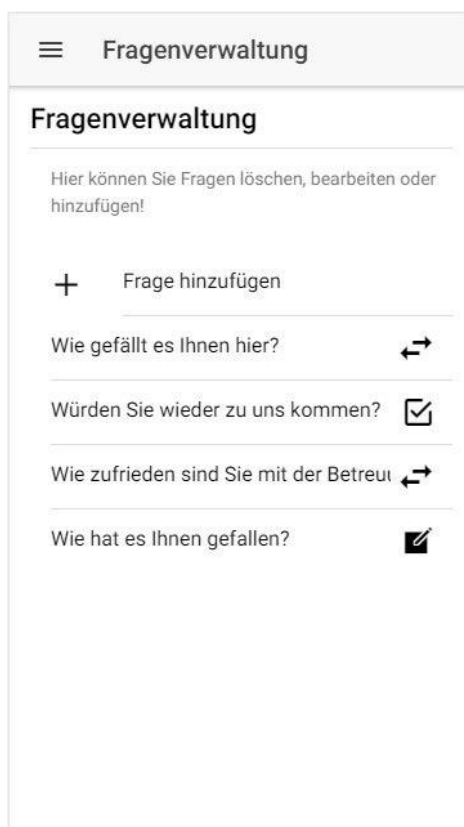


Abbildung 77 Fragenverwaltung Therapeut

Der Therapeut kann jetzt entweder eine neue Frage anlegen indem er auf das statische Element „Frage hinzufügen“ drückt, oder er öffnet das vorher bereits gezeigte Menü indem er auf ein Listenelement klickt.

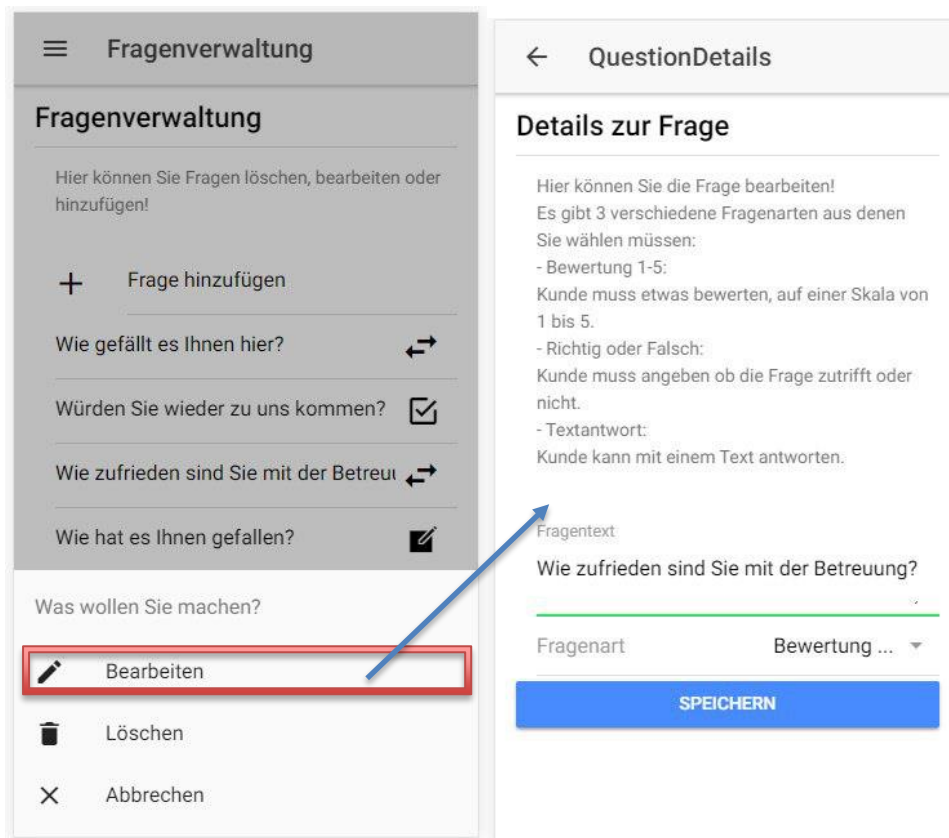


Abbildung 79 Fragenverwaltung Menü

Abbildung 79 Fragenverwaltung Bearbeiten

Wenn der Therapeut auf das Listenitem „Frage verwalten“ drückt öffnet sich auch die Seite auf Abbildung 77, diesmal ohne „Fragentext“ und „Fragenart“. Diese müssen neu vom Benutzer gewählt werden.

Die Fragenart ist ein Dropdown in der es nur 3 Möglichkeiten gibt:

- Bewertung 1-5
- Richtig oder Falsch
- Textantwort

9.3.6.3 Feedbackverwaltung

Das Menü der Feedbackverwaltung gleicht dem der Fragenverwaltung exakt, bis auf die Icons. Denn Icons wurden hier nicht benötigt.

Somit ist es auch wieder dasselbe Schema wie bei der Fragenverwaltung. Es gibt 1 statisches Element am Anfang der Liste, mit dem man ein neues Feedback anlegen kann. Alle anderen Listitems lösen wieder ein Event aus mit dem das Menü geöffnet wird:

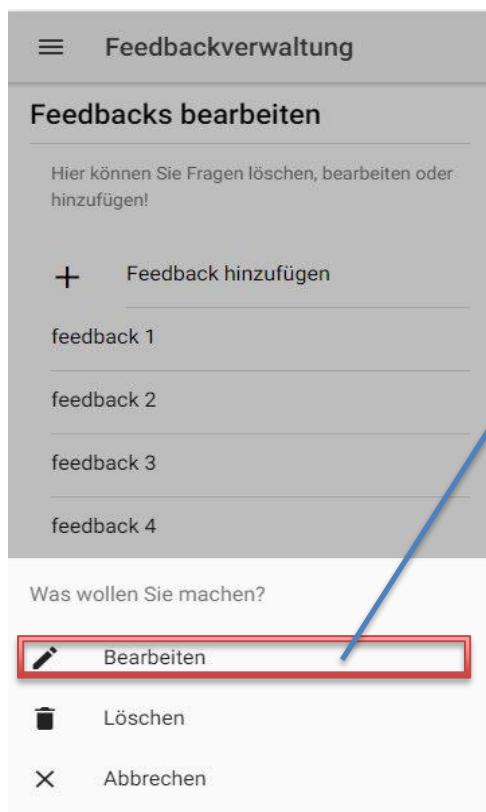


Abbildung 81 Feedbackverwaltung Menü

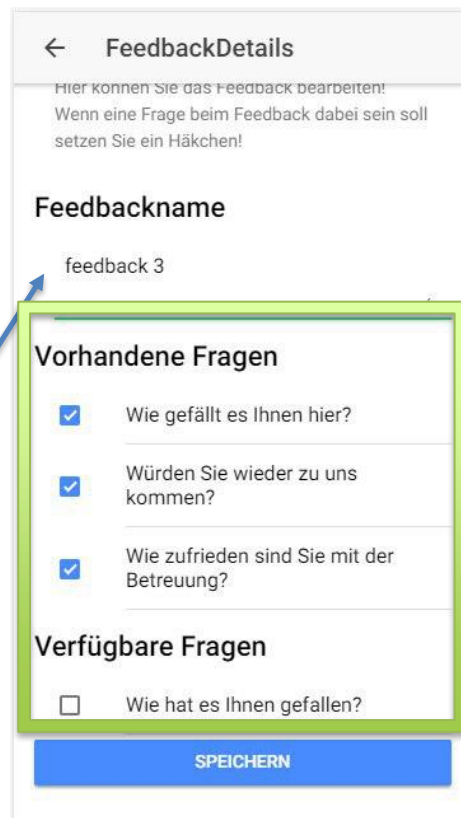


Abbildung 81 Feedback Details

Nachdem der Therapeut auf den Button „Bearbeiten“ gedrückt hat wird er auf die nächste Seite weitergeleitet. Dort hat er die Möglichkeit das Feedback zu bearbeiten. Der Feedbackname ist ein normaler String der beliebig geändert werden kann. Im grünen Bereich sind 2 Listen markiert: „Vorhandene Fragen“ und „Verfügbare Fragen“. Diese Listen beinhalten, wie der Name schon verrät, die Fragen die in dem Feedback enthalten sind und die Fragen die noch verfügbar sind. Beide Listen bestehen aus Items mit Checkboxelementen. Diese Elemente repräsentieren jeweils eine Frage. Wenn der Therapeut nun eine Frage nicht mehr haben will oder eine Frage hinzufügen will muss er nur auf die Checkbox drücken.

9.3.7 Feedbackstatistiken

Wenn der Therapeut eine Frage auswählt kommt er direkt zu den Ergebnissen:

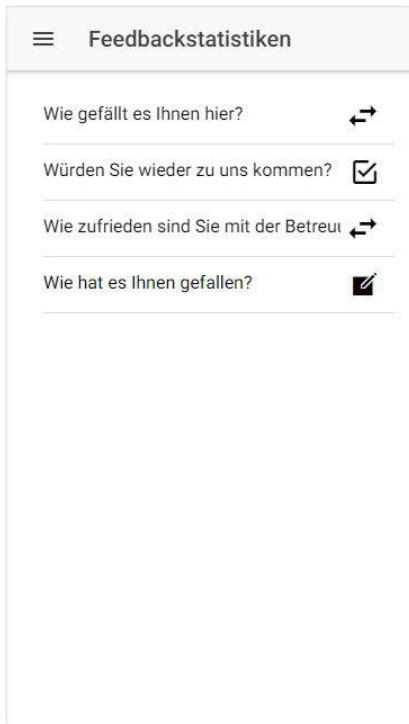


Abbildung 82 Fragen für die Ergebnisse vorliegen

9.3.7.1 Diagramme

Die Ergebnisse werden dann je nach Fragenart unterschiedlich dargestellt. Die Fragenarten 1 und 2 werden in Diagrammen dargestellt. Fragenart 3 wird mit Textfeldern visualisiert.

Für die Diagramme mussten noch zusätzlich ng2-charts und chart mit npm installiert werden:

- C:\Users\<Windows-username>>npm install ng2-charts --save
- C:\Users\<Windows-username>>npm install chart.js --save

Anschließend kann man mit der Implementierung der Diagramme beginnen.

9.3.7.2 Visualisierung der Diagramme

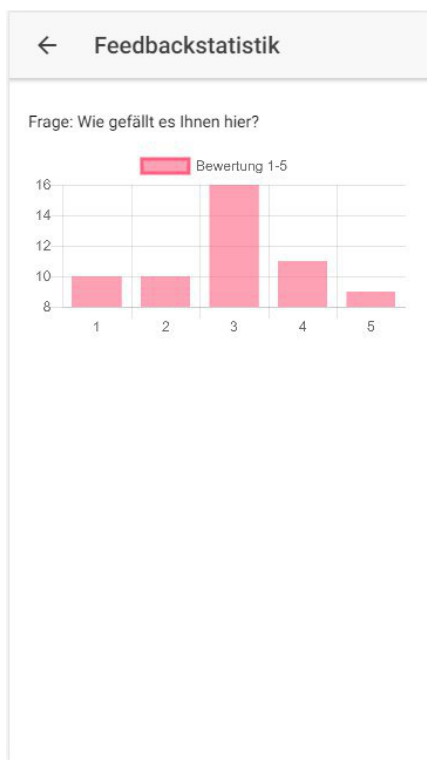


Abbildung 83 Balkendiagramm für Fragenart 1

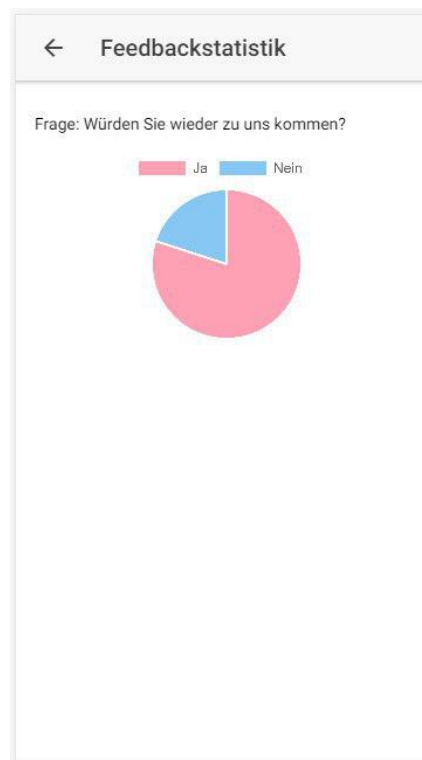
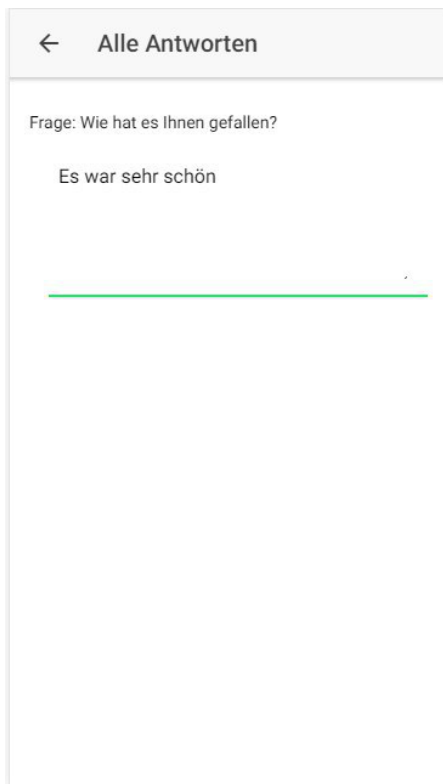


Abbildung 84 Kreisdiagramm für Fragenart 2

9.3.7.3 Visualisierung Fragenart 3

Die Ergebnisse für diese Fragen werden in einer Liste von Textfeldern mit Antworten dargestellt:



The screenshot shows a mobile application interface. At the top, there is a header bar with a back arrow icon and the text "Alle Antworten". Below this, the question "Frage: Wie hat es Ihnen gefallen?" is displayed. Underneath the question, the answer "Es war sehr schön" is shown. A green horizontal line is positioned below the answer, indicating the end of the current entry in the list.

Abbildung 85 Visualisierung Fragenart 3

10 Zusammenfassung

10.1 Testen

Zum Testen der Arbeit wurde eine neue Windows 10-VM erstellt, in der die benötigten Programme installiert wurden, um den Betrieb beim Physiotherapeuten zu simulieren.

10.1.1 Testen der App

Um die App auf einem Android-Smartphone zu testen, muss man zuerst den Entwicklermodus am Handy aktivieren. Dazu navigiert man in den Einstellungen in den Menüpunkt, in dem die Build-Nummer zu finden ist. Das ist auf jedem Handy unterschiedlich, auf Samsung-Smartphones findet man diese in der Regel unter Telefoninfo -> Softwareinfo. Danach klickt man 7 mal auf die Build-Nummer, daraufhin erhält man in den Einstellungen einen neuen Menüpunkt, in dem man die Entwickleroptionen aktiviert. Hat man diese nun aktiviert, führt man im Projektordner den Kommandozeilen-Befehl „ionic cordova run android—device“ aus und die App wird mittels Cordova in die native Programmiersprache übersetzt und ausgeführt.

Da Ionic bereits eine App, Ionic DevApp, zur Verfügung stellt, hat das Projektteam diese zum Testen verwendet. Die DevApp verbindet sich automatisch auf den Webserver, den Ionic beim Ausführen generiert, somit kann man live mitverfolgen, wo Probleme bei dieser App auftreten. Jedoch funktioniert diese App nur, wenn das Entwicklungsgerät und das mobile Gerät auf dem die App ausgeführt werden soll im gleichen WLAN sind, deshalb war dies in der Schule nicht wirklich praktisch umzusetzen und wir mussten auf die erste Variante zurückgreifen.

10.1.2 Testen der Computeranimationen

Das Testen der Computeranimationen erweist sich als sehr einfach, da man nach dem Rendern fertige MP4-Videos erhält. Diese wurden dann auf einen Online-Speicher hochgeladen, von dem aus sie über das Internet abgerufen werden können, in diesem Fall Google Drive. Dazu muss dann noch in der Datenbank zu jeder Übung der Google Drive Link zur jeweiligen Übung gespeichert werden und man kann sie über diesen mithilfe des HTML5-Video-Tags in

der App einbinden. Es wurde dann nur noch getestet, ob die Animationen auf allen mobilen Geräten und auch allen Browsern abgerufen werden konnte.

10.2 Probleme und Lösungen

Bei der Entwicklung der App, speziell beim Zugriff auf das REST-Service, gab es anfangs Probleme, da man mit der seit Angular 5 hinzugefügten Async-Pipe nicht auf den von uns entwickelten REST-Service zugreifen konnte. Nach Absprache mit Herrn Professor Rainer Stropek konnte dieses Problem behoben werden, er beschrieb einen anderen Weg ohne der Async-Pipe.

Bei den Computeranimationen gab es vor allem Probleme durch Verformungen des 3D-Modells. Nahezu bei jeder Bewegung eines Knochens verformte sich das Objekt unrealistisch und es mussten Korrekturen durchgeführt werden. Das konnte mithilfe des Weight-Paint-Modus vielfach korrigiert werden, bei runden Stellen wie der Schulter war dies jedoch trotzdem sehr schwierig, da es hierbei nicht so einfach war, genaue Gewichtungen zu definieren. Hierbei wurde dann mit dem Sculpt-Mode gearbeitet und die Verformungen je nach Typ ausgefüllt oder geglättet.

Zuerst war auch nicht klar, wie man die Animationen am besten in Blender rendert. Hierbei haben diverse Foren ([reddit.com](https://www.reddit.com) bzw. blender.stackexchange.com) weitergeholfen. Die generelle Vorgehensweise habe ich mir mithilfe von YouTube und der Dokumentation der drei verschiedenen 3D-Entwicklungssysteme angeeignet.

10.3 Resümee

Da wir im Projektentwicklungsunterricht bereits mit der Planung und Dokumentation eines Softwareentwicklungs-Projekts vertraut gemacht wurden, war eine rasche Durchführung der Planungsaufgaben, speziell die Erstellung von generellen Projektdokumenten, einer Zeitplanung oder von Use-Case-Diagrammen möglich.

Für die Durchführung der Diplomarbeit war es notwendig, sich in neue Technologien einzuarbeiten. Das Einlesen in neue Technologien mithilfe von Dokumentationen, Foren und YouTube war zwar teilweise sehr zeitaufwändig, jedoch mit der Hilfe von Professoren trotzdem ohne größere Probleme zu realisieren.

Ein hoher Einarbeitungsaufwand war hauptsächlich bei folgenden Technologien nötig:

- ✓ 3D-Computeranimationen
- ✓ Angular
- ✓ Ionic
- ✓ REST
- ✓ SlimFramework
- ✓ Serverkonfiguration

Eine große Unterstützung war auch unser Auftraggeber, welcher uns laufend mit Informationen zu verschiedenen Übungen und teilweise auch Design-Wünschen versorgte, wofür wir ihm sehr dankbar sind.

10.4 Zukunft

Das Projekt wird dem Physiotherapeuten im April, nach der Abgabe der Diplomarbeit, übergeben. Dieser wird die App dann seinen Kunden anbieten, welche somit die Tester der App sind. Somit kann der Physiotherapeut darüber entscheiden, ob die App gut bei seinen Kunden ankommt und uns Feedback darüber geben, was vielleicht noch verbessert oder geändert gehört. So können eventuell weitere Projektteams unser Projekt verändern und erweitern.

Da das Erstellen der Computeranimationen sehr aufwändig ist, konnten im Rahmen der Diplomarbeit nicht zu allen Übungen Animationen erstellt werden. Es ist also möglich, dass ein anderes Projektteam die restlichen Animationen fertigstellt.

11 Literatur- und Quellenverzeichnis

11.1 Abbildungsverzeichnis

Abbildung 1: HTL-Perg (fcp.at, 2017)	14
Abbildung 2: Desktop-Anwendung (Ist-Zustand)	16
Abbildung 3: Projektstrukturplan	19
Abbildung 4: Projekthierarchie	20
Abbildung 5: Use-Case	22
Abbildung 6: Architektur	22
Abbildung 7: Datenbankmodell	24
Abbildung 8: Willkommensfenster der App	28
Abbildung 9: Signup-Seite App	29
Abbildung 10: Loginfenster App	29
Abbildung 11: Homescreen-Kunde App	30
Abbildung 12: Homescreen-Therapeut	30
Abbildung 13: Meine Trainingspläne App	31
Abbildung 14: Detailansicht einer Übung	33
Abbildung 15: Detailansicht einer Übung mit Animation	33
Abbildung 16: Angular-Logo (de.wikipedia.org, 2018)	34
Abbildung 17: Ionic-Logo (de.wikipedia.org, 2018)	34
Abbildung 18: HTML-Logo (wikipedia.org, 2018)	35
Abbildung 19: TypeScript-Logo (wikipedia.org, 2018)	35
Abbildung 20: JS-Logo (wikipedia.org, 2018)	36
Abbildung 21: Logo-Node (wikipedia.org, 2018)	36
Abbildung 22: MySQL-Logo (de.wikipedia.org, 2018)	36
Abbildung 23 Installation phpMyAdmin	37
Abbildung 24 phpMyAdmin Menü	37
Abbildung 25 Exportieren der Datenbank	38
Abbildung 26 Importieren des Datenbankscripts	39
Abbildung 27 Synology Serverkonfiguration	40

Abbildung 28 API im Filesystem	41
Abbildung 29 DDNS Hostname.....	41
Abbildung 30: Blender-Logo (blender.org, 2018)	42
Abbildung 31: 3DS-Max-Logo (autodesk.com, 2018)	42
Abbildung 32: Maya-Logo (autodesk.com, 2018)	42
Abbildung 33: NVIDIA Mental Ray - Logo (nvidia.com, 2018)	44
Abbildung 34: Solidangle-Logo (vimeo.com, 2018)	44
Abbildung 35: Visual Studio-Logo (wikimedia.org, 2018)	45
Abbildung 36: Visual Studio Code-Logo	45
Abbildung 37 Atom Editor	46
Abbildung 38: Ionic DevApp-Logo (play.google.com, 2018)	46
Abbildung 39: Notepad++ Logo (de.wikipedia.org, 2018)	46
Abbildung 40: Postman (getpostman.com, 2018)	47
Abbildung 41: XAMPP Control Panel.....	47
Abbildung 42 POST Anfrage in Postman	50
Abbildung 43 PUT Anfrage im Postman	51
Abbildung 44 Ordnerstruktur SlimFramework	56
Abbildung 45: OpenGL-Logo (khronos.org, 2018)	57
Abbildung 46: Github-Logo (github.com, 2018).....	57
Abbildung 47: Blender-Formen	58
Abbildung 48: Hantelbank (Blender).....	59
Abbildung 49: Node-Editor (Blender).....	59
Abbildung 50: Material zuweisen (Blender).....	60
Abbildung 51: Materialien (3DS Max)	60
Abbildung 52: Material Editor (3DS Max)	61
Abbildung 53: Materialien (Maya)	61
Abbildung 54: Skelett (Blender)	62
Abbildung 55: Hand-Knochen (Blender)	63
Abbildung 56: IK Settings (Blender)	64
Abbildung 57: IK Knie (Blender)	64
Abbildung 58: Quick-Rig (Maya).....	66
Abbildung 59: Ausgangsposition Abduktoren in Seitenlage	67

Abbildung 60: Endposition Abduktoren in Seitenlage	67
Abbildung 61: Weight-Paint-Mode (Blender)	68
Abbildung 62: Weight-Paint-Mode (3DS Max).....	68
Abbildung 63: Weight Paint - Rechter Arm (Maya)	69
Abbildung 64: Kamera und Licht (Blender)	70
Abbildung 65: Kamera (Blender).....	71
Abbildung 66: Frame-Settings (Blender)	72
Abbildung 67: Output-Settings (Blender).....	73
Abbildung 68: H.264-Render (Blender).....	74
Abbildung 69: Video-Tag (Animation).....	75
Abbildung 70: Motion Capture - Planet der Affen (denofgeek.com, 2016)	76
Abbildung 71: Ionic-Menü App	87
Abbildung 72 Feedbackmenü.....	92
Abbildung 73 Feedbackverwaltung Kunde	93
Abbildung 74 Feedbackmenü Kunde	94
Abbildung 76 Beantwortung Feedback 1	95
Abbildung 76 Beantwortung Feedback 2	95
Abbildung 77 Fragenverwaltung Therapeut	96
Abbildung 79 Fragenverwaltung Menü.....	97
Abbildung 79 Fragenverwaltung Bearbeiten	97
Abbildung 81 Feedbackverwaltung Menü	98
Abbildung 81 Feedback Details	98
Abbildung 82 Fragen für die Ergebnisse vorliegen	99
Abbildung 83 Balkendiagramm für Fragenart 1	100
Abbildung 84 Kreisdiagramm für Fragenart 2.....	100
Abbildung 85 Visualisierung Fragenart 3	101

11.2 Literaturverzeichnis

angular.io. (2018). Von Angular: <https://angular.io/api/core/OnInit> abgerufen

autodesk.com. (2018). Von Autodesk:

<http://help.autodesk.com/view/MAYAUL/2017/ENU/?guid=GUID-1B5419EA-A848-4A92-A266-F6C7E7C7F6B3> abgerufen

autodesk.com. (2018). Von Autodesk:

<http://help.autodesk.com/view/3DSMAX/2017/ENU/?guid=GUID-458D9B76-4378-455D-B59B-E766F254A57A> abgerufen

blender.org. (2018). Von Blender: <https://www.blender.org/features/rendering/> abgerufen

blender.stackexchange.com. (kein Datum). Von Blender Stackexchange:

<https://blender.stackexchange.com/> abgerufen

computerstories.net. (2011). Von Computerstories: <https://computerstories.net/motion-capture-explained-982> abgerufen

cycles-renderer.org. (kein Datum). Von Cycles-Renderer: <https://www.cycles-renderer.org/about/> abgerufen

docs.blender.org. (2018). Von Blender Docs:

https://docs.blender.org/manual/en/dev/interface/window_system/introduction.html abgerufen

emn178. (2017). *github.com*. Von GitHub: <https://github.com/emn178/js-sha512> abgerufen

help.autodesk.com/view/3DSMAX. (2018). Von 3DS Max Docs:

<http://help.autodesk.com/view/3DSMAX/2017/ENU/?guid=GUID-F8326C68-F2F9-47F7-AC1D-BA41D7825C7C> abgerufen

knowledge.autodesk.com. (2018). Von Maya Docs:

<https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-F4FCE554-1FA5-447A-8835-63EB43D2690B-htm.html> abgerufen

reactive.io. (2018). Von Reactive: <http://reactivex.io/intro.html> abgerufen

reddit.com. (kein Datum). Von Reddit - Blenderhelp: <https://www.reddit.com/r/blenderhelp/> abgerufen

Rob--W. (2017). *github.com*. Von GitHub: <https://github.com/Rob--W/cors-anywhere> abgerufen

solidangle.com. (2018). Von Solidangle: <https://www.solidangle.com/arnold/> abgerufen

Trammell, K. (15. November 2016). *cgcookie.com*. Von cgcookie:

<https://cgcookie.com/articles/big-idea-blender-render-vs-cycles> abgerufen

Wikipedia. (2018). *wikipedia.org*. Von Wikipedia:

https://de.wikipedia.org/wiki/Hypertext_Markup_Language abgerufen

wikipedia.org. (2018). Von Wikipedia: <https://de.wikipedia.org/wiki/OpenGL> abgerufen

wikipedia.org. (2018). Von Wikipedia: <https://de.wikipedia.org/wiki/MySQL> abgerufen

wikipedia.org. (2018). Von Wikipedia:

[https://de.wikipedia.org/wiki/Python_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Python_(Programmiersprache)) abgerufen

wikipedia.org. (2018). Von Wikipedia: <https://de.wikipedia.org/wiki/Angular> abgerufen

wikipedia.org. (2018). Von Wikipedia: [https://de.wikipedia.org/wiki/Ionic_\(Framework\)](https://de.wikipedia.org/wiki/Ionic_(Framework))

abgerufen

wikipedia.org. (2018). Von Wikipedia:

https://de.wikipedia.org/wiki/VMware#VMware_Player abgerufen

wikipedia.org. (2018). Von Wikipedia: <https://de.wikipedia.org/wiki/TypeScript> abgerufen

wikipedia.org. (2018). Von Wikipedia: <https://de.wikipedia.org/wiki/JavaScript> abgerufen

wikipedia.org. (2018). Von Wikipedia: <https://de.wikipedia.org/wiki/Node.js> abgerufen

Wikipedia.org. (2018). Von Wikipedia:

https://de.wikipedia.org/wiki/Representational_State_Transfer abgerufen

workline, c. (2017). *Creative Workline*. Von

<https://www.creativeworkline.at/2015/09/hybrid-mobile-app-entwicklung-ionic/>:

<https://www.creativeworkline.at/2015/09/hybrid-mobile-app-entwicklung-ionic/>

abgerufen

12 Im Anhang

12.1 Lebenslauf Franz-Filip Schörghuber

Persönliche Daten

Geburtsdaten:	Amstetten, 6. Juli 1999
Familienstand:	Ledig
Religionskenntnis:	röm.-kath.
Staatsbürgerschaft:	Österreich
Eltern:	Franz Schörghuber, Controller Karin Schörghuber, MTF
Geschwister:	1 Bruder

Schulbildung

Derzeit. 5. Jahrgang	HTL für Informatik in Perg
4 Jahre	Hauptschule in Wallsee
4 Jahre	Volksschule in Wallsee

Kenntnisse und Fähigkeiten

Sprachen:	Deutsch (Muttersprache) Englisch (fließend)
EDV-Kenntnisse:	Programmieren in Java, C#, C, Typescript, Angular, Ionic MS Office Bildbearbeitung (GIMP) Webdesign (HTML, PHP, CSS, JavaScript) Vertiefte Kenntnisse im Designen von REST-APIs Verwalten von Datenbanken
Sonstiges:	ECDL Erste-Hilfe-Kurs B-Führerschein

12.2 Lebenslauf Michael Köck

Persönliche Daten

Geburtsdaten:	Amstetten, 1. Dezember 1997
Familienstand:	ledig
Religionskenntnis:	röm.-kath.
Staatsbürgerschaft:	Österreich
Eltern:	Karl Köck, Produktmanager Maria Köck, Selbstständig
Geschwister:	2 Schwestern

Schulbildung

Derzeit. 5. Jahrgang	HTL für Informatik in Perg
4 Jahre	Hauptschule in Grein
4 Jahre	Volksschule in Bad Kreuzen

Kenntnisse und Fähigkeiten

Sprachen:	Deutsch (Muttersprache) Englisch (fließend)
EDV-Kenntnisse:	Programmieren in Java, C#, Typescript, Angular, Ionic MS Office Bildbearbeitung (GIMP) Webdesign (HTML, PHP, CSS, JavaScript) Sound und Videobearbeitung (Pinnacle)
Sonstiges:	ECDL Erste-Hilfe-Kurs B-Führerschein

12.3 Lebenslauf Gregor Arbeithuber

Persönliche Daten

Geburtsdaten:	Linz, 31. Mai 1999
Familienstand:	ledig
Religionskenntnis:	röm.-kath.
Staatsbürgerschaft:	Österreich
Eltern:	Dietmar Arbeithuber, Polizeibeamter Ingrid Arbeithuber, Büroangestellte
Geschwister:	1 Bruder

Schulbildung

Derzeit. 5. Jahrgang	HTL für Informatik in Perg
4 Jahre	Hauptschule in Naarn
4 Jahre	Volksschule in Naarn

Kenntnisse und Fähigkeiten

Sprachen:	Deutsch (Muttersprache) Englisch (fließend)
EDV-Kenntnisse:	Programmieren in Java, C#, C, C++, Python MS Office Bildbearbeitung (GIMP) Webdesign (HTML, PHP, CSS, Javascript) Sound und Videobearbeitung (Pinnacle) Computeranimationen
Sonstiges:	ECDL Erste-Hilfe-Kurs B-Führerschein

